Managing Urban Traffic Networks Using Data Analysis, Traffic Theory, and Deep Reinforcement Learning

Noaeen, Mohammad


doctoral thesis

University of Calgary graduate students retain copyright ownership and moral rights for their thesis. You may use this material in any way that is permitted by the Copyright Act or through licensing that has been assigned to the document. For uses that are not allowable under copyright legislation or licensing, you are required to seek permission.

Downloaded from PRISM: https://prism.ucalgary.ca
UNIVERSITY OF CALGARY

Managing Urban Traffic Networks Using Data Analysis, Traffic Theory, and Deep Reinforcement Learning

by

Mohammad Noaeen

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA
JANUARY, 2021

© Mohammad Noaeen 2021
Abstract

Traffic congestion is a growing problem worldwide and is worsening from the continuous increase in urban population and thus the number of vehicles. Designing new roads to increase road capacity may seem an effective solution in relieving congestion. However, expanding new roads can be a temporary fix but not a sustainable working solution, especially when the new capacity is free of charge for road users. This is because such developments can attract more road users, and the congestion may return to its original state prior to the capacity addition. In turn, improvement of Intelligent Transportation Systems (ITS) efficiency has been found to lead to improved urban transportation and enhanced quality of life. Despite all the advances in traffic management methods and technologies, recent research indicates that it is still a challenge to manage traffic at the network level, due to several factors such as the complexity of traffic phenomenon, difficulties in short-term traffic prediction, and insufficiency of coverage of data-gathering devices. This dissertation proposes new methods to manage and mitigate traffic congestion in large-scale urban transportation networks. It looks at traffic management and traffic signal control areas from different viewpoints using different techniques: data analysis, traffic theory, and reinforcement learning.

Part I of this thesis presents exploratory studies of social media platforms and websites in the area of traffic engineering and management. The main goal of this part is the grasp knowledge from the data of crowd, not only from the theoretical publications, about traffic engineering and management problems. In general, this part helps understand concerns, challenges, requirements, and efficacy of social media data in traffic management using data mining methods. It is first attempted to obtain knowledge about the key areas of traffic engineering based on the crowd data. This part also explores traffic management services requirements from various sources of traffic data using different quantitative and qualitative data analysis and machine learning techniques. Finally, it investigates the efficacy of using social media data (Twitter data) in traffic management systems. The results of this Part suggest that the traffic environment and contextual factors are important
and should be considered in addition to the traffic-theory based methods (presented in Part II) to improve the performance of urban traffic networks.

Parts II and III aim to propose network-level control of large-scale urban networks. In Part II, the goal is to provide decentralized real-time traffic signal control methods using traffic theories. Centralized systems provide a unified control entity that acts centrally, based on the collected data from the sensors and the measurements and computation over the entire network. These approaches do not scale well when they are used to control large urban networks. As a solution to the scaling problem, distributed and decentralized systems present more than one decision-making unit where they do not contain any centralized controller that coordinates or generates traffic plans. Decentralized systems require local data only. In these systems, the local controllers have no interactions with other intersections in terms of both input and output data in decision-making computations. This way, the reliability of the system increases by removing the need for communications between intersections. Thus, communication issues, such as network delays, do not affect the control system.

In Part II, Chapter 5 proposes a traffic cycle time optimization method for the isolated intersection. Moving from the single intersection control towards the network-scale one, Chapter 6 presents a real-time decentralized method for large-scale urban traffic signal control considering the spillback condition. In the network-scale, avoiding and controlling spillback and reducing the time that intersections experience spillback is crucial. Spillback occurs when a queue of vehicles fills up the storage capacity of a link, and no vehicles can enter the link from the upstream approaches. This reduces the outflow of the network.

Part III, first conducts a systematic literature review on the application of reinforcement learning (RL) in the network-scale traffic signal control (in Chapter 7). Following the outcome of this chapter, Chapter 8 proposes a deep Reinforcement Learning-based bi-modal perimeter control, where public transit and car modes are considered. The proposed method is a hybrid control approach, which integrates Proportional Integral controller as the high level controller and deep Reinforcement Learning as the low level controller. The main goal of the proposed perimeter
control is to improve the performance of the entire network by controlling only a limited number of traffic signals along the perimeter of the protected region. The traffic signals learn the optimal decisions independently by interacting with the traffic environment.

**Keywords:** Large-scale urban network, traffic operations, natural language processing, Q&A websites, social media analysis, twitter mining, crowd sourcing, requirements elicitation, information extraction, decentralized signal control, queue spillover, shock wave model, hierarchical bi-modal perimeter control, passenger macroscopic fundamental diagram, public transport priority, deep reinforcement learning, DQN, neural networks, multivariable proportional integral control.
Acknowledgements

I would like to first say a very big thank you to my thesis supervisor, Dr. Behrouz Far, for his continuous support and guidance, and for providing me the opportunity to conduct this research with absolute freedom. Dr. Far is someone who really cares about and will look out for his students. I am very grateful for the opportunity of doing my PhD under his supervision. Thank you, Dr. Far!

I am also very grateful to Dr. Mohsen Ramezani for his scientific advice and insightful discussions and suggestions. Dr. Ramezani, it has been my absolute pleasure and good fortune to work with you on the transportation engineering aspects of my research. I genuinely hope that we will be given the opportunity to work even closer together in the future.

Moreover, I would like to thank Dr. Mahmood Moussavi, Dr. Miguel Goulão, Dr. Ahmad Radmanesh, and Dr. Diwakar Krishnamurthy, the other members of my defence committee, for taking the time to review this dissertation and providing their useful feedback.

Many thanks to Dr. Ana Bazzan, Dr. Amir Abbas Rassafi, Liana Goodman, Atharva Naik, Taimoor Abrar, and Jared Crebo; my other research collaborators who assisted me in carrying out this thesis. I truly enjoyed getting to know each and every one of you.

Finally, my deepest gratitude goes to my dearest mother, Fatemeh, and to my late father Aliakbar. Thank you both for showing me that the key to life is enjoyment while accepting its challenges. Dear Dad, I wished you had lived longer to celebrate this milestone together! I would also like to thank my brothers, Mehrdad and Kamshad, who deserve my wholehearted gratitude. Most importantly, I would like to thank Zahra, my wonderful wife, for her unconditional kindness, patience, support, encouragement and love in every step of the journey. Thank you for your boundless support in the past, in the present, and our future. I dedicate this thesis to my parents and my wife. I love you all dearly.

Calgary, January 21, 2021

Mohammad Noaeen
Preface

All chapters of this thesis (excluding Introduction and Conclusion Chapters) have been either published or submitted for publication as follows:

**Part I**


**Part II**


**Chapter 6: Mohammad Noaeen, Behrouz Far, and Mohsen Ramezani.** “Real-time Decentralized Traffic Signal Control for Congested Urban Networks Considering Queue Spillbacks.” under 2nd round of review in Transportation Research Part C.

Chapter 8: Mohammad Noaeen, Behrouz Far, and Mohsen Ramezani. “Bi-modal Perimeter Control: A Hybrid Control Approach Integrating Proportional Integral Controller and Deep Reinforcement Learning.”. submitted to Transportation Research Part C.
# Contents

1 Introduction  
1.1 Thesis Motivation and Background ........................................ 2  
1.2 Thesis Objectives and Methodology ......................................... 7  
1.3 Thesis Contributions ............................................................. 9  
1.4 Thesis Outline ................................................................. 12  

2 Transportation Engineering on Social Question and Answer Websites: An Empirical Case Study  
2.1 Introduction and Related Work .................................................. 20  
2.2 Study Procedure ................................................................. 23  
2.2.1 Research Questions .......................................................... 24  
2.2.2 Data Collection and Preparation ........................................... 24  
2.2.3 Data Preparation ............................................................. 32  
2.3 Data Analysis ................................................................. 33  
2.4 Evaluation ............................................................. 34  
2.5 Results and Findings ............................................................ 38  
2.5.1 RQ1. The main categories of discussion topics about TE among software practitioners .......................................................... 38  
2.5.2 RQ2. Implications for Researchers and Practitioners .................. 41  
2.6 Threats to Validity .............................................................. 42
3 Let’s hear it from RETTA: A Requirements Elicitation Tool for Traffic management systems
3.1 Introduction ............................................ 52
3.2 RETTA Prototype ........................................ 53
3.3 Conclusion and Future Work ................................ 56

4 The Efficacy of Using Social Media Data for Designing Traffic Management Systems
4.1 Introduction ............................................ 59
4.2 Related Work ........................................... 62
4.3 Methods .................................................. 64
  4.3.1 Research Questions ................................... 65
  4.3.2 Data Collection and Preparation ...................... 65
  4.3.3 Data Analysis ....................................... 68
  4.3.4 Image Analysis ..................................... 68
4.4 Results and Findings ...................................... 68
  4.4.1 RQ1- Information Type ............................... 69
  4.4.2 RQ2- Information Application ....................... 72
4.5 Threats to validity and limitations ........................ 73
4.6 Conclusion and Research Implications ..................... 74

5 Traffic Signal Timing Optimization by Modelling the Lost Time Effect in the Shock Wave Delay Model
5.1 Introduction ............................................. 83
5.2 Shock Wave Theory in Signalized Intersections ............. 84
  5.2.1 Related work ....................................... 84
5.2.2 Shock wave delay model ........................................ 86
5.2.3 Lost time ......................................................... 89
5.3 Proposed Method .................................................. 90
   5.3.1 Formulating the lost time effect ............................ 90
   5.3.2 The proposed optimization model ........................... 91
   5.3.3 Convexity .................................................... 94
5.4 Evaluation ......................................................... 94
5.5 Discussion .......................................................... 95
5.6 Conclusion .......................................................... 96

6 Real-time Decentralized Traffic Signal Control for Congested Urban Networks

   Considering Queue Spillbacks ........................................ 100
   6.1 Introduction ...................................................... 100
   6.2 Problem statement ............................................... 105
   6.3 Decentralized traffic signal control method ..................... 106
      6.3.1 Saturated green time: without spillback control ............ 107
      6.3.2 Saturated green time: considering spillback control ........ 109
      6.3.3 Phase time calculation .................................. 112
      6.3.4 Phase improvement ...................................... 114
   6.4 Numerical experiments ........................................... 116
      6.4.1 Experiment Setup ......................................... 117
      6.4.2 Isolated Intersection Case Study .......................... 120
      6.4.3 Network Case Study ...................................... 121
   6.5 Conclusion ........................................................ 126
   6.A NOMENCLATURE ............................................... 129
   6.B Arrival flow estimation ........................................ 130
7 Reinforcement Learning in Urban Network Traffic Signal Control: A Systematic Literature Review

7.1 Introduction ............................................ 141
7.2 Review Method ........................................ 144
  7.2.1 Search strategy, selection criteria, and data extraction ................. 144
7.3 Results and Findings ...................................... 149
  7.3.1 Publication and authors data ................................ 149
  7.3.2 Traffic simulation and evaluation ................................ 159
  7.3.3 Method identification and analysis .......................... 167
  7.3.4 Authors’ highlights and future works ........................ 188
7.4 Threats to Validity ........................................ 189
7.5 Conclusion .............................................. 190
7.A Performance measures ..................................... 192
7.B Elements in State definitions ................................ 193
7.C Elements in Reward Functions ............................... 194
7.D Detailed major first events ................................... 194

8 Bi-modal Perimeter Control: A Hybrid Control Approach Integrating Proportional Integral Controller and Deep Reinforcement Learning

8.1 Introduction ............................................ 227
8.2 Problem Statement ...................................... 229
8.3 Methodology ............................................. 230
  8.3.1 Constructing 3D-pMFDs to determine the set points ................. 230
  8.3.2 High-level controller: PI controller ................................ 232
  8.3.3 Deep RL controller ......................................... 233
8.4 Results .................................................. 240
  8.4.1 Experiment Setup ......................................... 240
8.4.2 PI controller ........................................ 241
8.4.3 Proposed controller .................................. 245
8.4.4 Training .............................................. 245
8.4.5 Evaluation ........................................... 247
8.5 Conclusion .............................................. 248

9 Conclusions and future research 257
9.1 Part I: Exploratory studies: application of data analysis ....................... 258
  9.1.1 Exploring the most popular traffic engineering-related topics of discussion amongst practitioners ........................................ 258
  9.1.2 A requirements elicitation tool for traffic management systems ............ 258
  9.1.3 Social media data analysis for traffic management systems ................ 259
9.2 Part II: Uni-modal network-scale traffic signal control: application of traffic theories 261
  9.2.1 Traffic signal timing optimization for the isolated intersection .............. 261
  9.2.2 Real-time decentralized traffic signal control for congested urban networks considering queue spillbacks ........................................ 261
9.3 Part III: Bi-modal network-scale traffic signal control: application of reinforcement learning ................................................................. 263
  9.3.1 A systematic literature review of Reinforcement Learning in network-scale traffic signal control ...................................................... 263
  9.3.2 Proposing a hierarchical bi-modal perimeter control by integrating proportional integral and deep RL methods ....................................... 264

A Permissions for Reusing Published Papers in PhD Thesis 266
List of Figures

1-1 Overview of the dissertation chapters and their inter-relation. TM: Traffic Management, TSC: Traffic Signal Control ........................................................... 13

2-1 The process of exploring main discussion topics among software practitioners about transportation engineering ................................................................. 23
2-2 The wordcloud representing the results of frequency analysis on the selected articles 27
2-3 Evaluation of the precision and relevance of the topic modeling results ......................... 37
2-4 Percent of total questions assigned to each topic .......................................................... 41

3-1 UML activity diagram of the core process of RETTA .................................................... 54
3-2 Detailed visual representation of classifying NFRs .......................................................... 54

4-1 Google trends for weather forecast, traffic, and traffic collision along five years (2014-2019) .................................................................................................................. 64
4-2 The keywords extracted using lexical association. ........................................................... 66
4-3 The meta data extracted from Google Vision API in JSON format ................................. 71
4-4 Labels extracted from collected traffic-related images from Twitter. We selected the top 10-15 labels extracted by Google Vision API based on their confidence score: (a) lane, road, vehicle, mode of transport, motor-vehicle, asphalt, traffic, street, snow, parking, family car; (b) transport, vehicle, lane, advertising, lighting, sky, commercial vehicle, road, public utility, traffic light, asphalt, signage; (c) mode of transport, motor-vehicle, commercial vehicle, car, automotive lighting, technology, road.

5-1 General shock wave analysis in the undersaturated condition.

5-2 The effect of lost times on cycle intervals in the shock wave analysis for two approaches.

6-1 Flowchart of the proposed traffic signal control method.

6-2 Shock wave analysis of the saturated green time estimation: a) estimated maximum queue extent is less than or equal to the link length and b) estimated maximum queue extent is greater than the link length.

6-3 Considering the available storage capacity of the downstream link as the available storage capacity of all the downstream lanes.

6-4 Shock wave analysis of the saturated green time estimation when the storage capacity of the downstream link is limited: a) the upstream estimated maximum queue extent is greater than the available storage capacity of the downstream movement, b) the upstream estimated maximum queue extent is equal to the available storage capacity of the downstream movement, and c) the upstream estimated maximum queue extent is less than the available storage capacity of the downstream movement.

6-5 Interphase at time step $K$ based on phase decisions at time steps $K-1$ and $K$: a) if phase $K-1$ is green and phase $K$ is red, the interphase is yellow, b) if phase $K-1$ is red and phase $K$ is red, the interphase is red, c) if phase $K-1$ is red and phase $K$ is green, the interphase is red, and d) if phase $K-1$ is green and phase $K$ is green, the interphase is green.
6-6 A typical intersection and traffic demands of microsimulation case studies: (a) An intersection with four incoming and four outgoing approaches, where each approach includes three distinct left-turn, through, and right-turn lanes, (b) total demand to all approaches in the isolated intersection test case, (c) medium demand in the network test case, and (d) high demand in the network test case.

6-7 A sample triangular FD for the right turns in an offline process: the extracted parameters are $S = 1800$ (veh/hr), $K^{\text{cri}} = 60$ (veh/km), $K^{\text{jam}} = 180$ (veh/km), $u^a = 30$ (km/hr).

6-8 Comparison of (a) average speed, and (b) average queue length, in each 30 (sec) interval over 2.5 (hr).

6-9 Average frequency of movement continuities between two successive phases in the isolated test case with the proposed method.

6-10 Average number of lanes in spillback in the network with (a) medium demand and (b) high demand.

6-11 Frequency of spillback duration in lanes in the network with (a) medium demand and (b) high demand.

6-12 MFD of the network with (a) medium demand, and (b) high demand.

6-13 Comparison of average speed and average queue length in the network over 120 minutes: (a) average speed (medium demand), (b) average speed (high demand), (c) average queue length (medium demand), and (d) average queue length (high demand).

6-14 Comparison of green times between the proposed method and Actuated method in the network with (a) medium demand and (b) high demand.

6-15 Average frequency of movement continuities between two successive phases in phase transition points in the network in the proposed method.

7-1 Directed graph of common bi-grams resulted from the NLP that is used to define the search strings.

7-2 The flowchart of systematic literature review process.
7-3 Distribution (solid line) and trend (dashed line) of number of studies over time. To make the comparison consistent, the 6 included papers published in 2020, until the end of March, are not included in this chart. ........................................ 156

7-4 Left) the distribution of studies based on country of the authors, and right) the number of authors. ......................................................... 158

7-5 Distribution of number of intersections in simulations: in all studies (pie chart), and over time (bar chart). Notes: 1) The data of arterial and network have been merged and the highest network size in each paper is reported, 2) NA represents the papers in which the number of intersections in the arterial or network is not explicitly revealed, 3) The years with no publication, i.e. 1995-1998 and 2001-2002, are not shown. ........ 160

7-6 Frequency of publications in each TSC application domain and variation of RL application. ................................................................. 168

7-7 Major first events in RL in the network-scale TSC collected from the included papers. [RM]: Ramp Metering, [VMS]: Variable Message Signs, [SI]: Signalized Intersection. 188

7-1 Performance measures and the number of their respective occurrences: (Top) The most frequently used performance measures. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All utilized performance measures (with their frequency). .......................................................... 192

7-2 Elements in state definitions: (Top) The most frequently used elements in state definitions. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All elements used in state definitions (with their frequency). ........ 193

7-3 Elements in reward definitions: (Top) The most frequently used elements in reward definitions. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All elements used in reward definitions (with their frequency). 194

8-1 Framework of the proposed method ............................................. 231
Demonstration of the reward function. \( x \) denotes \( \sum_{j=1}^{N_{ij}} \left( \frac{N_{ij}^c + N_{ij}^b}{C} \right) \) and \( r \) is the immediate reward.

Agent types including corner and non-corner agents. The top-right intersection and its adjacent intersection are shown to show the two types of the agents. The feeder lanes on the feeder links are shown in green. The outgoing lanes which discharge to outside region are depicted in blue. The lanes in outside region that discharge to the outside region but through the inside region are not considered as a feeder lane, see the west-north left turn movement on the feeder link in the corner agent.

Demand in the network over time.

MFD diagrams of inside region: passenger production versus car accumulation and passenger production versus bus accumulation diagrams.

The fitted surfaces of 3d-pMFDs of (a) inside region, and (b) outside region.

a) \( U_{12} \) over time in PI control, and b, c, d, e) comparison between fixed time control and PI control in terms of car and bus accumulations in inside and outside regions. The dashed lines indicate the set points.

RL control: Training process.

Comparison of the three methods: total passenger delay over time.
List of Tables

2.1 The identified articles for identifying the most frequently terms in the area of trans-  
portation engineering (#C: number of Citations) ........................................ 26
2.2 Search strings extracted by frequency analysis. ........................................ 30
2.3 Topic Modeling Results (K represents the number of topics) ....................... 31
2.4 Qualitative and quantitative information about the projects included in our study.  
WI and TI indicate word intrusion and topic intrusion. .................................. 38
4.1 Results of the BTM approach ..................................................................... 70
5.1 Results of the Proposed Signal Timing Method and Webster’s Method. .......... 95
6.1 Extracted FD parameters for the three types of movements ......................... 119
6.2 Comparison of the three methods in isolated intersection case. The numbers in  
parentheses show the improvements in terms of % with respect to the Webster method. 120
6.3 Comparison of the three methods in the network case. The numbers in parentheses  
show the improvements in terms of % with respect to the Webster method. .......... 126
7.1 Details of the search process ....................................................................... 145
7.2 Overall comparison of the included articles. [✓/x]: yes for discrete methods and no  
for continuous methods. ............................................................................. 151
7.3 Identification of the included papers in terms of data source, method identification, code, simulation, and evaluation. .................................................. 163

7.4 Methods used for comparison .................................................................. 166

7.5 Categorization of types of traffic signal based actions in the RL methods. ...... 182

7.1 The details of major first events in RL in the network-scale TSC collected from the included papers. [AV]: Autonomous Vehicle, [LIME]: Local Interpretable Model-Agnostic Explanations, [SHAP]: Shapley Additive exPlanation ........ 195

8.1 Comparison of the performance of the three controllers. The values in parentheses indicate the measures’ percentage of reduction of PI and proposed methods compared to the fixed time method. .................................................. 248
The task of optimization and control of traffic signals is a pivotal concern for most modern cities and is important to the management of traffic flow, energy consumption, air pollution, and safety in urban areas. Intelligent Transportation Systems (ITS) have been utilized to help with traffic control, in order to alleviate the detrimental impacts of vehicle traffic. By emergence of cutting-edge communication and processing technologies as well as development of sensing technologies and data gathering devices, traffic control has emerged as a potential field that recently has received a large amount of attention. This presents the problem of improving traffic flow and traffic signal control within the already existing infrastructure.

To address this problem, plenty of traffic management systems, policies, and technologies have been developed and implemented. This dissertation proposes and develops new methods for traffic control in large-scale urban networks. In essence, this dissertation looks at this multidisciplinary problem from different views, including data mining, crowd sourcing, requirements engineering, applied physics, traffic engineering, optimization, machine learning, and reinforcement learning (RL) methods.

In this chapter, Section 1.1 outlines the thesis motivation and briefly introduces the research background related to the main objectives of this thesis. A corresponding detailed state-of-the-art literature review is provided later in each chapter. Section 1.2 presents the objectives that this thesis
accomplishes. The main contributions of this thesis corresponding to each chapter are elaborated in Section 1.3. Finally, Section 1.4 outlines the structure of the thesis.

1.1 Thesis Motivation and Background

The cost of traffic in the world was estimated at $1 trillion in 2013 [13]. Worldwide, half a billion cars were on the road in 1985, and this number doubled by 2010 [2]. This figure is expected to hit two billion in 2020 [1]. With an explosion in urban and rural population rates, city transportation systems become less efficient at handling the ever-growing number of commuters. A lack of space and resources with which to improve infrastructure poses problems in accommodating the increasing urban population. The resulting congestion further leads to increased pollution caused by sitting idle in traffic jams, traffic delays and bottlenecks, and a rise in accidents. The secondary issues that arise are also severe, including economic loss, and an overall decrease in quality of life. This presents the problem of improving traffic flow and traffic signal control (TSC) within the already existing infrastructure.

Despite all the advances in traffic management, recent literature indicates that it is still a challenge to manage traffic at the network level, due to several factors such as the complexity of traffic phenomenon, difficulties in short-term traffic prediction, and insufficiency of coverage of data-gathering devices [46]. The amount of literature in this research area is considerable, and several researchers have widely attempted to contribute to traffic control in different ways, based on various assumptions, goals, and requirements. Hence, the goal of this dissertation is to move towards this challenging problem and develop methods for controlling large-scale urban traffic networks. It should also be noted that we discussed the research problem and the related requirements of the traffic management and signal control systems with traffic engineers and managers at the City of Calgary through a few meetings and learned from their field experience in using and installing existing real-world traffic control systems.
Part I contains three chapters, which mainly focuses on the exploratory studies to understand what the data can provide us with. In Chapter 2, we begin with exploring and better grasping what the most challenging topics related to traffic engineering and traffic management are among practitioners. Analyzing Q&A websites in other areas of research can help with finding the major topics, common challenges, and misconceptions that software practitioners and developers are most interested in discussions (e.g. [41, 40, 5]). Different from these research works that focused on developers (e.g. mobile developers and web developers), we conduct this exploratory study to identify the areas of traffic engineering that may require extra attention by researchers. This Chapter (Ch. 2) addresses the following research questions to seek a holistic view of traffic engineering, through the eyes of the social Q&A sites: (i) What are the key categories of topics of discussions about transportation engineering among software practitioners? This research question groups together common issues in transportation engineering (according to data from the Q&A sites) and finds any outstanding lessons or stories that can be derived from those groups, in order to identify concerns of the research community about traffic engineering. (ii) What are the key implications of this exploratory case study for researchers and practitioners?

Chapter 3 designs the RETTA Tool, a requirements elicitation tool for traffic management systems. Developing software-reliant systems for complex domains such as the traffic management domain requires a more challenging requirements engineering job. While, in recent decades, a wide range of requirements elicitation techniques has been developed to address complex systems, these techniques usually aim to work in a context-free spectrum and thus are not concerned about the complexity of a particular domain, such as the traffic management domain. In this domain, we deal with many constraints to elicit requirements, such as the diversity of stakeholders, a clear need for time-centric elicitation techniques, the legal issue for some types of elicitation techniques, the variability of the transportation demand, and knowledge of the road network and the traffic conditions. Moreover, some of the software tools in this domain are critical because they involve the life and safety of individuals (e.g. emergency control systems) and are very sensitive to mistakes. To address these issues, this chapter introduces the RETTA Tool, which aims to tackle problems
associated with eliciting requirements for traffic management services such as emergency medical services, traffic signal timing, and urban transportation planning systems. The main stakeholders in most software tools in the traffic management domain are the crowd participants in a traffic network, who use traffic management services and at the same time define the requirements of these services. Given this we can rely on them to cater for the stakeholders’ needs properly.

It has long been acknowledged in the context of developing dynamic and reactive systems that users’ input during different stages of the development process helps to quickly and incrementally adapt to changes in the system’s context and users’ needs. Given the data- and communication-intensive nature of developing transportation management systems, utilizing social media data provides a new route for a dynamic collection of needs and experiences in a timely and direct fashion. Concerning the usefulness of social media data in traffic management in urban networks, some may believe that social media data such as Twitter data is redundant to traditional sensors data. In contrast, a study by Chen et al. [7] supports the hypothesis that Twitter data offers a suitable complementary source to traditional sensor data. Although traditional sensors can reasonably provide traffic data in various traffic conditions, they cannot reflect the real cause of traffic conditions. This fact led to proposing methods in integrating physical and social sensors for traffic management and signal control. Crowdsourcing and dynamic information collected from the end-users of traffic management systems are key to developing functional and satisfying systems for all stakeholders. However, finding what these stakeholders and end-users desire from traditional system characterization processes and theory-based academic resources alone can be difficult, lengthy and inaccurate. By analyzing social media data, including text and image, Chapter 4 provides a systematic methodology to efficiently explore social media data in the context of traffic management and find out what can be understood from what the people post and share.

The scope of Chapters 5 and 6 in Part II is traffic signal control in the isolated intersection and large-scale network of intersections. Optimizing and controlling traffic signals is a long-standing concern for most urban cities. The type of traffic control method impacts the challenges of data gathering, processing, and transmission. Centralized systems provide a unified control entity that
acts centrally, based on the collected data from the sensors and the measurements and computation over the entire network. These approaches do not scale well when they are used to control large urban networks. As a solution to the scaling problem, distributed and decentralized systems present more than one decision-making unit where they do not contain any centralized controller that coordinates or generates traffic plans.

In distributed systems, every single intersection controller is an independent agent that makes its own decisions in interactions with other neighbor intersections. In these systems, there is a chance of information exchange between neighbor intersections and the decisions might be made by negotiation between the intersections. In contrast, decentralized systems require local data only. In decentralized systems, the local controllers have no interactions with other intersections in terms of both input and output data in decision-making computations. This way, the reliability of the system increases by removing the need for communications between intersections. Thus, communication issues, such as network delays, do not affect the control system. There are some research works in this class of traffic network signal control [18, 6, 29, 44, 43, 42, 45, 19, 28, 30]. In a recent research, Li et al. [31] proposed a decentralized Max Pressure based method, called position-weighted backpressure (PWBP) which captures the spatial distribution of vehicles along the links, and consequently, spillback conditions. The PWBP method employs a capacity-aware version of Max Pressure, proposed by Varaiya [45], which additionally accounts for the possibility of spillbacks by considering spatial distribution of vehicles through applying higher weights to queues that extend to the ingress of the link. In the PWBP method, a small number of phases (i.e. 4 - 8 possible phases) are used. In contrast to this work, a shock wave theory-based traffic model may better model the dynamic behaviour of traffic, especially in spillback conditions. Moreover, exploring all possible phases (with any number of non-conflicting movements) to determine the optimal phase can provide a better solution. Furthermore, in the network scale, avoiding and controlling spillback condition and reducing the time that intersections experience spillback is crucial.

The motivation of this part is to propose a new decentralized network-level traffic signal control method that addresses the effects of queue spillbacks. The method should be traffic-responsive
without any need for data communication between intersections’ controllers. The method should also be a feasible solution in all conditions in the entire network with any scale within a short amount of time, to make it favourable for real-time applications.

Finally, **Part III** includes two chapters. In **Chapter 7** of this part, we conduct a systematic literature review to dissect the existing research that applied reinforcement learning in the network-scale traffic signal control to explore the gaps and opportunities in this area of research.

Based on the open research problems found in the RL area, **Chapter 8** follows the idea of the perimeter control in large-scale urban networks. Regulating traffic signals along the perimeter of a controlled area in a network, known as the perimeter control, has proven efficient in reducing congestion and travel time within the network ([11, 16, 39, 21, 23, 36]). Given the existence of a well-defined Macroscopic Fundamental Diagram (MFD) for a network ([33, 15, 12]), the inflows can be controlled to keep the network accumulation close to the critical value, although this may come at the expense of an increased delay for the vehicles outside the controlled area. There have been extensive research efforts in distributing the inflows via a spatially-uniform perimeter plan; however, the methods that distribute inflows in a spatially-varying way offer lower network travel time ([39, 27, 26]). Recently, the use of reinforcement learning (RL) has gained attraction in traffic signal control. [37] proposed a reinforcement learning approach to control perimeter inflows via spatially-varying metering rates, which is, to the best of our knowledge, the first and so far the only use of reinforcement learning in the perimeter control. Generally, an important advantage of the perimeter control is that it does not entail high computational effort and it is supported by stability analysis ([3, 22, 14]). This work does not consider public transit mode in the network, while public transit (e.g. buses) is an inevitable part of a sustainable urban traffic management system. Compared to mass transit, as in urban rail transit, buses use street roads to provide more convenient access for the passengers. On the other hand, they can transfer more passengers using less space. Considering the substantial role of buses in urban networks, many researchers have provided different strategies in the area of bi-modal and multi-modal traffic control [10, 9, 25, 49, 47, 35, 38, 48, 20, 50]. However, only a few studies focused on developing bi-modal
strategies at the network level, specifically as a perimeter control, e.g. [4, 24]. The research in this recent area has been facilitated by extending the concept of MFD to multi-modal p-MFD [8], bi-modal 3D-MFD (or 3D-vMFD) and passenger 3D-MFD, also known as 3D-pMFD [17]. The 3D-MFD relates the accumulation of cars and buses, and the total circulating vehicle flow in the network, while in 3D-pMFD network passenger flow is taken into account to consider that buses carry more passengers. Similar to MFD, 3D-MFD exists empirically ([32, 34]).

The motivation of this part is to improve the efficiency of the network in terms of delay time and travel time by proposing a bi-modal perimeter control method that provides public transit priority in the perimeter of urban network using a deep reinforcement learning approach.

1.2 Thesis Objectives and Methodology

The two goals of this dissertation are to develop (i) synthesized understanding of the contextual factors in traffic control, and (ii) decentralized real-time control methods for large-scale congested transportation networks to improve the network performance. The first goal is followed in Part I and the second goal is researched in Parts II and III in the uni-modal and bi-modal traffic, respectively. The detailed objectives of the thesis according to the structure of the chapters are listed as follows:

- **Part I: Exploratory studies: application of data analysis**
  - The objective of Chapter 2 is to conduct a large-scale exploratory study of social Q&A communities (i.e. Stack Overflow, Programmers Stack Exchange) to observe the key areas of discussion about traffic engineering and management among practitioners. A better understanding of what are the most challenging topics related to this area of research among practitioners can help to identify the areas of traffic engineering and management that may require extra attention
- The objective of Chapter 3 is to leverage the wisdom of the crowd to help with the requirements elicitation and classification task in the traffic management domain. This bridges the gap among stakeholders from both areas of software development and transportation engineering.

- The objective of Chapter 4 is to efficiently explore social media data in the context of traffic management through a systematic methodology and find out what can be understood from what users post and share. We look for issues and relevant information that may assist authorities and software development teams in making decisions when designing and developing traffic management systems by leveraging lay people’s input.

- **Part II: Uni-modal network-scale traffic signal control: application of traffic theories**

  - The main objective of Chapter 5 is to optimize cycle time in the isolated intersection through finding the optimum cycle time and green splits. The number of required variables should be kept low and easily measurable. Although the goal is to present a solution for the isolated intersection case, the proposed method might be extended to coordinated intersections and network-scale traffic signal control.

  - The key objective of Chapter 6 is to develop a real-time decentralized traffic signal control method for large-scale urban networks. The method should address the effects of queue spillbacks. It should remove the need for communications between intersections to increase the reliability of the system. Thus, communication issues, such as network delays, do not affect the control system. The method should result in a feasible solution in all conditions in the entire network with any scale within a short amount of time to make it favourable for real-time applications.

- **Part III: Bi-modal network-scale traffic signal control: application of reinforcement learning**
- The main objective of Chapter 7 is to conduct a systematic literature review to analyze the existing research that applied reinforcement learning in the area of network-scale traffic signal control, in an effort to provide statistical and conceptual knowledge and identify past and present trends and provide a future roadmap for research in the area.

- The primary objective of Chapter 8 is to provide a novel hierarchical perimeter control to improve the network-wide performance using deep reinforcement learning. In this study, we consider two traffic modes: public transit and cars. To reduce the state space in RL while providing a good image of the entire network, the output of a macroscopic controller in high level control is used as one of the state elements. This method should provide public transit priority on the perimeter of the protected region as a consequence of the learning process in the RL method.

1.3 Thesis Contributions

Considering the thesis motivation and objectives, the contributions of the thesis in each chapter are as follows:

**Chapter 2** The exploration and identification of the main discussion topics of traffic engineering and management among practitioners, which provides insights into the different categories and problem areas of traffic engineering by conducting statistical analysis on the data set retrieved from Q&A websites to measure the proportions of the types of questions asked about traffic engineering. This helps to identify the challenges facing traffic engineers that require more attention from the traffic engineering and management research and development communities in the future.

**Chapter 3** An interactive solution for eliciting requirements and exploring functional and non-functional requirements of software-reliant systems in the area of traffic management. A tool is prototyped to leverage the wisdom of the crowd and combine it with machine learning
approaches such as Natural Language Processing (NLP) and Naïve Bayes to help with the requirements elicitation and classification task in the traffic management domain. This bridges the gap among stakeholders from both areas of software development and transportation engineering.

Chapter 4 By analyzing social media data, including text and image, this chapter provides a systematic methodology to efficiently explore social media data in the context of traffic management and find out what can be understood from what users post and share. We also explore how this understanding is similar to or differs from theory-based publications. To this end, we use manual qualitative analysis, automatic information extraction, and NLP to analyze social media data. Compared to the existing research on social media for traffic management, our proposed approach is distinct in three ways: (1) We use three different sources of data, including theory-based publications, Google Trends, and Twitter data; (2) We use a systematic search method to explore search strings for filtering data collected from social media, i.e. Biterm Topic Modelling (BTM), and Weighted Finite State Transducers (WFST), and (3) To address the problem from different perspectives, we analyze two different data types, i.e. text and image, using NLP, deep neural networks (DNN), and descriptive statistics.

Chapter 5 This chapter proposes an optimization model to find the optimum cycle and green splits by minimizing the shock wave delay model for a simple isolated signalized intersection as a pre-timed method. To do so, the lost time effect is considered in the shock wave delay model. The incremental effect of the lost time in extending the required green times in each cycle affects the extension of the red time in the other approach. This way, cycle time is optimized, based on the minimum required green and red times in both approaches, by establishing an appropriate balance between the two approaches, while providing the minimum delay for individual vehicles in the intersection. In this method, the key strategy is to keep both approaches in the undersaturated condition. To the best of our knowledge, this work is the
first to use the shock wave model to find the optimum cycle time and green splits, which differs from the cases that only the green splits are optimized given a cycle time.

**Chapter 6** This chapter proposes a decentralized network-level traffic signal control method addressing the effects of queue spillbacks. The method is traffic-responsive, does not require data communication between intersections’ controllers, uses lane-based queue measurements, and is acyclic. Each traffic controller operating on an intersection aims at maximizing the effective outflow rate locally and independently with the goal of maximizing global throughput of the entire network. At each intersection, the signal control method estimates and adopts the maximum possible phase time in which all active movements discharge at their full capacity. This is modeled using a shock wave-based queue length estimation model while capturing the spillback at the downstream links. The method demands real-time data, including the queue lengths, the arrival flows, and the downstream queue lengths in all the lanes at the control decision times. The proposed method results in a feasible solution in all conditions in the entire network with any scale within a short amount of time, which makes it favourable for real-time applications.

**Chapter 7** This chapter presents a comprehensive, systematic literature review on the application of reinforcement learning (RL) in the network-scale traffic signal control (TSC). The main goal of this chapter is to identify all eligible articles in the defined area, analyze the data of the included articles, and provide findings based on the data analysis. In other words, this chapter aims to provide statistical and conceptual knowledge and identify past and present trends and a future roadmap for research in this area. Only the network-scale papers that tested the proposed methods in networks with two or more intersections are targeted. This review covers 160 peer-reviewed articles published from 1994 to March 2020. The study in this chapter is the most comprehensive systematic literature review on the application of RL in the network-scale traffic signal control that addresses the most important and fundamental characters and components of the existing methods in the area. An in-depth analysis on
the important components is provided, including: the publication and authors’ data, traffic simulation and evaluation, the methods, authors’ highlights and implications for future study.

Chapter 8. This chapter proposes a hierarchical bi-modal network-level perimeter control method addressing the effects of queue spillbacks and facilitating the bus priority. In this chapter, we propose a PI control-based deep reinforcement learning method to control the perimeter in a large-scale urban network with 400 intersections. The contributions of this work are: (i) the application of PI control in an RL controller as a perimeter control in a bi-modal network, (ii) applying deep RL in a hierarchical system with designing both high-level and low-level controllers, (iii) considering various factors in designing the RL controller, including providing bus priority, addressing spillback conditions, the output of PI controller in designing the elements of the RL controller, and (iv) using microsimulation for evaluation of the proposed method.

1.4 Thesis Outline

This thesis is comprised of 9 chapters, including Chapter 1, Introduction, and Chapter 9, Conclusion and future research, and 7 main chapters, which are organized into 3 parts. Part I includes Chapters 2, 3, and 4, which focus on exploratory studies to leverage the wisdom of the crowd in the area of traffic control. Part II consists of Chapters 5 and 6, which propose traffic signal control methods for the isolated intersection as well as for large-scale congested urban traffic networks focusing on the uni-modal mode, i.e. cars. Part III includes Chapters 7 and 8 that elaborate on the application of reinforcement learning in the network-scale traffic signal control. Chapter 7 presents a systematic literature review in this area and Chapter 8 proposes a perimeter control method in the bi-modal large-scale network, including buses and cars. The proposed method integrates deep RL and PI control to utilize the efficiency of both control methods. Note that each main chapter is a complete stand-alone research article with its own notations. Figure 1-1 shows a visual overview of the dissertation and the inter-relation between chapters. In this figure, we illustrate how each chapter
(or part) motivates the research in other chapters, or provides information and insight for conducting research in other parts. Finally, two future works are recommended. The first is to integrate the proposed network-level traffic signal control method in Chapter 6 and the hierarchical RL-based bi-modal perimeter control method proposed in Chapter 8 to get the benefit of both in improving the network performance. The second is to consider the context-sensitive factors obtained from social media text and image data analysis as a metadata input to the above-mentioned integrated method.
Bibliography


[34] Monica Menendez, Lukas Ambühl, Allister Loder, and H Becker. Evaluating london’s congestion charge–an approach using the macroscopic fundamental diagram, transportation research arena (tra), vienna. 2018.


Part I

Exploratory Studies: Application of Data Analysis
Abstract

In the last decade, the area of Transportation Engineering (TE), and its underlying disciplines such as public transit, connected vehicles, road planning, and air traffic management, has become increasingly prominent. A better understanding of what the most challenging topics related to TE are among practitioners will greatly help to identify the areas of TE that may require extra attention by researchers and project managers. However, there has been very little experimental work in regards to identify true practitioners’ needs on the implementation and understanding of TE activities and tasks. Therefore, in this chapter, we use data from the popular social Q&A sites (e.g. Stack Overflow and Engineering Exchange), and analyze 2,457 questions and answers in order to examine the needs of transportation engineers, and their concerns and questions. We applied Latent Dirichlet Allocation-based (LDA) topic models and statistical analysis to explore the main related topics to TE.

Our findings show that practitioners are questioning the application of GIS tools, such as QGIS and ArcGIS for managing and implementing road planning. Further, we determined the popularity of each topic by conducting statistical analysis. Our findings help highlight the challenges facing transportation engineers, which require more attention from the Civil Engineering, Software Engineering, and specifically Data Analysis research communities and establish a novel approach for analyzing the content of social Q&A websites.

2.1 Introduction and Related Work

In the last decade, the area of Transportation Engineering (TE) and its underlying disciplines such as public transit, connected vehicles, road planning, and air traffic management has become increasingly prominent. A better understanding of the most challenging topics related to TE among
practitioners is required in order to help to identify gaps in knowledge and application. Therefore, we conducted a large scale exploratory study of social Q&A communities in order to observe the key areas of discussion about TE among software practitioners.

Regarding the related work that analyzed Q&A websites in other areas of research, Barua et al. [1] use topic modelling on StackOverflow posts to identify the main topics that developers discuss on the site, with results showing that developers are most interested in discussions involving web development, mobile applications, Git, and MySQL. Another study by Rosen and Shihab [2] follows a similar topic modelling technique but focuses on mobile developers. Two of their main focuses are the types of issues mobile developers discuss and questions they ask (i.e. what, how, or why). They found that mobile developers mainly ask questions about “how” something should be done. Similarly, Bajaj et al. also use topic modeling to determine what topics web developers are discussing on Stack Overflow [3], with the end goal of determining common challenges and misconceptions among web developers.

Our research is notably different from these research works because it directly addresses a new data intensive and media-rich field of research; “transportation engineering”. We seek a holistic view of TE to better evaluate its strengths and weaknesses, through the eyes of the social Q&A sites.

We applied Latent Dirichlet Allocation-based (LDA) topic models and statistical analysis in order to explore the main topics of discussions about TE among practitioners. In this chapter, we outline our process for collecting and preparing the data set related to TE. We analyzed our results by composing topic models from our collection, and conducted a survey with 26 participants to evaluate the accuracy of these topic models. Moreover, we conducted a statistical analysis by categorizing a sample of the titles we collected.

Our findings show that (a) one significant topic of interest among transportation engineers and developers is to better understand the ways they can customize and apply the existing GIS tools and technologies (e.g. Google Map) in order to perform geospatial data analysis (such as simulating highways and roads on a map and measuring the distance various values of data points (i.e.
(b) A large portion of the questions that directly asked for input from the Q&A community were about environmental aspects of transportation engineering (e.g. transportation planning for optimizing the travel time, and fuel consumption); (c) A common desire among developers and practitioners is to collect and analyze the traffic data sets, and produce information applicable for cities and their traffic problems; (d) One significant topic of interest among practitioners who work in the area of transportation engineering is planning urban roads, as well as the application of GIS tools (e.g. QGIS and ArcGIS) in order to alleviate the complexity of this task; and (e) Most of the practitioners found the application of GPS technology the most relevant and useful technique for managing and optimizing traffic signal timing.

Furthermore we found that, the topics of application of GIS tools (such as QGIS, and ArcGIS) significantly outweighed others in popularity. These results indicate the usefulness of these tools in practical road planning and transportation engineering, which can be applied in theoretical urban planning research. The two significant contributions of this study are as follows:

- Exploring and identifying the main discussion topics about TE among software development practitioners and transportation engineers, which provides insights into the different categories and problem areas of TE.
- Conducting statistical analysis on the data set retrieved from Q&A websites in order to measure the popularity of the main topics related to TE. This helps identify the topics for future research and development.

The rest of this chapter is structured as follows: Section 2.2 describes the procedure we followed to conduct this explorative case study, such as identifying the main goals and research questions, data collection and preparation, and data analysis. In this section we followed the same process we implemented in previous work [4] in order to explore the key discussion topics among practitioners about “Requirements Engineering”. Section 2.3 presents our analytical approach. Section 2.4 reports the process and the results of our evaluation. The key results and findings of this study are reported
in Section 2.5. Threats to the validity of the results of this research are discussed in section 2.6. In section 2.7, we conclude the chapter by reporting our key findings which address the research questions in section 2.2.1.

2.2 Study Procedure

This section elaborates upon the main steps of this study. Figure 2-1 illustrate the process we applied to implement our study.

![Diagram](image)

**Figure 2-1:** The process of exploring main discussion topics among software practitioners about transportation engineering
2.2.1 Research Questions

This chapter addresses the following research questions:

• **RQ1**: What are the key categories of topics of discussions about transportation engineering among software practitioners?

This RQ groups together common issues in transportation engineering (according to data from the Q&A sites) and finds any outstanding lessons or stories that can be derived from those groups, in order to identify concerns of the research community about TE.

• **RQ2**: What are the key implications of this exploratory case study for researchers and practitioners?

2.2.2 Data Collection and Preparation

STEP 1- Data Collection for Frequency Analysis

As stated by Shapiro and Pearse [5], citation counts can be used as relatively objective tools for evaluating scholarly feedback, as well as for demonstrating the writing styles that are most frequently used by other researchers and practitioners. Therefore, we identified the top 10 most frequently cited papers in the areas of TE and Big Data Analytics (BDA) in Google Scholar published in each of these periods: 2016, 2015, 2014, and from 2001 to 2013. We limited our search to certain common words in the area of transportation and traffic in the title of papers, without contextual consideration, in order to ensure the papers are related to the transportation field. On the other hand, to collect papers about new trends in the area of TE, we also used the string “Big Data” in our query in the title section, as Big Data Analytics is a new hotspot in this field. Furthermore, we manually reviewed the 40 papers to confirm that the context of all papers sampled are related to TE. We also examined the quality and found that publications occurred in high quality journals and conferences, such as IEEE, Science Direct, Springer, and ACM. Following query was used:
Allintitle: “traffic” OR “car” OR “transportation” OR “urban” OR “urbanism” OR “vehicle” OR “rural” OR “cities” OR “congestion” AND “big data”.

Table 2.1 lists the 40 papers collected from our search, including the title of papers, the year of publication, and their citation counts.

STEP 2- Identifying Search Strings

Stack Exchange Data Explorer (SEDE) is an interactive open-source web tool for sharing, querying, and analyzing the data sets from every website in the Stack Exchange network\(^1\). To define the search strings for crawling 11 sites in SEDE, we applied text mining techniques to extract the most frequently used words used in the articles identified in STEP 1. To this end, we used the Text Mining Package (tm) \(^2\) of R, an open-source programming language and software environment for statistical computing \(^3\). As stated by Keshav \([46]\), the title, abstract, introduction, section/subsection headings (not their content), and conclusion of a paper represent the general idea and the main contributions of a paper. Moreover, given the unstructured nature of natural language text and the negative impact of dirty data on the result of text analysis tasks, data cleansing is one of the most important steps in text analysis. Hence, before conducting the frequency analysis we applied the main steps of text pre-processing on these parts of the 40 papers identified in STEP 1.

For the purpose of pre-processing, we conducted the following steps iteratively:

1. **Manual Transformation:** In this step, we performed a few preliminary clean-up steps, such as removing hyphens (e.g. an-alysis, sys-tem) which appeared after converting the pdf files to plain text. We also removed the common words in research papers such as: paper, approach, present, focus, need, exist, address, propose, and data.

2. **Removing numbers and punctuations:** All of the numbers, such as years and in-text reference numbers, are removed in this step.

\(^1\)https://data.stackexchange.com/
\(^2\)http://tm.r-forge.r-project.org/
\(^3\)https://www.r-project.org/
Table 2.1: The identified articles for identifying the most frequently terms in the area of transportation engineering (#C: number of Citations)

<table>
<thead>
<tr>
<th># Papers</th>
<th>Year</th>
<th># C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Energy-efficient dynamic traffic offloading and reconfiguration of networked data centres for big data stream mobile computing: review, challenges, and a case study [7]</td>
<td>2016</td>
<td>15</td>
</tr>
<tr>
<td>3 Urban planning and building smart cities based on the Internet of Things using Big Data analytics [8]</td>
<td>2016</td>
<td>9</td>
</tr>
<tr>
<td>4 Big Data for Social Transportation [9]</td>
<td>2016</td>
<td>6</td>
</tr>
<tr>
<td>5 Big Data Considerations for Rural Property Professionals [10]</td>
<td>2016</td>
<td>5</td>
</tr>
<tr>
<td>8 Special issue on big data driven Intelligent Transportation Systems [13]</td>
<td>2016</td>
<td>2</td>
</tr>
<tr>
<td>9 Big traffic data processing framework for intelligent monitoring and recording systems [14]</td>
<td>2016</td>
<td>2</td>
</tr>
<tr>
<td>10 On Traffic-Aware Partition and Aggregation in MapReduce for Big Data Applications [15]</td>
<td>2016</td>
<td>2</td>
</tr>
<tr>
<td>12 Towards cloud based big data analytics for smart future cities [17]</td>
<td>2015</td>
<td>17</td>
</tr>
<tr>
<td>13 Big Data applications in real-time traffic operation and safety monitoring and improvement on urban expressways [18]</td>
<td>2015</td>
<td>14</td>
</tr>
<tr>
<td>14 Applications of big data to smart cities [19]</td>
<td>2015</td>
<td>14</td>
</tr>
<tr>
<td>15 Traffic zone division based on big data from mobile phone base stations [20]</td>
<td>2015</td>
<td>11</td>
</tr>
<tr>
<td>16 Detecting anomalies from big network traffic data using an adaptive detection approach [21]</td>
<td>2015</td>
<td>11</td>
</tr>
<tr>
<td>18 Mining the Situation: Spatiotemporal Traffic Prediction With Big Data [23]</td>
<td>2015</td>
<td>9</td>
</tr>
<tr>
<td>19 Big data for smart cities with KNIME a real experience in the SmartSantander testbed [24]</td>
<td>2015</td>
<td>9</td>
</tr>
<tr>
<td>20 Dazzled by data: Big Data, the census and urban geography [25]</td>
<td>2015</td>
<td>9</td>
</tr>
<tr>
<td>21 The real-time city? Big data and smart urbanism [26]</td>
<td>2014</td>
<td>269</td>
</tr>
<tr>
<td>22 A big data urban growth simulation at a national scale: Configuring the GIS and neural network based Land Transformation Model to run in a High Performance Computing (HPC) environment [27]</td>
<td>2014</td>
<td>62</td>
</tr>
<tr>
<td>23 Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop [28]</td>
<td>2014</td>
<td>38</td>
</tr>
<tr>
<td>24 The Uses of Big Data in Cities [29]</td>
<td>2014</td>
<td>26</td>
</tr>
<tr>
<td>26 The digital skin of cities: urban theory and research in the age of the sensored and metered city, ubiquitous computing and big data [31]</td>
<td>2014</td>
<td>24</td>
</tr>
<tr>
<td>27 From self-tracking to smart urban infrastructures: Towards an interdisciplinary research agenda on Big Data [32]</td>
<td>2014</td>
<td>23</td>
</tr>
<tr>
<td>28 Large-scale network traffic monitoring with DBStream, a system for rolling big data analysis [33]</td>
<td>2014</td>
<td>20</td>
</tr>
<tr>
<td>29 Visualizing big network traffic data using frequent pattern mining and hypergraphs [34]</td>
<td>2014</td>
<td>14</td>
</tr>
<tr>
<td>31 U-Air: when urban air quality inference meets big data [36]</td>
<td>2013</td>
<td>159</td>
</tr>
<tr>
<td>32 Big data, smart cities and city planning [37]</td>
<td>2013</td>
<td>96</td>
</tr>
<tr>
<td>33 Bootstrapping Smart Cities through a Self-Sustainable Model Based on Big Data Flows [38]</td>
<td>2013</td>
<td>66</td>
</tr>
<tr>
<td>34 Cloud Based Big Data Analytics for Smart Future Cities [39]</td>
<td>2013</td>
<td>29</td>
</tr>
<tr>
<td>35 RTIC-C: A Big Data System for Massive Traffic Information Mining [40]</td>
<td>2013</td>
<td>16</td>
</tr>
<tr>
<td>36 Big Data Processing and Mining for Next Generation Intelligent Transportation Systems [41]</td>
<td>2013</td>
<td>9</td>
</tr>
<tr>
<td>37 Study on Big Data Center Traffic Management Based on the Separation of Large-Scale Data Stream [42]</td>
<td>2013</td>
<td>9</td>
</tr>
<tr>
<td>38 Smart cities will need big data [43]</td>
<td>2013</td>
<td>5</td>
</tr>
<tr>
<td>39 An efficient transportation architecture for big data movement [44]</td>
<td>2013</td>
<td>4</td>
</tr>
<tr>
<td>40 Smart Cities, Urban Sensing and Big Data: Mining Geo-location in Social Networks [45]</td>
<td>2013</td>
<td>1</td>
</tr>
</tbody>
</table>
3. **Removing stop words**: These words, such as articles (e.g. a, an, the), conjunctions (e.g. and, or, for, so, etc), common verbs (e.g. is, have, has, do, does, etc), and qualifiers (e.g. may, often, some, etc) are the most frequent words in English with little semantic meaning.

4. **Stemming**: We used this technique to reduce the related words to their common roots, such as using *analyze* for *analysis*, *analyzing*, and *analyzed*.

Next, we performed the **frequency analysis** on the resulting text files and extracted the most frequent words of the identified papers in **STEP 1**. Figure 2-2 shows these frequently used terms.

**STEP3- Data collection from Q&A sites**

In this step, to collect questions and answers related to all aspects of TE we crawled 11 websites in SEDE. In order to select these 11 websites generally related to TE, first we found 30 websites in SEDE based on the relevency of the name and purpose (briefly explained in each site), which we recognized were most likely related to TE. After certain general queries, 19 sites were removed
because they include less than 10 questions related to TE (e.g. User Experiences, Hardware recommendations, Code review, Earth Science, Academia, Travel Answers, Web Applications, Electrical Engineering, Computational Science). The remaining relevant 11 websites were used for data collection and analysis include:

1. Ask Patents 4
2. Computer Science 5
3. Data Science 6
4. Database Administrators 7
5. Engineering 8
6. GIS 9
7. Open Data 10
8. Programmers 11
9. Software Recommendation Stack Exchange 12
10. Stack Overflow 13
11. Statistical Analysis 14

4http://patents.stackexchange.com/
5http://cs.stackexchange.com/
6http://datascience.stackexchange.com/
7http://dba.stackexchange.com/
8http://engineering.stackexchange.com/
9http://gis.stackexchange.com/
10http://opendata.stackexchange.com/
11http://programmers.stackexchange.com/
12http://softwarerecs.stackexchange.com/
13http://stackoverflow.com/
14http://stats.stackexchange.com/
Using the SEDE, we collected a set of questions that practitioners asked about various aspects of TE. Given that our goal in this experiment is to identify the main question topics surrounding TE, we included all of the answered and unanswered questions in our study. We conducted 3 iterations of querying the network, in order to refine our query, and to obtain our final data set for analysis. To collect data relevant to TE, our logic for the queries was to find posts containing the word “traffic” or “transport” and at least one other word from a set of keywords related to TE. These words were acquired from our text mining analysis in the previous step. If the query found a relevant post, it would return the entire discussion by collecting every post in that thread.

A post can be either a question or an answer, and certain properties are exclusive to questions, such as Titles and Tags. Among 11 websites in the SEDE, Stack Overflow’s database size is far larger than the others; it was too large to impose our query’s logic on the body of all posts. In order to prevent the query from timing out, we had to filter Stack Overflow by the Title or Tags containing our main criteria “traffic” or “transport”, and then the body of the question containing one of the keywords. This meant we collected threads where only the question was found to be relevant. For the other sites, it was feasible to look for “traffic” plus a keyword in the body of a post, which let us examine relevant answers as well. To crawl Stack Overflow the set of keywords for this first iteration is indicated in Table 2.2.

By manually analysing 1,000 posts from iteration #1, we discovered there were a great deal of irrelevant results collected. “Traffic” was a common word used in general discussion, specifically in Computer Engineering; for instance, “traffic data” and “traffic analysis” are common between Transportation Engineering and Computer Engineering. As a result, topics not related to Transportation Engineering were also returned despite being irrelevant. For the next iteration we removed the words “smart”, “city”, and “cities” from the query, because they had too many common usages. We also removed “social”, “media”, “mobile”, “cloud”, “infrastructure”, and “knowledge”, which were most commonly used in Computer Engineering related questions. We also added words that were used in relevant posts, in order to collect them in the absence of the broad terms we removed, rather than TE. On the other hand, we added next group of frequently used words, including “bus”,

29
Table 2.2: Search strings extracted by frequency analysis.

<table>
<thead>
<tr>
<th>Iteration #</th>
<th>Search strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Iteration</td>
<td>smart, urban, city, cities, analysis, comput, social, mobile, cloud, infrastructure, public, knowledge, vehicle, car, road, data</td>
</tr>
<tr>
<td>Second Iteration</td>
<td>flow, urban, volume, visualiz, environ, street, highway, analysis, public, vehicle, car, road, google map, gis, simulat, bus, data, accident</td>
</tr>
<tr>
<td>Third Iteration</td>
<td>simulat, urban, street, highway, analysis, public, vehicle, car, road, data, google map, gis</td>
</tr>
</tbody>
</table>

“simulat”, “flow”, “visualiz”, “environ”, “street”, “highway”, “google map”, “gis”, “accident”, and “volume” to our query’s set of keywords. Our second iteration reduced; however, the data collected was more relevant. We manually analysed 500 random results from the second iteration and found that they were generally related to TE, which - despite being an improvement - remained too broad.

Our keyword set contained 18 words, so we further refined it to include only words more exclusive to TE. Our keyword set for the final iteration is indicated in Table II. In this iteration, “road” intentionally has a space before and after it because we found that road was used in combination with other words, to convey meanings other than road, such as roadmap and broadcasting. We also removed “flow”, “volume”, “visualiz”, “environ”, “bus”, and “accident” because they retrieved a great deal of posts unrelated to TE. We used the results of the third iteration as the main data set of our analysis. A summary of the search terms related to each iteration is presented in Table 2.2.
Table 2.3: Topic Modeling Results (K represents the number of topics)

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>k=3</strong></td>
<td><strong>k=4</strong></td>
<td><strong>k=5</strong></td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td><strong>Map</strong></td>
<td><strong>Map</strong></td>
</tr>
<tr>
<td><strong>Vehicle</strong></td>
<td><strong>Road</strong></td>
<td><strong>Vehicle</strong></td>
</tr>
<tr>
<td><strong>Road</strong></td>
<td><strong>Data</strong></td>
<td><strong>Data</strong></td>
</tr>
<tr>
<td><strong>City</strong></td>
<td><strong>Time</strong></td>
<td><strong>Time</strong></td>
</tr>
<tr>
<td><strong>Traffic</strong></td>
<td><strong>City</strong></td>
<td><strong>City</strong></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td><strong>signal</strong></td>
<td><strong>Network</strong></td>
</tr>
<tr>
<td><strong>Area</strong></td>
<td><strong>Time</strong></td>
<td><strong>Distance</strong></td>
</tr>
<tr>
<td><strong>Map</strong></td>
<td><strong>Map</strong></td>
<td><strong>Street</strong></td>
</tr>
<tr>
<td><strong>URBAN</strong></td>
<td><strong>Highway</strong></td>
<td><strong>Highway</strong></td>
</tr>
<tr>
<td><strong>Signal</strong></td>
<td><strong>Traffic</strong></td>
<td><strong>Time</strong></td>
</tr>
<tr>
<td><strong>Route</strong></td>
<td><strong>Model</strong></td>
<td><strong>Area</strong></td>
</tr>
<tr>
<td><strong>Information</strong></td>
<td><strong>Traffic</strong></td>
<td><strong>Time</strong></td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td><strong>Location</strong></td>
<td><strong>Information</strong></td>
</tr>
<tr>
<td><strong>Segment</strong></td>
<td><strong>Fuel</strong></td>
<td><strong>Segment</strong></td>
</tr>
<tr>
<td><strong>Transportation</strong></td>
<td><strong>Model</strong></td>
<td><strong>Signal</strong></td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td><strong>Street</strong></td>
<td><strong>Location</strong></td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td><strong>Large</strong></td>
<td><strong>Map</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>k=6</strong></td>
<td><strong>k=7</strong></td>
</tr>
<tr>
<td><strong>Topic#1</strong></td>
<td><strong>Topic#1</strong></td>
</tr>
<tr>
<td><strong>Vehicle</strong></td>
<td><strong>Traffic</strong></td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td><strong>Map</strong></td>
</tr>
<tr>
<td><strong>Signal</strong></td>
<td><strong>Road</strong></td>
</tr>
<tr>
<td><strong>Map</strong></td>
<td><strong>Traffic</strong></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td><strong>Time</strong></td>
</tr>
<tr>
<td><strong>Highway</strong></td>
<td><strong>Area</strong></td>
</tr>
<tr>
<td><strong>Traffic</strong></td>
<td><strong>Speed</strong></td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td><strong>Distance</strong></td>
</tr>
<tr>
<td><strong>Segment</strong></td>
<td><strong>Model</strong></td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td><strong>User</strong></td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td><strong>Street</strong></td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td><strong>API</strong></td>
</tr>
<tr>
<td><strong>segment</strong></td>
<td><strong>Traffic</strong></td>
</tr>
<tr>
<td><strong>Map</strong></td>
<td><strong>Qgis</strong></td>
</tr>
<tr>
<td><strong>Google map</strong></td>
<td><strong>Source</strong></td>
</tr>
<tr>
<td><strong>Street</strong></td>
<td><strong>Map</strong></td>
</tr>
<tr>
<td><strong>Transportation</strong></td>
<td><strong>Current</strong></td>
</tr>
<tr>
<td><strong>Path</strong></td>
<td><strong>Source</strong></td>
</tr>
<tr>
<td><strong>Street</strong></td>
<td><strong>map</strong></td>
</tr>
<tr>
<td><strong>Accident</strong></td>
<td></td>
</tr>
</tbody>
</table>
We used the keywords obtained in the third iteration to crawl other 10 sites, in the body of the answers, in addition to the title.

2.2.3 Data Preparation

The data returned from Stack Exchange was not immediately ready for analysis. Stack Exchange returned the data from each query in CSV format and with HTML syntax. We had to convert each row into a text file in order to prepare the corpus for our algorithms. This copied the title of each column into each text file, and there were a number of other formatting problems which remained from the CSV file. We also approached our data cleaning process in an iterative manner. The first time we ran and analyzed our topic modelling and word frequencies, we did minimal cleaning. Each step performed in the third and final iteration is described below as well as its evolution throughout our iterations.

**Step 1** - Convert alphabetical text to lowercase: We performed this step for all three iterations. We removed case sensitivity to ensure we did not analyze the capitalization of a word as a separate case than its lower-case counterpart.

**Step 2** - Remove HTML tags: This step needed to be performed before punctuation was removed because angled brackets were indicators of HTML tags. This was noticed after the first iteration. For the second and final iterations, we removed everything between angled brackets. This removed the Tags from question posts as well.

**Step 3** - Manual transformation: We replaced words that were synonymous with each other, because they would appear too scattered, due to their independence. Some of these words had to be united before the corpus was stemmed. For the second iteration, for example, “gas” and “gasoline” was replaced with “fue” because they would refer to “fuel”. Other examples are replacing “datastores”, “dbstore”, and “store” by “storage”, or replacing “visualize”, “visualise”, “visualizations”, “visualisation”, and “visualisations” by “visualization”.

32
**Step 4** - Removing numbers and punctuations: We performed this step for all three iterations. This joined hyphenated words together, rather than separate them. For example, “google-map” became “googlemap”, and “map-reduce” became “mapreduce”.

**Step 5** - Removing stopwords: Stopwords are common words that provide no meaning on their own, such as “the”. Here we used the default set of stopwords in the tm_map package for R.

**Step 6** - Strip Whitespace: We removed excessive whitespace such as newlines, double spaces and tabs.

**Step 7** - Stemming: We performed this step for all 3 iterations. Stemming is the process of reducing words to their origins by removing suffixes. For example, “environmental”, “environment” and “environmentally” would all become “environ”.

**Step 8** - More manual transformation: In the second iteration, “junction” were replaced with “intersection”; “pollution” was replaced with “environ”.

**Step 9** - Remove additional words: For the second iteration we deleted an additional 86 words. For the final iteration, our additional set contained 142 removed words, most of which were common and provided no meaning in a TE context.

### 2.3 Data Analysis

Topic modeling is an unsupervised text analysis technique to summarize a large volume of unlabelled text with a smaller number of distributions over words [47, 48]. These hidden distributions are called *topics*. In this step, we used Latent Dirichlet Allocation (LDA), a well-known topic modeling algorithm, to analyze the content of the 2,457 posts retrieved in the previous step, and produced five main topics. A topic for the LDA approach is a probability distribution over a vocabulary [48]. To this end, we used the *topicmodels* package in R. To implement the topic modeling approach, we used the Gibbs sampling option, as it is more accurate than the variational algorithm [49]. The details of the algorithm we used to explore the most frequently discussed topics related to “Big Data” in TE is represented in Algorithm 1. In this algorithm, we define the requisite variables,
assign them a random value, and then run a loop for the desired number of times. In each loop, each word instance of our corpus is assigned to a sample topic. We defined the initial value of the variables used in this algorithm as follows:

- $n_{d,k}$ (the number of words assigned to topic $k$) = 6
- $k$ (number of topics) = (3..7)
- $\text{iteration} = 2000$
- $n_{\text{start}}$ (defining different starting points for the sampling purpose) = 5
- As we started the algorithm randomly, it is necessary to discard the initial iterations. To this end, we discarded the first 4,000 iterations, which is referred to as burn-in period.
- $\text{thin} = 500$. This parameter is used to reduce the correlation between samples (e.g. over 2,000 iterations, we took every $500^{th}$ iteration for further analysis)

In this algorithm, $z$ represents the topic assignment for each of the $N$ words $w$ in our data set.

2.4 Evaluation

The results of running the LDA algorithm for various values of $k$ are illustrated in Tables 2.3 (a-e). There is no gold standard list of topics for technology-related transportation engineering questions on Q&A sites to compare the results of our analysis with. So, in an attempt to judge the merit of our explored topic models, we used word intrusion and topic intrusion, two quantitative evaluation techniques proposed by Chang et al. [51]. In the remainder of this section, we elaborate on the details of each of these techniques, and report on the results of our evaluation.
**Input:** words $w \in$ documents $d$, nstart, burn-in, thin, $n_{d,k}$  
**Output:** topic assignments $z$

```
begin
  randomly initialize $z$ and increment counters
  foreach iteration do
    foreach word $w$ do
      foreach topic $k$ do
        $\theta_{d,w,k} =$ calculating the document/topic distribution for topic $k$, word $w$ in document $d$
      end
      topic $\leftarrow$ sample from $\text{multnomial}(\theta_{d,w})$
      $z[w] \leftarrow$ topic
      update counts according to new assignments
    end
  end
return $z$
end
```

**Algorithm 1:** LDA Gibbs Algorithm [50] for exploring the most frequently used topics related to “Transportation Engineering”

---

**Word intrusion**

This method measures the quality of the inferred topics by calculating their “cohesiveness”. To measure the coherence of the explored topics in our study, we followed the steps proposed by Chang et al. [51] as follows:

**Step 1:** As illustrated in Table 2.3 (a-e), we selected and represented the six most probable words from each topic. In this step, we added another random word from a list of words with low probability in the current topic (this list is one of the outputs of the LDA algorithm). Then, we randomly ordered the seven words of each topic and asked participants to select the word in each topic that does not fit with the others. As we selected the new word from the list of less probable words, we reduced the probability that this intruder semantically belongs to the other words of each topic.

**Step 2:** Next, we measured the *model precision* for different values of $k$ by calculating how well they match human concepts. For this purpose, we used the following equation proposed in [51]:

---

35
\[ MP_k = \left( \sum_s 1(i_{k,s} = \omega_k) \right) / S \] (2.1)

Where \( MP_k \) represents the model precision of our model for a specific value of \( k \). Further, \( \omega_k \) denotes the index of the intruding word for topic \( k \), \( i_{k,s} \) represents the intruder selected by \( s^{th} \) subject from a set of words listed in topic \( k \), and \( S \) denotes the number of subjects.

**Topic intrusion**

This method measures how the inferred topics that are assigned to a corpus of documents agree with human judgments about the content of the documents.

**Step 1:** In this step, we first selected the three topics with the highest probability that were assigned to each document. Then, we added an *intruder topic* to this list, which is selected randomly from the list of low-probability topics. Next, for each document, we asked participants to choose a topic that is not related to the document. The judgment was made based on the title and a snippet from each document (or question).

**Step 2:** In order to measure how well our explored topic models assigned the topics to the 2,457 extracted documents, we used the *Topic Log Odds (TLO)* [51] parameter, which quantifies the agreement between the model and human judgment. For this purpose, we used equation 2.2:

\[ TLO_d = \left( \sum_s (\log \hat{\theta}_{d,j_{d,s}} - \log \hat{\theta}_{d,j_{d,s}}) \right) / S \] (2.2)

Where \( \hat{\theta}_d \) represents the probability that document \( d \) belongs to each topic, \( j_{d,s} \in \{1...k\} \) represents the intruding topic detected by participant \( s \) for document \( d \), and \( j_{d,s} \) denotes the *true* intruder. We asked 26 undergraduate students to evaluate the precision and relevance of the explored topics of our study. Each participant conducted a total of 13 or 14 tasks, with 8 or 9 tasks completed for word intrusion and 5 for topic intrusion. For word intrusion, the topics were split up into three groups of 8:
Figure 2-3: Evaluation of the precision and relevance of the topic modeling results

- Group 1: 3 topics from k=3, 4 topics from k=4, 1 topic from k=5
- Group 2: 4 topics from k=5, 4 topics from k=6
Table 2.4: Qualitative and quantitative information about the projects included in our study. WI and TI indicate word intrusion and topic intrusion.

<table>
<thead>
<tr>
<th>K</th>
<th>#participants (WI)</th>
<th>#Documents</th>
<th>#participants (TI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>26</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>41</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>68</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>total</td>
<td>242</td>
<td>130</td>
<td>130</td>
</tr>
</tbody>
</table>

- Group 3: 2 topics from k=5, 6 topics from k=7

Table 2.4 lists the statistical details of the participants and their assigned tasks. Figure 2-3 represents Boxplots of the precision for various values of K (i.e. 3, 4, 5, 6, and 7). The figures illustrate that with a larger number of topics, their meaning becomes less coherent to the reader. This may be because the topics become too niche to be consistently associated with a large set of documents. With this in mind, we have concluded that having 5 topics is most appropriate for further analysis, because the intruder is more consistently identified than in other values of K. This implies that these 5 topics are the most coherent and applicable topics with regards to our data.

2.5 Results and Findings

2.5.1 RQ1. The main categories of discussion topics about TE among software practitioners

Following the results of topic and word intrusion analysis, topic models when k=5 have been selected as the most cohesive topics. These topics will be discussed further in this section.

*Topic 1- map, road, distance, highway, API, Google Map:* We determined Topic 1 to be related to Transportation and Geospatial Analysis. However, the words listed in this topic are not
fully supporting the area of *geospatial data analysis*. Instead, the most common geospatial analysis
task in the area of *transportation engineering* is **distance measurement**, which is addressed by
this topic.

**Finding 1.1:** One significant topic of interest among transportation engineers and developers
is to better understand the ways to customize and apply existing GIS tools and technologies
(e.g. Google Map) to perform geospatial data analysis. These include simulating highways
and roads on a map, measuring the distance, and various values of data points (i.e. latitude,
longitude).

**Topic 2- vehicle, model, speed, time, fuel, resource:** This topic represents questions and answers
which directly addressed the area of transportation planning and modeling with the aim of reducing
travel times (i.e. vehicle, speed, time), and fuel consumption (i.e. vehicle, speed, fuel, resource).
By looking at a randomly selected set of questions, and the answers assigned to this category of
topics, we found that most of the questions which asked for input related to “fuel” keyword, were
primarily about very technical details of fuel consumption and its optimization.

**Finding 1.2:** A large portion of the questions that directly asked for input from the Q&A
community were primarily about environmental aspects of *transportation engineering*, such as
transportation planning for optimizing the travel time, and fuel consumption.

**Topic 3- data, city, street, information, software, SQL:** This topic encompasses keywords related
to **data analysis**. More specifically, this topic describes Q&As that elaborated on analyzing data
sets related to cities, and their traffic and transportation. This is illustrated by keywords such as
*SQL*, *data*, and *city*. 

39
Finding 1.3: A common desire among developers and practitioners is to collect and analyze traffic data sets and produce information applicable for cities and their traffic.

Topic 4- road, network, area, segment, QGIS, Urban: This topic is most directly related to road transportation management, and the application of new technologies, such as QGIS. “QGIS” (previously known as Quantum GIS) is a cross-platform free and open-source desktop geographic information system (GIS) application that provides data viewing, editing, and analysis. Keeping in mind that QGIS provides data analysis, we also found this topic to be related to data analysis aspects of road transportation management.

Finding 1.4: One significant topic of interest among practitioners who work in the area of transportation engineering is planning urban roads, as well as the application of GIS tools (e.g. QGIS and ArcGIS) to alleviate the complexity of this task.

Topic 5- traffic, vehicle, time, signal, location, gps: We found this topic is significantly related to signal timing and its role in managing urban traffic. In addition, GPS, location, and vehicle are keywords, which helped us to relate this topic to signal timing optimization as well. This is illustrated by the application of GPS technology for tracking the location and speed of vehicles.

Finding 1.5: Most of the practitioners found the application of GPS technology the most relevant and useful technique for managing and optimizing traffic signal timing.

15http://www.qgis.org/en/site/
2.5.2 RQ2. Implications for Researchers and Practitioners

With regards to implications of this study for researchers and practitioners, in this section, we elaborate on the statistical findings of this study as well as the implications that arise from these findings.

As illustrated in Figure 2-4, more than 50% of the questions were related to planning urban roads, as well as the application of GIS tools (e.g. QGIS and ArcGIS) to alleviate the complexity of this task (topic #4). This is followed by analysis of transportation data sets with 24.2% of questions and answers (topic #3). Questions related to environmental aspects of transportation management (topic #2), the application of GIS tools (topic #1), and Managing traffic signal timing and the application of GPS technology (topic #5) have been addressed by 17.8%, 16.6%, and 15% of the questions and answers, respectively.

![Figure 2-4: Percent of total questions assigned to each topic](image)

**Finding 2.1:** The application of GIS tools significantly outweighed the others in popularity. These were primarily comprised of questions relating to QGIS, and ArcGIS tools. This implies the usefulness and popularity of these tools in practical road planning and transportation engineering, which can be applied in theoretical urban planning research.

**Finding 2.2:** There are less questions among transportation engineers and developers that are related to big data analysis such as data storage, computation, and visualization. This
implies a clear need for more collaboration among researchers, data analysts, and transportation engineers (or practitioners).

2.6 Threats to Validity

Two main threats to the validity of our results are as follows:

A major threat to the validity of our results is that to keep the searching process unbiased, we used frequency analysis as an automated method to select search terms for Topic Modelling. However, in using this method we may have selected search terms that resulted in loss of posts on Q&A sites.

Many of the questions on the Stack Exchange sites are considered “closed” by the communities. That is to say, they are part of the archive of the site, but may no longer be interacted with by the community. However, there was no way to mitigate the selection of these posts in SEDE, so we included them in our study. This calls into question the fairness of evaluation of social Q&A sites, and whether the discussion archived in a “closed” post can be considered relevant to contemporary TE issues.

Additionally, the authors of posts on StackExchange may or may not be professional practitioners of TE. While the large quantity of anecdotal posts that discuss TE issues suggests that the majority of posts are written by such practitioners, StackExchange is an open community where people of any background may post recommendations. This affects our study because any number of questions may be asked by students and beginners, rather than practitioners. Once again, we found no way to mitigate this without some method of manual selection.

This leads to the final point: we have chosen not to go through our corpus manually to best keep the effect of our Topic Modelling process. However, this means that posts that were only tangentially related to TE were allowed into data set, which may have inflicted a bias on our final results. On a more extreme level, some posts may not have been related to TE at all, but may
have entered our corpus from use of keywords during the Data Collection process. In this case, we effectively reduced the irrelevance of the posts selected through the iterative collecting process, which narrowed the scope of our data significantly.

2.7 Discussion and Conclusion

In this section, we conclude the main findings of our study by addressing the research questions of this chapter as follows:

- One significant topic of interest among transportation engineers and developers is to better understand the ways they can customize and apply the existing GIS tools and technologies (e.g. Google Map), in order to perform geospatial data analysis (including: simulating highways and roads on a map and measuring the distance various values of data points (i.e. Latitude, longitude)).

- A large portion of the questions that directly asked for input from the Q&A community were primarily focused on environmental aspects of transportation engineering, such as transportation planning for optimizing the travel time, and fuel consumption.

- A common desire among developers and practitioners is to collect and analyze the traffic data sets, and produce information applicable for cities and their traffic.

- One significant topic of interest among practitioners who work in the area of transportation engineering is planning urban roads, as well as the application of GIS tools (e.g. QGIS and ArcGIS) to alleviate the complexity of this task.

- Most practitioners found the application of GPS technology the most relevant and useful technique for managing and optimizing traffic signal timing.

- The application of GIS tools such as QGIS, and ArcGIS topic significantly outweighed the others in popularity. This implies the usefulness and popularity of these tools in practical road
planning and *transportation engineering*, which can be applied in theoretical urban planning research.

- There are fewer questions among transportation engineers and developers related to **big data analysis** such as data storage, computation, and visualization. This implies a clear need for more collaboration among researchers, data analysts, and transportation engineers (or practitioners).


Abstract

The area of Traffic Management (TM) is characterized by uncertainty, complexity, and imprecision. The complexity of software systems in the TM domain which contributes to a more challenging Requirements Engineering (RE) job mainly stems from the diversity of stakeholders and complexity of requirements elicitation in this domain. This work brings an interactive solution for exploring functional and non-functional requirements of software-reliant systems in the area of traffic management. We prototyped the RETTA tool which leverages the wisdom of the crowd and combines it with machine learning approaches such as Natural Language Processing and Naïve Bayes to help with the requirements elicitation and classification task in the TM domain. This bridges the gap among stakeholders from both areas of software development and transportation engineering. The RETTA prototype is mainly designed for requirements engineers and software developers in the area of TM and can be used on Android-based devices.

3.1 Introduction

Developing software-reliant systems for complex domains such as the Traffic Management (TM) domain requires a more challenging RE job. While, in recent decades, a wide range of requirements elicitation techniques has been developed to address complex systems, these techniques usually aim to work in a context-free spectrum and thus are not concerned about the complexity of a particular domain, such as the TM domain. In this domain, we deal with many constraints to elicit requirements, such as the diversity of stakeholders, a clear need for time-centric elicitation techniques, the legal issue for some types of elicitation techniques, the variability of the transportation demand,
and knowledge of the road network and the traffic conditions. Moreover, some of the software tools in this domain are critical because they involve people’s life (e.g. emergency control systems) and are very sensitive to mistakes.

To address these issues and following our recent study on exploring transportation engineers’ concerns on social Q&A websites [1], we introduce the RETTA Tool, which aims to tackle problems associated with eliciting requirements for TM services such as Emergency Medical Services (EMS), Traffic Signal Timing (TST), and Urban Transportation Planning (UTP) systems. Given that in most of the software tools in the TM domain, the main stakeholders are the crowd participants in a traffic network, who use TM services and at the same time define the requirements of these services, we can rely on them to cater for their needs properly. Thus we apply the *crowdsourcing* [2] technique (e.g. social networks mining) as one of the requirements elicitation techniques to prototype the RETTA Tool. Moreover, we used various quantitative and qualitative data analysis and machine learning techniques to explore TM services requirements from various sources of traffic data, such as real-time camera and drone data, signal light sensors’ data, and historical traffic data.

### 3.2 RETTA Prototype

The RETTA tool prototype is designed based on the lessons learned in more than five years working with transportation engineers and traffic planners. It is designed and developed using Android platform for mobile phones and tablets. Following the complexity of the computation and data analysis in the TM domain, we designed the RETTA tool in a way to be easy to develop. Thus, to design and develop the tool, we used the Model View ViewModel (MVVM) [3] architecture and wrapped all of the technical details and analytical computations (e.g. Shockwave model [4, 5] and machine learning algorithms) to the Model layer. Moreover, this technology reduces the coupling between the ModelView (i.e. the Controller in the Model View Controller Architecture) and the View layers, which simplifies developing tools with a high level of data analysis. Figure 3-1 illustrates the dynamic flow and the sequence of activities of the RETTA prototype.
Figure 3-1: UML activity diagram of the core process of RETTA

Figure 3-2: Detailed visual representation of classifying NFRs
Moreover, Figure 3-2 presents the main screens of the RETTA tool and a summarized interaction flow for identifying functional and non-functional requirements of a sample traffic management system. Once the main screen containing traffic management services such as the EMS, UTP, and TST are loaded, the user can choose the target service for which they intend to develop a software-reliant system. While the services listed in Figure 3-2 (a) all are tightly related to the TM domain, they differ in terms of the crowd that characterizes them and the data types that can be used to elicit their requirements. Figure 3-2 (b) shows the situation where the user aims to elicit the requirements of the TST service. Following the theoretical logics and technical parameters required for each service, there are specific data sources associated with each service. We summarize the main features of the RETTA tool by describing an elicitaton scenario for eliciting the TST system requirements as follows:

TST is a TM system with the goal of optimizing signal timing in a traffic network. Once a user initializes the tool by defining the geographical location in which the TM software will be deployed (Activity 1, Figure 3-1), the tool will present a list of services (Figure 3-2 (a)) that are qualified for implementing the requirements elicitation process (Activities 2-3, Figure 3-1). In situations where the tool cannot collect adequate information for exploring the requirements of a specific service, that service will not be offered to the user. As illustrated in Figure 3-2 (b), the user selects the TST service (Activity 4, Figure 3-1). Following this action, the RETTA tool lists available data sources which can be identified based on the geographical location of the user or the initial settings of the tool (Activities 5, Figure 3-1). For instance, in an area, in which the Traffic Signal Sensor data is not available, the tool does not list this data source. To elicit the requirements for the TST service, the following sources can be used to gather traffic data: (1) real time traffic data including the data from traffic cameras, drones, signal sensors, connected vehicles, and cellphones (e.g., via the Cities, TomTom, GoogleMap), (2) historical traffic data (e.g. via NGSIM, the Cities and transportation agencies), and (3) social networks (i.e. Twitter). The requirements engineer can select any of these data sources depending on the type of the service or the context for which they are developing a software. Given the context-specific nature of TM services, in this stage, the tool asks the user to
characterize the specific features of the context under development (Activity 6, Figure 3-1). Figure 3-2 (c) represents a sample screen of characterizing Twitter data collection. The details of this screen vary by the type of the data source selected by the user. For instance, the tool asks for a specific geographical area when a user selects the camera data source. Once all the required data for conducting the requirements elicitation task are collected and specified, the RETTA tool deploys various domain specific data analysis techniques (e.g. spatial-temporal traffic data analysis), and machine learning methods such as NLP, topic modelling (i.e. Latent Dirichlet Allocation (LDA)), association rule mining and Naïve Bayes algorithms to elicit and classify TM services’ requirements (Activity 7, Figure 3-1 and Figure 3-2 (d-e)). Given that due to the sparsity of word co-occurrence patterns in short texts such as Twitter and microblogs [6], the RETTA tool combines all of the retrieved tweets related to a specific service and then applies the LDA algorithm. The main data pre-processing steps such as removing English stop-words, numbers, HTML tags, and stemming process will be applied before applying the topic modeling and NLP approaches on the retrieved tweets. Moreover, to classify the requirements, we defined some domain specific regular expressions [7] and increased the weighting of valuable words for each TM service. For instance, malfunction, signal, light, traffic, accident are the keywords that can characterize the reliability of TST service.

3.3 Conclusion and Future Work

The RETTA tool provides an interactive environment for eliciting requirements in the traffic management domain. A short informal evaluation of the tool has been carried out in Intelligent Software Systems Laboratory at the University of Calgary. Software developers found the RETTA remarkably easy to use and very thought provoking for eliciting traffic domain requirements. Future work will concentrate on improving the efficiency and completing the text analysis and classification approaches for exploring and classifying requirements. Moreover, we aim to improve the applied requirements elicitation process in the RETTA tool to address the complexity and the scale of the crowd and to ensure that we record their requirements efficiently and precisely.


The Efficacy of Using Social Media Data for Designing Traffic Management Systems

Abstract

It has long been acknowledged in the context of developing dynamic and reactive systems that users’ input during different stages of the development process helps to quickly and incrementally adapt to changes in the system’s context and users’ needs. Given the data- and communication-intensive nature of developing transportation management systems, utilizing social media data provides a new route for a dynamic collection of needs and experiences in a timely and direct fashion. In this chapter, we will explore how and to what extent social media data can support urban traffic management systems. To this end, we have conducted a mixed-method study including both manual qualitative analysis, and automatic information extraction using weighted finite-state transducers (WFST), natural language processing (NLP), and deep neural networks (DNN) on Twitter data. We utilize Canadian traffic information from Twitter to look for issues and relevant information that may assist authorities and software development teams in making decisions when designing and developing traffic management systems by leveraging lay people’s input. Data triangulation will also be used to help compare our results against other data sources such as Google Trends and scientific material. We found that the self-reported traffic information with lay users on Twitter can be a valuable source to characterize traffic management systems. Moreover, we found that although theory-based publications in the context of traffic management systems can help with traffic estimation, control, and prediction, they are insufficient to characterize the context-sensitive aspects of these systems.

4.1 Introduction

Traffic congestion is a growing worldwide problem and is worsening from the continuous increase in urban population and the number of cars across the world. Increased urbanization has impacted
traffic conditions worldwide, leading to delays and accidents because of commuters rushing to reach their destination. Half a billion cars were on the road in 1985, and this number doubled by 2010 [1]. It is expected to hit two billion in 2020 [2]. The cost of traffic in the world was estimated at $1 trillion in 2013 [3]. Even multi-billion dollar investments in new roads to increase road capacity are not a working solution to relieve congestion, especially when the new capacity is unpriced. This is because such developments can attract more road users, and the congestion may become as it was before the capacity addition [4]. Consequently, there has been an increase in extensive research on traffic trends to predict road traffic conditions for commuters and make their commute more comfortable and quick. Accordingly, the design and development of traffic management systems have been demanding and long-lasting multidisciplinary problems in various research areas, such as transportation engineering, software development, data mining, and machine learning.

Despite all the advances in traffic management, recent literature indicates that it is still a challenge to manage traffic at the network level, due to several factors such as the complexity of traffic phenomenon, difficulties in short-term traffic prediction, and insufficiency of coverage of data-gathering devices [5]. Over the years, researchers have proposed various analytical and theory-based traffic models and control methods (e.g. [6, 7, 8]) to tackle traffic control problems. All these published methods in the area of traffic engineering (which, in this chapter for brevity, are referred to as theory-based publications) require traffic data such as traffic arrival flow rates and queue lengths as input. The output, i.e. control decisions, of these methods may not be efficient without accurate and timely traffic data [9]. Traditionally, traffic data is collected by two categories of physical traffic sensors, including fixed sensors (e.g. inductive loop detectors, traffic cameras) and moving sensors (e.g. probe vehicles, cellphones, global position systems (GPSs)). Despite the relatively high reliability of these types of sensors, they have some limitations. Fixed sensors are costly to be installed in an urban network [10]. The drawback of moving sensors is their sparsity and limited penetration rate. Furthermore, because the data from these sensors are mostly noisy in urban arterials [11], it may lead to incorrect traffic congestion estimation and control.
With the ever-growing interest in social media in people’s daily lives, another type of sensor, called social traffic sensor, has joined traffic data acquisition sources. The advantages of this new type of sensor over traditional traffic sensors are its high volume, cost-effectiveness, and area coverage [12]. Social sensors build on new layers of knowledge about traffic behaviour by providing significant insights about the causes of non-recurrent and recurrent traffic conditions [13]. Moreover, people show a high propensity to post and share real-time traffic-related information through social media when facing traffic events (e.g. sports games), incidents (e.g. accidents, road closure), and traffic conditions (e.g. traffic jam). Unlike physical traffic sensors, individuals (i.e. pedestrians, drivers, and passengers) are treated as human (or social) sensors [14], which freely post their observations and provide an opportunity to extract useful information for traffic management. This potential has been widely studied by researchers in the area of traffic management, for instance, to explore transportation engineers’ concerns [15] and enhance traffic incident detection and awareness [16, 17] from social media.

Considering this cost-effective opportunity provided by end-users of traffic management systems, in this chapter, we aim to investigate how and to what extent social media data can help with the aim to explore the requirements of traffic management systems. In other words, we want to explore which aspects of these systems can/cannot be addressed and improved using social media data. To this end, we use manual qualitative analysis, automatic information extraction, and Natural Language Processing (NLP) to analyze social media data. Compared to the existing research on social media for traffic management, our proposed approach is distinct in three ways: (1) We use three different sources of data, including theory-based publications, Google Trends, and Twitter data; (2) We use a systematic search method to explore search strings for filtering data collected from social media, i.e. Biterm Topic Modelling (BTM), and Weighted Finite State Transducers (WFST), and (3) To address the problem from different perspectives, we analyze two different data types, i.e. text and image, using NLP, deep neural networks (DNN), and descriptive statistics.

The rest of this chapter is structured as follows: we begin by reviewing relevant studies on the application of social media mining in traffic management systems in Section 4.2. Section 4.3
overviews our research questions, followed by the data analysis methods conducted in this study. Section 4.4 presents the results of our data analysis and related findings. Threats to the validity and limitations of our study are discussed in Section 4.5. Finally, Section 8.5 provides some concluding remarks and directions for future work.

4.2 Related Work

Concerning the usefulness of social media data in traffic management in urban networks, some may believe that social media data such as Twitter data is redundant to traditional sensors data. In contrast, a study by Chen et al. [18] supports the hypothesis that Twitter data offers a suitable complementary source to traditional sensor data. Although traditional sensors can reasonably provide traffic data in various traffic conditions, they cannot reflect the real cause of traffic conditions. This fact led to proposing methods in integrating physical and social sensors for traffic management and signal control. Daly et al. [13] showed that this combination could highly improve the real-time perception of the reasons behind traffic conditions and the variation of the congestion level in the cities. Also, sharing of events via social media can complement the vehicle count data obtained from traffic sensors by suggesting new timing plans for traffic signals in the network to adapt to the event’s influence. In this regard, Ramadhan et al. [14] proposed a traffic signal control method that creates traffic signal configuration using social media data and the number of vehicles while considering vehicles entering or exiting a point of interest, such as sports games, on the road.

The literature on social media data mining mainly adapted to large-scale event detection with extensive spatial and temporal data, such as an earthquake [19]. However, detecting a traffic accident that impacts a small area and attracts a limited number of tweets is challenging [20]. To this end, employing the Support Vector Machines method, D’Andrea et al. [21] developed a near real-time detection system that detects traffic-related events from twitter stream analysis with high accuracy (more than 88%), often faster than online news websites and local newspapers. Wojtowicz et al. [22] assessed how social media is used to help with traffic management operations in special
events (e.g. concerts and sporting events) and disruptive events (e.g. accidents and weather events). Furthermore, Nallaperuma et al. [23] proposed a traffic management platform to capture dynamic patterns from traffic data streams that can successfully and in a timely manner detect recurrent and non-recurrent events.

It is more challenging when the objective is to monitor all the congested links in an urban network to detect traffic congestion at a network level. Chen et al. [18] proposed a topic modelling-based language model and a collaborative inference model to efficiently identify and monitor traffic congestion locations in two U.S. major cities through using Twitter and INRIX probe speed data sets. Gong et al. [24] developed a platform for Twitter data harvesting for traffic patterns used to extract information to identify near real-time potential traffic congestion. This work was implemented for three cities in Australia and suggested using Twitter as a free source of data for traffic congestion detection in a near real-time process.

Understanding of daily traffic conditions is also essential for urban planning and policy-making [25]. The potential influence of human mobility and activity patterns on daily traffic congestion is studied using social media data. It is found that social media data can provide a better understanding of traffic conditions [25]. Furthermore, as a requirements elicitation tool for traffic management systems, RETTA [26] leverages the knowledge of the crowd and combines it with machine learning approaches to help with the requirements elicitation and classification task in the transportation management domain. This bridges the gap among stakeholders from both areas of software development and transportation engineering.

Another application of social media data is in the context of traffic information prediction, particularly in short-term traffic flow prediction [27], traffic speed prediction [28], and traffic condition prediction [29]. For instance, He et al. [30] investigated the correlation between traffic volume and the number of tweets to forecast traffic flow for the next hour or beyond. The results revealed that their method provides better performance than an existing auto-regression based traffic flow prediction model.
Figure 4-1: Google trends for weather forecast, traffic, and traffic collision along five years (2014-2019).

The image data is also used for information extraction from social media. For instance, the spatiotemporal patterns of tourists’ accommodation are explored through a photo analysis on Geotagged Flickr photos [31].

By analyzing social media data, including text and image, in this chapter, we provide a systematic methodology to efficiently explore social media data in the context of traffic management and find out what can be understood from what the people post and share. We also explore how this understanding is similar to or different from theory-based publications.

4.3 Methods

In this Section, we begin by introducing the research questions followed by the data analysis process used in this chapter.
4.3.1 Research Questions

• **RQ1: Information type**- *What are the main traffic-related topics that people communicate the most on social media?*

• **RQ2: Information application**- *How does social media analysis contribute to explore the requirements of a traffic management system?*

4.3.2 Data Collection and Preparation

To answer our RQs, we selected Twitter as it is one of the largest social media platforms [32, 33] and largely coupled with our daily activities and social communications. The collected data included Tweets, spanned from December 1\textsuperscript{st}, 2018 to January 10\textsuperscript{th}, 2019, from the entire world. To explore the time interval for our data collection task, we used Google Trends to retrieve weekly data on the number of Google searches using weather forecast, traffic, and traffic collision, during the last five years (2014-2019), see Fig 4-1. Google Trends data tracks the popularity of specific search terms on Google in the form of popularity score and temporal trends. As illustrated in Fig 4-1, during the time interval of [December-January], people have more concern/interest in traffic conditions or information.

**Data Streaming**

Twitter is a micro-blogging platform that allows users to broadcast or receive short messages of 280 characters or fewer called 'tweets' to seek or share information. As the first step, to ensure we can scale up the scope of our analysis in later stages of our study, we used the Twitter Streaming API and collected a random 1% sample of public tweets [34], without applying any constraint on the content of the data being streamed. To implement our data collection process, we used the rtweet package of R (https://cran.r-project.org/web/packages/rtweet/rtweet.pdf). The collected data was properly formatted to JSON and stored in MongoDB on a local machine.
Search Term Selection

To explore unbiased content-carrying keywords and search strings, we used a context-sensitive lexical association technique, presented in [35]. We modelled our datasets (i.e. from technical and social media repositories) using Weighted Finite State Automata (WFST) [36] and statistical language models. A WFST is a Finite State Automaton (FSA) in which each transition is labelled with input data, output data, and a weight representing the cost of each transition. The lowest the weight, the highest the probability of a term’s occurrence in a specific context. Fig 4-2 illustrated the overall structure of a WFST generated for 60 traffic-related theory-based publications. As illustrated in this figure, the term ‘increase transportation system efficiency’ is one of the most probable expressions in this context. In the rest of this Section, we elaborate on the application of WFSTs extracting content-carrying terms from technical repositories (i.e. theory-based publications). These terms will be used as search strings for filtering out the collected twitter data based on their relevance to the traffic context.

Considering that the traditional keyword extraction methods such as TF-IDF apply basic document features (e.g. frequency and length of terms, length of document, and the availability of a specific term in a repository), relevant terms stay independent of other content-carrying terms in the
document which contributes to losing the context surrounding terms when measuring their contextual relevance. This is a weakness shared by all bag of words approaches. To address this problem and leverage the richness of the information generated by contextual relationships, in this step, we apply the lexical association technique on traffic-related theory-based publications. This approach quantitatively determines the strength of association between two or more words (or terms) based on their co(occurrence) in a dataset \[37, 38\] and will assign different weights to terms depending on the context in which they appear.

To select this approach, we followed the underlying assumption that “a context in which a word is used can often influence its meaning” \[39\]. Thus, in a dataset, the terms that are highly related to each other and occur together more often than expected by chance have a particular meaning and can be considered as relevant terms. On the contrary, the irrelevant (background or contextual stop-words) terms will have a very low association with the other terms in a corpus \[38\]. The dataset that we used to implement this approach contained papers in the area of traffic management systems published in the high-quality journals of Transportation Research Part A to Part F, with the much focus on addressing challenging traffic management problems.

**Data Pre-processing**

In this step, we used the search terms that were identified in the previous section and implemented the following pre-processing steps:

1. **Language and Location Filtering:** We excluded all non-English language Tweets and Tweets that originated from outside of Canada.

2. **Keyword Filtering:** Looking at Fig 4-2, most of the content-carrying terms, returned from theory-based publications, are very technical and might not be useful for filtering the language that people use on social media. Thus, in addition to the search string generated in Section 4.3.2, we used the following 20 predefined vocabularies obtained from domain experts and
similar works in the context of social media analysis for traffic engineering and control [18, 17, 40]:

| Keywords | crowded, pedestrian, traffic, accident, incident, driver, stuck, block, crash, lane, street, roadway, congestion, avenue, highway, station, construction, vehicle, mile, and delay. |

4.3.3 Data Analysis

To answer the RQs of this study, we used the following data analysis approaches:

**Biterm Topic Modeling (BTM)**

To answer RQ1, we use BTM, an unsupervised topic modelling approach designed for short text documents [41]. Given the short length of Tweets (i.e. ≤ 280 characters), conventional topic modelling approaches such as Latent Dirichlet Allocation [42] and Probabilistic Latent Semantic Analysis [43] that exploit word co-occurrence patterns at document-level, are highly sensitive to the length of each document and suffer from the severe data sparsity problem. Instead, BTM aggregates all the co-occurrence patterns in the entire corpus, and it is less dependent on the frequency of words in individual documents.

4.3.4 Image Analysis

To analyze the content of collected images from Twitter, we used Google Vision API [44], which implements a DNN to classify the images by providing scored-tags for different objects included in the image.

4.4 Results and Findings

This Section presents the main results of this study concerning our research questions.
4.4.1 RQ1- Information Type

Text

As pointed in Section 4.3.3, we applied the BTM approach to explore traffic-related information from Twitter. A biterm denotes an unordered pair of words in a small and fixed-size context (e.g. every single tweet in our study). For example, if after pre-processing and cleaning the data, a tweet contains three words $w_1$, $w_2$, and $w_3$, BTM generates the following biterms to model the corpus: \{(w_1, w_2), (w_2, w_3), (w_1, w_3)\}. This generative model defines the distribution of biterm $b = (w_i, w_j)$ in topic $z$ as: $P(b) = \sum_k P(w_i | z) \times P(w_j | z) \times P(z)$, where $k$ defines the number of topics and should be defined upfront. This procedure also required the Gibbs sampling option, as it provides greater accuracy than the variational algorithm [45]. We ran the algorithm for different values of $k$ (i.e. $k = 2 \ldots 7$) and, based on our evaluation of the results, made a choice as to the most cohesive $k$. Table 4.1 presents the results of applying the BTM approach on our collected dataset, for $k = 2, 3$. Moreover, we found that with a large number of topics (i.e. $k = 5, 6, 7$), the topics become too niche to be consistently associated with a specific concept, and, consequently, their meanings become less cohesive. Looking at Table 4.1, we can see that the topics are more meaningful and related when $k = 3$ and can be interpreted as follow:

**Topic 1**– We determined *topic1* to be related to *road traffic*. However, *traffic* alone does not completely cover the idea behind this topic; *safety* and *time* (i.e. time stuck in traffic, and daytime) are a much closer approximation of the topic’s idea, which can be considered when developing traffic management systems.

**Topic 2**– This topic goes through keywords related to *traffic signal*, illustrating a large focus on the long queues at traffic signal intersections (i.e. stop, park, light, and close). After further analyzing the tweets assigned to this topic, we found that the words *truck* and *park* are not intended to explain parking conditions on the street, but rather explain the heavy traffic caused by traffic signals. As such, this topic is less focused on static information or vehicles types and more on...
<table>
<thead>
<tr>
<th></th>
<th>k=2</th>
<th>k=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic1</td>
<td>traffic</td>
<td>traffic</td>
</tr>
<tr>
<td>Topic2</td>
<td>accid</td>
<td>light</td>
</tr>
<tr>
<td>Topic1</td>
<td>mile</td>
<td>home</td>
</tr>
<tr>
<td>Topic2</td>
<td>polic</td>
<td>condit</td>
</tr>
<tr>
<td>Topic3</td>
<td>home</td>
<td>accid</td>
</tr>
<tr>
<td>hour</td>
<td>minut</td>
<td>hour</td>
</tr>
<tr>
<td>hour</td>
<td>stop</td>
<td>truck</td>
</tr>
<tr>
<td>line</td>
<td>day</td>
<td>safeti</td>
</tr>
<tr>
<td>line</td>
<td>due</td>
<td>park</td>
</tr>
<tr>
<td>day</td>
<td>due</td>
<td>home</td>
</tr>
</tbody>
</table>

Dynamic aspects of traffic signal timing that developers can leverage to build dynamic and context-sensitive traffic management systems.

**Topic 3**— Topic 3 seems more appropriate when tied with traffic accidents and the crowd’s report about the scene and timing of the accidents. The terms police, accid, lane, and hour as keywords vouch for this. due and home are also relevant words, as they hint at drivers’ destination and the cause of the accident.

**Image**

The results of our preliminary image analysis on the collected images from Twitter using Google Vision API [44] show that while the explored information from social text snippets is a powerful tool for characterizing traffic management systems, it still needs more context-sensitive information about the local characteristics of the environment and traffic conditions; exhibited very nicely in extracted features from these images.

The code snippet shown in Figure 4-3 represents sample features that can be collected from traffic-related images on Twitter, using Google Vision API. The impact of weather on road conditions, mode of transportation, traffic congestion, car type (e.g. family, compact, mid-size), car model (e.g. luxury), road conditions (i.e. asphalt, snow, sand) and plate information are sample features/labels that can be collected from this type of data.
After further analyzing the Tweets that have been assigned to which can be considered when developing TM systems. Where the collected images from Twitter, using Google Vision API as they hint at drivers' destination and the cause of the accident. Clearly vouches for this. Of the accident.

Traffic accident and crowd's report about the scene and timing developers can leverage to build dynamic and context-sensitive for explaining the heavy traffic caused by traffic signals. As intended to explain parking conditions on a street, but rather this topic, we found that the words traffic signal intersections (i.e. stop, park, light, and close).

However, related, when consequently, their meanings become less cohesive. Looking niche to be consistently associated with a specific concept, and, dataset, for the results of applying the BTM approach on our collected made a choice as to the most cohesive option, as it provides greater accuracy than the variational upfront. This procedure also required the Gibbs sampling.

A. RQ1- Information Type

2) Image:

This topic goes through keywords related to police, accid, lane and can be interpreted as follow: Moreover, we found that with a large alone does not completely cover the idea behind this topic; however, related images posted on Twitter is as effective as traffic cameras in capturing on-the-fly traffic-related events and conditions, combined with its low cost; means that it would be a mistake to dismiss this source of information when characterizing traffic management systems and their needs. Additionally, most drivers do not have enough time to report traffic conditions with enough details and often only post a photo to communicate their message. Fig 4-4 represents three examples from our dataset, including the extracted information for each image.

Figure 4-3: The meta data extracted from Google Vision API in JSON format

The results of the Google vision API show that analyzing traffic-related images posted on Twitter is as effective as traffic cameras in capturing on-the-fly traffic-related events and conditions, combined with its low cost; means that it would be a mistake to dismiss this source of information when characterizing traffic management systems and their needs. Additionally, most drivers do not have enough time to report traffic conditions with enough details and often only post a photo to communicate their message. Fig 4-4 represents three examples from our dataset, including the extracted information for each image.
Figure 4-4: Labels extracted from collected traffic-related images from Twitter. We selected the top 10-15 labels extracted by Google Vision API based on their confidence score: (a) lane, road, vehicle, mode of transport, motor-vehicle, asphalt, traffic, street, snow, parking, family car, (b) transport, vehicle, lane, advertising, lighting, sky, commercial vehicle, road, public utility, traffic light, asphalt, signage, (c) mode of transport, motor-vehicle, commercial vehicle, car, automotive lighting, technology, road.

4.4.2 RQ2- Information Application

To explore the application of the extracted information from social media in developing and characterizing traffic management systems, we leveraged the systematic process of open [46] and axial coding [47] methods. After coding the extracted labels from image data, using Google Vision API, and content-carrying terms from text data, using the BTM approach and lexical association, we categorized them based on their common properties and similarities. During this axial coding process, we generated higher-abstraction level type categories from the extracted labels/terms to find the relationship and links between categories and their subcategories. Our key findings in applying this process are summarized below:

Finding 1– real-time feature: While no specific topic significantly outweighed the others in frequency, the real-time nature of the extracted information from both image and textual data types often overlapped between these topics. This implies that self-reported information on social media can be used to explore real-time information about an on-going event, regardless of the topic under discussion.
Finding 2– context-sensitive features: The results of our comparison between the coded results associated with each of the three studied data sources (i.e. Twitter’s text data, Twitter’s image data, theory-based papers), indicate that while scientific information presented in theory-based publications can be used to form the core and basics of the traffic management strategies, there is a clear need for crowdsourcing to adjust and customize these techniques for specific contexts.

Moreover, comparing the coding results associated with *image* and *text* data collected from Twitter, we found that while textual data can augment the process of characterizing traffic management systems, the extracted information from *image* data type adds beneficial information about the context in which the system at hand will be deployed. We found that image data is the only source of information, compared to the other two sources, that can be leveraged to extract *demographic and life-style information* for a specific location/context (e.g. residential/non-residential, road conditions, the car makes and models (income level), signage information, etc.). This information can be used to avoid crunch through a customized design for the system, better communication with the end-users of the product, and more detailed information about the systems’ non-technical aspects, which significantly impact its applicability.

4.5 Threats to validity and limitations

Our study has several limitations. Not all twitter users make their location available on twitter. Unavailability of exact tweets makes it challenging to identify traffic problems of a specific region. Therefore, it is not easy to develop it into a real-time system that could give traffic-related information for a specific location. Topic modelling assigns a very low priority to some very location-specific keywords; hence, it is required to look at even lower priority keywords when labelling data for classification or looking for traffic trends. When performing image data extraction, there is the challenge that images are not always directly related to traffic and are relevant only in the context of the particular tweet, thus making it difficult to identify traffic information.
Moreover, as Twitter data is produced as a side-effect of communication between users, rather than as a representation of real information, people manage their online identities, which makes the content of the data highly biased. Also, Twitter data is not representative of the general population, and our observations are made from a sample of Twitter users, not from a random population sample. In other words, the generalizability of the study population can be affected by specific qualities that make someone more or less likely to share information in public media. In the same vein, the selected list of theory-based publications for keyword extraction and comparison purposes may pose another threat to the validity of the results of this study. To mitigate this risk, we included 60 publications from different aspects of traffic management, including both scientific and practical aspects.

4.6 Conclusion and Research Implications

Crowdsourcing and dynamic information collected from the end-users of traffic management systems are key to developing functional and satisfying systems for all stakeholders. However, finding what these stakeholders and end-users desire out of traditional system characterization processes and theory-based academic resources alone can be difficult, prolonged, and inaccurate. Using social media analysis, we tried to simplify that process by looking at different types of data (i.e. text, images), communicated by the crowd on social media. To this end, we used manual qualitative analysis (e.g. open coding) and automatic information extraction and NLP to analyze social media data. Following the results of these analyses, we found that while theory-based scientific works in the area of traffic management such as traffic signal timing, urban planning, and transportation management contribute to this process to a great extent, the end-users and stakeholders seldom mention those concepts when they communicate their concerns about traffic road conditions.

Given the flexibility of weighted finite-state transducers in exploring the lexical association between documents with different lengths, this data structure can be used effectively to filter unstructured and short-length information generated by the public on social media platforms. The
application of this technique in our information extraction process helps to effectively filter out social media data to a subset relevant to context-sensitive information, such as public transit and event-based urban network traffic control.

Concerning future steps and practical implications of this research, we intend to use Amazon Mechanical Turk (AMT) [48] to label the collected data for further sentiment analysis (i.e. emotions, anxiety level, cultural characteristics, etc.). We are also planning to conduct a parallel study with a larger number of theory-based papers and related books, rather than social media platforms alone, and compare results from the two. Another direction to take is replicating this study on more social media platforms, such as Instagram, Facebook, and more static Q&A/review platforms such as Yelp, Reddit, and Quora. Developing a tool that enhances the application of crowdsourcing in developing traffic management systems, focusing on dynamic tracking of generated information on social media over time through novel forms of natural language and image processing methods, is another possible extension to this work.
Bibliography


[23] Dinithi Nallaperuma, Rashmika Nawaratne, Tharindu Bandaragoda, Achini Adikari, Su Nguyen, Thimal Kempitiya, Daswin De Silva, Damminda Alahakoon, and Dakshan Pothuhera. Online incremental machine learning platform for big data-driven smart traffic


[27] Ming Ni, Qing He, and Jing Gao. Using social media to predict traffic flow under special event conditions. In *The 93rd Annual Meeting of Transportation Research Board*, 2014.


[31] Yeran Sun, Hongchao Fan, Marco Helbich, and Alexander Zipf. Analyzing human activities through volunteered geographic information: Using flickr to analyze spatial and temporal pat-


Part II

Uni-modal Network-Scale Traffic Signal Control: Application of Traffic Theories
Traffic Signal Timing Optimization by Modelling the Lost Time Effect in the Shock Wave Delay Model

Abstract

Over the years, studies presented on shock wave model optimization have been limited to the proposal of optimization control policies using queue length constraints in oversaturated conditions, and also finding the optimum cycle time and green splits based on either a known cycle time from the field or an optimum cycle time obtained from other methods. To our best knowledge, we can say after reviewing the literature that no attempt has been made to use the shock wave model to find the optimum cycle time for a general isolated intersection, because minimizing this model generates very small values, close to zero for an optimum cycle time, which is unacceptable. In this chapter, we propose an optimization model that provides the optimum cycle time and green splits when the total average delay at a general isolated signalized intersection is minimized for all vehicles present. To do so, we model the lost time effect in the shock wave delay model, which creates the most desirable optimum cycle time values. In our optimization process, the key strategy is to keep both approaches in the undersaturated condition. Therefore, our model works when the total amount of volume-to-capacity ratio of both approaches is less than 2.0; otherwise, where both approaches are oversaturated, other control policies should be considered and utilized. A comparison of the results with a widely-used formula in the literature reveals that our model is superior.

5.1 Introduction

Shock wave theory has been enhanced for use in different types of transportation facilities. One of the most common applications of this model is in the estimation of queue length and delay at signalized intersections, which primarily serves the purpose of managing traffic congestion and optimizing the traffic signal timing. Signal timing calculation, optimization, and control are highly
cost-effective methods of traffic management, especially when compared to physically modifying roads. Thus, we planned to find a simple and quick solution to optimize the cycle time and green splits, such that it would work as a fixed-time method in a general isolated signalized intersection, based on a widely-used and accurate model in delay estimations called the shock wave model [1]. As a fixed-time method, Webster [2] proposed a framework for signal timing calculation at an isolated signalized intersection to determine optimal cycle and appropriate duration for green time in each phase.

Although the use of traffic responsive methods is rising, fixed-time methods are still widely used [3], such as in the majority of signalized intersections in the U.S. [4] and Athens [5]. New fixed-time methods, like the model we propose, can be versatile and extended for use in traffic responsive control systems, since the design procedure of both methods is identical [6].

In this chapter, we modify the shock wave delay function to use as the objective function in the optimization model. In addition, we consider the fixed lost times in both approaches, model their effect in the signal timing, and use them in the constraints of the model in order to get desirable optimum cycle and green splits.

5.2 Shock Wave Theory in Signalized Intersections

5.2.1 Related work

Kinematic wave theory, also known as the Lighthill-Whitham-Richards (LWR) shock wave theory, was established on an analogy between traffic flow dynamics and fluid dynamics. In signalized intersections, the theory is commonly applied to estimate different measures of effectiveness (MoE), such as travel times, delays, queue lengths, and number of stops.

The primary development of the LWR shock wave model at signalized intersections was started by Rorbech [7], who developed the theory to find queue lengths at specific intervals during the green time. Following this, [8] presented the dynamics of queue formation and dissipation in isolated intersections, which were widely utilized by subsequent researchers. In their paper, a qualitative
analysis and a mathematical modelling were provided. The qualitative analysis is simpler, comprised of linear waves and triangular areas of queue formation and dissipation due to simplification assumptions. This method was later used in several publications (e.g. [9, 1, 10]). The second method, mathematical modelling, contains complicated mathematical computations and involves different curves and fan-out form of characteristics lines in each area. This method was also used by some researchers (e.g. [11]). In the qualitative analysis, it is assumed that flow and density change to capacity conditions immediately at the beginning of the effective green. A similar instantaneous transition is also assumed at the end of green time when clearing the intersection. However, in practice there is a gradual transition from one state to the other, which is supposed in the second method. [8] suggested that if the lost times during the phase transitions are considered and effective green times are used in place of nominal green times, the assumption of instantaneous transitions will not be unrealistic. This suggestion is also utilized by [12], who developed a real-time control policy to minimize total delay at isolated intersections, subject to queue length constraints in an oversaturated condition.

Using the simplistic method, [1] presented a shock wave delay model, which will be reviewed in the following section. They proved that the delay estimation from their shock wave model is accurate and consistent with results from other existing methods and a traffic simulation software. Since we aim to present a fast processing method in this chapter, we use the simplistic method of shock wave delay model presented by [1]. In other publications, the shock wave delay model has been used in the optimization framework in the following ways:

1. To determine optimum green splits based on fixed predetermined cycle time (e.g. [10])

2. To determine optimum cycle time and green splits through a control system based on queue length constraints (e.g. [12])
3. To determine optimum cycle time and green splits in an adaptive signal optimization starting with a predetermined cycle time (e.g. [13])

5.2.2 Shock wave delay model

Shock wave theory is founded on the three traffic parameters of flow, speed, and density, which are related by the equation below:

\[ q_\eta = k_\eta \cdot u_\eta \]  

(5.1)

Where \( q_\eta \), \( k_\eta \), and \( u_\eta \) are average traffic flow rate, average traffic density, and average traffic speed in region \( \eta \), respectively. In this theory, any macroscopic change in traffic characteristics is separated by a line in a space-time diagram which is called a “shock wave”. This line separates two distinct traffic regions and depicts the speed of the change propagating along a roadway. Shock wave speed is defined as below (\( \eta 1 \) and \( \eta 2 \) show two different regions):

\[ SW = \frac{q_{\eta 2} - q_{\eta 1}}{k_{\eta 2} - k_{\eta 1}}. \]  

(5.2)

The formation of different shock waves in a signalized intersection is illustrated in Figure 5-1. For three regions A (jam), B (discharge), and C (arrival) in this figure, the parameters are defined as follows. In the jam region, speed \( (u_j) \) and flow rate \( (q_j) \) are both assumed zero. Also, saturation \( (q_d) \) and arrival \( (q_a) \) flow rates are commonly shown by \( S \) and \( q \), respectively.

\( k_j \): jam density (veh/m)
\( k_d \): discharge density (veh/m)
\( k_a \): arrival density (veh/m)
\( q_j \): jam flow rate (veh/sec)
\( q_d \): saturation flow rate (veh/sec)
\( q_a \): arrival flow rate (veh/sec)
Figure 5-1: General shock wave analysis in the undersaturated condition.

- $u_j$: jam speed (m/sec)
- $u_d$: discharge speed (m/sec)
- $u_a$: free flow speed (m/sec)

While approaching the intersection, vehicles reach the stop bar in the beginning of the red time, and the first shock wave is formed, called $SW_I$. This wave shows the boundary between arriving vehicles and stopped vehicles. $SW_R$ emerges at the end of the red time, moves to the upstream of the stop bar and separates stopped vehicles and discharging vehicles. At the intersection of both waves ($SW_I$ and $SW_R$), the queue length is in the maximum extent, which is shown by $X_m$. The time to reach the maximum extent of the queue is given by $t_m$. Another shock wave ($SW_N$), separating arriving and discharging vehicles, is generated at the end of the time $t_m$. The queue which reaches its maximum at the end of $t_m$, requires a portion of green time to be cleared, which is called $t_c$. The relevant equations are as follows:

$$ SW_I = \frac{q}{k_a - k_j}, \quad (5.3) $$

$$ SW_R = \frac{S}{k_d - k_j}, \quad (5.4) $$

87
\[ SW_N = \frac{S - q}{k_d - k_a}, \quad (5.5) \]

\[ t_m = \frac{q \cdot r \cdot (k_j - k_d)}{S \cdot (k_j - k_a) - q \cdot (k_j - k_d)}, \quad (5.6) \]

\[ X_m = -t_m \cdot SW_R = \frac{q \cdot r \cdot S}{S \cdot (k_j - k_a) - q \cdot (k_j - k_d)}, \quad (5.7) \]

\[ t_c = \frac{X_m}{SW_N} = \frac{q \cdot r \cdot S \cdot (k_d - k_a)}{(S - q) \cdot [S \cdot (k_j - k_a) - q \cdot (k_j - k_d)]}. \quad (5.8) \]

The overall delay \((D)\) is determined by calculating the difference between the total travel time in the presence and the absence of traffic signals. The total travel time \((TTT)\) is calculated as the multiplication of the area of each region in Figure 5-1, and the difference between the density of each region and the approach density.

\[ D = TTT_{\text{signals}} - TTT_{\text{no-signals}} = \sum_{\eta=A,B,C} A_\eta \cdot k_\eta - \sum_{\eta=A,B,C} A_\eta \cdot k_C = A_A \cdot (k_A - k_C) - A_B \cdot (k_B - k_C), \quad (5.9) \]

\[ D = \frac{X_m}{2} \cdot [r \cdot (k_j - k_a) + (t_m + t_c) \cdot (k_d - k_a)]. \quad (5.10) \]

The average delay \((d)\) over a specific cycle time is given by the total delay divided by the total arrivals in an approach, and has units of seconds.

\[ d = \frac{\sum_{\phi=1}^{n} D_\phi}{C \cdot \sum_{\phi=1}^{n} q_\phi}, \quad (5.11) \]

where \(\phi\) indicates the number of approaches and \(C\) is the cycle time (sec).
5.2.3 Lost time

At the beginning of green time, while departing the intersection there is an interval that vehicles require to accelerate to reach capacity-condition speed. This time is called start-up lost time \((L_s)\) and is considered as an average nominal time, about 2 to 3 seconds for typical intersections. Another lost time occurs at the end of the cycle due to the considerable decrease in the departing flow rate of vehicles in a part of yellow time, and this is called end-gain or clearance lost time \((L_c)\); 2 to 3 seconds is also allocated to this nominal time. Therefore, in calculations for signalized intersections, effective green time \((g)\) and lost time \((L)\) are usually used instead of displayed green and yellow times as below:

\[
C = \sum_{\phi=1}^{n} g_\phi + L_\phi = g_1 + g_2 + L_1 + L_2, \tag{5.12}
\]

where \(g_1\) and \(g_2\) are effective green times, and \(L_1\) and \(L_2\) are total lost times in approaches 1 (i.e. \(L_1 = L_{s1} + L_{c1}\)) and 2 (i.e. \(L_2 = L_{s2} + L_{c2}\)), respectively. The lost time is usually considered in other existing delay models like Queuing model (Michalopoulos and Stephanopoulos 1977). Webster (1958) used lost time in estimating the optimum signal cycle time \((C_{opt})\) as below:

\[
C_{opt} = \frac{1.5L + 5}{1 - \sum_{\phi=1}^{n} \frac{q_\phi}{S_{\phi}}}. \tag{5.13}
\]

In this equation, lost time plays a key role in finding the optimum cycle time. Both equations above reveal that, when a simplified method is sought, lost time should be considered both in cycle intervals and in signal timing optimization.
5.3 Proposed Method

5.3.1 Formulating the lost time effect

Figure 5-2 explores the effect of lost times on cycle intervals in the shock wave analysis for two approaches. Start-up lost time \((L_s)\), in addition to its effect on producing delay in each cycle, changes the position of the starting shock wave ahead (point A to point B), making the queue length in the back of the queue larger (point C to point D). For instance, if 2 seconds is determined for this lost time (as the length of the line AB), then more than 2 seconds is needed to discharge the added queue in this period (the line EF or similarly the line LG). By adding the time \(L_s\) to the end of the displayed red interval, a wave, BD, is formed and considered instead of AC. This wave extends the maximum queue from point C to point D. The line EF shows the incremental effect of lost time and is defined as the minimum green time that must be provided to clear the queue length due to the start-up lost time. We call the line EF as \(L_{ec}\) (in the unit of time). Since the lines EF and CG are the same length, we can formulate \(L_{ec}\) as below:

\[
L_{ec} = CL + LG = L_s + \left( \frac{L_s}{SW_{R}} + \frac{L_s}{SW_{N}} \right) \cdot \left( \frac{SW_{N} \cdot SW_{I}}{SW_{N} - SW_{I}} \right) .
\]

(5.14)

By simplifying this equation we have:

\[
L_{ec} = q \cdot \frac{L_s}{S - q}.
\]

(5.15)

Also, the clearance lost time from the previous cycle (line NO) adds more delay to this effect (line FM), and the total effect is shown by \(L_{ex}\) (line EM). According to Figure 5-2, the total extra delay time because of the lost time in the approach is equal to the line CK (similarly EM in Figure 5-2). Supposing that start-up and clearance lost times are identical, and that the triangles \(\Delta \text{CDG}\) and \(\Delta \text{HIJ}\) are equal, then the lines LG, DJ, and GK are all the same length. Therefore, the minimum total extra time required to clear the extra queue length due to both the start-up lost time and clearance lost time is given by:
Figure 5-2: The effect of lost times on cycle intervals in the shock wave analysis for two approaches.

\[ L_{ex} = L_{ec} + L_G = L_s + 2 \cdot L_G. \]  \hspace{1cm} (5.16)

By simplifying it, we have:

\[ L_{ex} = \frac{L_s \cdot (S + q)}{S - q}. \]  \hspace{1cm} (5.17)

The equation above is used in the constraints of the proposed optimization model.

### 5.3.2 The proposed optimization model

In signal timing optimization problems, three objective functions are frequent: capacity maximization, cycle time minimization and delay minimization [14]. In this chapter, to find the optimum
cycle time, we aim to minimize the shock wave delay function. To optimize the signal timing of an isolated traffic signal, we consider a signalized intersection with two approaches (a one-lane eastbound, and a one-lane northbound). According to Figure 5-2, in the shock wave delay function, for the first approach we replace \( r_1 \) by \( R_1 + L_{s1} + L_{c1} \), and for the other approach we replace \( r_2 \) by \( R_2 + L_{s2} + L_{c2} \). In the delay function, we could also replace \( t_{m1} + t_{c1} \) by \( R_2 - (L_{s1} + L_{c1}) \) and \( t_{m2} + t_{c2} \) by \( R_1 - (L_{s2} + L_{c2}) \). However, since the results are similar, the change was not applied in the delay function. Then, the two shock wave delay functions below are used in the optimization model for two approaches:

\[
D_1 = \frac{X_{m1}}{2} \cdot [(R_1 + L_{s1} + L_{c1}) \cdot (k_{j1} - k_{a1}) + (t_{m1} + t_{c1}) \cdot (k_{d1} - k_{a1})].
\]

(5.18)

\[
D_2 = \frac{X_{m2}}{2} \cdot [(R_2 + L_{s2} + L_{c2}) \cdot (k_{j2} - k_{a2}) + (t_{m2} + t_{c2}) \cdot (k_{d2} - k_{a2})].
\]

(5.19)

Based on Equation 5.12, the objective function, which is the average delay for two approaches, is given by:

\[
d = \frac{D_1 + D_2}{(R_1 + R_2) \cdot (q_1 + q_2)},
\]

(5.20)

where \( R_1 + R_2 = C \). If we substitute \( D_1 \) and \( D_2 \) in the equation above and simplify the equation, \( d \) is obtained as below:

\[
d = \frac{q_1 \cdot S_1 \cdot (S_2 - q_2) \cdot (R_1 + L_{s1} + L_{c1})^2 + q_2 \cdot S_2 \cdot (S_1 - q_1) \cdot (R_2 + L_{s2} + L_{c2})^2}{2 \cdot (S_1 - q_1) \cdot (S_2 - q_2) \cdot (R_1 + R_2) \cdot (q_1 + q_2)}.
\]

(5.21)

The significance of this equation is that to calculate \( d \), only the data of arrival and departure flow rates (i.e. \( q_1, S_1, q_2, \) and \( S_2 \)) in both intersection approaches is needed, while \( L_{s1}, L_{c1}, L_{s2}, \) and \( L_{c2} \) are all initially assumed as a fixed known value, about 2 seconds each. Then, our proposed optimization on the basis of lost time analysis is written as below:
\[ Min \quad d = \frac{q_1 \cdot S_1 \cdot (S_2 - q_2) \cdot (R_1 + L_{s2} + L_{c2})^2 + q_2 \cdot S_2 \cdot (S_1 - q_1) \cdot (R_2 + L_{s2} + L_{c2})^2}{2 \cdot (S_1 - q_1) \cdot (S_2 - q_2) \cdot (R_1 + R_2) \cdot (q_1 + q_2)} . \]

s.t.

1. \[ R_1 \geq L_{ex2} + L_{s2} + L_{c2} \]
2. \[ R_2 \geq L_{ex1} + L_{s1} + L_{c1} \]
3. \[ R_1 \geq t_{m2} + t_{c2} + L_{s2} + L_{c2} \]
4. \[ R_2 \geq t_{m1} + t_{c1} + L_{s1} + L_{c1} \]
5. \[ g_1 \geq L_{ex1} \]
6. \[ g_2 \geq L_{ex2}(5.22) \]

The defined constraints are obtained from the geometry in Figure 5-2. In the above setting, the first and second constraints provide the minimum duration for the displayed red times based on the other approach’s lost times and lost times effects. Constraints 3 and 4 provide minimum duration based on the required time to form and clear the queue in the other approach, with the purpose of keeping the intersection in the undersaturated condition. The third constraint has been derived from the combination of two constraints \[ R_1 \geq g_2 + L_{s2} + L_{c2} \] and \[ g_2 \geq t_{m2} + t_{c2} \]. The similar way, the fourth constraint is derived from the two constraints \[ R_2 \geq g_1 + L_{s1} + L_{c1} \] and \[ g_1 \geq t_{m1} + t_{c1} \].

Finally, in constraints 5 and 6, \( L_{ex1} \) and \( L_{ex2} \) have been considered as a minimum for the green times in each intersection.

Regarding the strategy of keeping the intersection in the undersaturated condition, it should be noted that the degree of saturation is obtained by the volume-to-capacity \( v/c \) ratio (see equation below) and is dependent on the signal timing, or more specifically, the green to cycle time ratio.

\[ \frac{v}{c} = \frac{q}{S \cdot \frac{g}{c}} . \]  \hspace{1cm} (5.23)

In each approach, \( v/c \) can be less than 1.0, 1.0 or more than 1.0, where they are called undersaturated, saturated, and oversaturated, respectively. If we aim to keep the intersection in the undersaturated condition, we must optimize the cycle time so the \( v/c \) ratio in both approaches will
be less than 1.0, and the total ratio less than 2.0. For example, if $S_1$, $q_1$, and $q_2$ are supposed 1800 veh/hr, 90 veh/hr, and 1620 veh/hr, and if $g/C = 0.5$ is assumed, then the $v/c$ ratios of the approaches will be 0.1 and 1.8, respectively. Hence, with this timing, one approach is undersaturated and the other is oversaturated. Since the total $v/c$ ratio of the approaches is less than 2.0 (i.e. 0.1 + 1.8 = 1.9), our method is able to find the optimum cycle time. In this case, suppose that the $g/C$ ratios for two approaches are 0.075 and 0.925. Then the $v/c$ ratios will be 0.67 and 0.97. This way, both approaches are in the undersaturated condition. Thus, our method works in the case that the total $v/c$ ratio of both approaches is less than 2.0.

5.3.3 Convexity

An important feature of the convex optimization problem is that the optimal solution is unique (Luenberger et al. 1984). A minimization problem that involves a convex objective function and a convex feasible region is referred to as a convex optimization problem. Our proposed two-variable ($R_1$ and $R_2$) optimization model is convex and provides one feasible solution. The first derivative test implies that the objective function is twice differentiable and continuous. The second derivative test of convexity, by calculating the determinant of Hessian for a given set of parameters, shows that the function is positive definite (convexity condition) in the feasible convex domain ($R_1 > 0$ and $R_2 > 0$).

5.4 Evaluation

To evaluate the proposed method, we consider hypothetical parameters as below:

$S_1 = S_2 = 1800 \, \text{veh/hr} = 0.5 \, \text{veh/sec}$

$k_{d1} = k_{d2} = 60 \, \text{veh/km} = 0.06 \, \text{veh/m}$

$k_{j1} = k_{j2} = 120 \, \text{veh/km} = 0.12 \, \text{veh/m}$

$L_{s1} = L_{c1} = L_{s2} = L_{c2} = 2 \, \text{sec}$. 

94
Table 5.1: Results of the Proposed Signal Timing Method and Webster’s Method.

<table>
<thead>
<tr>
<th>Case</th>
<th>Approach 1</th>
<th>Approach 2</th>
<th>The proposed method</th>
<th>Webster’s method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>qi (veh/hr)</td>
<td>ka1 (veh/km)</td>
<td>q2 (veh/hr)</td>
<td>ka2 (veh/km)</td>
</tr>
<tr>
<td>1</td>
<td>90</td>
<td>2</td>
<td>90</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>2</td>
<td>1620</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>180</td>
<td>4</td>
<td>360</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>180</td>
<td>4</td>
<td>1260</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>270</td>
<td>6</td>
<td>720</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>450</td>
<td>10</td>
<td>450</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>450</td>
<td>10</td>
<td>900</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>450</td>
<td>10</td>
<td>1260</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>810</td>
<td>18</td>
<td>900</td>
<td>18</td>
</tr>
</tbody>
</table>

To investigate the model for different situations, we use different values of $q$ and $k_a$ for both approaches, where $q$ is ranging from 90 veh/hr (indicating low $v/c$ of 0.1 for $g/C=0.5$) to 1620 veh/hr (indicating high $v/c$ of 1.8 for $g/C=0.5$). The results for 9 combinations using Maple software are presented in Table 5.1. In this Table, the results from Webster’s method are also provided for evaluation purposes. Since the shock wave delay model has been proven accurate for calculating delay, we used this model to calculate the delay related to signal timing provided by Webster’s method. In Table 5.1, we show displayed green ($G$) times instead of displayed red ($R$) times, as is prevalent in signal timing studies.

5.5 Discussion

The results given in Table 5.1 show that our model, in all cases, produces lower optimum cycle time. Generally, green to cycle time ratios are comparable in both methods, with a maximum difference of approximately 12%. Also, in all cases, the average delay of our proposed method is lower than Webster’s method, where the constraints are not considered when finding the delay of both methods. When the initial arrival flow rates are low in both approaches (i.e. the total $v/c$ is less than 0.5), there is no significant difference between the results of the two methods (e.g. cases 1 and 3) but when the total $v/c$ is around 2.0 (i.e. 1.9 in cases 2, 8, and 9) the difference is remarkable (an optimum cycle time of 160 seconds vs. 340 seconds). Webster’s method is considered inefficient in
high $v/c$ ratios (Ahn 2004). This is evident in the delays in cases 8 and 9, which are almost twice the amount of delay in our solution. Furthermore, as recommended by the Canadian capacity guide for signalized intersections, the maximum cycle time used will depend largely on the control method, but should not exceed 160 seconds [6]. Thus, in high $v/c$ ratios, the results of our model is superior, where our method produces exactly 160 seconds at the highest total $v/c$ ratios. It is prudent to mention that the total ratio of higher than 2.0 means both approaches are in the oversaturated condition. While these conditions (e.g. bottlenecks) cannot be prevented, various control policies are presented by researchers to manage these situations, which are beyond the scope of this research.

5.6 Conclusion

In this chapter, we proposed an optimization model to find the optimum cycle and green splits by minimizing the shock wave delay model for a simple isolated signalized intersection as a pre-timed method. To do so, we modelled the lost time compliant with delay calculations in the shock wave model. The incremental effect of lost time in extending the required green times in each cycle affects the extension of the red time in the other approach. This way, cycle time is optimized, based on the minimum required green and red times in both approaches, by establishing an appropriate balance between the two approaches, while providing the minimum delay for individual vehicles in the intersection. In this method, the key strategy is to keep both approaches in the undersaturated condition. Therefore, our model works when the total $v/c$ ratio of both approaches is less than 2.0; otherwise, where both approaches are oversaturated, other control policies should be considered.

The results show that our proposed method outperforms Webster’s method, particularly in high $v/c$ ratios. Without considering the lost time and its incremental effect, the comparable model produces very small values for the optimum cycle time, close to zero, which is impractical; the triangles in the model are originally formed based on the red time of each approach, hence the model tends to shrink the area of triangles to provide the minimum possible delay, which occurs when the red times are shrunk to near-zero values in both approaches.
Moreover, without the constraints presented in our research work, the ratio of green to cycle time will be similar to both the proposed method and Webster’s method if we input the optimum cycle time as a known value to find the optimum green splits. Nevertheless, to optimize the cycle time and provide reasonable values, the proposed restrictions are required to provide low boundaries to prevent green times from reaching zero. In conclusion, considering the lost time as a constant, defining the relations in cycle time based on the displayed red times, the effective green times, and the lost times, and embedding the lost time in both the delay function and constraints aids the model to generate appropriate, real-world results.

This method, even without further development, can provide insight by supplying an appropriate approximation about optimum signal settings. It can be considered a starting point, while for future work it can be adapted and applied in coordinated intersections, network scale or traffic adaptive systems. Also, this model can be improved by adding queue length control constraints, pedestrian constraints, and number of vehicle stops, using them in a multi-objective optimization problem. To investigate the applicability and strength of the proposed optimization model, different types of traffic signals (e.g. with two or more lanes) in various conditions will be examined via access to real data.
Bibliography


Real-time Decentralized Traffic Signal Control for Congested Urban Networks Considering Queue Spillbacks

Abstract
This chapter proposes a decentralized network-level traffic signal control method addressing the effects of queue spillbacks. The method is traffic-responsive, does not require data communication between intersections’ controllers, uses lane-based queue measurements, and is acyclic. Each traffic controller operating on an intersection aims at maximizing the effective outflow rate locally and independently with the goal of maximizing global throughput of the entire network. At each intersection, the signal control method estimates and adopts the maximum possible phase time in which all active movements discharge at their full capacity. This is modeled using a shock wave based queue length estimation model while capturing the spillback at the downstream links. The method demands real-time data including, the queue lengths, the arrival flows, and the downstream queue lengths in all the lanes at the control decision times. The proposed method results in a feasible solution in all conditions in the entire network with any scale within a short amount of time, which makes it favourable for real-time applications. Numerical results demonstrate that the proposed method outperforms other well-known benchmark methods in both isolated intersection and network configurations.

6.1 Introduction
Optimizing and controlling traffic signals is a long-standing concern for most urban cities. The amount of literature in this research area is considerable, and several researchers have widely attempted to contribute to traffic control in different ways, based on various assumptions, goals, and requirements. The existing research works can be broken down from different aspects, including: (i)
the scale (isolated [1], arterial [2, 3], and network [4, 5, 6]), (ii) the level of responsiveness to traffic (fixed-time or pre-timed [7, 8], actuated [9], and real-time [10, 11, 12]), (iii) the modeling approach (traffic theory based [5, 13], simulation based [14], and data-driven [15, 16]), (iv) the type of control method (centralized [17, 18, 19], and decentralized and distributed [20, 21, 22, 23, 24]), (v) the level of congestion condition (under-saturated [25], over-saturated [26, 27], and spillback [28, 29]), (vi) the type of estimation method of the input data to the control method (queueing theory [30], and shock wave model [31]), and (vii) the data communication level (traditional [32] and connected vehicle environment [33, 10]).

The first step of any traffic-responsive control system is gathering and estimating the input data based on the requirements of the control method. The queue length is an input data that is the base requirement of most traffic signal control methods. Literature suggests two modeling categories to estimate the queue length, (i) the cumulative traffic input-output based models [7, 34, 35, 36], and (ii) shock wave theory-based models [31, 37, 38, 39]. The models in the first category are not inherently able to provide the spatial distribution of queue dynamics [13], suffer from measurement errors, and are not suitable to accurately estimate queue lengths in oversaturated conditions [40]. Alternately, the shock wave based models provide the temporal-spatial queue dynamics with input data from loop detectors or probe vehicles. The temporal-spatial feature is essential specifically in oversaturated and spillback conditions. Thus, we develop our control method based on a shock wave based queue length estimation model, which needs arrival flow and initial queue length data in real-time and is reasonably accurate as it captures spatial and temporal queue length evolution. It is worth noting that the queue spillback phenomenon can be captured by the Cell Transmission Model (CTM) [41], link transmission model (LTM) [42], deep learning approaches for queue length estimation [43], and shock wave models. However, the shock wave model requires less information compared to the two other models [44].

In addition to the level of accuracy, the number of required data types, and the capabilities of capturing the details of traffic behaviours, there are other factors that impact the effectiveness of the traffic signal control method. These factors include the volume of the data, the cost-effectiveness
of data collection in the real world, the time and the degree of simplicity of the data processing for estimation and signal timing, and the amount of data communication. The type of traffic control method identifies the amount of impact of these factors, specifically on the challenges of data gathering, processing, and transmission. With regards to the type of traffic control methods, centralized systems provide a unified control entity that acts centrally, based on the collected data from the sensors and the measurements and computation over the entire network. These approaches do not scale well when they are used to control large urban networks. As a solution to the scaling problem, distributed and decentralized systems present more than one decision-making unit where they do not contain any centralized controller that coordinates or generates traffic plans. Although there is not a general agreement on the usage of distributed and decentralized in the literature of network traffic signal control and the two terms are used interchangeably, we use them as follows.

In distributed systems, every single intersection controller is an independent agent that makes its own decisions in interactions with other neighbor intersections. In these systems, there is a chance of information exchange between neighbor intersections and the decisions might be made by negotiation between the intersections. In contrast, decentralized systems require local data only. In decentralized systems, the local controllers have no interactions with other intersections in terms of both input and output data in decision-making computations. This way, the reliability of the system increases by removing the need for communications between intersections. Thus, communication issues, such as network delays, do not affect the control system. Decentralized systems in the literature of network traffic signal control may sometimes be referred to as fully distributed system. For instance, Gershenson (2007) [45] proposed a local self-regulation approach in which a phase in an agent receives green time if the number of vehicles on an approach meets a predefined threshold. In another work, Burguillo-Rial et al. (2012) [46] proposed a self-organizing control method relying on stop-bar detectors. Furthermore, Lämmer et al. (2008) [20] proposed a self-control method that equalizes the degree of saturation and queue length. Another approach was proposed by Xie et al. (2012) [47], who modeled the single intersection control problem as a single machine-scheduling
problem. The approach operates based on aggregating arriving vehicles in clusters, where clusters are defined as jobs in the scheduling problem.

The pressure-driven traffic control policies $P_0$ and Max Pressure (MP) are remarkable research works in the area of distributed and decentralized traffic control. Smith (1980) [48] introduced the $P_0$ control method, which considers variable route choices and maximizes network capacity under certain conditions, where demand is within network capacity, and there is no spillback. This method was revisited by Smith (2011) [49] and Smith et al. (2019) [50]. MP was originally presented by Tassiulas et al. (1990) [51] in a wireless packet switching network. This method only requires the queue lengths data in serviceable buffers of each processor and at adjacent downstream buffers. Varaiya (2013) [52] applied and developed MP in an urban traffic network. This local feedback controller controls traffic signals based on the difference in traffic load on the upstream and downstream links of the intersection. Varaiya (2013) [52] formally proved that MP can stabilize a traffic network when the network inflows are within the network capacity and there is no spillback. Since then, many research works attempted to adapt the method to real-world applications. For instance, Gregoire et al. (2014) [53], proposed a method to consider finite link capacity, i.e., bounded queue, while the assumption in the preliminary method relies on infinite road capacities. The MP method is also employed to stabilize the queues in signalized arterial network [54]. The preliminary method (i.e., [52]) performs an acyclic control, where cycle length and composition can vary during the process. An extension is a cyclic MP-based algorithm proposed by Le et al. (2015) [21] to provide a minimum phase in a cycle for the movements with low traffic demand. Li et al. [55] proposed a decentralized MP-based method, called position-weighted backpressure (PWBP) which captures the spatial distribution of vehicles along the links, and consequently, spillback conditions. The results showed that PWBP method outperforms the standard and the capacity-aware MP methods, specifically with higher demand levels.

Having considered the requirements discussed above, our proposed control method is fully decentralized, which makes it independent of data communication among intersections in the network. This independence is essential from two aspects. First, during the estimation step, the method re-
moves the need for exchanging input (estimated) data between intersections. The second aspect is that each intersection does not need to share its control decisions with other intersections. These two factors contribute to decreasing the processing time of the method.

Another feature of the proposed method is that it is acyclic, which removes the need to determine cycle times and pre-determined sequence of phases in a cycle. This increases the frequency of decision points during the signal timing process and helps provide higher flexibility in traffic-responsive solutions.

The proposed control method is a discrete-time optimization that explores the maximum possible phase time in which all active movements discharge at their full capacity. This is fulfilled based on finding the minimum saturated green time (greater than zero) for all possible phases. The proposed method averts from dealing with minimizing or maximizing any variables. In turn, the control method simply finds the minimum and maximum of estimated parameters, such as saturated green time and effective outflow rate in each lane. Therefore, it is not trapped in the optimization process’ challenges and complexities. This feature of the method leads to have a feasible solution in all traffic conditions. Moreover, it ensures a calculation time of less than a second, which empowers the method for real-time applications.

In the network scale, avoiding and controlling spillback condition and reducing the time that intersections experience spillback is crucial. Spillback occurs when a queue of vehicles fills up the storage capacity of a link, and no vehicles can enter the link from the upstream approaches. This reduces the outflow of the network. Therefore, in the proposed method to avoid the spillback condition, we consider the available storage capacity of the downstream link as a constraint and limit the saturated green time of each upstream lane to the related constraint. To this end, with regards to all downstream lanes, we estimate the time during which a downstream lane can accommodate vehicles from upstream links based on its available storage capacity and the length of the link. The available storage capacity of the downstream link is calculated based on the queue length in the downstream link, which is read at the time of decision-making.
Finally, it should be pointed out that the PWBP method proposed by Li et al. [55] employs a capacity-aware version of MP which additionally accounts for the possibility of spillbacks by considering spatial distribution of vehicles through applying higher weights to queues that extend to the ingress of the link. In the method, a small number of phases (i.e. 4 - 8 possible phases) are used. In our proposed method, the phase with the highest estimated effective outflow rate receives the priority. The notable difference is that our proposed method also optimizes the phase duration while the underlying traffic model is different. Our method is explicitly based on shock wave theory. Moreover, in our proposed method, all possible phases (with any number of non-conflicting movements) are explored to determine the optimal phase.

6.A lists a nomenclature for the different variables and parameters used in this chapter. The rest of this chapter is organized as follows. Section 6.2 describes the problem statement. Section 8.3 presents the proposed decentralized traffic control method. In Section 8.4, the simulation setup and results are discussed. This Section evaluates the efficiency of the proposed control method in both isolated intersection and network configurations. Finally, Section 8.5 provides some concluding remarks as well as directions for further research.

6.2 Problem statement

Let us consider an urban network with multiple intersections, in which each intersection is controlled independently in a fully decentralized system without any coordination methods. Let each intersection be a general 4-way signalized intersection with lanes in which only one movement is served, whether left-turn, through, or right-turn movement. The traffic movements $j \in \{1, 2, \ldots, J\}$ can be activated, i.e. receive green signal indications, or deactivated, i.e. receive red signal indications. A phase is defined as a set of non-conflicting movements that indicates whether the movements in the intersection are activated or deactivated during the time called phase time $p$. During the phase time, one or several non-conflicting movements are activated and simultaneously receive the right
of way and all other movements are deactivated. All possible phases can be defined as \( P = (\lambda_{ij}) \), \( i \in \{1, 2, \ldots, I\} \), where \( i \) denotes the number (i.e. index) of possible phases; \( I \) is the total number of possible phases; \( \lambda_{ij} \in \{0, 1\} \), 0 and 1, respectively, represent red and green signal indications; and \((\lambda_{ij})\) denotes a \( i \times j \) matrix with the elements \( \lambda_{ij} \). \( P_i \) is a possible phase in the set \( P \). We assume that the discrete time index is \( k \), where \( k \in \{0, 1, \ldots\} \). \( K \) is defined as a given decision point.

The control decisions of each intersection are made independently of other intersections at each decision point \( k \). The time instant in which the control decisions are made is called the phase transition point. At each phase transition point, the following real-time input data in all the lanes of the intersection are collected: (i) the queue lengths, (ii) the arrival flows, and (iii) the downstream queue lengths. The control decisions include phase time \( p^*(K) \), phase \( P^*(K) \), and interphase \( I^*(K) \). The three control decisions are integrated and represent the ideal phase, \( \psi^*(K) = (p^*(K), P^*(K), I^*(K)) \). The objective is to find an ideal phase \( \psi^*(K) \) at each decision point to maximize the number of vehicles that exit the approaches of the intersection, i.e. the intersection throughput. The premise is that the ideal phase would enable movements to discharge at their maximum capacity. Accordingly, it can be conjectured that no other phase can lead to more intersection throughput.

### 6.3 Decentralized traffic signal control method

In this section, the details of the proposed traffic signal control method for a general network are presented. The flowchart in Figure 6-1 helps clarify the process of the proposed method, which is applied to each intersection independently. The proposed method is explained according to the control decisions, (i) phase time calculation and (ii) phase improvement and interphase determination. Prior to that, we provide the details of estimating saturated green time without considering spillback effect (Section 6.3.1) and in general case considering the effect of spillback (Section 6.3.2).
6.3.1 Saturated green time: without spillback control

The saturated green time, $G_{j}^{\text{sat}}(K)$, which is estimated for each movement at the decision point $K$ is defined as the maximum green time that movement $j$ in an intersection can discharge vehicles at its full capacity (i.e. the saturation flow). Theoretically, saturated green time guarantees the maximum effective outflow rate of the vehicles, assuming the arrival rate to the back of the queue is smaller than the saturation flow. The saturated green time is calculated based on the LWR (Lighthill-Whitham-Richards) shock wave theory \[56, 57\] as follows:

$$G_{j}^{\text{sat}}(K) = \frac{x_{j}^{0}(K) \cdot (K_{j}^{\text{jam}} - K_{j}^{\text{arr}}(K))}{S_{j} - q_{j}^{\text{arr}}(K)}, \quad (6.1)$$
where $K_j^{jam}$ and $S_j$ are the jam density and the saturation flow of movement $j$, respectively. $x_j^0(K)$, $K_j^{arr}(K)$, and $q_j^{arr}(K)$ respectively denote queue length, and estimated arrival density and flow into movement $j$ at the decision point $K$.

To estimate the saturated green time, three parameters and two variable inputs are required. The parameters include (1) saturation flow $S_j$, (2) critical density $K_j^{cri}$, and (3) jam density $K_j^{jam}$. The parameters can be readily estimated from the flow-density Fundamental Diagrams (FDs) that are constructed individually for all three turns (i.e. left, through, and right turns) in an offline process. A common assumption in the literature is to assume triangular flow-density FDs. Consequently, $K_j^{arr}(K)$ is:

$$K_j^{arr}(K) = \frac{q_j^{arr}(K) \cdot K_j^{cri}}{S_j}. \quad (6.2)$$

With limited link length, $l$, the propagation of shock waves in queue buildup evolution is constrained. This occurs in spillback conditions and affects the saturated green time. In this regard, let us define the estimated maximum queue extent, $x_j^M(K)$, as:

$$x_j^M(K) = \frac{x_j^0(K) \cdot (K_j^{jam} \cdot S_j - q_j^{arr}(K) \cdot K_j^{cri})}{K_j^{jam} \cdot (S_j - q_j^{arr}(K))}. \quad (6.3)$$

As shown in Figure 6-2, when the estimated maximum queue extent, $x_j^M(K)$, exceeds the link length, the two backward-moving shock waves, $SWI_j(K)$ and $SWR_j(K)$, cannot theoretically intersect each other because of the link length limitation. To accommodate this condition, $G_j^{sat}(K)$ based on the triangular FD is:

$$G_j^{sat}(K) = \begin{cases} \frac{x_j^M(K) \cdot K_j^{jam}}{S_j} & \text{if } x_j^M(K) \leq l \\ \frac{l \cdot K_j^{jam}}{S_j} & \text{if } x_j^M(K) > l. \end{cases} \quad (6.4)$$

Note that, $G_j^{sat}(K)$ in the first condition of Equation 6.4 is equivalent to Equation 6.1.

Besides the three FD parameters, queue length $x_j^0(.)$ and arrival flow $q_j^{arr}(.)$ are two variable inputs required to make the control decision. They are assumed to be either available, e.g. via
Figure 6-2: Shock wave analysis of the saturated green time time estimation: a) estimated maximum queue extent is less than or equal to the link length and b) estimated maximum queue extent is greater than the link length.

traffic cameras, loop detectors, or estimated based on historical and/or real-time data. In the microsimulation tests, to measure the arrival flow the vehicle counts are collected through detectors at the far end upstream of each approach. For arrival flow measurement, a discretization method is used that is presented in 6.B. To estimate queue length in each lane we use the available data during the simulation at each decision point. The choice of queue length measurement devices and estimation methods are out of scope of this chapter. It should be noted that although the proposed method is decentralized and the traffic signal controller in each intersection only uses the data of its incoming and outgoing links, the queue length data of incoming and outgoing links of each intersection is shared with four immediate adjacent intersections. In other words, the queue length data of a link simultaneously affect the decision making of two intersections.

6.3.2 Saturated green time: considering spillback control

In isolated intersections, it is assumed that there are no storage capacity restrictions in downstream links. However, in the network, storage capacities of links are limited and the discharge time of the upstream approach is restricted based on the available storage capacity of the downstream
link. Hence, in spillback control, the impact of the downstream queues on the signal timing control
decisions is considered.

Let us assume that $x_{dj}^d(K)$ is the available storage capacity of the link downstream of movement
$j$ at the decision point $K$, $G_{dj}^d(K)$ is the maximum time that the link downstream of movement
$j$ can let vehicles in at the decision point $K$, and $l^d$ is the downstream link length. Note that
we define $x_{dj}^d(K)$ for the downstream link, not the downstream lanes. This definition is chosen as
a conservative measure because discharged vehicles from the upstream link might use any of the
downstream lanes. In addition, discharge interruptions might happen when the vehicles get stuck
in the downstream lanes while trying to change lanes. Therefore, we consider that the available
storage capacity of a downstream lane depends on the available storage capacity of the downstream
link, which is the minimum available storage capacity in all the lanes in the link. For this purpose,
let $x_{0d}^d(K)$ be the queue length of the movement $j$ in the downstream link. We also define $x_{0dL}^d(K)$
to represent the maximum queue length among all the lanes in the downstream link. Consequently,

$$x_{dj}^d(K) = l^d - x_{j}^{0dL}(K).$$  \hspace{1cm} (6.5)

As illustrated in Figure 6-3, the available storage capacity of all the downstream lanes is considered
to be the same and be equal to the available storage capacity of the link. The available storage
capacity of the link is calculated as the difference of the link length and the maximum of downstream
lanes’ queue length. Note that if more details of real-time information on lane-choice of vehicles is
available, this can be readily incorporated in the proposed method. The current assumption is a
simple solution to lane-choice parameters as this is very detailed and hard to obtain in practice.

The spillback constraint states that if the estimated saturated green time of upstream movement
$j$, $G_{j}^{sat}(K)$, is greater than the available time capacity of the link downstream of movement $j$,
$G_{dj}^d(K)$, $G_{j}^{sat}(K)$ is limited to $G_{dj}^d(K)$ (see Figure 6-4), i.e. $G_{j}^{sat}(K) = \min \{G_{j}^{sat}(K),G_{dj}^d(K)\}$, where:

$$G_{dj}^d(K) = \frac{x_{j}^{dl}(K) \cdot K_{j}^{jam}}{S_j}.$$  \hspace{1cm} (6.6)
As stated earlier, upstream lane is taken into account. The reason is that during the selected phase time, the only available storage capacity of the downstream movements and only the queue length evolution of the downstream lanes or links is not considered in estimating the maximum possible queue extent that can be discharged from the upstream lane.

Note that \( G_j^a(K) \) can also be written as:

\[
G_j^a(K) = \begin{cases} 
\frac{x_j^d(K) - K_j^{jam}}{S_j} & \text{if } x_j^M(K) > x_j^d(K) \\
\frac{x_j^a(K) - K_j^{jam}}{S_j} & \text{if } x_j^M(K) \leq x_j^d(K).
\end{cases}
\] (6.7)

Figure 6-3: Considering the available storage capacity of the downstream link as the available storage capacity of all the downstream lanes.

Note that \( x_j^d(K) \) is the maximum queue length that downstream link can accommodate the vehicles traveling from the upstream lane. When the entire queue of the upstream lane cannot be discharged due to the downstream capacity restriction, in Equation 6.6, \( x_j^d(K) \) is considered equivalent to the maximum possible queue extent that can be discharged from the upstream lane. \( S_j \) and \( K_j^{jam} \) are then related to the upstream lane.

\( G_j^a(K) \) is estimated based on two assumptions, (i) capacity flow from the upstream link and (ii) zero outflow for the downstream link. With regards to the second assumption it is worth mentioning that the queue length evolution of the downstream lanes or links is not considered in estimating the available storage capacity of the downstream movements and only the queue length evolution of the upstream lane is taken into account. The reason is that during the selected phase time, the only source of queue length growth is the upstream queue (which has been considered). As stated earlier,
the destination of all movements in either of the possible phases are distinct; thus, during a phase, there is no inflow to a downstream link except from only one movement in one of the upstream links. Considering the queue discharge in downstream lanes may result in determining a longer phase as it provides more storage capacity. However, we ignore this useful storage capacity in our method because, to estimate it, we need to collect extra data from all immediate downstream traffic signals, i.e. traffic signal states and timing. Nevertheless, two considerations in the proposed acyclic method mitigate the downside of ignoring the possible queue discharge in downstream movements, which are movement continuities and shorter time interval between the decision points. Therefore, the estimation method is conservative and yet simple and effective to estimate $G_{ij}^d(K)$. Being conservative is consistent with the previous assumption on estimation of $x_{ij}^d(K)$. It should be pointed out that although the proposed method is presented with one lane per movement, it can be easily extended to multiple lanes per movement.

Eventually, we can combine the conditions in estimating the saturated green time $G_{ij}^{\text{sat}}(K)$ in non-spillback and spillback conditions as below:

$$G_{ij}^{\text{sat}}(K) = \begin{cases} \frac{x_{ij}(K) - K_{ij}^\text{jam}}{S_j} & \text{if } x_{ij}(K) \leq x_{ij}^d(K) \text{ and } x_{ij}(K) \leq l \\ \frac{lK_{ij}^\text{jam}}{S_j} & \text{if } l \leq x_{ij}(K) \text{ and } l \leq x_{ij}^d(K) \\ \frac{x_{ij}(K) - K_{ij}^\text{jam}}{S_j} & \text{if } x_{ij}^d(K) \leq x_{ij}(K) \text{ and } x_{ij}^d(K) \leq l. \end{cases}$$  \tag{6.8}$$

### 6.3.3 Phase time calculation

Here, based on the estimated saturated green time in Section 6.3.2, the minimum saturated green time greater than zero for possible phase $i$ at the decision point $K$, $G_i(K)$, is:

$$G_i(K) = \min_{j=1..J} \left( G_{ij}^{\text{sat}}(K) \cdot \lambda_{ij} \right), \quad i \in \{1, \ldots, I\}. \tag{6.9}$$
After finding $G_i(K)$, the next step is calculating the maximum of the estimated effective outflow rate of vehicles in each possible phase. To this end, first the outflow of movement $j$ in possible phase $i$ based on the shock wave model is obtained as:

$$F_{ij}^{m}(K) = \begin{cases} S_j \cdot G_i(K) \cdot \lambda_{ij} & \text{if } G_{j}^{\text{sat}}(K) \geq G_i(K) \\ \left(S_j \cdot G_{j}^{\text{sat}}(K) + q_j^{\text{tr}}(K) \cdot (G_i(K) - G_{j}^{\text{sat}}(K))\right) \cdot \lambda_{ij} & \text{if } G_{j}^{\text{sat}}(K) < G_i(K). \end{cases} \quad (6.10)$$

Figure 6-4: Shock wave analysis of the saturated green time estimation when the storage capacity of the downstream link is limited: a) the upstream estimated maximum queue extent is greater than the available storage capacity of the downstream movement, b) the upstream estimated maximum queue extent is equal to the available storage capacity of the downstream movement, and c) the upstream estimated maximum queue extent is less than the available storage capacity of the downstream movement.
Next, for possible phase \( i \) the total estimated effective outflow rate during the determined phase time of the possible phase (including the phase’s total lost time), \( F^{p_i}(K) \), is calculated and then the maximum value of estimated total effective outflow rates at the decision point \( K \), \( f^p(K) \), is given as:

\[
f^p(K) = \max_{i=1}^I F^{p_i}(K) = \max_{i=1}^I \sum_{j=1}^J F^{m_{ij}}(K) G_i(K) + L, \quad i \in \{1, \ldots, I\},
\]

(6.11)

where \( f^p(K) \) is the maximum effective outflow rate during the nominal phase time (including the phase’s total lost time) that is estimated at the decision point \( K \); and \( L \) is the phase lost time including start-up and clearance lost times. We denote the phase associated with the maximum value of \( F^{p_i}(K) \) as \( i^* \). Hence, the selected phase time, \( G_{i^*}(K) \), is the ideal phase duration that is denoted by \( p^*(K) \).

The phase time, \( p^*(K) \), is the ideal phase duration. Nevertheless, the selected phase is not final as it may change in the two iterations introduced in Section 6.3.4 which are considered to improve the selected phase with regards to increasing the outflow of the vehicles and decreasing the effect of the long queues.\(^1\) Thus, in this step, we call the selected phase as \( P_{\text{Iter1}}(K) \). It should be pointed out that in the first iteration, only the first condition of Equation 6.10 is used to calculate the outflow of each active movement because the estimated saturated green times of all movements in each phase are greater than or equal to the selected minimum saturated green time for the phase. The second condition may be used in the following in Section 6.3.4.

### 6.3.4 Phase improvement

A phase can, at most, support a specific number of movements, called maximum movements. Suppose that the selected phase in Section 6.3.3 would contain a lower number of movements than

\(^1\)In this step, there may be two or more phases with the same or different number of movements that provide the same effective outflow as a maximum effective outflow. To reduce the unnecessary computation efforts, we ignore finding several phases to improve. Instead, we select the phase associated with the first occurrence of maximum effective outflow, and improve it later in further steps.
the maximum movements. Allowing other non-conflicting movements to be activated can provide an opportunity to increase the intersection outflow during the selected phase time $p^*(K)$. By this consideration, movements in under-saturated conditions might also appear in a phase and here the second condition of Equation (10) is used in the calculation of the outflow of the extra undersaturated movements in a phase. Consequently, we do the second iteration to explore the ideal phase with a higher number of movements (if available) based on the finalized phase time, $p^*(K)$.

To this end, first we find the possible phases that contain at least the same active movements of the selected phase in the first iteration. Here, we end up with a new set of possible phases, $P_{\text{Iter2}}(K)$, such that $P_{\text{Iter2}}(K) \subseteq P_i$. Then, we apply the same process provided in Section 6.3.3 over the new set of phases determined in this step (i.e. $P_{\text{Iter2}}(K)$) to find a phase or a set of phases with a higher effective outflow rate, i.e. $P^*_{\text{Set-Iter2}}(K)$. We should note that the additional movements added to the selected phase in Section 6.3.3, $P^*_{\text{Iter1}}(K)$, can contain queues which exceed the available storage capacity of their related downstream lane. As vehicles do not enter the intersection when facing a spillover in the downstream lane, we do not remove the phases that include these movements from the possible phases of $P_{\text{Iter2}}(K)$.

Following the process in increasing the number of activated movements in the selected phase, the method may find more than one possible phase in the set $P^*_{\text{Set-Iter2}}(K)$. To help with reducing the long queues issue, which can negatively impact the performance of the signal timing method, the method selects a phase among the phases in the set $P^*_{\text{Set-Iter2}}(K)$ that contains a larger total queue length. This is done by finding the total queue length of each possible phase in the set of possible phases, $P^*_{\text{Set-Iter2}}(K)$. Next, the phase with maximum total queue length is selected as the ideal phase.

Finally, an ideal phase is obtained at decision point $K$. However, before the phase is activated, sub-phase called interphase, $I(k)$, is required to be activated. The interphase time is equal to yellow time (e.g. 3 sec). The possible signal indications in an interphase are red, green, and yellow. Figure 6-5 provides a visual illustration of the four possible cases. The interphase is determined based on the selected phase and previous phase. States $a$, $b$, and $c$ might regularly happen in a cyclic signal...
In this section, the details of the proposed traffic control. However, in the proposed acyclic method, when the selected phase and previous phase are both green, a green interphase lets the vehicles in a movement remain discharging without any interruption and maintains the movement continuity between the two successive phases. This reduces the number of unnecessary yellow times, hence, vehicles’ stops (see Figure 6-5d). If the selected and previous phases are, respectively, red and green, the traffic light is considered to stay red during the interphase to manage the movements’ conflicts, see Figure 6-5c. We should point out that the decision point $K$ is determined through the calculation of the phase time in the previous decision point, i.e. $K - 1$. The sum of the phase and interphase times at the decision point $K - 1$ specifies the next decision point, i.e. $K$. It is apparent that decision points for each intersection are unique.

### 6.4 Numerical experiments

In this section, we conduct microsimulation experiments and provide a comprehensive quantitative evaluation by comparing our method with two benchmark methods, including a fixed-time signal control.
control, based on the Webster method [7], and the actuated signal control. The fixed signal timing in the Webster method is determined based on the average traffic demand during the study period. However, in the actuated traffic signal control method (which we call Actuated method in short form), traffic signal timing is continuously adjusted in response to real-time measures of traffic obtained from detectors installed on all the approaches. Different detectors are used to distinguish different lanes. By receiving a call from detectors, the Actuated controller, based on some pre-defined thresholds, decides whether to extend or terminate the green phase in response to the actuation source. Although the Actuated method has a better performance than the fixed-time method in most cases, it does not offer any real-time optimization to adapt to traffic fluctuations and possible spillbacks properly. Some of the Actuated method’s parameters used in this chapter are: (i) distance of detectors from the stoplines = 20 (m), (ii) minimum green time = 7 (sec), (iii) maximum green time = 53 (sec), and (iv) passage time = 3 (sec).

6.4.1 Experiment Setup

The experiments are conducted in AIMSUN microsimulation environment. To evaluate the proposed method, two cases are studied: (i) an isolated intersection and (ii) a 5 × 5 network including 25 intersections. In both cases, a typical intersection is considered, in which each approach consists of three lanes with distinct movements, including left-turn, through, and right-turn lanes, as shown in Figure 6-6a. The movements are ordered from 1 to \( J \) counterclockwise, starting at eastbound left-turn. The total number of movements in an intersection, \( J \), is 12. Moreover, the maximum number of simultaneously non-conflicting movements in a phase is 4, and the total number of identified possible phases, \( I \), is 111. The possible phases include 12 one-movement, 38 two-movement, 44 three-movement, and 17 four-movement phases.

In the isolated intersection test case, all approaches are equally 300 (m) long. A 2.5-hour time-varying demand scenario is employed, as depicted in Figure 6-6b. Thirty simulation replications
Figure 6-6: A typical intersection and traffic demands of microsimulation case studies: (a) An intersection with four incoming and four outgoing approaches, where each approach includes three distinct left-turn, through, and right-turn lanes, (b) total demand to all approaches in the isolated intersection test case, (c) medium demand in the network test case, and (d) high demand in the network test case.

are run with various random seeds with a 30-minute warm-up period at the beginning of each experiment.

In the network case, 20 external centroids at the outer links are used to generate and attract trips. In addition, 25 generative-only internal centroids are also designed to feed the network internally, which operate similar to side streets entering each link. The length of all internal links in the network is set to 350 (m). The network case study is evaluated with two different demand levels, medium and high, in 2-hour time-varying demand scenarios, see Figure 6-6c and 6-6d. 10 simulation replications are run with a 15-minute warm-up period.
Table 6.1: Extracted FD parameters for the three types of movements

<table>
<thead>
<tr>
<th>Movement</th>
<th>$S$ (veh/hr)</th>
<th>$K^{cri}$ (veh/km)</th>
<th>$K^{jam}$ (veh/km)</th>
<th>$u^a$ (km/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left turn</td>
<td>1650</td>
<td>55</td>
<td>180</td>
<td>30</td>
</tr>
<tr>
<td>Through</td>
<td>2200</td>
<td>55</td>
<td>180</td>
<td>40</td>
</tr>
<tr>
<td>Right turn</td>
<td>1800</td>
<td>60</td>
<td>180</td>
<td>30</td>
</tr>
</tbody>
</table>

Saturated green time is a key component in our proposed method, see Section 6.3.1. It is estimated based on queue lengths, arrival flow data, and FD. In both isolated and network cases, queue lengths are estimated by counting the number of stopped vehicles in each lane and multiplying it in $1/K^{jam}_j$. Stopped vehicles are defined as vehicles with a speed that is less than or equal to 5 (km/hr). During a high traffic load in the network, where recognizing the end of the queue is crucial, we use the position of the last stopped vehicle, as the end of the queue. In the simulation setup, the initial queue length is read through AIMSUN software; however, in real-world applications, this data can be collected through different methods such as traffic cameras, detectors, and probe data.

In the network case, the arrival flow estimation method described in 6.B is employed. For this purpose, a detector at the beginning of each link is installed to read the count of vehicles in each 10 sec interval and follow the arrival flow estimation process. It is assumed that the estimated arrival flow is distributed equally among the three movements (i.e., $1/3$ for each lane). The presented arrival flow estimation method for the network case in Appendix A is also applicable in the case of the isolated intersection.

The traffic signal control requires FDs; hence, we simulated 30 replications to determine three FDs for each type of movement, including left, through, and right turns. In Figure 6-7, the FD for right turns is depicted as a sample. The extracted triangular FD parameters for different movements are provided in Table 6.1.

It should be noted that all tests include en-route dynamic traffic assignment. Moreover, yellow time and lost time are considered as 3 (sec) and 4 (sec), respectively.
Figure 6-7: A sample triangular FD for the right turns in an offline process: the extracted parameters are $S = 1800$ (veh/hr), $K^{cri} = 60$ (veh/km), $K^{jam} = 180$ (veh/km), $u^a = 30$ (km/hr).

Table 6.2: Comparison of the three methods in isolated intersection case. The numbers in parentheses show the improvements in terms of % with respect to the Webster method.

<table>
<thead>
<tr>
<th></th>
<th>Proposed method</th>
<th>Actuated method</th>
<th>Webster method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average travel time (sec/km)</td>
<td>87.9 (25.7%)</td>
<td>114.2 (3.5%)</td>
<td>118.3</td>
</tr>
<tr>
<td>Average speed (km/hr)</td>
<td>42.4 (22.5%)</td>
<td>35.7 (3.2%)</td>
<td>34.6</td>
</tr>
</tbody>
</table>

6.4.2 Isolated Intersection Case Study

First, we study the isolated intersection test case. Results are shown in Figure 6-8 where the average speed and average queue length during every 30 seconds are plotted over time. By the proposed method, the average speed increases 18.8% and 22.5% compared to the Actuated method and Webster method, respectively, which is a significant improvement (see Table 6.2). Average travel time has been reduced by 23.0% and 25.7% compared to the Actuated method and Webster methods, respectively.

In Figure 6-8b, we observe that the proposed method is more efficient than other methods in reducing the average queue length considered over all lanes of the intersection. As shown in Figure 6-8b, the proposed method controls the average queue lengths under 6 (m) while the average
Figure 6-8: Comparison of (a) average speed, and (b) average queue length, in each 30 (sec) interval over 2.5 (hr).

Queue lengths in Actuated and Webster methods range from 2 (m) to 18 (m), and 2 (m) to 20 (m), respectively. In other words, a maximum of 12 vehicles are in all approaches during the experiments on average using the proposed method, while with the Actuated and Webster methods, this increases to 36 and 40 vehicles, respectively. These comparisons indicate that the proposed method is more efficient than the two other methods to control the isolated traffic signal.

Figure 6-9 demonstrates that during the 2.5-hour experiment, 3307 movements out of total of 6428 activated movements are prevented from unnecessary signal switches from green to yellow. This accounts for 51.4% of total activated movements, which have a significant impact on the intersection performance.

6.4.3 Network Case Study

Here, an experiment is performed to evaluate the effectiveness of the proposed method in the network settings. In this experiment, spillback control is integrated into the method. The results
in Figure 6-10 indicate the improved performance of the proposed method in reducing the number of spillbacks in the lanes in the network with both medium and high demands. With medium demand, the proposed method clears all spillbacks in lanes while the average number of lanes with spillback at the end of simulation in Actuated and Webster methods are 5 and 11, respectively. These numbers for the high demand for the proposed, Actuated, and Webster methods are 2.5, 10, and 17.5, respectively.

We also studied the impact of the three methods on the length and frequency of spillbacks at the lane level in the network for both medium and high demands. Eight different ranges of spillback duration were considered. Results shown in Figure 6-11 clearly support the efficiency of the proposed method in reducing the time and frequency of the spillbacks in the network. In all ranges of spillback durations, the frequency of spillbacks with the proposed method is less than the two other methods for both medium and high demands.
Comparison of network macroscopic fundamental diagrams (MFDs) of the network with the three control methods with both medium and high traffic demands show a higher production of the network when the proposed signal control method is applied (see Figures 6-12a and 6-12b). Additionally, the proposed traffic signal control method prevents the network from experiencing the congested regime.

Results depicted in Figures 6-13a and 6-13b for the average speed and 6-13c and 6-13d for the average queue length in the network also confirm the effectiveness of the proposed method. Average speed in the network using the proposed method is 19 (km/hr) with the medium demand (ignoring the data for the first minute) and 14 (km/hr) with the high demand. However, the average speed falls down to about 9 (km/hr) with the medium demand and 7 (km/hr) with the high demand using both two other signal control methods. Moreover, the average queue length in the network with medium demand peaks at 19 (m), 55 (m), and 75 (m) using the proposed, Actuated, and Webster method. For the high demand, the maximum of average queue length using the proposed, Actuated,
and Webster method are 38 (m), 83 (m), and 103 (m), respectively. Average travel time with the proposed method in medium and high traffic demands are 43.3% and 37.7% lower than the Actuated method, respectively. Compared to the Webster method, these reductions are 52.9% and 46.4%, respectively (see Table 6.3). A similar comparison shows a significant increase in average speed in the network in the proposed method compared to the Actuated method in medium demand, 58.5%,
Figure 6-13: Comparison of average speed and average queue length in the network over 120 minutes: (a) average speed (medium demand), (b) average speed (high demand), (c) average queue length (medium demand), and (d) average queue length (high demand).

and high demand, 47.9%. Compared to the Webster method, average speed in the network in the proposed method increases 90.8% in medium demand and 73.9% in high demand.

The distributions of green time duration in all replications for the proposed and Actuated methods are shown in Figures 6-14a and 6-14b. The green times are limited to the range of 7 to 53 (sec) in the Actuated method, whereas, as depicted, the proposed method provides a wider range of green times compared to the Actuated method. The maximum green times of a movement in the proposed method for medium and high demands are 249 (sec) and 411 (sec), respectively. Evidently,
Table 6.3: Comparison of the three methods in the network case. The numbers in parentheses show the improvements in terms of % with respect to the Webster method.

<table>
<thead>
<tr>
<th>Demand level</th>
<th>Proposed method</th>
<th>Actuated method</th>
<th>Webster method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average travel time (sec/km)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>medium</td>
<td>136.0 (52.9%)</td>
<td>239.6 (17.0%)</td>
<td>288.6</td>
</tr>
<tr>
<td>high</td>
<td>176.1 (46.4%)</td>
<td>282.6 (14.1%)</td>
<td>328.8</td>
</tr>
<tr>
<td>Average speed (km/hr)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>medium</td>
<td>29.0 (90.8%)</td>
<td>18.3 (20.4%)</td>
<td>15.2</td>
</tr>
<tr>
<td>high</td>
<td>24.7 (73.9%)</td>
<td>16.7 (17.6%)</td>
<td>14.2</td>
</tr>
</tbody>
</table>

with higher traffic demand, the green times required to accommodate the traffic increases, which is confirmed by the performance of the proposed signal control method. Note that one movement may remain active for 249 seconds over several phases, while the other simultaneous non-conflicting movements may change during the 249 seconds. Furthermore, the minimum green time is 2 (sec) for both medium and high demand. Although the proposed method provides a broader range of green times, the average of green time durations in the proposed method for medium and high demands are 13.9 and 16.1 (sec) lower than the average green times in the Actuated method which are 29.6 and 31.8 (sec), respectively. The shorter green time in the proposed method reflects shorter red times for conflicting movements, and hence provides less probability of spillback.

Figures 6-15a and 6-15b show a large number of times that yellow phase can be avoided by the proposed method. During a 2-hour experiment, 26147 movements out of total of 73736 activated movements of medium demand and 21826 movements out of total of 64144 activated movements of high demand were prevented from unnecessary signal switches from green to yellow. These numbers account for 35.5% and 34.0% of total activated movements, which are considerable portions of the activated movements and have a significant impact on the network performance reducing the extent of acceleration and deceleration of platoons of vehicles.

6.5 Conclusion

In this chapter, we have proposed a real-time decentralized traffic signal control method for urban networks. The proposed method is acyclic and based on lane-based queue measurements. There is no communication between intersections and all control decisions in the network are entirely local.
Figure 6-14: Comparison of green times between the proposed method and Actuated method in the network with (a) medium demand and (b) high demand.

Each intersection is independently controlled based on estimated arrival flow and queue length at the end of the previous time step collected in all lanes of the intersection. The control decisions for each intersection in the network include phase time, active movements in the phase, and interphase.

In brief, the method first finds the minimum saturated green time in all possible phases. This is to guarantee that vehicles discharge at their full capacity during a phase. In determining the minimum saturated green time, a spillback control is also applied, which considers the impact of the storage capacity of the downstream links on the control decisions. Then, the estimated effective
outflow rate during the minimum saturated green time of each phase is obtained, and the phase and phase time with maximum estimated effective outflow rate is determined. Without changing the selected phase time, the process of phase improvement is done to increase the outflow of the intersection by increasing the number of active movements in the selected phase and prioritizing longer queues. Lastly, based on the selected ideal phase and previous phase, the interphase is determined to facilitate movement continuity.

Using microsimulation, we compared our acyclic decentralized method against other well-known benchmark methods (Actuated and Webster methods) from different aspects, including average speed, average queue length, and average travel time in both isolated and network scenarios. In addition, we mainly evaluated our method in the network case in terms of lane spillback, frequency of

Figure 6-15: Average frequency of movement continuities between two successive phases in phase transition points in the network in the proposed method.
lane spillbacks, and MFD of the network. The results showed that our proposed method outperforms other benchmark methods in cases of both isolated intersection and network.

For future work, we recommend the study of the proposed method in integration with a route recommendation system. Enhancing this method by incorporating the proposed method within a hierarchical control scheme like Perimeter control is a future research direction. The impact of link length and noise in the queue length and arrival flow estimations can also be studied. Considering lanes with mixed movements is also an extension of this work. Furthermore, other elements of traffic such as transit priority, Emergency Medical Services (EMS) priority, and multi-modality can be considered to provide a comprehensive method as a solution for a complex city-scale network.

6.A  NOMENCLATURE

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J$</td>
<td>Traffic movements, $j \in {1, 2, \ldots, J}$</td>
</tr>
<tr>
<td>$J$</td>
<td>Total number of movements, for the defined test case $J = 12$,</td>
</tr>
<tr>
<td>$i$</td>
<td>Index of possible phases, $i \in {1, 2, \ldots, I}$,</td>
</tr>
<tr>
<td>$I$</td>
<td>Total number of possible phases, for the defined test case $I = 111$,</td>
</tr>
<tr>
<td>$P$</td>
<td>A pre-determined set of all possible phases in an intersection,</td>
</tr>
<tr>
<td>$P_i$</td>
<td>A phase in a pre-determined set of all possible phases in an intersection,</td>
</tr>
<tr>
<td>$p$ (unit of time)</td>
<td>Phase time,</td>
</tr>
<tr>
<td>$\lambda_{ij}$</td>
<td>Signal indications (0 for red and 1 for green signal indication) of movement $j$ in phase $i$,</td>
</tr>
<tr>
<td>$k$</td>
<td>Discrete time index, $k \in {0, 1, \ldots}$,</td>
</tr>
<tr>
<td>$I_i(k)$</td>
<td>Set of possible interphases,</td>
</tr>
<tr>
<td>$\psi^*(k)$</td>
<td>Ideal phase,</td>
</tr>
<tr>
<td>$K$</td>
<td>Decision point,</td>
</tr>
<tr>
<td>$G_i(K)$ (unit of time)</td>
<td>Minimum saturated green time greater than zero for phase $i$ at the decision point $K$,</td>
</tr>
<tr>
<td>$G^\text{sat}_j(K)$ (unit of time)</td>
<td>Saturated green time of movement $j$ at the decision point $K$,</td>
</tr>
<tr>
<td>$K_j^\text{jam}$ (veh/unit of distance)</td>
<td>Jam density of movement $j$,</td>
</tr>
<tr>
<td>$S_j$ (veh/unit of time)</td>
<td>Saturation flow of movement $j$,</td>
</tr>
<tr>
<td>$x_j^0(K)$ (unit of distance)</td>
<td>Queue length in movement $j$ at the decision point $K$,</td>
</tr>
<tr>
<td>$K^\text{arr}_j(K)$ (veh/unit of distance)</td>
<td>Arrival density of movement $j$ at the decision point $K$,</td>
</tr>
<tr>
<td>$q^\text{arr}_j(K)$ (veh/unit of time)</td>
<td>Arrival flow of movement $j$ at the decision point $K$,</td>
</tr>
<tr>
<td>$K^\text{crit}_j$ (veh/unit of distance)</td>
<td>Critical density of movement $j$,</td>
</tr>
<tr>
<td>$SWL_j(K)$ (unit of distance/unit of time)</td>
<td>Backward-moving estimated shock wave formed between arriving vehicles and queued vehicles,</td>
</tr>
<tr>
<td>$SWR_j(K)$ (unit of distance/unit of time)</td>
<td>Backward-moving estimated shock wave formed between queued vehicles and vehicles leaving queue at saturation,</td>
</tr>
<tr>
<td>$SWN_j(K)$ (unit of distance/unit of time)</td>
<td>Forward-moving estimated shock wave formed between arriving vehicles and vehicles leaving queue at saturation,</td>
</tr>
<tr>
<td>$l$ (unit of distance)</td>
<td>Link length,</td>
</tr>
<tr>
<td>$x_j^M(K)$ (unit of distance)</td>
<td>Maximum queue extent in movement $j$ at the decision point $K$,</td>
</tr>
<tr>
<td>$G^\text{d}_j(K)$ (unit of time)</td>
<td>Available time capacity of the link downstream of movement $j$</td>
</tr>
</tbody>
</table>
6.B Arrival flow estimation

For arrival flow estimation, we use a discretization approach in which we consider two effective factors: (i) the queue length in each lane at phase transition point and (ii) the time that it takes the platoon of vehicles to reach the end of the queue. First, we consider a detection interval, $d$ (sec).

Based on the detection interval and free flow speed $u^a$ (m/sec) of the link, the maximum distance, $D$ (m), that can be travelled in each link during each detection interval is $D = d \cdot u^a$. Accordingly, we divide the link length to $\eta$ sections, where $\eta = \frac{D}{d}$. For each section, a matrix is considered. When the count of vehicles are read at the upstream of the link at the end of each detection interval, the count data is converted to flow data and is stored in the matrix $q_{j,\eta}^{arr}(K)$. At the end of the next detection interval, this data is moved to the matrix in the adjacent downstream section, $q_{j,\eta-1}^{arr}(K)$.
and the new read data is stored in the matrix $q_{\text{arr}}(K)$. This process continues to build $q_{\text{arr-set}}(K) = \{q_{\text{arr}}^{1}(K), q_{\text{arr}}^{2}(K), ..., q_{\text{arr}}^{\eta}(K)\}$, including $\eta$ matrices as below:

$$
q_{\text{arr-set}}(K) = \begin{cases} 
q_{\text{arr}}^{1}(K) := q_{\text{arr}}^{2}(K - 1), \\
q_{\text{arr}}^{2}(K) := q_{\text{arr}}^{3}(K - 1), \\
\vdots \\
q_{\text{arr}}^{\eta-1}(K) := q_{\text{arr}}^{\eta}(K - 1), \\
q_{\text{arr}}^{\eta}(K) := q_{\text{arr}}^{\text{input}}(K),
\end{cases}
$$

(6.12)

where $q_{\text{arr-set}}(K)$ is the set of arrival flow matrices at the time step $K$ for each lane; and $q_{\text{arr}}^{1}(K)$, $q_{\text{arr}}^{\eta}(K)$, $q_{\text{arr}}^{\text{input}}(K)$, respectively, are the matrices for storing the data of the most upstream section, the data of the most downstream section, and the data which is read at the end of each detection interval at the upstream of the link.

Depending on the position of the last queued vehicle at the end of the detection interval, i.e. the estimated queue length $x^{0}_{j}(K)$, the related matrix, $q_{\text{arr}}^{0}(K) \subseteq q_{\text{arr-set}}(K)$, based on the following conditions is selected to determine the arrival flow, which ensures to a high degree that the approaching platoon of vehicles will reach the end of the queue.
\[
q_j^{\text{arr}}(K) = \begin{cases} 
q_j^{\text{arr}}(K) & \text{if } 0 \leq x_j^0(K) < D_{j,1}(K) \\
q_j^{\text{arr}}(K) & \text{if } D_{j,1}(K) \leq x_j^0(K) < D_{j,2}(K) \\
& \ldots \\
q_{j,n-1}^{\text{arr}}(K) & \text{if } D_{j,n-2}(K) \leq x_j^0(K) < D_{j,n-1}(K) \\
q_{j,n}^{\text{arr}}(K) & \text{if } D_{j,n-1}(K) \leq x_j^0(K) < D_{j,n}(K). 
\end{cases}
\]

(6.13)


Part III

Bi-modal Network-scale Traffic
Signal Control: Application of Reinforcement Learning
Abstract
Improvement of traffic signal control (TSC) efficiency has been found to lead to improved urban transportation and enhanced quality of life. Recently, the use of reinforcement learning (RL) has gained traction as the next most viable improvement in TSC. We conducted a systematic literature review to dissect the existing research that applied RL in the area of TSC, in an effort to provide statistical and conceptual knowledge and identify past and present trends and a future roadmap for research in this area. We only targeted the network-scale papers that tested the proposed methods in networks with two or more intersections. Based on our study, the first article in this area was published in 1994. This review covers 160 peer-reviewed articles published from 1994 to March 2020. The current research is the most comprehensive systematic literature review that addresses the most important and fundamental characters and components of the existing methods in the area. We provide an in-depth analysis on the important components, including: the publication and authors' data, traffic simulation and evaluation, the methods, authors' highlights and implications for future study. This chapter is aimed at paving the way for new researchers, while also providing current researchers with a background on which to build in order to advance the field of study. The findings are obtained from the qualitative and descriptive data analysis on data extracted from the included articles.

7.1 Introduction
With an explosion in urban and rural population rates, city transportation systems become less efficient at handling the ever-growing numbers of commuters. A lack of space and resources with which to improve infrastructure poses problems in accommodating the increasing urban population.
The resulting congestion further leads to increased pollution caused by sitting idle in traffic jams; traffic delays and bottlenecks; and a rise in accidents. The secondary issues that arise are just as severe, including: economic loss, and an overall decrease in quality of life. This presents the problem of improving traffic flow and traffic signal control (TSC) within the already existing infrastructure.

Traffic signal control is most often regularised through a fixed time, actuated, or adaptive signal (whether the state-of-the-art methods or the methods deployed in the real-world, such as SCATS [1], SCOOT [2], and TUC [3] signal controls).

Fixed time signal control involves a repeating pattern that does not change with the live traffic situation and which continues through its cycles regardless of the traffic in that area. The actuated control method controls and also operates traffic signals based on real-time data of loop detectors. Despite being traffic responsive, the actuated control method is not designed to fully address fluctuating traffic demands, thereby rendering it less than optimal, specifically in highly saturated volumes. Conversely, an adaptive signal is a more efficient solution as it has the built-in capacity to adapt to traffic changes without the restrictions that plague the actuated method. Reinforcement Learning (RL) [4] in TSC is located in the category of the adaptive methods, but it operates differently. While adaptive signals focus on fulfilling constraints on a success-only path, RL allows failure in an effort to learn an optimal solution (like humans and animals).

The literature presents numerous studies exploring intelligent methods of optimizing TSC, including RL, which allows natural and artificial systems to learn and adapt to their environment. Derived from the natural learning processes observed in animals, RL is based on a reward system that promotes long-term goals in an unknown or changing environment. Using Pavlov’s behaviour experiments, RL has proven to be successful in training animals. RL has many different aspects (also referred to as building blocks) that, when combined in a multitude of ways, can result in a program where the learning is drastically different from its competitor. Therefore, there are a large variety of combinations to explore as potential solutions to the presented TSC problem. The changing aspects relate to how the algorithm learns, to how it thinks, or to how it uses data to arrive at an optimal policy.
To train an RL agent, they are placed in an unfamiliar environment requiring interaction, and observed. The agent is given little to no information about their environment and is required to complete tasks therein or presented with an overall goal that must be achieved. All incoming information is gleaned through sensors, cameras and other informational devices directly in the vehicles. Rewards are set based on predetermined long-term goals, multiple of specific, that the agent works towards. Multiple rewards may even be delivered for different aspects of each action. Learning, for the agent takes place through the interactions in the environment and by receiving or being denied rewards; essentially, a feedback cycle of state, action and reward.

Some large-scale environments have a single agent in the RL environment, however it is common to have multiple agents work either cooperatively or competitively, in what is called Multi-Agent Reinforcement Learning (MARL). The benefit here is that agents work across a large environment while still having the precision of a single agent. Often a multi-agent learning approach assigns an agent to the smallest breakdown of the environment, such as an intersection. The agents are organized for data sharing and cooperation, resulting in converging on an optimal solution faster than a single agent is capable of on their own.

Due to the rising popularity of RL in TSC, we aim to thoroughly characterize the existing research in the area of urban traffic networks where RL is applied, and to provide a complete account of what has already been explored. To this end, we exclude the research that has only proposed or tested for single isolated intersection control. To the best of our knowledge, there is no such systematic literature review aimed at examining the existing research of RL in TSC. The primary difference between our work and current review papers is in the area of a comprehensive systematic literature review methodology - seen in our use of well-defined inclusion/exclusion criteria, along with extracting insights and interesting findings using data analysis techniques from the data reviewed.

It is worth noting, however, that there are several surveys and review papers that do cover this area. For instance, [5, 6, 7, 8, 9, 10] all present general reviews or surveys on TSC methods, compiling a list of the most recent methods and algorithms related to RL in TSC. Additionally,
two very recent papers, [11, 12], discuss the applications and opportunities regarding Deep RL in TSC, while a number of relevant studies that do not exclusively focus on RL in TSC, e.g. [13, 14, 15, 16, 17, 18, 19, 20] were found to exist.

After completing a literature review, this chapter is organized as follows: Section two will explore the methodology used to conduct this review, including the search strategy, selection criteria, and data extraction; Section three delves into the results and findings presented in section one; Section four covers threats to the validity of our work; and Section five wraps up with a look at future implications of the current research.

7.2 Review Method

In this study, search strategy, inclusion and exclusion criteria, and data extraction are intertwined, and the selection criteria and data extraction are performed within the search and search evaluation steps; hence, we explore this as a whole, rather than separately. It should be noted that to identify the relevant literature we conducted both manual and automated searches, and included all articles published up to the end of March 2020. We designed and implemented our review based on the guidelines provided by [21].

7.2.1 Search strategy, selection criteria, and data extraction

To identify the most appropriate search terms and strings for the automated search, we used literature review, manual content analysis, and Natural Language Processing (NLP) in the steps that follow.

1: Based on knowledge in the area of the study, we identified seven venues, including five journals and two conferences as listed in Table 7.1, and manually searched for and reviewed the relevant published studies from 2017 to 2019 based on their title, abstract, and keywords. In very limited cases, we also examined the conclusion and searched for specific, relevant key terms to help with the inclusion decision. Fifteen pertinent articles were found during this first stage search. These
papers were also used in the fourth step to form the Quasi-Gold Standard (QGS) [22] to evaluate the quality of the search strings.

2: We performed NLP, including language modelling and lexical association analysis [23], to extract the most frequently used terms in the 15 retrieved articles for the purpose of identifying search strings. Figure 7-1 shows the directed graph of common bi-grams formed from the QGS set, based on the frequency analysis on the pre-processed texts collected from the title, abstract, introduction, section/subsection headings, and conclusion of the 15 articles. Based on the results from the NLP, several inspections and investigations, and using various combinations of the most related search terms, the following single search string was chosen: \((\text{reward AND action AND (traffic light OR traffic signal)} \text{ AND reinforcement learning)}\).

3: Using the identified search string we queried Google Scholar, which served as our main search base, yielding 2,887 articles. To increase the reliability of our findings, the search was complemented by searching within Web of Science, IEEE Xplore, ACM DigitalLibrary, SpringerLink, and ScienceDirect databases after the data extraction. This accelerated our search process by providing a view of those papers that were included and excluded, based on the defined criteria.

4: We formed the Quasi-Gold Standard. Our automated search retrieved all 15 articles, indicating the quasi-sensitivity of 100%.

5: We began the process of inclusion and exclusion by carefully defining the selection criteria. This iterative process continued even during data extraction, (i.e. the last step) to ensure the correctness of the included and excluded papers. Figure 7-2 depicts the flowchart of the article
selection process in this chapter. Publications that meet the following criteria were excluded: (1) those articles not related to RL in TSC, (2) duplicate papers, (3) review and survey papers, (4) presentations, abstracts, extended abstracts, viewpoints, letters, reports, technical reports, projects, table of contents, any papers that have not been peer reviewed, theses and dissertations, books, and book chapters, (5) publications in any language other than English, (6) those papers where a better or updated version was found in another journal or publication and used, and (7) unavailability of the full-text of the paper on the internet.

Figure 7-1: Directed graph of common bi-grams resulted from the NLP that is used to define the search strings.
Since the scope of our research is network-scale, henceforth referred to as TSC (i.e., any scales involving two or more intersections), another important exclusion criterion used was those papers that propose or test a proposed method in a single isolated intersection only.

We included those papers containing an RL method, whether it was the core or combined method applied in the context of TSC. These were included whether or not they provided evaluation and simulation, and if they provided only a framework.

This investigation is designed to cover the papers that apply RL in controlling traffic signals, and as such, if RL is used to control only vehicles, it is out of scope for our chapter. In essence,
as long as RL is being applied in TSC, whether vehicle control is involved or not, the paper was included in the current study. Hence, the connected and autonomous vehicle (CAV) environment is covered in our research as long as TSC is involved. The CAV in the current study is inclusive of all environments of cases of CAV, including: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-vehicle (I2V).

Despite the fact that below-mentioned research areas have traffic components (including: public transit, bikes, and pedestrians) in common with our research, they are out of scope since TSC is not their main focus as it is in the current study. The research areas include: ramp metering, freeway, traffic control (not TSC), public transit (where the focus is on bus control), route choice, routing systems, pedestrian routing, reactions of cyclists to speed advice, ride-sharing, best path selection, lane changing, autonomous intersection, traffic congestion detection, driver behaviour, traffic signal control simulation, simulator, simulation environment, online calibration, traffic assignment problems, couriers management in express systems, fleet management, toll plaza, traffic analytics, traffic control architecture based on fog computing paradigm where the focus is on fog paradigm (not TSC), image-based learning, image processing, and optimizing the sensor installation locations in a traffic network. It must also be noted that if a study includes both TSC and either ramp metering, public transit, emergency vehicles, or fog computing in TSC, it is included.

6: After applying the selection criteria and identifying the included papers, we used both, backward snowballing by screening the reference list of the included papers, and forward snowballing [24] by scanning the citation of the included papers; see Figure 7-2.

7: In addition to Google Scholar, five other electronic databases were searched, yielding no new articles. All articles found here had already been covered in the search using Google Scholar and the snowballing method.

8: From the included papers, data were extracted and grouped into the following four categories: (1) publication and authors’ data: publication venue, type of publication venue, year of publication, number of publications, authors’ country, number of authors, authors’ university department, and authors’ affiliation, (2) traffic simulation and evaluation: number of intersections, road
network type, type of map, location of real-world map, lanes and turns configuration, data collection method, traffic simulator, presence of evaluation, compared methods, performance measures, saturation and spillback considerations, and (3) method identification and analysis: identifying core and combined RL methods, core non-RL methods, and combined non-RL methods, centralization (centralized, decentralized, and hierarchical methods), state, action, and reward (including elements in state definition, elements in reward definition, and types of actions), action selection methods and parameters, state- and action-space discretization, tabular and approximation based methods, deep reinforcement learning based methods, and availability of code/implementation, and (4) authors’ highlights and future works: authors’ special and distinctive statements usually stated as "to the best of our knowledge", and future work.

In the following section, we provide an analysis of the data we extracted. In Table 7.2, we present an overall comparison along with details of some of the features mentioned above for all 160 papers included in our study. The papers that were reviewed are listed in the section entitled Included Articles, and are referenced as [A1], [A2], etc. These numbers can be directly mapped to Table 7.2, the main text of the manuscript, as well as the Appendix section. Other resources cited in this chapter are listed in the reference section.

7.3 Results and Findings

7.3.1 Publication and authors data

Publication venue and type of publication venue

The 160 included articles mentioned above have been published in 104 different publication venues. The International Conference on Intelligent Transportation Systems (ITSC) (18 papers), IEEE Transactions on Intelligent Transportation Systems (7 papers), Transportation Research Part C (6 papers), Engineering Applications of Artificial Intelligence (5 papers), International Conference on Machine Learning (4 papers), and Journal of Intelligent Transportation Systems (4 papers)
contained the greatest numbers of publications. 57% and 39% of the papers appeared in conferences and journals, respectively, whereas the remaining 4% were published in workshops and symposia.
Table 7.2: Overall comparison of the included articles. [✓/✗]: yes for discrete methods and no for continuous methods.

<table>
<thead>
<tr>
<th>Citation</th>
<th>Method</th>
<th>Discretization</th>
<th>Approx. Method</th>
<th>Action Selection</th>
<th>Action Type</th>
<th>Discrimination</th>
<th>Spillback or Sat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mikami et al., 1994 [25]</td>
<td>AD1,V4</td>
<td>LA</td>
<td>Genetic Alg.</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
</tr>
<tr>
<td>Cao et al., 1999 [26]</td>
<td>AD1,V4</td>
<td>LCS</td>
<td>Fuzzy</td>
<td>–</td>
<td>Credit Assignment</td>
<td>7</td>
<td>✗</td>
</tr>
<tr>
<td>Cao et al., 2000 [27]</td>
<td>AD1,V4</td>
<td>LCS</td>
<td>–</td>
<td>–</td>
<td>Credit Assignment</td>
<td>7</td>
<td>✗</td>
</tr>
<tr>
<td>Wiering, 2000 [28]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Camponogaro et al., 2003 [29]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Wiering et al., 2004 [31]</td>
<td>AD1,V4</td>
<td>Model-based RL</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>5</td>
<td>✗</td>
</tr>
<tr>
<td>Dowling et al., 2004 [32]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>IA</td>
<td>6</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Ling et al., 2005 [33]</td>
<td>AD7,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>IA</td>
<td>7</td>
<td>✗</td>
<td>Neither</td>
</tr>
<tr>
<td>Da Silva et al., 2006 [34]</td>
<td>AD1,V4</td>
<td>Model-based RL</td>
<td>–</td>
<td>–</td>
<td>3</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>De Oliveira et al., 2006 [36]</td>
<td>AD1,V2</td>
<td>RL</td>
<td>–</td>
<td>Linear and Non-linear FAs</td>
<td>Softmax</td>
<td>7</td>
<td>Spillback</td>
</tr>
<tr>
<td>Richter et al., 2007 [37]</td>
<td>AD1,V2</td>
<td>Q-Learning</td>
<td>SensorGrid</td>
<td>CMAC</td>
<td>Softmax</td>
<td>6</td>
<td>✗</td>
</tr>
<tr>
<td>Su et al., 2007 [38]</td>
<td>AD1,V4</td>
<td>Policy-Gradient</td>
<td>CHF</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Zhang et al., 2007 [39]</td>
<td>AD1,V4</td>
<td>LA</td>
<td>–</td>
<td>–</td>
<td>Multiple</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Bazzan et al., 2007 [40]</td>
<td>AD1,V4</td>
<td>Policy-Gradient</td>
<td>Linear FA</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Richter et al., 2007 [41]</td>
<td>AD1,V4</td>
<td>Policy-Gradient</td>
<td>Linear FA</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Karyer et al., 2008 [42]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>Max-plus</td>
<td>e-greedy</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Salkham et al., 2008 [43]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>Softmax</td>
<td>5</td>
<td>✗</td>
<td>Neither</td>
</tr>
<tr>
<td>Li et al., 2008 [44]</td>
<td>AD1,V4</td>
<td>ADP</td>
<td>–</td>
<td>–</td>
<td>3</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Li et al., 2008 [45]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>Multiband</td>
<td>e-greedy</td>
<td>6</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Li et al., 2008 [46]</td>
<td>AD1,V4</td>
<td>ADP</td>
<td>–</td>
<td>–</td>
<td>3</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Xinhai et al., 2009 [47]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Cai et al., 2009 [48]</td>
<td>AD1,V4</td>
<td>ADP</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Dusparic et al., 2009 [49]</td>
<td>AD1,V4</td>
<td>W-Learning</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Zhao et al., 2009 [50]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>GT</td>
<td>BPNN</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Dusparic et al., 2009 [51]</td>
<td>AD1,V4</td>
<td>W-Learning</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Wu et al., 2009 [52]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Dusparic et al., 2009 [53]</td>
<td>AD1,V4</td>
<td>W-Learning</td>
<td>–</td>
<td>–</td>
<td>5</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Prothmann et al., 2009 [54]</td>
<td>AD1,V4</td>
<td>LCS</td>
<td>Evolutionary Alg.</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
</tr>
<tr>
<td>Medina et al., 2010 [55]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Multiple</td>
<td>7</td>
<td>Spillback</td>
</tr>
<tr>
<td>prashanth et al., 2010 [56]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Bazzan et al., 2010 [57]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>3</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Ard et al., 2010 [58]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>FFNN</td>
<td>–</td>
<td>6</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>Danarynejad et al., 2010 [59]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>Fuzzy Granulation</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Dai et al., 2010 [60]</td>
<td>AD1,V4</td>
<td>ADP</td>
<td>–</td>
<td>NN</td>
<td>7</td>
<td>✗</td>
<td>Spillback</td>
</tr>
<tr>
<td>Bhat et al., 2010 [61]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>5</td>
<td>✗</td>
<td>Spillback</td>
</tr>
<tr>
<td>Natarajan et al., 2010 [62]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>Softmax</td>
<td>7</td>
<td>✗</td>
<td>Neither</td>
</tr>
<tr>
<td>Waskow et al., 2010 [63]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>Tile Coding</td>
<td>3</td>
<td>✗</td>
<td>Sat</td>
</tr>
<tr>
<td>El-Tantawy et al., 2010 [64]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>GT (Coaviour)</td>
<td>–</td>
<td>7</td>
<td>✗</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Citation</td>
<td>Method</td>
<td>RL Method</td>
<td>Combined Method</td>
<td>Approx. Method</td>
<td>Act. Selection</td>
<td>Action Type</td>
<td>Optimization</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>---------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Salkham et al., 2010 [65]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>e-greedy</td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>Houli et al., 2010 [66]</td>
<td>AD1,V1</td>
<td>Model-based RL</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Prashanth et al., 2011 [67]</td>
<td>AD1,V1</td>
<td>Multiple RL</td>
<td>Average Cost Alg</td>
<td></td>
<td></td>
<td>6</td>
<td>✓</td>
</tr>
<tr>
<td>Song et al., 2011 [68]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Li et al., 2011 [69]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>LU et al., 2011 [70]</td>
<td>AD1,V4</td>
<td>Actor-Critic</td>
<td>Swarm Optimization</td>
<td></td>
<td></td>
<td>3</td>
<td>✓</td>
</tr>
<tr>
<td>Natarjan et al., 2011 [71]</td>
<td>AD1,V5</td>
<td>Inverse RL</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Heinen et al., 2011 [72]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Abdoos et al., 2011 [73]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Dusparic et al., 2012 [75]</td>
<td>AD1,V1</td>
<td>RL</td>
<td>FFNN</td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Medina et al., 2012 [76]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>Max-plus</td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Khamis et al., 2012 [77]</td>
<td>AD1,V2</td>
<td>Model-based RL</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Khamis et al., 2012 [78]</td>
<td>AD1,V1</td>
<td>Model-based RL</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Jadhao et al., 2012 [79]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Xu et al., 2013 [80]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>Softmax</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Moghadam et al., 2013 [81]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>Random</td>
<td>3</td>
<td>×</td>
</tr>
<tr>
<td>Chin et al., 2013 [82]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>e-greedy</td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>Nuli et al., 2013 [83]</td>
<td>AD1,V1</td>
<td>Actor-Critic</td>
<td></td>
<td></td>
<td>e-greedy</td>
<td>5</td>
<td>×</td>
</tr>
<tr>
<td>El-Tantawy et al., 2013 [84]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Abdoos et al., 2013 [85]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>Softmax</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Box et al., 2013 [86]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>Tile Coding</td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Pham et al., 2013 [87]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Brys et al., 2014 [88]</td>
<td>AD1,V3</td>
<td>Q-Learning</td>
<td>Tile Coding</td>
<td></td>
<td></td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Prabuchandnan et al., 2014 [89]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Chanloha et al., 2014 [90]</td>
<td>AD1,V3</td>
<td>Q-Learning</td>
<td>CTM</td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Faghi et al., 2014 [91]</td>
<td>AD1,V2</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Marsetti et al., 2014 [92]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Khamis et al., 2014 [93]</td>
<td>AD1,V1</td>
<td>Model-based RL</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>El-Tantawy et al., 2014 [94]</td>
<td>AD1,V1</td>
<td>Multiple RL</td>
<td></td>
<td></td>
<td>Multiple</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Sadigh et al., 2014 [95]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>Edge Computing</td>
<td>Gradient-descent</td>
<td></td>
<td>3</td>
<td>✓</td>
</tr>
<tr>
<td>Abdoos et al., 2014 [96]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Lin et al., 2014 [97]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>Edge Computing</td>
<td>Gradient-descent</td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Jadhao et al., 2014 [98]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>Random</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Tao et al., 2014 [99]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>Xiang et al., 2015 [100]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>Grey Theory</td>
<td>BPNN</td>
<td>Greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Xu et al., 2015 [101]</td>
<td>AD1,V1</td>
<td>RL</td>
<td></td>
<td></td>
<td>Not clear</td>
<td>–</td>
<td>No evaluation</td>
</tr>
<tr>
<td>Yin et al., 2015 [102]</td>
<td>AD1,V1</td>
<td>ADP</td>
<td>Gradient-descent</td>
<td></td>
<td></td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Araghchi et al., 2015 [103]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td></td>
<td></td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Ajalou et al., 2015 [104]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>CTM</td>
<td></td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Citation</td>
<td>Method</td>
<td>RL Method</td>
<td>Combined Method</td>
<td>Approx. Method</td>
<td>Act. Selection</td>
<td>Action Type</td>
<td>Quantization</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Ozan et al., 2015 [105]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Random</td>
<td>6</td>
<td>×</td>
</tr>
<tr>
<td>Zhu et al., 2015 [106]</td>
<td>AD1,V2</td>
<td>RL</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>El-Tantawy et al., 2015</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Prabuchandran et al., 2015</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>2</td>
<td>×</td>
</tr>
<tr>
<td>Abdelgawad et al., 2015</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-softmax</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Mashayekhi et al., 2015</td>
<td>AD1,V3</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Abdos et al., 2015 [111]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>5</td>
<td>×</td>
</tr>
<tr>
<td>Tahfo et al., 2015 [112]</td>
<td>AD1,V4</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>3</td>
<td>×</td>
</tr>
<tr>
<td>Van der Pol et al., 2016</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Dusparik et al., 2016</td>
<td>AD1,V1</td>
<td>W-Learning</td>
<td>–</td>
<td>–</td>
<td>Softmax</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Gaikwad et al., 2016</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>3</td>
<td>✓</td>
</tr>
<tr>
<td>Prashanth et al., 2016</td>
<td>AD1,V3</td>
<td>Actor-Critic</td>
<td>–</td>
<td>–</td>
<td>SPSA, SF</td>
<td>Softmax</td>
<td>7</td>
</tr>
<tr>
<td>Iyer et al., 2016 [117]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Greedy</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al., 2016 [118]</td>
<td>AD1,V2</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Pro-CEP Model</td>
<td>DQF</td>
<td>Not clear</td>
</tr>
<tr>
<td>Prashanth et al., 2016</td>
<td>AD1,V3</td>
<td>RS RL</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Chu et al., 2016 [120]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>UCB</td>
<td>7</td>
</tr>
<tr>
<td>Shen et al., 2016 [121]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Cellular Automata</td>
<td>WNN</td>
<td>7</td>
</tr>
<tr>
<td>Chu et al., 2016 [122]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Linear Q-function</td>
<td>e-greedy</td>
<td>7</td>
</tr>
<tr>
<td>Rosyadi et al., 2016</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Yin et al., 2016 [124]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Linear FA</td>
<td>–</td>
<td>×</td>
</tr>
<tr>
<td>Chen et al., 2017 [125]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Kristensen et al., 2017</td>
<td>AD5,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>JADE</td>
<td>Greedy</td>
<td>7</td>
</tr>
<tr>
<td>Darmoul et al., 2017</td>
<td>AD1,V2</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Immune Network</td>
<td>–</td>
<td>6</td>
</tr>
<tr>
<td>Jin et al., 2017 [128]</td>
<td>AD1,V1</td>
<td>SARSAA</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Liu et al., 2017 [129]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Dynamic Clustering</td>
<td>Gradient-descent, Tile Coding</td>
<td>7</td>
</tr>
<tr>
<td>El Hatri et al., 2017</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>4</td>
<td>×</td>
</tr>
<tr>
<td>Lu et al., 2017 [131]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Softmax</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Chu et al., 2017 [132]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Max-sum</td>
<td>LR, SR</td>
<td>e-greedy</td>
</tr>
<tr>
<td>Liu et al., 2017 [133]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Vidhate et al., 2017</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Aslani et al., 2017</td>
<td>AD1,V1</td>
<td>Actor-Critic</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Tile Coding, RBF</td>
<td>e-greedy</td>
</tr>
<tr>
<td>Al Islam et al., 2018</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>×</td>
</tr>
<tr>
<td>Daeichian et al., 2018</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Fuzzy, GT</td>
<td>–</td>
<td>3</td>
</tr>
<tr>
<td>Wei et al., 2018 [138]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>Phase Gate</td>
<td>e-greedy</td>
<td>7 ×</td>
</tr>
<tr>
<td>Aslani et al., 2018</td>
<td>AD1,V1</td>
<td>Multiple RL</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Tile Coding, RBF</td>
<td>e-greedy</td>
</tr>
<tr>
<td>Aslani et al., 2018 [139]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>e-greedy</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Li et al., 2018 [142]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>DAE</td>
<td>e-greedy</td>
<td>5 ×</td>
</tr>
<tr>
<td>Kao et al., 2018 [143]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>5 ×</td>
<td>×</td>
</tr>
<tr>
<td>Liu et al., 2018 [144]</td>
<td>AD1,V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>NN</td>
<td>e-greedy</td>
<td>5 ×</td>
</tr>
<tr>
<td>Citation</td>
<td>Method</td>
<td>RL Method</td>
<td>Combined Method</td>
<td>Approx. Method</td>
<td>Act. Selection</td>
<td>Mixes Type</td>
<td>Quantization</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------</td>
<td>---------------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Torabi et al., 2018</td>
<td>AD1-V1</td>
<td>RL</td>
<td>Self-adaptive TSC</td>
<td>–</td>
<td>Act 1</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al., 2018</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>Bayesian Optimization</td>
<td>–</td>
<td>–</td>
<td>6</td>
<td>x</td>
</tr>
<tr>
<td>Shi et al., 2018</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>Transfer Learning</td>
<td>RNN</td>
<td>ε-greedy</td>
<td>7</td>
<td>x</td>
</tr>
<tr>
<td>Jin et al., 2018</td>
<td>AD1-V1</td>
<td>SARSA</td>
<td>–</td>
<td>KNN</td>
<td>ε-greedy</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Vintis et al., 2018</td>
<td>AD1-V1</td>
<td>RL</td>
<td>–</td>
<td>NN</td>
<td>–</td>
<td>Multiple</td>
<td>x</td>
</tr>
<tr>
<td>Lemus et al., 2018</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>–</td>
<td>ε-greedy</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Nishi et al., 2018</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>GCN</td>
<td>–</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Jin et al., 2019</td>
<td>AD1-V1</td>
<td>SARSA</td>
<td>–</td>
<td>LC,KNN,CMAC,FFNN,CQF</td>
<td>ε-greedy</td>
<td>3</td>
<td>x</td>
</tr>
<tr>
<td>Reda et al., 2018</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>NN,GAN</td>
<td>Softmax</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Lemos et al., 2018</td>
<td>AD1-V1</td>
<td>RL</td>
<td>–</td>
<td>–</td>
<td>Multiple</td>
<td>6</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al., 2018</td>
<td>AD1-V1</td>
<td>RL</td>
<td>Fog Computing</td>
<td>–</td>
<td>CNN</td>
<td>ε-greedy</td>
<td>7</td>
</tr>
<tr>
<td>Chen et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>CNN</td>
<td>Greedy</td>
<td>7</td>
<td>x</td>
</tr>
<tr>
<td>Rizzo et al., 2019</td>
<td>AD1-V1</td>
<td>Policy Gradient</td>
<td>–</td>
<td>DNN</td>
<td>Softmax</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Aslani et al., 2019</td>
<td>AD1-V1</td>
<td>Actor-Critic</td>
<td>–</td>
<td>–</td>
<td>Comparison</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Higuera et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>DNN</td>
<td>–</td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>Xu et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>Transfer Learning</td>
<td>LSTM</td>
<td>–</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Rizzo et al., 2018</td>
<td>AD1-V1</td>
<td>Policy-Gradient</td>
<td>–</td>
<td>DNN</td>
<td>Softmax</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Zheng et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>FRAP Model Design</td>
<td>DNN</td>
<td>–</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Xu et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>DCN</td>
<td>–</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Zhou et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>Edge Computing</td>
<td>DNN</td>
<td>ε-greedy</td>
<td>Act 1</td>
<td>✓</td>
</tr>
<tr>
<td>Hodari et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>Cellular Automata</td>
<td>–</td>
<td>Not clear</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Niu et al., 2019</td>
<td>AD1-V1</td>
<td>Actor-Critic</td>
<td>–</td>
<td>GCN</td>
<td>ε-greedy</td>
<td>Act 2</td>
<td>x</td>
</tr>
<tr>
<td>Ge et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>DCN</td>
<td>ε-greedy</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Shu et al., 2019</td>
<td>AD1-V1</td>
<td>RL</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>x</td>
</tr>
<tr>
<td>Kim et al., 2019</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>DNN-CNN</td>
<td>Softmax</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Lee et al., 2020</td>
<td>AD1-V1</td>
<td>SARSA</td>
<td>–</td>
<td>DNN</td>
<td>–</td>
<td>2</td>
<td>✓</td>
</tr>
<tr>
<td>Reda et al., 2019</td>
<td>AD1-V1</td>
<td>RL</td>
<td>Multiple RL</td>
<td>FFNN</td>
<td>Random</td>
<td>7</td>
<td>x</td>
</tr>
<tr>
<td>Xu et al., 2020</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>RNN</td>
<td>ε-greedy</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Qu et al., 2020</td>
<td>AD1-V1</td>
<td>RL</td>
<td>RMS-NE, CTM</td>
<td>–</td>
<td>Softmax</td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>Zhao et al., 2020</td>
<td>AD1-V1</td>
<td>RL</td>
<td>Junction Tree Alg.</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>✓</td>
</tr>
<tr>
<td>Kim et al., 2020</td>
<td>AD1-V1</td>
<td>Q-Learning</td>
<td>–</td>
<td>Dueling NN</td>
<td>ε-greedy</td>
<td>7</td>
<td>x</td>
</tr>
</tbody>
</table>
The results of our analysis show that the first paper [25] in the area was published in 1994 and proposed a learning scheme that combines RL of a local agent with global optimization by Genetic Algorithms where the idea is the periodic modification of parameters for RL by a genetic search. The paper aimed to blend two conflicting objectives, including distributed learning of agents and cooperation among agents in a large-scale and multi-agent type real-time planning situation to achieve cooperation in the long term without focusing on autonomy. This approach was tested in a network of nine intersections.

The second paper [26] was published in 1999 to provide a faster method than hierarchical control algorithms to react to varying conditions by designing an intelligent local controller primarily for distributed traffic control systems. To this end, a classifier system with a fuzzy rule representation, with both evolutionary and reinforcement learning, is used to determine useful junction control rules within the dynamic environment. Instead of using fuzzy rules, this work utilizes a fuzzy coding strategy for rule generation.

In 2000, two papers were published. One [27] intended to determine useful control rules within the dynamic traffic environment, and thus improve the traffic conditions by incorporating learning classifier systems and TCP/IP based communication server (supporting the communication in the control system) into a distributed learning control strategy for traffic signals. The objective is to increase the speed of response of the local controller to changes in the environment.

The second one [28] intended to minimize the overall waiting time of vehicles in a city (tested in a 6-intersection network) by proposing a set of multi-agent model-based RL systems for TSC, which can also be used for optimizing driving policies for vehicles. This paper presents the first model-based RL in the area to learn to estimate waiting times of vehicles given particular input states. Among the 160 included papers, 6 papers [31, 42, 66, 77, 78, 93] used or extended this model-based method. Furthermore, [34] proposed a model-based method that is based on a mechanism for creating, updating and selecting one among several partial models of the environment.
From 2001 to 2006, only 8 papers were published. In 2007, for the first time, 5 papers were published per year, and this number has been maintained in the following years.

As shown in Figure 7-3, the number of studies regarding RL in TSC is increasing. This topic continues to gain momentum, and hence, importance as the world and specifically, urban populations increase and this is demonstrated in the large number of papers published recently. During the first 15 years, from 1994 to 2008, a total of 22 papers on the subject were published, compared to 27 papers in the year 2019 alone. These statistics further attest to the fact that an in-depth review like ours is important at this time as it provides future works a comprehensive view of the past 25 years. Past trends notwithstanding, the Figure also allows us to determine where the majority of our information comes from and whether or not it is still relevant. As the majority of the distribution is concentrated around the past 10 years, we propose that the information presented here will, undoubtedly inform and possibly, shape future studies.
Authors’ country

Figure 7-4 exhibits our data of the distribution of research papers among each country. 160 papers from a total of 30 different countries were included in this review, with most of the research papers coming from 7 countries: China, USA, India, Iran, Ireland, Canada, and Brazil. These papers represent 67% of our pool of papers, strongly suggesting that as of 2020, these countries are the global leaders in the research of RL in network-scale TSC.

An interesting fact to note is that all seven of the countries mentioned have a traffic index\(^\text{1}\) above 140.45 according to [185], with Iran having the highest traffic index of the group, at 216.09. The motor data company, INRIX [186], finds Ireland’s capital city of Dublin to be the 7th worst city in the world in terms of hours lost due to traffic congestion based on 2019 data (154 hours). And, a common factor found in all of these countries is the extensiveness of their road networks. According to the Central Intelligence Agency (CIA) World Factbook [187], the USA, India, China, and Brazil rank in the top four countries with the longest road network, respectively, with Canada at sixth. Along with their large road infrastructures, these countries have consistently been producing the most research papers annually. The National Science Foundation (NSF) [188], Science and Engineering Indicators show that other than Ireland, these countries also rank in the top 15 countries that publish scholarly articles, with the USA and China ranking first and second place, respectively. This is directly in correlation to these countries producing the largest percentage of papers in our review. Finally, the solution to traffic congestion is coveted by many institutions, so, research into this topic could yield massive financial returns to the researchers. With the governments of China and the United States being the first and second and Canada not far behind on the list of capital spent on transport infrastructure in 2018 as per [189], the incentives are clear to the researchers in these regions.

\(^{1}\)Metric that is a composite index of time consumed in traffic due to job commute, estimation of time consumption dissatisfaction, CO2 consumption estimation in traffic and overall inefficiencies in the traffic system.
Number of authors

The number of authors of the papers published on RL in TSC ranges from 1 to 10. 92% of the papers have 2 to 5 authors, while papers with 3 authors alone accounted for has the highest proportion (36%) of papers. See Figure 7-4.

Authors’ university department and authors’ affiliation

This study also took into consideration the department and affiliations of authors, and identified four groups of departments, as follows: (1) computer, IT, and electrical engineering or related departments, (2) civil engineering and transportation engineering, (3) industrial, mechanical, material, and geomatics (i.e. other engineering departments), and (4) science, business, management, astronautics, and English. It was found that 62% of the authors were from information technology and computer related departments (group 1), and 26% were from civil engineering and transporta-
tion engineering departments (group 2), while researchers from other departments (groups 3 and 4) accounted for 12% of the total.

Notably, authors from these four groups of departments had very low research collaboration with each other, and had conducted research in only 20 out of 160 articles, generally. The computer-related and civil engineering-related departments collaborated in only 11 articles, while they contributed independently in 98 and 36 papers, respectively.

This investigation also uncovered that academia and research institutes have low research collaborations with industry and government, with 11 instances of collaboration between academia and industry and only one appearance of government in research papers that collaborated with academia in a paper in 2018. Suffice it to say that research exploring the potential benefits of the collaborations between these three types of departments is needed, as the findings suggest that increased collaboration in these domains may boost the efficiency of the proposed methods in real-time, real-world applications.

7.3.2 Traffic simulation and evaluation

Number of intersections and road network types

Figure 7-5 shows the distribution of the number of intersections studied. Though many studies are still currently being held with fewer than a dozen intersections, there is also work on 25+ intersections. And while 44% of the papers run simulations in small-scale networks with 8 intersections or lower, we observed that the recent trend is to test the proposed methods in medium and large-scale networks. The size of the network is also important as it may impact the exponential growth of state-space and action-space and generally the complexity and computational effort required to reach a solution through an RL method, whether for training or testing stages.

We categorized the type of the testbed network into three main groups, including: (1) the network of intersections, (2) arterial network, and (3) the signalized roundabout networks. As already mentioned, the isolated intersection case is excluded from our study. The arterial network
Figure 7-5: Distribution of number of intersections in simulations: in all studies (pie chart), and over time (bar chart).

Notes: 1) The data of arterial and network have been merged and the highest network size in each paper is reported, 2) NA represents the papers in which the number of intersections in the arterial or network is not explicitly revealed, 3) The years with no publication, i.e. 1995-1998 and 2001-2002, are not shown.

is an open network, as compared to a closed network. Arterial network control is applied on a sequence of intersections to provide preference to progressive traffic flow along the arterial. Unlike the isolated intersections, the intersections in the arterial network operate as a system and the system coordinates timing of adjacent intersections. On the other hand, the network of intersections is considered as a close loop. Being a closed loop, the network demands at least four intersections. The networks of two and three intersections may have the characteristics and behaviour of both groups (the network of intersections and arterial network); therefore, we considered them separately as two sub-groups under the first group, i.e. the network of intersections. By doing this, the arterial network group is limited to four or more intersections.

Finally, the signalized roundabout networks deals with both approaching and circulatory lanes, which are fundamentally different from the first two groups. The signalized roundabout networks involve more than one set of traffic signals in a node. [168, 159], both published in 2019 are only
two papers in the literature that studied the signalized roundabouts, which indicates a potentially open research problem on the RL area. And, only 17 papers worked on arterial networks (out of 189 network configurations in 160 papers), all between 4-8 intersections except one with 16 intersections, which is the maximum number in an arterial network. The highest numbers of intersections in a network are 225 ([118, 174]), 196 ([153]), and 127 ([171]), respectively. The paper [118] is the first paper in the literature that tested the proposed method in a network with the highest number of intersections: 225 intersections in 2016. In 5 of the papers that we came across, the number of intersections of the network or arterial network is not released, and in 6 papers there is no simulation used. See Table 7.3.

**Type of map**

With regard to the type of maps used to test the proposed RL methods by the included studies, of those that used maps, 100 (63%) papers used a synthetic map, 45 (28%) papers used a real-world map, and 9 (5%) papers used both types. The 6 studies that did not use any simulations to evaluate their proposed approach accounted for 4%.

**Location of real-world maps**

49 real maps are used from 14 countries for simulation purposes, with China topping the list at 10 maps, followed by the USA (8 maps), Ireland (8 maps) and Canada (6 maps). These maps are mostly large-scale networks of intersections.

**Lanes and turns configuration**

It is important that the considered map as a testbed would be consistent with a real-world set-up. For example, one-lane or one-way crossing links cannot replicate the common cases in real-world scenarios. This may dramatically impact the usability and performance evaluation of the methods, specifically in terms of computation efficiency. Based on the collected data, most of the papers used
a good level of complexity in the number of lanes and turns. 24% of the papers used real-world maps with multiple lanes, 28% used two or more lanes (including 13% using three or more lanes) with a good level of complexity, and 20% used the synthetic maps with multiple lanes but without significant information on the number and types of turns. By a good level of complexity, we mean that in a regular driving style the through and left lanes are involved regardless of the right turn. The through and left lanes in the opposite approaches have conflicting right of ways and add to the state and action spaces, which increases the complexity of the problem. In the regular driving style, right turns can usually be accommodated simultaneously with either through or left lanes and do not impact the state and action dimensions. Also, in a right-side driving style, the through and right lanes are considered regardless of a left turn (similar logic to the regular driving applies here). In 25 (16%) of the papers, a low level of lane/turn complexity, including one-lane links, one-way crossing links, or multiple-lane links with only through lanes is represented. 9% of the papers did not reveal any or enough information about the lanes and turns.

**Data collection method**

We identified four categories of data source for the proposed methods, including general detection devices (65%), loop detectors (17%), vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) and infrastructure-to-infrastructure (I2I) (11%), and camera/image (7%). By general detection devices, we mean that the authors did not specify any specific data-source. The papers that use loop detectors provide either no specific design or specific designs like (1) one detector at the stop-line, (2) one detector at a distance from the stop-line, (3) one detector at the upstream (end of the lane), (4) two detectors at the stop-line and at a distance from the stop-line, or (5) two detectors at the stop-line and at the end of the lane. The third category presents the methods that are inherently image-based or just use traffic cameras as a data source. The papers that used V2I, V2V, I2I, and image/camera as the data source are identified in Table 7.3.
Table 7.3: Identification of the included papers in terms of data source, method identification, code, simulation, and evaluation.

<table>
<thead>
<tr>
<th>Category</th>
<th>Data source</th>
<th>citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2I/V2V/I2I</td>
<td></td>
<td>[115, 140, 143, 93, 42, 157, 66, 136, 80, 129, 150, 97, 77, 28, 126, 149, 86]</td>
</tr>
<tr>
<td>Image/Camera</td>
<td></td>
<td>[109, 138, 133, 171, 180, 179, 170, 178, 98, 120, 155]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Identification</th>
<th></th>
<th>citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor-critic</td>
<td></td>
<td>[116, 139, 135, 180, 165, 178, 174, 91, 83, 70, 39, 37, 160, 60]</td>
</tr>
<tr>
<td>Deep Learning</td>
<td></td>
<td>[138, 142, 144, 171, 181, 147, 179, 163, 164, 165, 166, 167, 168, 169, 170, 184, 178, 174, 175, 156, 154, 113, 159, 158, 120, 149, 155, 161]</td>
</tr>
<tr>
<td>Model-based methods</td>
<td></td>
<td>[93, 42, 31, 66, 124, 77, 28, 32, 78, 60, 34]</td>
</tr>
<tr>
<td>RL and GT</td>
<td></td>
<td>[182, 80, 50, 137, 84, 64, 47]</td>
</tr>
<tr>
<td>ADP</td>
<td></td>
<td>[91, 44, 102, 60, 155]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code, Simulation, and Evaluation</th>
<th></th>
<th>citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code is available</td>
<td></td>
<td>[138, 119, 144, 171, 180, 165, 166, 169, 153, 88, 149]</td>
</tr>
<tr>
<td>No Simulation</td>
<td></td>
<td>[101, 155, 32, 64, 59, 47]</td>
</tr>
<tr>
<td>No Evaluation/Self-comparison</td>
<td></td>
<td>[101, 155, 32, 64, 59, 47, 53, 119, 144, 54]</td>
</tr>
</tbody>
</table>
Traffic simulator

16 traffic simulator software or platforms are identified in these studies to simulate the traffic. These include SUMO [190] (37 papers), VISSIM [191] (19 papers), PARAMICS [192] (15 papers), GLD [193] (13 papers), AIMSUN [194] (6 papers), ITSUMO [195] (4 papers), CityFlow [196] (3 papers), TRANSYT [197]/TRANSYT-7F [198] and MATLAB [199] and AIM [200] and TSIS [201] (2 papers each), MATISSE [202] and USTCMTS (used in [147]) and SMPL (used in [38]) and SeSAm (used in [40]) (1 paper each). Since the source of the last three simulation software tools is not found, we referred the reader to the papers in which they are used, for further information.

Of interest is that SUMO was used for the first time in 2015 and has since become the most frequently used software in this area, with 17 out of 27 occasions in 2019 alone. The second most frequently used software, (i.e. VISSIM) started being used in 2010, and PARAMICS, the first well-known traffic simulator has been in use since 2003. Prior to this the tools were either custom-built or not clearly outlined in the research. In 18 studies, the authors designed or used a custom-built environment for simulations. What is surprising is that 27 studies (16.7%) did not state the simulation tools that they used for the test, not including the those that did not use any simulations.

Presence of evaluation

150 (94%) of the studies provided an evaluation, while 7 (4%) did not. Three (2%) papers provided self-comparison, meaning that they compared the variations of the proposed methods with each other, but not with other TSC methods. The papers that did not provide an evaluation are identified in Table 7.3.

Compared methods and performance measures

The authors used different TSC methods and performance measures to compare and validate their proposed method. Fixed time methods alone are inefficient in evaluating other methods because
they are unable to adapt to the traffic flow changes, however, they are sufficient to use in exploring the feasibility of a proposed method or the proof of concept. We found that 27 (17%) of the studies used only fixed time or random methods/policies for comparisons. In 39 (24%) cases, the TSC methods are used where no RL method such as actuated and adaptive methods are included, as they provide for better evaluation. In 76 (47%) studies, RL has been used either alone or with other types of methods. Involving RL methods for evaluation, however, cannot guarantee a perfect evaluation every time. Generally, if an RL method is used along with actuated and adaptive methods, it can provide a great evaluation, specifically when the comparison is made with state-of-the-art RL methods proposed by the other authors in the field. We collected these RL methods as a reference for the readers in Table 7.4. The table also provides the citation of referenced papers that used evolutionary and meta-heuristics algorithms, real-world, fixed time, and adaptive methods. It might be of interest to the reader to know the number of methods that are used in these papers for evaluation/comparison purposes: 1 (33%), 2 (31%), 3 (14%), 4 (5%), and 5 to 8 (5%), demonstrating that comparison of a method with only one other is the most common.

Among the performance measures, delay is the most frequent performance measure in the papers with 71 occurrences (20%), followed by travel time and waiting time and queue size (each 12%), number of stops and speed and throughput (e.g. the number of the vehicles passed the intersection) (each 6%), and environmental measures (5%), which accounts for 80% of the papers. We found 33 unique performance measures that are listed in 7.A. Each unique measure delegates several similar measures. The list shows a variety of measures that authors may consider or use in their following research works. 7.A also depicts the top ten performance measures.

**Saturation and spillback considerations**

Although there are numerous methods listed in the literature, for traffic signal control, they are mostly effective in low traffic loads. The main challenge in traffic control is managing the high load of traffic representing rush-hour in the morning or afternoon in a network that may lead to a
<table>
<thead>
<tr>
<th>Compared Methods</th>
<th>citation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RL methods</strong></td>
<td></td>
</tr>
<tr>
<td>Distributed and multi-agent RL</td>
<td>[58, 84, 80, 49, 42, 89, 103, 29, 148], [203, 204, 205, 206, 207, 208, 209]</td>
</tr>
<tr>
<td>Actor-critic RL</td>
<td>[210, 211, 212]</td>
</tr>
<tr>
<td>Distributed Intersection Management Protocol</td>
<td>[217]</td>
</tr>
<tr>
<td>Junction Tree Algorithm</td>
<td>[106]</td>
</tr>
<tr>
<td>TILDE</td>
<td>[218]</td>
</tr>
<tr>
<td>Propositional Function-Gradient Boosting</td>
<td>[219]</td>
</tr>
<tr>
<td>Wiering Controller (TC-1)</td>
<td>[28]</td>
</tr>
<tr>
<td>Traffic Controller with State Bit for Congestion</td>
<td>[220]</td>
</tr>
<tr>
<td>SPSA-based multi-agent</td>
<td>[221]</td>
</tr>
<tr>
<td><strong>Evolutionary and meta-heuristics methods</strong></td>
<td></td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>[31]</td>
</tr>
<tr>
<td>Cooperative Ensemble</td>
<td>[222]</td>
</tr>
<tr>
<td><strong>Real-world methods</strong></td>
<td></td>
</tr>
<tr>
<td>GLIDE (SCATS-like method in Singapore)</td>
<td>[223]</td>
</tr>
<tr>
<td><strong>Fixed time methods</strong></td>
<td></td>
</tr>
<tr>
<td>Optimized fixed time (TRANSYT)</td>
<td>[224]</td>
</tr>
<tr>
<td>Fixed Time</td>
<td>[225]</td>
</tr>
<tr>
<td>Fixed time with random offsets</td>
<td>[226]</td>
</tr>
<tr>
<td>Webster method</td>
<td>[227]</td>
</tr>
<tr>
<td><strong>Adaptive methods</strong></td>
<td></td>
</tr>
<tr>
<td>Saturation balancing method</td>
<td>[228]</td>
</tr>
<tr>
<td>Self-Organizing Traffic Light Control (SOTL)</td>
<td>[229, 230]</td>
</tr>
<tr>
<td>Max Pressure</td>
<td>[231]</td>
</tr>
<tr>
<td>Longest Queue First - Maximal Weight Matching</td>
<td>[232]</td>
</tr>
<tr>
<td>Anticipated All Clearing policy</td>
<td>[233]</td>
</tr>
<tr>
<td>Cluster based adaptive methods</td>
<td>[234, 235]</td>
</tr>
</tbody>
</table>

spillback condition in different links. Our research found that 74% of the studies simulated the traffic condition in high demands, close to saturation, saturated, or oversaturated conditions to test the efficiency of the proposed methods in traffic congestion (labelled as Sat in Table 7.2). Nevertheless, among these, only 5% explicitly applied and addressed a spillback prevention strategy, and 6% analyzed or mentioned that the proposed method is able to prevent spillback (labelled as Spillback in Table 7.2). 6% of the papers explained spillback but never considered or applied it, whereas in 56% of the studies with high demand, spillback is not even mentioned. 22% neither addressed high
demand and spillback, nor explicitly mentioned them (labelled as *Neither* in Table 7.2), and 4% provided no evaluation (labelled as *No Evaluation* in Table 7.2).

### 7.3.3 Method identification and analysis

In this section, we elaborate on the proposed methods from different angles. We start with the categorization of the proposed methods in terms of application domain and the way that RL is applied, whether alone or in combination with other methods from the same or different fields. We identified 7 categories of TSC application domain, including general (or regular) TSC, computing paradigms in TSC, TSC in roundabouts, TSC for public transit, TSC for emergency vehicles, TSC for perimeter control, and automating streetcar bunching control in TSC. We also categorized the variations of RL applications, i.e. the ways that RL method can be applied, into five groups that can be applied to each TSC application domain, see Figure 7-6. In Table 7.2, these categories are specified for each article.

- **AD1**: RL in general TSC domain

  In contrast to other domains, general TSC domain is defined as a category that covers the general idea of controlling all traffic signals (not only the signals along the perimeter of a network) in a regular network (open and closed network, compared to roundabouts) and regular components (regular cars, compared to buses, vehicles bunching control, and emergency vehicles) without a focus on communication delay problem and computing paradigms consideration. As this category, i.e. general TSC domain, is the main focus of most of the studies (93%) and contain all five variations of RL application, in the following, we address these variations separately for this category. The other categories contain only the first variation (see Figure 7-6 again) and we just explain those categories without focusing on the variation of how RL is applied.

  - *(AD1,V1)*: RL is used alone or in combination with a machine learning (ML), game theory (GT), or dynamic programming (DP) in TSC
The vast majority of the studies (130) are classified in this category where RL is the only applied method, or is used in combination with other ML, GT, and DP methods. Different innovative designs were proposed to tackle the problems of TSC, which has a continuous state space, infinite horizon, and is only partially observable and difficult to model. In this context, most of the papers try to improve the performance (e.g. [81]), dimensionality (e.g. by means of function approximation [63, 56]), complexity (e.g. by organizing agents in groups of limited size [57], and using holonic RL methods [85]), scalability (e.g.
stability (e.g. [141]), speed of optimization (e.g. through Transfer Learning models [167]), state and/or action space manageability or generalizability (e.g. [103, 115]), and centrality in applying RL in TSC. Some studies invest in solving convergence and oscillation problems that commonly appear in the multi-agent context. For instance, [155] proposed a model based on Double Deep Q-Learning, with Experience Replay and cooperation between agents. They used NN to reduce the correlation between agents and improve performance. Improving the accuracy, optimum functioning, and efficiency of the results is also of high interest to the researchers, through hierarchical methods [146, 163], for example. There are several papers where the main focus of research is on studying the coordination between agents (e.g. [173]), and the integrated network, specifically signalized intersections and ramp metering (e.g. [64, 84, 107]). Famous methods proposed in traffic theory context, such as CTM [90, 104, 182], Max-plus [42, 76], and Max Pressure [166], are also applied in some research, and in other studies multiple traffic optimization goals are simultaneously optimized (multi-policy RL), e.g. [114]. In 2017, for the first time, a distributed multi-agent RL method proposed by [133] considered both vehicular and pedestrian traffic in the network-scale. Since 2018, the number of papers that focused on improving deep RL models in TSC has increased (e.g. [138, 142, 144]). The co-learning problem of both classes of learning agents, traffic signals and drivers, with different goals (minimizing individual travel times vs minimizing the queues locally), different nature (driver agents learn in episodes that are asynchronous, while traffic light agents learn continuously (non-episodic)), and the nontrivial task of microscopic modelling and simulation (whose actions are highly coupled) is another area of research that was addressed [150]. Analyzing what specifically RL does differently (i.e. analysis of the learned policies) than other TSC methods is conducted by [180].

- \(AD1, V2\): A method is used from a different field (rather than ML, GT, and DP) in RL or RL framework
To cope with non-stationary environments, [34] proposed, formalized and showed the efficiency of a method called the RL-CD, or Reinforcement Learning with Context Detection, which performs well in non-stationary environments, and better than classic RL algorithms (Q-Learning and Prioritized Sweeping). In a similar work, [36] assessed the feasibility of applying RL-CD approach in a more realistic scenario, implemented by means of a microscopic traffic simulator. [39] showed how to use Conditional Random Fields (CRFs) to model control processes, where CRFs model joint actions in a decentralized Markov decision process and define how agents can communicate with each other to choose the best joint action. The CRF model clearly outperformed the independent agents approach. [77] enhanced the single-objective controller by developing a multi-agent TSC system based on a multi-objective sequential decision-making framework using Bayesian interpretation and some innovative reward design. [106] proposed a JTA based algorithm to obtain an exact inference of the best joint actions in traffic signal coordination that outperformed independent learning (Q-learning), real-time adaptive learning, and fixed timing plans. To predict future system states and avoid unwanted states, [118] applied Proactive Complex Event Processing method (in processing proactive traffic congestion control) that uses RL to find the optimal joint policy. This method works well when used to control congestion. To develop learning and adaptation mechanisms to deal with disturbances, [127] proposed a distributed TSC system based on hybridization between Case-Based Reasoning (CBR) and an adaptation of the reinforcement principle within artificial immune networks (a mechanism inspired by biological immunity). The method controls interrupted flow at signalized intersections. [153] studied how the attention mechanism helps cooperation (to minimize the average queue length) via using graph attentional networks to facilitate communication, incorporating the temporal and spatial influences of neighbouring intersections to the target intersection, and building up index-free modelling of neighbouring intersections.
– *(AD1,V3)*: RL is used in the methods from a different field or in a specific design/model/framework for TSC context

[38] integrated *sensor networks* and *grid computing* and the usage of web services to implement this integration and used Q-learning algorithms in distributed Stargates (a computer with sensor signal processing capabilities) to TSC. [111] modelled a *holonic* multi-agent system and proposed a holonic RL multi-agent system method that improves the performance of individual Q-learning in TSC. [110] showed the applicability and efficacy of using *auction theory* combined with an RL in a multi-agent system. In low traffic volume the proposed method outperforms actuated and pre-timed control strategies, but in heavy volume the pre-timed control strategy outperforms the proposed method. [117] designed a distributed multi-agent method with coordination between agents through the communication of decision data. In this paper, the effectiveness of integrating *fuzzy logic controller* (to deal with continuous states and actions) and Q-learning (for learning during the process) is studied.

– *(AD1,V4)*: The application of RL in optimization problems or optimization in an RL method or framework

All of the proposed methods in this category are designed to handle the growing complexity and course of dimensionality and to increase the speed of TSC by using RL in the optimization problems or applying the optimization methods in an RL method or framework. To achieve cooperation in the long term, as stated earlier, [25] combined RL of a local agent with global optimization through *Genetic Algorithms* by which the RL parameters are modified. [26] used a classifier system with a fuzzy rule representation, with both evolutionary and RL methods to provide a faster method than hierarchical control. [27] incorporated LCS and TCP/IP based communication server into a distributed learning control strategy to increase the speed of control. [54] handled the growing complexity by using an organic approach to TSC and proved the feasibility of the proposed approach. [70] proposed a multi-agent TSC by using *Swarm Intelligence* and *Neuro-Fuzzy RL* to combine the better attributes of
both with improving the learning speed and performance. [97] optimized the TSC for V2I networks by proposing a cooperative distributed Q-learning algorithm with a fast gradient-descent function approximation. [105] applied an RL method in the optimization problem to reach good solutions for TSC. The results were better than the genetic algorithm and hill-climbing methods in low demand but could not outperform them in medium and high demands. [112] showed that Swarm Q-learning performs better than standard Q-learning in increasing the speed of TSC. To alleviate traffic congestion and limit the effects of incidents on traffic flow, [130] proposed a Q-Learning based traffic management model, which simultaneously optimizes vehicle re-routing and TSC based on the Multi-Objective Particle Swarm Optimization (MOPSO) method.

- (AD1,V5): Theoretical RL method development with feasibility analysis in TSC
  To provide a simple and efficient method to implement, [71] put into operation a functional gradient boosting approach to imitation learning in relational domains. The proposed approach outperforms both learning a single relational regression tree and performing a propositional functional gradient to represent the policy in all domains. To provide a solution to multi-objective problems with correlated objectives rather than typical multi-objective problems, [88] proposed an RL-based method combining multiple correlated rewards and shaping signals by measuring confidence (i.e. combining the feedback from all objectives, instead of only looking at a single one), called adaptive objective selection. They formally defined a new class of multi-objective problems called correlated multi-objective problems (CMOP), whose set of solutions being optimal for at least one objective is so restricted that the decision-maker is least concerned about which of these is found, and more so about how fast one is found, or how well one is approximated. [95] proposed a method for synthesizing a control policy for an MDP such that traces of the MDP satisfy a control objective expressed as a linear temporal logic (LTL) formula through using an RL algorithm that finds the policy optimizing the expected utility of every state in the Rabin-weighted product.
MDP. They prove that the method is guaranteed to find a controller that satisfies the LTL property with probability one if such a policy exists, and they suggest empirically with a case study in traffic control that their method produces reasonable control strategies even when the LTL property cannot be satisfied with probability one. \cite{116} optimized variance-related risk measures in rewards and demonstrated its usefulness in a TSC application. The \textit{risk-sensitive algorithms} result in lower variance but higher long-term cost compared to their risk-neutral counterparts. \cite{119} illustrated the usefulness of modeling \textit{human decisions} by \textit{Cumulative Prospect Theory (CPT) paradigm} in RL and suggested that CPT-based criteria is useful in a TSC application. \cite{177} proposed a \textit{dynamic correlation matrix based MARL} approach where the meta-parameters are evolved using an evolutionary algorithm in a distributed manner. This was done to provide meaningful theoretical verification by using both agent-level implementation and system-level convergence verification. Agents using the proposed learning algorithm reach optimal behaviours faster than other canonical learning techniques.

- \textit{AD2}: RL in computing paradigms domain (e.g. edge, fog, and cloud computing)

  - \textit{(AD2,V1)}: In 2014, \cite{98} proposed a Q-learning approach to effectively prevent the queue spillovers by applying an Edge detection algorithm on images from simulation and calculating waiting time in each link accordingly. In 2019, \cite{162} designed a framework of edge computing under the TSC scene and proposed a cooperative TSC algorithm based on MARL to avoid the curse of dimensionality, provide minimal response time, and reduce network load. In 2019, \cite{171} proposed a large-scale Edge-based RL (ERL) solution to better alleviate congestion in complex traffic scenarios based on Edge Computing nodes for traffic data collection. They concluded that ERL in distributed edge servers has much better scalability and faster DNN training than the cloud service. In 2019, \cite{157} proposed a traffic control architecture based on fog computing paradigm and a distributed RL algorithm (that connects traffic signals, vehicles, Fog nodes and traffic cloud) to overcome communication bandwidth

173
limitation and reduce communication delay, make real-time traffic condition information available to vehicles, and lower the probability of traffic congestion in the city through generating traffic signal control flow and communication flow for each intersection. This is not only suitable for current vehicles but also more useful for driverless vehicles anticipated in the future, as it will be able to plan its route much more intelligently with information from the Fog node.

- **AD3**: RL is used in the domain of roundabout control

  - *(AD3, V1)*: The dynamics of signalized roundabouts is complex because it deals with both approaching and circulatory lanes. The conflicts between approaching and circulatory flows cannot be solved by metering only the approaching lanes and reacting consequently because the circulatory lanes may be occupied. [168] proposed a deep RL method for signalized roundabouts in congested networks to maximize traffic flow while being able to avoid traffic jams in connected junctions. In a particular study conducted in this domain, [159] explored the possibility of deriving explanations from a neural network agent (trained using Policy Gradient) for TSC in a signalized roundabout. How the agent learns to react differently based on each specific lane’s traffic by implicitly predicting the route of the traffic and thus its future circulatory occupancy is explored. This is done by analyzing the relation between the agent phase preferences and the actual traffic, assessing the agent capability of reacting to the current detectors state, and estimating the effect of the road detectors state on the agent selected phases through the SHAP model-agnostic technique. The results reveal that it is possible to extract meaningful explanations on the decisions taken by the policy. Further research involves the study of the trade-off of accuracy in comparison with a complex deep learning controller.

- **AD4**: RL in public transit domain
– *(AD4, V1)*: In 2014, [90] developed a distributed CTM-Based MARL for network-scale signal control with transit priority that outperforms pre-emptive and differential priority control methods because of the improved awareness of the signal switching cost. To eliminate the need for feature extraction in the state space and to directly use available information received from the high-detailed traffic sensors [156] proposed a multimodal Deep RL based traffic signal controller that combines both regular traffic and public transit and minimizes the overall travellers’ delay through the intersection.

• *AD5*: RL in the emergency vehicles domain

– *(AD5, V1)*: In 2017, to detect and give priority to emergency vehicles, [126] proposed a reinforced traffic control policy that reduces the waiting time of emergency vehicles at intersections as well travel time of other vehicles using a multi-agent system development framework (JADE).

• *AD6*: RL in the cordon (or perimeter) control domain

– *(AD6, V1)*: In 2019, [174] explored how RL can be used to re-time traffic signals on the perimeter by developing an RL based controller with NN architectures that controls perimeter with spatially-varying metering rates.

• *AD7*: RL in automating streetcar bunching control domain

– *(AD7, V1)*: [33] conducted research to mitigate the effects of car bunching along transit routes through automating car bunching control by means of multiple RL agents that act on a series of successive signalized intersections. The RL agents are the bunch-splitting, holding, and expressing agents, which work cooperatively to break up a car bunch if one is detected, and to build a reasonable headway between the paired cars. Typical control strategies include vehicle holding, expressing, and short turning, all of which are usually
implemented through manual means via field supervisors or central control centers. The
designed multiple RL agents could split up a streetcar bunch and prevent it from forming
again with a high success rate.

In the following, we introduce the types of RL methods used in the articles we included in
this chapter. Furthermore, we extracted data about other methods that an RL method has been
integrated with to provide a solution, whether as a core or combined method.

RL methods

Q-learning \[203\] ranks first on the list of used RL methods in 96 (60\%) studies. 13 papers do not
provide any specific RL method, especially when the goal of the paper is the proposal of a framework
or when an RL concept is used. In this case, we used “RL” in Table 7.2. The remaining papers
include methods such as Actor-critic (10 papers), Model-based RL method which is mostly based
on the well-known Wiering’s method \[28\](8 papers), approximate dynamic programming (ADP)
(6 papers), W-Learning (5 papers), Multiple RL (including Q-learning, SARSA, and actor-critic)
(4 papers), Policy-Gradient (4 papers), Learning Classifier System (LCS) (3 papers), SARSA (3
papers), Temporal Difference (TD) (2 papers), Learning Automata (LA) (2 papers), Inverse RL
(2 papers), continuous residual RL (CR RL) (1 paper), and Risk-sensitive RL (RS RL) (1 paper).
Below is a synopsis of these RL methods.

- **Q-learning and SARSA**: Q-learning was first investigated in 1989 by combining the Bellman
equation and Markov decision process with temporal-difference learning. Although effective for
many real-world problems, this has its limitations with the increasing complexity of data and
the environment \[236\]. Both Q-learning and SARSA (State, Action, Reward, State, Action)
methods are critic-only where they use Q-tables to decide which action to take. The biggest
difference between the two is that Q-learning is an off-policy RL method, while SARSA is an
on-policy method. Q-Learning and SARSA Q-table update equations are respectively presented
below, where the policy is TD.
Q(s, a) ← Q(s, a) + α[r + γ max_a Q(s', a) − Q(s, a)]  \hspace{1cm} (7.1)

Q(s, a) ← Q(s, a) + α[r + γ Q(s', a') − Q(s, a)] \hspace{1cm} (7.2)

A simply put summary of Q-learning [211] shows the algorithm to follow these steps:

1. s is the current state in which the agent resides.
2. The agent chooses action a from the available or acceptable actions.
3. In response, the agent receives a reward r for action a and the next state s'. It is important to note that this state s' is merely a projection or estimation of the next state, rather than a known value.
4. Q(s,a) is then updated using equation 7.1.
5. The entire process is repeated.

- **Critic-only, Actor-only, and Actor-Critic:** The critic uses the calculated Q-values (or function values) to choose its action while the actor uses the policy to decide. Actor-only methods work to improve the policy. Actor-critic being a combination of both, allows both calculations of the Q-values and the policy to choose appropriate action. The papers that used an actor-critic method are identified in Table 7.3.

- **W-Learning:** W-Learning is a multi-policy self-organising action-selection technique proposed in [237] that builds on Q-Learning. In W-Learning, there is a competition among selfish Q-learners where agents learn Q-values for state-action pairs for each policy and W-values for each of the states of each of their policies to explore what happens if the nominated action is not followed.

- **Approximate Dynamic Programming (ADP):** The computational complexity of DP algorithms due to excessive system state and their need for an exact algorithm and true value function
makes the algorithm impractical in solving large-scale TSC problems. ADP aims not to fall into the predicament of computational complexity by replacing the true value function of the DP with an approximation function. In other words, it is similar to the model-based RL with function approximation. The research papers that used ADP method are identified in Table 7.3.

- **Policy-Gradient:** The policy gradient method does not need to estimate the state or action value functions. It learns parameterized policy functions directly by searching policy space to maximize a measure based on the accumulated reward. In this way, it averts the convergence problems of estimating value functions.

- **Inverse RL:** In Inverse RL [238], the reward function of an agent (that the agent tries to optimize) is learned and determined by observing the agent’s behaviour over time, the environment model, and the environment measurements. This approach is akin to learning from an expert and is helpful in the domains where the reward may not be easily accessible, like TSC [62]. This method has its origins in Imitation learning (also called as apprenticeship learning, learning by observation, or learning from demonstrations). It is comparable to supervised learning, with the key difference being that the examples are not i.i.d, but instead, follow a meaningful trajectory [239].

- **Learning Classifier Systems:** A learning classifier system (LCS) [240] is a rule-based RL system in which each rule (or classifier) is composed of a condition, an action, and a reward (or evaluation). LCS combines an evolutionary process (e.g. a genetic algorithm), with a learning process (e.g. RL), wherein a rule is constructed as \{IF ‘condition’ THEN ‘action’\}. A genetic algorithm tries to improve condition-action rule space by generating new classifiers from current strong classifiers and removing the weak ones. RL is responsible for selecting the action with the best-rewarded response or evaluation to be executed.
• **Learning Automata:** The action selection in learning automata is performed based on the last selected action and the received reward. A learning automata method forms from a vector of probabilities over the set of actions, which are updated (i.e. increase or remain the same) based on the reward.

• **Model-based and Model-Free:** Model-based methods provide the agent with part or all of a model in which the agent must work. In model-free methods the agent develops its own model in which to work and has fewer restrictions. Essentially, in model-based methods the transition and reward functions are assumed to be available to construct a model, unlike in model-free methods where the agents do not need to have access to information regarding how the environment works. The papers that used model-based methods are identified in Table 7.3.

**Core non-RL methods**

RL is the core method in 149 (93%) of the studies. In 11 studies, other methods including Fuzzy, Auction theory, Proactive Complex Event Processing model (Pro-CEP), Self-adaptive TSC system, Cellular Automaton, Immune Network algorithm, Neural Networks (NN), Multiband, SensorGrid, and Fog Computing were employed as the core method in which RL is used as a combined method.

**Combined non-RL methods**

There are also other non-RL methods that were used in combination with the RL methods or frameworks, including Junction Tree Algorithm (JTA), Cumulative Prospect Theory paradigm (CPT paradigm), Self-Organizing Map, model-based and Bayesian optimization algorithms, Transfer Learning, FRAP model design (invariant to symmetric operations like Flipping and Rotation and considers All Phase configurations), Cell Transmission Model (CTM), regional mixed strategy Nash-Equilibrium (RMS-NE), Max-plus, Conditional Random Fields (CRF), Genetic Algorithm (GA), Grey theory model, qualitative method, Evolutionary algorithm, Game theory (GT), Dynamic Clustering algorithm, Neuro-Fuzzy, Java Agent Development Framework (JADE), Coalition,
Fuzzy granulation, Edge computing framework, Max-sum message passing algorithm, and Swarm optimization.

- **Game Theory**: RL and Machine Learning are not the same thing as Game Theory. When approaching a learning algorithm, one can choose to use RL, Game Theory or a combination of the two. Game Theory, which has its roots in economics (as opposed to the biological background of RL), factors in the competition of other "players" in the environment that may wish to undermine the agent in its task. The environment acts upon the agent as much as the agent acts upon the environment. If the environment is not threatening or does not directly affect the agent itself, there is no need to employ Game Theory. An example of such would be a self-driving vehicle vs a traffic signal controller. For the vehicle, interacting with traffic directly, where each surrounding motorized vehicle, bicycle and pedestrian can pose a threat to its safety, Game Theory can give the agent a competitive edge in keeping the other vehicles in consideration as other players. The traffic signal controller agent, though it affects traffic and traffic affects its reward, is not directly impacted by other vehicles or "player". The papers that used GT and RL methods are identified in Table 7.3.

**Centralization (centralized, decentralized, and hierarchical methods)**

A multi-agent system must be regulated. It can be centralized where all information goes to and is stored in a central location, and agents are supervised by this central agent. At large scale implementation, multiple agents tackle the task given while communicating with each other. This usually results in quicker optimization as each agent learns from its neighbours as well as from itself [112]. With multiple agents, the collected data and actions can be stored centrally in a location that all agents can access to function as one agent. In this setup, the central agent often makes all decisions for the system, and this can slow down the learning process while coordinating the unit [241]. Conversely, a decentralized approach can be used to store the local information around multiple agents, allowing each one to make its own decision while still communicating with its
neighbours [242]. In this approach, all agents would be considered “equal” [243]. Another setup that uses a combination of both centralized and decentralized methods is a hierarchical system, which can be categorized as a centralized method since it involves centralization. It forms a hierarchy of sorts, where the lower agents may have limited to no ability to enact upon the environment without permission from the “leader”. A hierarchical control allows agents to perform micro-actions between tasks to improve the finesse of the agents.

Most (65%) of the proposed methods in the area of RL in the network-scale are designed in a decentralized way. There are 7 papers proposing holonic or hierarchical methods while the rest are centralized methods that might be rarely applicable in real-world scenarios in a real-time process.

**State, action, and reward**

There are normally three main components in RL: state, action, and reward. There are several elements that can define state and action in various papers. The elements of the state can be similar or different from those of the rewards. Based on our collected data of 160 studies, we identified 35 distinct elements of state and 30 of reward. The top 5 frequently used elements in state are queue size with 73 occurrences (38%), phase state (11%), number of vehicles (10%), the position of the vehicles (6%), and speed (6%). In reward, the top 5 are queue size with 71 occurrences (30%), delay (13%), waiting time (9%), the number of vehicles (6%), and number of vehicles passed the intersection (or generally throughput) (4%). The elements of the state in 16 (5%) research papers and those of reward in 18 (7%) papers are not available. The list of all the elements of the state (38 unique elements) and reward (39 unique elements) found in the papers are available in 7.B and 7.C. This might help the new researchers get some idea in defining these components. 7.B and 7.C also depict the most frequently used elements in states and reward, respectively.

In addition to state and reward, action needs to be defined. The majority of the research papers define refer to switching the traffic signal when talking about action. However, there are some that will use a different definition than the signal switch. These actions include (Act1) set the value
Table 7.5: Categorization of types of traffic signal based actions in the RL methods.

<table>
<thead>
<tr>
<th>cycle length</th>
<th>phase duration</th>
<th>phases order</th>
<th>cycle-/phase-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 fixed</td>
<td>fixed</td>
<td>fixed</td>
<td>cycle-based</td>
</tr>
<tr>
<td>2 fixed</td>
<td>fixed</td>
<td>variable</td>
<td></td>
</tr>
<tr>
<td>3 fixed</td>
<td>variable</td>
<td>fixed</td>
<td></td>
</tr>
<tr>
<td>4 fixed</td>
<td>variable</td>
<td>variable</td>
<td></td>
</tr>
<tr>
<td>5 variable</td>
<td>variable</td>
<td>fixed</td>
<td></td>
</tr>
<tr>
<td>6 variable</td>
<td>variable</td>
<td>variable</td>
<td></td>
</tr>
<tr>
<td>7 variable</td>
<td>variable</td>
<td>-</td>
<td>phase-based</td>
</tr>
</tbody>
</table>

of a threshold metric for each traffic signal, (Act2) a link specific metering rate in the perimeter (or cordon) control, (Act3) select a route as a driver’s action, (Act4) set the acceleration of the vehicles, and (Act5) set the maximum speed of the vehicles. The last three options are used in a mixed environment where the vehicles are also considered as well as traffic signals. When two or three types of actions were used, it was reported as “Multiple” in Table 7.2.

The actions directly related to control traffic signals are categorized into two main groups (phase-based and cycle-based), and seven classes where each class is defined based on cycle length, phase duration, and phase order. Each of these three elements can be fixed or variable. See Table 7.5. The decision point in cycle-based methods is the end of the cycle where cycle length, phase duration, or phase order are determined. In phase-based methods, the decision is made at the end of a phase, which includes phase duration determination and phase selection. In this case, phase duration can be set as fixed for the entire phase or can be allowed to be extended at the end of the phase. Note that in phase-based methods, the cycle and phase order are not applicable. In Table 7.5, the first class is not applicable in RL design because it is counter-intuitive to optimize while all the three elements are fixed.

62% of the papers proposed phase-based methods (i.e. class 7), while 32% constructed their methods on cycle-based methods. The number of research papers in each class of the cycle-based methods are 2, 13, 4, 18, and 11, respectively, for classes 2 to 6. This indicates that only 2 papers focused on using RL, assuming that the phase duration is fixed. Moreover, only 20 (12%) papers
worked on applying RL in a fixed cycle-length setup. There are also 3 papers with multiple actions or sets of actions, and 6 papers in which the action is not clearly or completely defined. The details related to each paper are given in Table 7.2.

**Action selection methods and parameters**

To select the actions, various action selection methods can be used. Based on the data we retrieved, 71 (44%) papers did not state which action selection they used, which is a significant number. \( \epsilon \)-greedy has the highest usage and was observed in 51 (32%) papers. Softmax (or Boltzmann) and greedy methods are used in 17 (11%) and 7 (5%) papers, respectively. Other action selection methods with a frequency of 4 or less are Random, Distributed W-Learning (DWL), credit assignment algorithms, \( \epsilon \)-softmax, and Upper Confidence Bound (UCB). Two papers employed multiple or combined action selection methods, while two others compared various action selection methods.

- **\( \epsilon \)-greedy strategy and SoftMax**: \( \epsilon \)-greedy used the epsilon term to balance exploration and exploitation of the environment, encouraging the former early on and switching to the later as the algorithm learns. It randomly selects actions for the next round based on the values of the exploration rate (\( \epsilon \)), discount factor (\( \gamma \)) and learning rate (\( \alpha \)). SoftMax behaves similarly but with weight parameters either assigned or learned to each action. They can be quite sensitive to changes and encourage an outcome where important parameters have more value. The sensitive nature of the weights makes them tricky to learn/find and can affect the performance of the algorithm significantly.

As part of the research, we also collected data about the RL parameters, including: exploration rate, discount factor and learning rate. Upon compilation of the data for action selection methods for all parameters, for example, it was discovered that the information was not displayed in a number of cases. 121, 60, and 86 out of 160 papers did not reveal the information about the exploration rates, discount factors, and learning rates, respectively.
In papers where the data are presented, the most common option was for the authors to acknowledge that these parameters exist within the range of (0,1). While it is a piece of information, it is not especially useful seeing as this information is well known among common practitioners of reinforcement/machine learning.

An interesting fact to note is that in the papers where the values of the exploration rate and learning rate are specified, there is a high tendency amongst authors to keep these values as low as possible or decreasing near the end of the process. The low learning rate values may indicate that the step size is small and that the authors want to get the best possible solutions to their reinforcement learning problem. A low exploration rate suggests that over time, the traffic controllers stop looking for new solutions and rely on the solutions already discovered.

**State space and action space discretization**

Reducing state space (for the states such as queue size, flow rate, and density), action space, and even reward space is one of the ways that can be used to reduce computational cost to make the methods more applicable in real-time process in the real world. To this end, the continuous states are grouped, for example, in 3 levels: low, medium, and high. Another way is using the comparison between the states with the previous step. This way the space is divided into two groups: better or worse. Reducing the space using grouping the data may come at the expense of the lower accuracy of the results, thus lowering the efficiency or optimality. Nonetheless, 59 (37%) of the papers used discretization while 91 (57%) did not discretize the state or action spaces. 3 (2%) papers used discretization only for the discrete methods, but not for the continuous methods that were evaluated.

**Tabular vs approximation-based methods**

Another point that impacts the efficiency of the performance of the methods in real-world is using tabular RL methods where a look-up table is used to map the spaces. Approximation methods
are used to provide a good approximation of the states that were not experienced in training. Interestingly, the number of the papers that used either of these two methods are eerily close: 74 (47%) papers used a tabular method while 72 (45%) used approximation methods.

The approximation methods that are used in our pool of research studies include the linear model tree, simultaneous perturbation stochastic approximation (SPSA), smoothed functional (SF), phase gate, tile coding, decomposable Q-function (DQF), radial basis functions (RBF), triangular-shaped functions (TF), neural networks (NN), linear function approximation (Linear FA), average cost algorithm, iterative approximation, KNN, linear combination (LC), cerebellar model articulated controller (CMAC), connectionist Q-learning framework (CQF), gradient-descent linear function approximation, double deep Q-learning, fuzzy granulation, linear regression (LR), sigmoid regression (SR), long short-term memory (LSTM), and linear Q-function. The applied neural networks in the papers consist of a denoising auto encoder (DAE), deep NN (DNN), feed forward NN (FFNN), recurrent NN (RNN), convolutional NN (CNN), deep convolutional network (DCN), perceptron, graph attentional network (GAN), dueling NN, graph convolutional network (GCN), incremental Gaussian mixture network (IGMN), back propagation NN (BPNN), fuzzy NN (FNN), neural fitted Q-iteration (NFQI), and wavelet NN (WNN). 38 papers used different neural network approximation methods, and the second-most used group is the Tile Coding, observed in 7 papers. One of these applied its own proposed method [56].

Neural networks are loosely designed after the brain and are constructed for tasks such as pattern recognition, labelling, and processing of data. They are commonly used for clustering, classification, and predictions. In the case of traffic control, prediction is the prominent use, though other use does occur. Neural networks consist of 3 main sections: input section, hidden section, and output section. Usually, there is only one of each input and output layers, but the hidden section may contain more. Each layer contains nodes. In the output layer, the number of nodes often corresponds with the number of possible outputs. In the input layer, the number of nodes often corresponds with the different types/sources of input data. The number of nodes within the hidden layers depends on
what each layer is designed to do, as well as changes with the purpose of the network. There are many types of neural networks in deep RL.

- Artificial Neural Networks (ANN): Known as a feed forward network, all incoming information is only processed and pushed in the forward direction.

- Recurrent Neural Networks (RNN): Unlike ANN, this neural network pushed processed data back to previous layers and nodes. It shares parameters across different time steps and results in fewer overall parameters. The fewer parameters allows for a smaller network.

- Convoluted Neural Networks (CNN): These networks use an extra step called convolution (for which it was named), which involves applying different "filters" to reduce the complexity of the input data. As the information filters through the network, these filters can be applied to highlight specific features of the data. This is one of the most common neural networks and is used in many disciplines.

With neural networks, deep RL becomes highly suited for complex environments presented by intersections and the dynamic changes that occur within a day of traffic.

Tile Coding is another well-known function approximator. Unlike the continuous methods such as radial basis functions (RBF), tile coding is a discretization method which is used in RL. It is a piece-wise constant approximation method that approximates the action-value functions by partitioning the state space into small regions with a constant reward value. Tilings design considers three main components: width of tiles, the resolution, and the number of required tilings based on the hyper-volume of the whole state space. For more information about Tile coding method, the reader can refer to [96].

**Deep reinforcement learning based methods**

Deep RL takes a different approach when dealing with the complex influx of data associated with traffic. It incorporates neural networks into RL algorithms and combines the advances in training
layered neural networks into abstract high-level representations of the raw input data, giving non-linear methods [107], [142]. These "layers" allow the agent to look at smaller, more reduced versions of data to extract information without an overload. The neural networks are what allow an algorithm to go "deep" and work with large or complex data input more efficiently.

Deep learning is also a tool to boost performance that was used in 27 (17%) papers. This is certainly a reasonable portion of the study, and the point to be noted is that using deep learning in RL in the network-scale began in 2016 with 2 papers. After a year with no cases, deep learning was used in 5 studies in 2018, followed by a sharp increase in 2019 with 18 papers out of a total of 27 published that year. It is expected that this trend will continue into the future, as in the first season of 2020, 3 out of 6 research papers used deep learning. SUMO has the frequency of 16 (out of 28) in conducting simulations for the deep learning related methods. VISSIM and CityFlow followed with 3 and 2 frequencies respectively. AIMSUN, PARAMICS, USTCMTS, and a custom-built tool have each been used once. The remaining three studies either did not state their methods or incorporated no traffic simulation. The papers that used deep RL methods are identified in Table 7.3.

**Code availability**

Still another feature that we investigated is the availability of the code, which we considered a good resource for those new to the field, and useful for reproducing the research. We found that in 11 papers the authors made their code available, and these can be found in Table 7.3. [88] is the first paper that made the code available in 2014. 7 of these 11 papers are deep methods published in 2018 and 2019. 2 papers provide actor-critic methods, while the rest are based on Q-learning. The codes are written for SUMO (5 papers), CityFlow (3 papers), GLD (1 paper), and AIM (1 paper). The approximation methods include NN, SPSA, Phase gate, and Tile coding.
7.3.4 Authors’ highlights and future works

Authors’ highlights

Figure 7-7 is an infographic timeline that aids in identifying past and current trends, specifically highlighting the research areas and challenges that have come to the fore most recently. It essentially indicates major first events in RL in the network-scale TSC. In addition, 7.D includes key statements made by the authors of the studies included in this chapter.

![Infographic Timeline](image)

**Figure 7-7: Major first events in RL in the network-scale TSC collected from the included papers.**

[RM]: Ramp Metering, [VMS]: Variable Message Signs, [SI]: Signalized Intersection.

Common future works suggested by the authors

During the course of our investigation, a few recurring steps the authors took in order to advance their research into the future, were noticed. One of the most recommended areas for future investigation involves testing the traffic signal controllers in the real world. Given that the final hurdle...
from theory to implementation is to see if the concept can successfully direct traffic at busy and unpredictable intersections and not just in simulations, this comes as no surprise. Authors also look to expand their work to a bigger network and to increase the number of phases that controllers could select, in addition to making traffic signal control a multi agent system and adapting to bigger intersections. In the same vein, the diversification of the proposed traffic signal formulations so that they could potentially be of greater use to more people, is also important. Adapting plans to different modes of transport, including motorized traffic such as public and mass transit, taxis, and freight vehicles, and non-motorized traffic like pedestrians and bikes are challenging at best. Incorporating better communication methods between their controllers, accounting for delays in communication and addressing noise (unwanted data) that their sensors might pick up are other points of focus for the future. Still another popular theme that arose out of our study is improving the performance of controllers. Specifically, a common direction is to change the definition of the reward function and obtain an improved state space, thereby allowing the controller to render a decision. The final future implication from this study is online learning, which is a popular choice since its strength lies in the controller’s ability to continuously adapt to traffic signal conditions. Although some efforts are already underway to achieving this, such as focusing on reducing the time required for learning during this continuous process, the area is still in its infancy.

7.4 Threats to Validity

There are threats to the validity of the results and findings of our review, which will now be discussed. Although attempts were made to select our search systematically and so that they capture the existing articles in the area under investigation, part of the included articles were retrieved during the forward and backward snowballing process. This suggests that the possibility of losing part of the existing evidence is real. Moreover, research papers that may be relevant based on our criteria might have been excluded. The authors of this research strove to collect as much relevant data as possible, and to cross reference the information for accuracy, inaccuracy remains a
possibility due to the large number of research papers and features we were dealing with. Differences in our understanding, as well as the intersection of concepts with RL, such as GT, ADP, DP, and LA may have also lead to omitting some relevant research papers that the key term "reinforcement learning" in our search string could not capture.

7.5 Conclusion

This chapter presented a comprehensive, systematic literature review on the application of reinforcement learning (RL) in the network-scale traffic signal control (TSC). The main goal of this research is to identify all eligible articles in the defined area, analyze the data of the included articles, and provide findings based on the data analysis. Considering all the published review papers we uncovered in this area (in our literature review), ours covers the highest number of articles. In this review, we studied the publication and authors data, traffic simulation and evaluation, the proposed and applied methods, and authors’ highlights and future works. And, we scrutinized and summarized the most significant elements in this area.

This chapter has shown some distinguishing points. First, it allows us to have a comprehensive view of the past 25 years of research on applying RL to TSC. This view allows us to see that the community has employed classical approaches (e.g., Q-learning) in the vast majority of the investigations. Thus, we see a large avenue for extensions, especially given that Q-learning is a tabular method and, as such, it is not fully equipped to deal with continuous spaces and/or with centralized approaches, in which the state space tends to be vast.

Second, and related to the first issue, we also note that deep learning is advancing to fill the gap left by methods that do not fully deal with huge state (and possibly, action) spaces. The number of papers employing deep learning is increasing, as demonstrated in our literature review.

Third, the use of non-commercial microscopic traffic simulators is on the rise, with SUMO being used more and more, especially within the computer science community. Associated with this trend,
is that there has been an increase in the exchange of code and experiences (e.g., SUMO has an active mailing list), which is certainly a positive trend.

Fourth, we have noted that there is a lack of interaction between traffic and transportation engineering practitioners and researchers investigating the use of RL on TSC. One of the consequences of this is that a high number of papers does not include or deal with real data, thus challenging the proper validation of the experimental results. Furthermore, no testbeds are being proposed. In fact, real-world scenarios are lacking, for which one could find at the very least a detailed map (including geometry), actual demand, fixed-time signal timings and target measurements to be used for comparison purposes. Moreover, the creation of testbeds would likely bring different communities together around common goals.

The review of the literature for the application of RL in single isolated intersections can be considered as an implication for future practice, which is complementary to this review. The integration of the results and findings of both scales can provide useful insights.
7.A Performance measures

Figure 7-1: Performance measures and the number of their respective occurrences: (Top) The most frequently used performance measures. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All utilized performance measures (with their frequency).
### Elements in State Definitions

<table>
<thead>
<tr>
<th>Elements in State Definition</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue size or average queue size (in phase, lane, downstream lane, upstream lane, link, intersection, network)</td>
<td>11</td>
</tr>
<tr>
<td>Congestion level or queue level (in phase, link, intersection, neighboring intersection, predecessor queue size (in direction), number of waiting vehicles (in lane, link, downstream link, upstream link, intersection, network))</td>
<td>7</td>
</tr>
<tr>
<td>Number of stopped vehicles (in phase, link, maximum queue size (associated with each phase, associated with each direction), saturation level in the current cycle, congestion based on comparison between consecutive phases, no threshold is set in congestion level (in link))</td>
<td>3</td>
</tr>
<tr>
<td>Distance of position of a particle with previous best position and position of its leader in Swarm</td>
<td>8</td>
</tr>
<tr>
<td>Properties of particles (in phase, lane, position of the best position encountered, leader, number of neighbors, position of the particle in the queue, index of the particle in the queue, index of the adjacent intersection with maximum queue length, index of the intersection that has been updated, phase information)</td>
<td>7</td>
</tr>
<tr>
<td>Number of vehicles (in phase, lane, position of the vehicle in the queue, position of the vehicle in the link, link, intersection, neighboring intersection)</td>
<td>6</td>
</tr>
<tr>
<td>Position of the vehicle (in intersections, adjacent intersections)</td>
<td>5</td>
</tr>
<tr>
<td>Position of the vehicle with minimum distance from the stop line, position of the vehicle in the network, position of the vehicle in the intersection, index of the phase with maximum queue length for the intersection, index of the adjacent intersection with maximum queue length, index of the intersection that has been updated, phase information)</td>
<td>4</td>
</tr>
<tr>
<td>Number of vehicles enter the network</td>
<td>3</td>
</tr>
<tr>
<td>Congestion level (in link, vehicle accumulation, average number of vehicles per lane associated with the lanes that share the signals governed by the phase agent, position of the vehicle in the network, position of the vehicle in the intersection, index of the phase with maximum queue length for the intersection, index of the adjacent intersection with maximum queue length, index of the phase that has to be set green, phase information)</td>
<td>2</td>
</tr>
<tr>
<td>Number of vehicles (in phase, lane, link, upstream neighboring intersections, downstream link, link, intersection, network)</td>
<td>1</td>
</tr>
<tr>
<td>Number of arriving/approaching vehicles (in link, intersection)</td>
<td>1</td>
</tr>
<tr>
<td>Maximum queue size (phase, associated with each phase, associated with each direction), saturation level in the current cycle, congestion based on comparison between consecutive phases, no threshold is set in congestion level (in link)</td>
<td>1</td>
</tr>
<tr>
<td>The code of combination phase</td>
<td>1</td>
</tr>
<tr>
<td>Current time of the day</td>
<td>1</td>
</tr>
<tr>
<td>Elapsed time (in phase, lane)</td>
<td>1</td>
</tr>
<tr>
<td>Elapsed green time, the time that current phase has been green since the last signal change, the time since the signal turned green, the time since the second-to-last signal change, the time since the last detecion sent by the short/long detector, the time that the current phase will last</td>
<td>1</td>
</tr>
<tr>
<td>Distance to the intersection of the n nearest vehicles, position of the vehicles in the queue, position (in terms of segments) for the first vehicle approaching the intersection</td>
<td>1</td>
</tr>
<tr>
<td>Speed or average speed (in green signal phase lane, lane in neighboring link, link, intersection, network)</td>
<td>1</td>
</tr>
<tr>
<td>Average vehicle speed (in link)</td>
<td>1</td>
</tr>
<tr>
<td>Speed of the vehicles (in phase, lane, link, intersection, neighboring intersection)</td>
<td>1</td>
</tr>
<tr>
<td>Velocities of the n nearest vehicles</td>
<td>1</td>
</tr>
<tr>
<td>Traffic light (id)</td>
<td>1</td>
</tr>
<tr>
<td>Inter-arrival times</td>
<td>1</td>
</tr>
<tr>
<td>Presence of public transport vehicles (in link, intersection)</td>
<td>1</td>
</tr>
<tr>
<td>Presence of public transport vehicles (in network)</td>
<td>1</td>
</tr>
<tr>
<td>Number of priority vehicles (in network)</td>
<td>1</td>
</tr>
<tr>
<td>Detector activation state</td>
<td>1</td>
</tr>
<tr>
<td>Number of incoming lanes (in intersection)</td>
<td>1</td>
</tr>
<tr>
<td>Total stop time</td>
<td>1</td>
</tr>
<tr>
<td>Image-like representation (of vehicles’ position, of queues, of signal)</td>
<td>1</td>
</tr>
<tr>
<td>Acceleration of the vehicles</td>
<td>1</td>
</tr>
<tr>
<td>Destination of vehicles</td>
<td>1</td>
</tr>
<tr>
<td>Static attributes of links, capacity, link indicator, jam density</td>
<td>1</td>
</tr>
<tr>
<td>Number of on-ramps/approaching vehicles (in link, intersection)</td>
<td>1</td>
</tr>
<tr>
<td>Maximum vehicles in the green phase</td>
<td>1</td>
</tr>
<tr>
<td>Value of an objective function</td>
<td>1</td>
</tr>
<tr>
<td>Number of vehicles (in lane, link, network)</td>
<td>1</td>
</tr>
<tr>
<td>Distance of position of a particle with previous best position and position of its leader in Swarm</td>
<td>1</td>
</tr>
<tr>
<td>Value of an objective function</td>
<td>1</td>
</tr>
<tr>
<td>Number of vehicles enter the network</td>
<td>1</td>
</tr>
<tr>
<td>Distance of position of a particle with previous best position and position of its leader in Swarm</td>
<td>1</td>
</tr>
<tr>
<td>Value of an objective function</td>
<td>1</td>
</tr>
<tr>
<td>Number of vehicles (in lane, link, network)</td>
<td>1</td>
</tr>
<tr>
<td>Distance of position of a particle with previous best position and position of its leader in Swarm</td>
<td>1</td>
</tr>
<tr>
<td>Value of an objective function</td>
<td>1</td>
</tr>
<tr>
<td>Number of vehicles enter the network</td>
<td>1</td>
</tr>
<tr>
<td>Distance of position of a particle with previous best position and position of its leader in Swarm</td>
<td>1</td>
</tr>
<tr>
<td>Value of an objective function</td>
<td>1</td>
</tr>
<tr>
<td>Number of vehicles (in lane, link, network)</td>
<td>1</td>
</tr>
<tr>
<td>Distance of position of a particle with previous best position and position of its leader in Swarm</td>
<td>1</td>
</tr>
<tr>
<td>Value of an objective function</td>
<td>1</td>
</tr>
<tr>
<td>Number of vehicles enter the network</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 7-2: Elements in state definitions: (Top) The most frequently used elements in state definitions. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All elements used in state definitions (with their frequency).
7.C Elements in Reward Functions

![Figure 7-3: Elements in reward definitions: (Top) The most frequently used elements in reward definitions. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All elements used in reward definitions (with their frequency).](image)

7.D Detailed major first events
<table>
<thead>
<tr>
<th>Citation</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Su et al., 2007 [38]</td>
<td>This application of Q-learning and SensorGrid can be seen as the first step towards expanding the usage of SensorGrid.</td>
</tr>
<tr>
<td>Kuyer et al., 2008 [42]</td>
<td>The first application of max-plus to a large-scale problem (not in small applications) and thus verifies its efficacy in realistic settings.</td>
</tr>
<tr>
<td>El-Tantawy et al., 2010 [64]</td>
<td>The first study that has tackled the Integrated traffic control problem (Ramp Metering (RM), Variable Message Signs (VMS), and Signalized Intersections (SI)) to find a closed-loop optimal control solution using a coordination mechanism that minimizes the communication requirements.</td>
</tr>
<tr>
<td>Prashanth et al., 2010 [56]</td>
<td>The first application of RL with function approximation for TSC.</td>
</tr>
<tr>
<td>Waskow et al., 2010 [63]</td>
<td>The first attempt to tackle the dimensionality problem in MARL by means of function approximation.</td>
</tr>
<tr>
<td>Natarajan et al., 2011 [71]</td>
<td>The first adaptation of a Statistical Relational Learning framework for the problem of learning relational policies from an expert (imitation learning).</td>
</tr>
<tr>
<td>Prashanth et al., 2011 [67]</td>
<td>The first to design RL-based TSC algorithms that minimize a long-run average cost criterion.</td>
</tr>
<tr>
<td>Nuli et al., 2013 [83]</td>
<td>No traffic adaptive control model exists to account for traffic heterogeneity and limited lane discipline.</td>
</tr>
<tr>
<td>El-Tantawy et al., 2014 [94]</td>
<td>The first study to investigate the effect of $\text{TD}(\lambda)$ methods for TSC as a continuing task (i.e., not a finite episode) with a discounted reward in which looking ahead to future steps is less important compared to a finite episodic task with undiscounted reward.</td>
</tr>
<tr>
<td>Zhu et al., 2015 [106]</td>
<td>Junction Tree Algorithm has not been applied to address the coordinated signal control problem.</td>
</tr>
<tr>
<td>Wang et al., 2016 [118]</td>
<td>Other works with the same functionality (predicting future system states by Proactive Complex Event Processing) have not been found.</td>
</tr>
<tr>
<td>Prashanth et al., 2016 [119]</td>
<td>The first work to combine cumulative prospect theory (CPT) with RL, and to investigate (and define) human-centered RL.</td>
</tr>
<tr>
<td>Darmoul et al., 2017 [127]</td>
<td>The first to integrate Case-based reasoning and RL for TSC and integrate immune features within MARL to achieve disturbance management.</td>
</tr>
<tr>
<td>Wei et al., 2018 [138]</td>
<td>None of existing studies have used the real traffic data to test their methods.</td>
</tr>
<tr>
<td>Li et al., 2018 [142]</td>
<td>The study of the applicability of deep RL on the road network has not yet been carried out.</td>
</tr>
<tr>
<td>Torabi et al., 2018 [145]</td>
<td>Validated on the largest realistic simulated traffic network published to date for collaborative multi-agent based TSC.</td>
</tr>
<tr>
<td>Vinitsky et al., 2018 [149]</td>
<td>The first to propose a standard set of benchmarks for traffic control in a micro-simulator and a framework for simultaneously learning control for a mixture of AVs interacting with human drivers and infrastructure in which deep RL can be applied to the control task.</td>
</tr>
<tr>
<td>Zhou et al., 2019 [171]</td>
<td>Edge based RL is the first RL proposal to optimize traffic signals on neighborhood scale.</td>
</tr>
<tr>
<td>Tan et al., 2019 [163]</td>
<td>The first attempt to use hierarchical deep RL models in large-scale TSC.</td>
</tr>
<tr>
<td>Chu et al., 2019 [165]</td>
<td>The first paper to present a fully scalable and decentralized MARL algorithm for the state-of-the-art deep RL agent: independent advantage actor critic (IA2C), within the context of ATSC by extending the idea of Independent Q-Learning on A2C.</td>
</tr>
<tr>
<td>Wei et al., 2019 [166]</td>
<td>The first time that the individual RL model automatically achieves coordination along arterial without any prior knowledge.</td>
</tr>
<tr>
<td>Xu et al., 2019 [167]</td>
<td>The first work to consider the impact of slow learning in RL on real-world applications by the effective transfer of RL algorithms trained on simulated traffic to the real-world traffic to reduce the mistakes to be made in the real world.</td>
</tr>
<tr>
<td>Rizzo et al., 2019 [168]</td>
<td>The first to address signalized roundabouts in congested network, as a complex TSC scenario using a deep RL method.</td>
</tr>
<tr>
<td>Zheng et al., 2019 [169]</td>
<td>The first work to reduce the problem space and explore different scenarios more efficiently, so that the RL algorithm can find the optimal solution within a minimal number of trials, instead of blindly exploring on repeated situations.</td>
</tr>
<tr>
<td>Wei et al., 2019 [153]</td>
<td>The first work to use GAN in RL for TSC and to conduct experiments on the large-scale road network with hundreds of traffic signals.</td>
</tr>
<tr>
<td>Ni et al., 2019 [174]</td>
<td>The first to extend RL to the <strong>cordon-control</strong> problem.</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Rizzo et al., 2019 [159]</td>
<td>The first to consider <strong>model explanation methods</strong> such as LIME and SHAP for the explanation and interpretation of RL agents decisions that can be verified by domain experts (<strong>RL with Explainability</strong>).</td>
</tr>
<tr>
<td>Chen et al., 2019 [158]</td>
<td>This study is among the earliest to apply <strong>deep RL</strong> for <strong>arterial</strong> adaptive signal control.</td>
</tr>
</tbody>
</table>


[23] Zahra Shakeri Hossein Abad, Vincenzo Gervasi, Didar Zowghi, and Behrouz H Far. Supporting analysts by dynamic extraction and classification of requirements-related knowledge. In *2019


[103]


215


[166] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In


Abstract

This chapter proposes a hierarchical perimeter control method in a large-scale urban network consisting of car and public transit modes. In this chapter, we propose a hierarchical proportional-integral (PI) control based deep reinforcement learning (deep RL) method to control each intersection on the perimeter independently. The PI control in the high-level of our controller is designed to reduce the state space and at the same time reflect the perimeter control input in the reward function of the deep RL method. The PI control is established on a passenger-based MFD of the network, called 3D-pMFD, which provides the set points, i.e., optimal points, of both inside and outside regions. Using the multivariable PI controller’s parameters based on the obtained set points, a reinforcement learning method is applied in the low-level controller to time each traffic signal on the perimeter without receiving any data from other traffic signals in the network, except vehicles accumulations in the network. The results show a significant improvement of the proposed method compared to the fixed time and PI control methods in terms of total passenger delay.

8.1 Introduction

Regulating traffic signals along the perimeter of a controlled area in a network, known as the perimeter control, has shown efficiency in reducing congestion and travel time within the network ([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]). Given the existence of a well-defined Macroscopic Fundamental Diagram (MFD) for a network ([13, 14, 15, 16, 17, 18, 19, 20, 21]), the inflows can be controlled to keep the network accumulation close to the critical value, although this may come at the expense of
the increased delay for the vehicles outside the controlled area. There have been extensive research
efforts in distributing the inflows via spatially-uniform perimeter plans; however, in comparison, the
methods of spatially-varying distributing the inflows offer better performance than spatially-uniform
methods ([4, 22, 23]).

RL algorithms are currently well suited to develop adaptive traffic signal control systems. More
specifically, with the advances in deep learning and deep RL, the application of deep RL approaches
in this area is appealing, where the issue of dimensionality is managed. Recently, [24] proposed a
reinforcement learning approach to control perimeter inflows via spatially-varying metering rates,
which is the first use of reinforcement learning in the research area of perimeter control. Generally,
an important advantage of the perimeter control is that it does not entail high computational effort,
and it is supported by stability analysis ([25, 26, 27, 28, 29]). A large body of work in this area is
dedicated to the uni-modal networks. However, public transit (e.g. buses) is an inevitable part of a
sustainable urban traffic management system. Compared to mass transit, as in urban rail transit,
buses use street roads to provide more convenient access for the passengers. On the other hand, buses
can transfer more passengers using less space. Considering the substantial role of buses in urban
networks, many researchers have provided different strategies in the area of bi-modal and multi-
modal traffic control. For example, some researchers elaborated on traffic signal control to maximize
passenger flows at the local level by providing bus priority ([30, 31, 32]). The bus priority may be
devoted through intermittent and dedicated bus lanes ([33, 34, 35]), flexible-sharing strategies such
as queue jumper lanes [36] and pre-signals ([37, 38]).

Distributing urban space between different modes to maximize passenger flows has also been
found effective as another solution [39]. However, only a few studies focused on developing bi-modal
strategies at the network level, specifically as perimeter control, e.g. [40]. The research in this recent
area has been facilitated by extending the concept of MFD to multi-modal p-MFD [41], bi-modal
3D-MFD (or 3D-vMFD) and passenger 3D-MFD, also known as 3D-pMFD [42]. The 3D-MFD
relates the accumulation of cars and buses and the vehicle production (or in some cases, the total
circulating vehicle flow) in the network, while in 3D-pMFD, the network passenger production (or
flow) is taken into account to consider that buses carry more passengers. Similar to MFD, 3D-MFD is shown that exists empirically ([43, 44]). In most bi-modal perimeter control methods, the total inflow of the modes is regulated ([40]). In contrast, [45] proposed a bi-modal perimeter control where the inflows of the two modes can be regulated independently. As indicated by the authors of the paper, this is the first strategy in the literature that enables buses to bypass the car queues in the perimeter of the network.

The focus of our work is on the bi-modal urban networks consisting of cars and buses. This research aims at proposing a bi-modal perimeter control method to provide public transit priority in the perimeter of an urban network by using a deep reinforcement learning approach. Each traffic signal on the perimeter has its own controller and has only access to the data of the immediate approaches at the intersection and operates independently of other traffic signals. To the best of our knowledge, this is the first research to integrate PI and deep RL control methods as a hierarchical method to control the perimeter in a bi-modal network.

8.2 Problem Statement

Consider a bi-modal urban network with multiple intersections, in which each intersection on the perimeter is controlled independently. The traffic network is partitioned into two regions $\rho_i$, $i=1,2$, including outside region $\rho_1$ and inside region $\rho_2$. The network consists of the car mode and public transit mode. We denote the modes with $\mu \in \{c,b\}$, where c and b delegate the car and public transit modes, respectively. The vehicle accumulation in each region of the network is denoted with $N_i = (N_i^c, N_i^b)^T$ where $N_i^\mu$ is the accumulation of the respective mode in each region. Also, the passenger production in each region of the network is $p_i = (p_i^c, p_i^b)^T$ where $p_i^\mu$ is the passenger production of the respective mode in each region. The total passenger production in each region of the network, $P_i$, is obtained based on the car outflow $q_i^c$, the car passenger occupancy $h_i^{cp}$, the actual bus passenger outflow $q_i^{bp}$, and the link length $l$ as follows.
\[ P_i = p_i^c + p_i^b = (h_{cp} \cdot q_i^c + q_i^{bp}) \cdot l. \] (8.1)

For the car mode, we assume that the number of passengers per car is \( h_{cp} = 1.2 \). Except for the perimeter, each intersection is a 4-way signalized intersection consisting of 3 lanes with left-turn, through, and shared straight and right-turn movements. On the feeder links of the perimeter, at the distance of 100 m from the stopline the 3-lane setup alters to the 4-lane one with a left-turn bay, through, through dedicated bus lane, and shared through and right-turn movements, see Figure 8-1.

The following section proposes a hierarchical system, including the high-level and low-level controllers. First, we explain the details of the high-level controller where a PI control is designed. In the low-level of our system, each intersection on the perimeter is controlled by an RL agent, which observes part of the entire system’s condition. Each agent operates without any communication with other agents except for receiving the data of accumulations in the network. The objective is to minimize the average travel time and delay in the network. To this end, the control decision for all the intersections on the perimeter is made at the beginning of the cycles based on the observation at that moment.

### 8.3 Methodology

The proposed methodology consists of three main steps: (i) constructing 3D-pMFDs to determine the set points, (ii) designing the PI controller as input to deep RL controller, and (iii) designing deep RL controller. The proposed framework is provided in Figure 8-1 to clarify the process.

### 8.3.1 Constructing 3D-pMFDs to determine the set points

Management of two region MFDs can enhance urban mobility and mitigate congestion in cities [2]. The reasoning behind this is to keep the network production about the optimal (maximum) values by controlling the inflows to the inner region to maintain the vehicles’ accumulation around their
optimal points. The optimal accumulation values of cars and buses in the two regions, namely the set points, are obtained from the 3D-pMFDs of these two regions. Various control methods can be leveraged to keep the network around this optimal state. To this end, we design a multivariable proportional integral discrete controller with 8 parameters and 4 set points for the two regions and two modes of transport. $\hat{N}_2^c$, $\hat{N}_2^b$, $\hat{N}_1^c$, $\hat{N}_1^b$ represent the controller set points extracted from the 3D-pMFDs in this step, for cars inside the region, buses inside the region, cars outside the region, and buses outside the region, respectively.

Figure 8-1: Framework of the proposed method
8.3.2 High-level controller: PI controller

Integral control and PI control have been found suitable in previous research work in the perimeter control [46, 4] for uni-modal systems, i.e. cars. Considering the usefulness of this control method, in this article, we devise the following control law as a multivariable PI controller:

\[
U_{12}(k) = U_{12}(k - 1) - \left( K_{P2}^c \cdot (N_2^c(k) - N_2^c(k - 1)) + K_{P2}^b \cdot (N_2^b(k) - N_2^b(k - 1)) \right) \\
+ \left( K_{P1}^c \cdot (N_1^c(k) - N_1^c(k - 1)) + K_{P1}^b \cdot (N_1^b(k) - N_1^b(k - 1)) \right) + \left( K_{I2}^c \cdot (\hat{N}_2^c - N_2^c(k)) + K_{I2}^b \cdot (\hat{N}_2^b - N_2^b(k)) \right) \\
- \left( K_{I1}^c \cdot (\hat{N}_1^c - N_1^c(k)) + K_{I1}^b \cdot (\hat{N}_1^b - N_1^b(k)) \right), \tag{8.2}
\]

where \(U_{12}(k)\) and \(U_{12}(k - 1)\) are perimeter controllers in the current and previous time steps, \(K_{P2}^c, K_{P2}^b, K_{P1}^c, K_{P1}^b\) denote proportional gains for cars inside the region, buses inside the region, cars outside the region, and buses outside the region, respectively. Also, \(K_{I2}^c, K_{I2}^b, K_{I1}^c, K_{I1}^b\) denote integral gains for cars inside the region, buses inside the region, cars outside the region, and buses outside the region, respectively.

The output of the PI controller, i.e. \(U_{12}(k)\), is given to the RL controller (explained in the next section) to be considered as one of the state and reward elements of the controller. The main logic behind considering \(U_{12}(k)\) as a state is that \(U_{12}(k)\) can reflect the network control requirement and status. The value of 0 means not letting more vehicles into the inner region, while 1 means the inside region should attract more vehicles to increase the network production. The continuous values between 0 and 1 indicate the tendency to each of these extremums. We also include \(U_{12}(k)\) in the reward function as an indication of goodness or badness of decisions that are made during the RL training.
8.3.3 Deep RL controller

A Markov Decision Process (MDP) is a mathematical framework well suited to optimize decision-making processes under uncertainty. An MDP is a four-tuple \( (S, A, R, T) \), including, respectively, state space, action space, reward function, and transition function. An MDP satisfies the Markov Property if the transition function, whether known or unknown, depends only upon the current state and the action taken, not on the sequence of events that preceded it.

If the reward and transition functions are known, the optimal policy can be found using dynamic programming methods via the recursive definitions of the value function. However, when the environmental dynamics are not known, i.e. unknown reward and transition functions, the agent has to estimate the value of taking action in a state without using knowledge about the reward function and transition probabilities. In this situation, Reinforcement Learning (RL) is suitable. RL can be model-based where the agent samples from the environment to estimate the reward and transition functions and find an optimal policy. Unlike model-based RL, in model-free RL algorithms, the agent directly estimates the Q-function from experience while the reward and transition functions are unknown beforehand.

In model-free RL, an agent tries to learn the optimal way of interacting with an environment. RL learns how an agent should map the states to actions to maximize a numerical reward. At each time step (or decision point) \( k \), based on a policy \( \pi \) that is intended to be optimized during the learning process towards reaching an optimal policy \( \pi^* \), the agent takes action \( a_k \) from a set of possible actions \( A \) in response to the current state \( s_k \) from a set of possible states \( S \); i.e. \( a_k = \pi(s_k) \). Simply put, a policy is a rule that the agent follows in selecting actions based on its current state. At the end of step \( k \), the agent receives a reward \( r_k \) from the environment based on a reward function \( R \) where the elements of the reward can be collected through sensors. A sequence of state, action, and reward is a history of an agent saved in memory. At each time step, the RL agent tries to learn an optimal policy, from its history of interactions with the environment, that maximizes the discounted cumulative reward:
where $\gamma \in [0,1]$ is a discount factor. The discount factor is associated with time horizons and is used to balance immediate and future rewards. It determines how much the RL agent cares about rewards in the distant future compared to those in the immediate future. If $\gamma$ is 0, the agent only cares about the most immediate reward ($R_k = r_k$). If $\gamma$ is 1, the reward is not discounted and the distant future reward is considered ($R_k = r_k + r_{k+1} + r_{k+2} + ...$). As we set $\gamma$ closer to 1, future rewards are given greater emphasis relative to the immediate reward. For more details, see [47].

One of the most frequently used and successful RL methods in traffic signal control is Q-Learning [47]. Q-Learning is a model-free RL. It is also an off-policy RL algorithm that uses a different policy for estimating Q-values than for action-selection. It updates the Q-values of the current state-action pair using the greedy policy to estimate the Q-value of the optimal policy of the next state-action pair. In other words, the optimal policy $\pi^*$ is learned by estimating a second function, called Q-function, that specifies the value of an action (following a given policy $\pi$) given the current state. Q-function calculates the quality of a state-action combination. Assuming the agent continues to follow the optimal policy, the Q-value is defined as the expected discounted future reward of taking action $a_k$ in state $s_k$:

$$Q^\pi(s_k, a_k) = \mathbb{E}_\pi [R_k | s_k, a_k] = \mathbb{E}_\pi [\sum_{\phi=k}^{\infty} \gamma^{(\phi-k)} r_\phi | s_k, a_k].$$

The Q-value is estimated by iterative Bellman updates:

$$Q^\pi(s_{k+1}, a_{k+1}) = Q^\pi(s_k, a_k) + \alpha (\Psi_k - Q^\pi(s_k, a_k)),$$

where $\alpha \in [0,1]$ is the learning rate that is set through experimentation, and the target is $\Psi_k = r_k + \gamma \max_{a_{k+1}} Q^\pi(s_{k+1}, a_{k+1})$. Hence:
Here, we should note that \( R_{k+1} \) in Equation 8.3 is the reward that is obtained at the next time step (after taking action), but it is not necessarily known upfront. That is why Q-Learning is a model-free RL that does not know about the model of the environment, i.e. the reward and transition functions.

In the learning process, there is the exploration-exploitation dilemma. The agent tries to exploit based on what it already learned to achieve the reward, and at the same time, it also must explore possible actions for each state to find the one that has received the highest reward for exploiting it.

In brief, Q-learning, is the process of iteratively updating Q-Values for each state-action pair using the Bellman Equation until the Q-function eventually converges to \( Q^* \). Instead of estimating the Q-value of each state-action pair separately in Q-Learning, deep reinforcement learning algorithms [48] use deep neural networks as function approximators to map from states to Q-values. This makes possible the use of a larger and/or continuous state space through parameterization [49]. The integration of artificial Neural Nets (NNs) into the Q-learning process is referred to as Deep Q-Learning, and a network that uses NNs to approximate Q-functions is called a Deep Q-Network (or DQN). In other words, DQN is a Q-learning, which is parameterized with deep NN with parameters \( \theta \), i.e. \( Q(s, a; \theta) \). The neural network input is the state, the number of output neurons is the number of the possible actions, and the targets are the Q-values of each of the actions.

Unlike Q-learning, whose convergence in the limit (infinity) is guaranteed, we do not have such guarantees for DQN. This is because (i) the data set is not i.i.d. and (ii) as the agent learns, the targets move [50]. To ameliorate this issue, we use a dueling architecture [51] and target network[52].

As discussed prior, in DQN, the Q-value indicates how effective it is to take a certain action given a certain state. However, in dueling DQN [51], the Q-value is decomposed into two values, including the value function \( V^\pi(s; \theta) \) and advantage function \( A^\pi(s, a; \theta) \); \( Q^\pi(s, a; \theta) = V^\pi(s; \theta) + A^\pi(s, a; \theta) \). The value function specifies how good it is to be in any given state. The advantage function A(a)
indicates the advantage of taking a certain action in a certain state compared to the other actions. In dueling DQN, the network separately computes the value and advantage functions and combines them back into a single Q-function only at the final layer to estimate the Q-function as below:

\[
Q^\pi(s, a) = V^\pi(s; \theta) + A^\pi(s, a; \theta) - \frac{1}{N} \sum_{a'} A^\pi(s, a'; \theta),
\]

where \( N \) is the number of actions. The purpose of the third term is to improve stability. Note that the output of the model is the sum of the value and advantage functions. However, for training the model, the same Q-value for targets as before is used.

In addition to the dueling DQN, target network freezing is also used to solve the overoptimistic problem. By freezing the target network for a period of time, the targets are partially stabilized. For target network freezing, the Q-value estimation is split into two different networks, a value network to estimate the Q-value of the current state-action pair, \( Q^\pi(s, a) \), and a target to compute the targets \( \Psi_k \).

This chapter implemented and customized the framework proposed by Rodrigues et al. [53] for a single isolated intersection. We used a neural network architecture with 2 fully-connected hidden layers with 16 neurons and ReLU activations [54] in each layer and 1 output layer of size \( 2 \times n_p + 1 \), where \( n_p \) is the number of phases used in action set. We used Dropout layers [55] to make the controller more robust and prevent the neural network from overfitting. Adam optimizer [56] is also used with a learning rate of 0.0005 and the update rate of the target network of 0.01. In addition, we employed experience replay with a memory size of 5000 observations. Exploration is done through an \( \epsilon \)-greedy policy, with the \( \epsilon \) parameter between 0.5 and 0.1 during the first 1500 steps.

In the following we define state and action spaces and the reward function.

**State space** (S)—The observation (or state space S) includes the local observation and the perimeter controller, \( U_{12}(k) \). The perimeter control gain is obtained from PI-based perimeter control.
based on car accumulation, bus accumulation, and mixed passenger production in both inside and outside regions. The perimeter control parameter provides knowledge of the system from a macroscopic control to the local controller in the training stage. This manages and reduces the state space of the RL agents while learning and testing. The local states include the number of cars and buses in the feeder links; that is where the dedicated bus lanes are set, during each cycle. We also consider the downstream impact of spillback on the feeder links. This is done to ensure there should be enough space for the feeder links to discharge into downstream links.

Consequently, we define four states, including (i) number of cars in the feeder links during a cycle time, (ii) number of buses in the feeder links during a cycle time, (iii) the sum of empty spaces in the downstream links of the feeder links, and (iv) the filtered value of $U_{12}$ considered for timing. Here, we should clarify the filtered value of $U_{12}$. The obtained actual value of $U_{12}$ may be less than zero or more than one. For the timings, we transform the values greater than 1 to 1, and the values less than 0 to 0.

**Action space (A)**—The action space consists of $2 \times n_p + 1$ actions and is designed based on a time extension control in the decision points, which are made at the end of each cycle. The cycle time is variable, and the actions are (i) to keep the phase timings of the previous cycle for the current cycle, or (ii) to increase/decrease the phase time of one (out of four) of the phases of the previous cycle for the current cycle. The minimum and maximum phase times are 10 and 50 sec. Other intersections in the network are controlled by a fixed time method similar to the one used in PI control and in retrieving the 3D-pMFDs. Similarly, the sequence of the phases is fixed. A fixed 3-sec yellow time is also considered between each cycle.

**Reward function (R)**—For the reward function, we consider different elements, including the number of cars and buses crossed the intersection (on the perimeter) in the incoming lanes from the outside region to the inside region, saturation flow rate $S$, the previous cycle length (which had been determined in the previous decision point), $C$, and $U_{12}$. The cycle time is the total of the phase times and yellow times.
Figure 8-2: Demonstration of the reward function. $x$ denotes $\frac{\sum_{j=1}^{n_f} (S_j \cdot g_j)}{C}$ and $r$ is the immediate reward.

We consider the dependency between the number of crossed vehicles during a cycle and $U_{12}$; for instance, in the case that the number of crossed vehicles in a cycle is close to the saturation outflow (that is the highest outflow rate), and $U_{12} = 1$, the reward should be maximum. Also, if the number of the crossed vehicles in a cycle is low and close to zero, and $U_{12} = 0$, the reward should be maximum. Additionally, when $U_{12}$ is between the range of 0 and 1, the reward function should change linearly according to Figure 8-2. As a result, the reward function is defined as below:

\[
 r = \begin{cases} 
 -(\sum_{j=1}^{n_f} (N^c_j + N^b_j)) + (\sum_{j=1}^{n_f} (S_j \cdot g_j)) \cdot (1 + U_{12}) 
 & \text{if } U_{12} = 0 \\
 (\sum_{j=1}^{n_f} (N^c_j + N^b_j)) + (\sum_{j=1}^{n_f} (S_j \cdot g_j)) \cdot (1 - U_{12}) 
 & \text{if } 0 < U_{12} < 1 \quad \& \quad \sum_{j=1}^{n_f} (N^c_j + N^b_j) \geq U_{12} \cdot \sum_{j=1}^{n_f} (S_j \cdot g_j) \\
 -(\sum_{j=1}^{n_f} (N^c_j + N^b_j)) + (\sum_{j=1}^{n_f} (S_j \cdot g_j)) \cdot (1 + U_{12}) 
 & \text{if } 0 < U_{12} < 1 \quad \& \quad \sum_{j=1}^{n_f} (N^c_j + N^b_j) \geq U_{12} \cdot \sum_{j=1}^{n_f} (S_j \cdot g_j) \\
 (\sum_{j=1}^{n_f} (N^c_j + N^b_j)) + (\sum_{j=1}^{n_f} (S_j \cdot g_j)) \cdot (1 - U_{12}) 
 & \text{if } U_{12} = 1,
\end{cases}
\]

where $j$ accounts for the feeder links, $n_f$ is the number of the feeder links, and $g$ denotes the green time (or phase time) of the feeder links.
Figure 8-3: Agent types including corner and non-corner agents. The top-right intersection and its adjacent intersection are shown to show the two types of the agents. The feeder lanes on the feeder links are shown in green. The outgoing lanes which discharge to outside region are depicted in blue. The lanes in outside region that discharge to the outside region but through the inside region are not considered as a feeder lane, see the west-north left turn movement on the feeder link in the corner agent.

It should be noted that we have two types of agents: corner agents and non-corner agents. The reason is that the intersections on the corner of a perimeter have 2 feeder links, while the non-corner intersections have only 1 feeder link, see Figure 8-3.

Also, note that $r$ is the immediate reward that is calculated in each step. In the proposed method current reward is compared with the immediate reward of the past step. This is called the relative reward $r_r$ and is obtained as $r_r(k) = r(k) - r(k-1)$. The relative reward is used as the reward in the process. Also, it is worth noting that the first term of the reward in Equation 8.8 is actually the outflow rate during the cycle.
8.4 Results

8.4.1 Experiment Setup

The experiments are conducted in AIMSUN microsimulation environment. To evaluate the proposed method, a 20 × 20 network including 400 intersections is used. The link lengths inside and outside of the cordon are deemed to be different. Inside the cordon, the links are 120 m long with three lanes, including left-turn, through, and shared right-turn and through lanes. The center region contains 64 intersection. The lane configuration outside the cordon is kept intact except for the links where a through bus reserved lane is added to dedicate priority to the buses when entering/approaching the center region. In terms of link length, the intersections outside the cordon are designed in two ways: the intersections with 350 m links (144 intersections), and the intersections with both 120 m and 350 m links (192 intersections), as shown in Figure 8-1.

A 4-hour time-varying demand scenario is employed, as depicted in Figure 8-4. Simulation replications are run with various random seeds with a 15-minute warm-up period at the beginning of each experiment. Trip origins and destinations were distributed over the entire network, inside and outside the cordon, as both internal and external centroids. The internal centroids are designed to operate similar to side streets entering each link. The setup let the network experience the range of saturation from low to high demand conditions.

The inflows to and outflows from the controlled region are measured by loop detectors located at all the incoming and outgoing links of the intersections on the perimeter boundary.

Pre-timed control with 96-sec cycle time without any designed progression (i.e. off-set is zero for all intersections) is used to denote the strategy without cordon metering. The cycle of all intersections in the network except the intersections on the perimeter is composed of four phases, including two 30-sec phases for the through and right turns and two 10-sec phases for the left turns, plus four 4-sec interphases. For the intersections on the perimeter the cycle is designed with four
Figure 8-4: Demand in the network over time.

20-sec phases each of which manages a complete link with four 4-sec interphases, i.e. $C = 96$ sec. The tests include en-route dynamic traffic assignment.

We designed 58 bus lines with 1168 bus stops in various bus route patterns. The bus stops regularly distanced at most 350 m away from each other. Most of the bus lines (~75%) pass through or end in the protected region. The bus departure times schedule are designed such that the time interval between departures starts with 20 min interval, goes down to 10 min and rises up to 25-30 min interval towards the end of the test. Different variants of dwell times are assigned to each stop, but we cared for them to be consistent with the cars demand. All feeder links are 350 m and each contains a 120 m left-turn bay. The reserved lane for public transit is 75 m long.

8.4.2 PI controller

In our problem, we deal with both cars and buses. Different from uni-modal 3D-MFDs in that the target is the maximum vehicles production, we highlight the importance of public transit through a 3D-pMFD by considering the maximum passenger production of both car and public transit modes, which we call “mixed passenger production”. The controller set points are the accumulation of cars and buses inside and outside the cordon. To obtain the PI controller’s set points, we first construct the 3D-pMFD of the network by conducting 10 simulations in distinct car and bus
demands under the already mentioned fixed-time control. The 10 replications’ demands vary +/- 5%. For the inside region, the 2D passenger production diagrams versus car accumulation and bus accumulation separately are shown in Figure 8-5. The figures depict the data points hourly for the 4-hour scenarios in four different colors. We used the following functional forms to fit surfaces to the same data points and construct 3D-pMFDs of the inside region:

\[
P_i = \alpha_1 N_{2}^{b3} + \alpha_2 N_{2}^{b2} + \alpha_3 N_{2}^{b} + \alpha_4 N_{2}^{c3} + \alpha_5 N_{2}^{c2} + \alpha_6 N_{2}^{c} + \alpha_7 N_{2}^{b} N_{2}^{c3} + \alpha_8 N_{2}^{b2} N_{2}^{c},
\]

(8.9)

The estimated parameters for inside region are as follows: \( \alpha_1 = -3.837 \), \( \alpha_2 = 270.9 \), \( \alpha_3 = -502.2 \), \( \alpha_4 = 0.0001486 \), \( \alpha_5 = -0.08987 \), \( \alpha_6 = 156.4 \), \( \alpha_7 = -0.0000000569 \), and \( \alpha_8 = 0.0002126 \). The resulted 3D-pMFDs with fitted surfaces are shown in Figure 8-6.

Based on the estimated parameters, the controller set points are obtained as follows: \( N_{2}^{c} = 1265 \), \( N_{2}^{b} = 49 \), \( N_{1}^{c} = 31656 \), and \( N_{1}^{b} = 322 \). The root mean square error (RMSE) and R-squared of the fitted surfaces for the inside region 3d-pMFD are 4687 (pax.m/sec) and 0.3948, respectively.
Figure 8-6: The fitted surfaces of 3d-pMFDs of (a) inside region, and (b) outside region.
indicates the passenger unit). The values of these measures of error for the outside region 3d-pMFD are 3971 and 0.4116, respectively.

Following the extracted values of the set points, we select one of the 10 replications to set the $K_P$ and $K_I$ parameters (8 parameters). To set these parameters, we used the trial and error method and tried various values to find the most desirable results of the PI controller. In the process of determining the parameter, we considered the following points: (i) the value of $K_I$ should be more than $K_P$, because the difference between the set point and the current point is usually more important than the difference between the current and previous points, (ii) the impact of buses should be more than cars; therefore, $K_P^b$ should be greater than $K_P^c$ and $K_I^b$ should be greater than $K_I^c$, (iii) the average of changes of the values during the simulation and the set points can give hints to get as much closer as to the best performance of the PI controller, which is the main goal of this process, (iv) the $U_{12}$ should not go far beyond 0 and 1, (v) the values of $U_{12}$ should react as soon as possible to the changes of the traffic condition, (vi) the initial values of $U_{12}$ should be 1, as in the beginning of the simulations the protected region is uncongested, and (vii) the ratio of $K_P$ and $K_I$ for buses and cars should be proportional.

Considering all the points mentioned above and running different simulations, we found the following values the most suitable: $K_{P2}^b = 0.0002$, $K_{P2}^c = 0.008$, $K_{P1}^b = 0.00001$, $K_{P1}^c = 0.002$, $K_{I2}^c = 0.00015$, $K_{I2}^b = 0.007$, $K_{I1}^c = 0.000002$, and $K_{I1}^b = 0.0002$.

Considering the set points and PI control parameters, the PI control is applied to the network. As shown in Figure 8-7, the results illustrate that the PI controller could reasonably keep the car and bus accumulations in both inside and outside regions about the set points.

Also, $U_{12}$ could successfully react to the congested and uncongested conditions inside and outside the perimeter. For instance, about at time 40 minutes, where car accumulation is about 2000 veh, going far beyond its set point, i.e. 1265 veh, $U_{12}$ reacts to the congestion in the inside region with a 25 minutes lag and could reduce the inflow to the inside region to control the congestion in the protected region. At about time 80 minutes, the car accumulation inside the region is still above the congestion level, while the bus accumulation inside the region is still above the car and bus accumulation in
the outside region are all in uncongested condition. However, compared to the time at 40 minutes, the car and bus accumulations in the outside region are closer to their set points, about 24000 and 250 veh, respectively, and the bus accumulation in the inside region is also closer, about 30 veh, to its set point. In this case, $U_{12}$ also reacts and tries to attract more buses (and forcibly cars) to the inside region. This time the reaction is applied with 10 minutes lag and $U_{12}$ touches the value of 0 at about 90 minutes. Then, for about 135 minutes, the PI control keeps the value of $U_{12}$ between about 0 and 0.4 to let the vehicles go inside the region but very smoothly, not in the maximum possible flow rate. This way, the number of cars and buses in both inside and outside regions are controlled to be as much as possible close to the set points. Finally, at the time point of 225 minutes, $U_{12}$ reacts to the uncongested condition in the inside region and rises up to let more, towards the maximum flow rate, vehicles in. This continues to the end of the experiment.

Also, the PI controller outperformed the fixed time controller, which will be discussed in the next section after experimenting with the proposed PI-based deep RL controller.

8.4.3 Proposed controller

8.4.4 Training

Following the discussed details of the RL method and the experiment setup, we conducted the training process simulating 4-hour experiments for 80 episodes. Figure 8-8 presents the results. This figure shows the passenger total travel time across the network vs the number of episodes used in the training as the proposed controller’s performance. At the beginning of the training phase, an increasing trend is observed. This may be because the controller is learning through increasing/decreasing the phase time for only 5 sec in each phase and the results are close to the fixed time method. The decreasing trend of the measure starts from episode 21 until 67.
Figure 8-7: a) $U_{12}$ over time in PI control, and b, c, d, e) comparison between fixed time control and PI control in terms of car and bus accumulations in inside and outside regions. The dashed lines indicate the set points.
Figure 8-8: RL control: Training process.

Figure 8-9: Comparison of the three methods: total passenger delay over time

8.4.5 Evaluation

After the training process, the proposed method is tested and the outcomes are compared with the fixed time (or no perimeter control) and the designed PI control. The results of total mixed passenger delay over time are presented in Figure 8-9. Table 8.1 shows the network-wide average of mixed passenger delay time (sec) and mixed passenger average travel time (sec/km). The PI control provided 8.6% and 6.1% reductions in delay and average travel time, respectively, compared to the fixed time controller. Furthermore, the proposed RL controller outperformed the PI controller by 9.3% and 7.4% reduction, respectively.
Table 8.1: Comparison of the performance of the three controllers. The values in parentheses indicate the measures’ percentage of reduction of PI and proposed methods compared to the fixed time method.

<table>
<thead>
<tr>
<th>Control Method</th>
<th>Mixed Passenger Delay (sec)</th>
<th>Mixed Passenger Average Travel Time (sec/km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed time controller</td>
<td>368.94</td>
<td>722.58</td>
</tr>
<tr>
<td>PI controller</td>
<td>337.05 (8.6%)</td>
<td>678.49 (6.1%)</td>
</tr>
<tr>
<td>Proposed controller</td>
<td>302.79 (17.9%)</td>
<td>627.13 (13.5%)</td>
</tr>
</tbody>
</table>

The difference in the performance between the PI controller and the proposed RL controller shows that the impact of the time lag between the time of sensing the congestion and the time that $U_{12}$ requires to reach either 0 or 1 in the PI controller, has been alleviated in the proposed RL controller.

8.5 Conclusion

In this chapter, we proposed a hierarchical PI control based deep reinforcement learning method to control the perimeter in a large-scale urban network. The PI controller is used as one of the state elements as well as in the reward function. The RL method is applied to each intersection independently. This application of PI control in designing the RL method helps to provide the RL method with a good knowledge of the entire network facilitated by the PI controller. This also helps to reduce the state space. The results showed that the proposed method could significantly outperform the fixed time and PI control methods. As future work, the proposed method can be applied to a network with variable perimeters.
Bibliography


253


This thesis studied the traffic control problem from different angles using different tools, including data analysis, traffic theories, and deep reinforcement learning methods. Synthesized understanding of the contextual factors in traffic control is explored in Part I (Chapters 2, 3, and 4). Part II and III concentrate on developing real-time control methods for large-scale congested transportation networks to improve the network performance. In Part II, with the focus on uni-modal traffic, Chapter 5 studied the isolated intersection and Chapter 6 proposed a real-time decentralized traffic signal control for congested urban networks considering queue spillbacks. Part III, after conducting a systematic literature review on the application of reinforcement learning in the network-scale traffic signal control in Chapter 7, proposed a hierarchical bi-modal perimeter control with priority based on a multivariable proportional-integral discrete control for high-level control and a deep reinforcement learning for low-level control in Chapter 8. The current chapter briefly summarizes the findings and main contributions of each part, elaborates on the potential field applications, and outlines the directions for future research. The detailed conclusions and contributions are also provided at the end of each chapter.
9.1 Part I: Exploratory studies: application of data analysis

9.1.1 Exploring the most popular traffic engineering-related topics of discussion amongst practitioners

The main contributions of Chapter 2 are summarized as follows:

- Exploring and identifying the main discussion topics about traffic engineering among practitioners, which provides insights into the different categories and problem areas of traffic engineering. These topics are classified into: (i) road transportation management, and application of GIS tools and technologies (e.g. Google Map) to perform geo-spatial data analysis, (ii) environmental aspects of transportation engineering, such as transportation planning and signal timing for optimizing the travel time, and fuel consumption, and (iii) traffic data collection and analysis.

- Conducting statistical analysis on the data set retrieved from Q&A websites to measure the proportions of the types of questions asked about traffic engineering. This helps identify the challenges facing traffic engineers that require more attention from the traffic engineering and management research and development communities in the future.

We used data from the popular social Q&A sites (e.g. Stack Overflow and Engineering Exchange), and analyzed 2,457 questions and answers in order to examine the needs, concerns and questions of transportation engineers. We applied Latent Dirichlet Allocation (LDA) based topic models and statistical analysis to explore the main related topics to traffic engineering.

One possible direction to extend the research of this chapter is to conduct a parallel study with academic papers rather than Q&A sites, and comparing results from the two.

9.1.2 A requirements elicitation tool for traffic management systems

The main contributions of Chapter 3 are listed as following:
• We prototyped a requirements elicitation tool for traffic management systems which aims to tackle problems associated with eliciting requirements for traffic management services such as emergency medical services, traffic signal timing, and urban transportation planning systems. We used machine learning methods, such as Natural Language Processing and Naïve Bayes to help with the requirements elicitation and classification task in the traffic management domain.

• The tool bridges the gap among stakeholders from both areas of software development and transportation engineering. The RETTA prototype is mainly designed for requirements engineers and software developers in the area of traffic management.

• The tool is designed and developed using Android platform for mobile phones and tablets. Following the complexity of the computation and data analysis in the traffic management domain, we used the Model View ViewModel (MVVM) architecture to design the tool in a way to be easy to develop.

Future work will concentrate on improving the efficiency and completing the text analysis and classification approaches for exploring and classifying requirements. Moreover, the applied requirements elicitation process in the RETTA tool can be improved to address the complexity and the scale of the crowd and to ensure that we record their requirements efficiently and precisely.

9.1.3 Social media data analysis for traffic management systems

By analyzing social media data, including text and image, Chapter 4 provided a methodology to efficiently explore social media data in the context of traffic management and find out what can be understood from what users post and share.

• We used three different sources of data, including theory-based publications, Google Trends, and Twitter data.

• We used a systematic search method to explore search strings for filtering data collected from social media. We have conducted a mixed-method study including both manual qualitative
analysis, and automatic information extraction using weighted finite-state transducers (WFST), Biterm Topic Modelling (BTM), and deep neural networks (DNN) on Twitter data.

- To address the problem from different perspectives, we analyzed two data types, i.e. text and image, using NLP, deep neural networks, and descriptive statistics.

- We utilized Canadian traffic information from Twitter to look for issues and relevant information that may assist authorities and software development teams in making decisions when designing and developing traffic management systems by leveraging lay people’s input.

- We found that the self-reported traffic information by lay users on Twitter can be a valuable source to characterize traffic management systems. Moreover, we found that although theory-based publications in the context of traffic management systems can help with traffic estimation, control, and prediction, they are insufficient to characterize the context-sensitive aspects of these systems.

Concerning future steps and practical implications of this research, Amazon Mechanical Turk (AMT) is an appropriate option that can be used to label the collected data for further sentiment analysis (i.e. emotions, anxiety level, cultural characteristics, etc.). Another direction to take is replicating this study on more social media platforms, such as Instagram, Facebook, and more static Q&A/review platforms such as Yelp, Reddit, and Quora. Developing a tool that enhances the application of crowdsourcing in developing traffic management systems, focusing on dynamic tracking of generated information on social media over time through novel forms of natural language and image processing methods, is another possible extension to this work.
9.2 Part II: Uni-modal network-scale traffic signal control: application of traffic theories

9.2.1 Traffic signal timing optimization for the isolated intersection

Chapter 5 proposed an optimization model that provides the optimum cycle time and green splits when the total average delay at a general isolated signalized intersection is minimized for all vehicles present. To this end, we contributed the followings:

- We modeled the lost time effect in the shock wave delay model, which creates the most desirable optimum cycle time values. In our optimization process, the key strategy is to keep both approaches in the saturated condition. Our method outperformed the Webster method in the isolated intersection.

- The number of required variables is low and easily measurable.

Although the goal is to present a solution for the isolated intersection case, the proposed method can be extended in the coordinated systems and network-scale traffic signal control.

9.2.2 Real-time decentralized traffic signal control for congested urban networks considering queue spillbacks

The contributions of Chapter 6 are as below:

- We proposed a real-time decentralized network-level traffic signal control method for urban networks addressing the effects of queue spillbacks.

- The proposed method is traffic-responsive, does not require data communication between intersections’ controllers, uses lane-based queue measurements, and is acyclic.

- All control decisions in the network are entirely local. In decentralized systems, the local controllers have no interactions with other intersections in terms of both input and output data.
in decision-making computations. This way, the reliability of the system increases by removing the need for communications between intersections. Thus, communication problems, such as network delays, do not affect the control system.

- Our proposed method is designed in such a way that it comes with a high level of accuracy, requires a low number of data types in the queue length estimation, and a low volume of the required data in the decision making process. The method is cost-effective in data collection, data processing for both estimation and signal timing, and data communication in the real world.

- The proposed method resulted in a feasible solution in all conditions in the entire network with any scale in a short amount of time, which makes it favourable for real-time applications.

Numerical results demonstrated that the proposed method outperforms other well-known benchmark methods in both isolated intersection and network configurations.

The proposed method can open doors to different research areas. It can be (i) integrated with a route recommendation system, (ii) incorporated within a hierarchical control scheme like Perimeter control, (iii) combined with transit priority, Emergency Medical Services (EMS) priority, and multimodality to provide a comprehensive method as a solution for a complex city-scale network, (iv) enhanced by studying the impact of link length and noise in the queue length and arrival flow estimations, (v) developed by considering lanes with mixed movements, and (vi) used in reinforcement learning based methods.
9.3 Part III: Bi-modal network-scale traffic signal control: application of reinforcement learning

9.3.1 A systematic literature review of Reinforcement Learning in network-scale traffic signal control

This chapter presented a comprehensive systematic literature review on the application of reinforcement learning (RL) in the network-scale traffic signal control.

- The main goal of this research is to identify all eligible articles in the defined area, analyze the data of the included articles, and provide findings based on the data analysis.

- Considering all the published review papers we uncovered in this area (in our literature review), ours covers the highest number of articles.

- In this review, we studied the publication and authors data, traffic simulation and evaluation, the proposed and applied methods, and authors’ highlights and future works. We also surveyed and summarized the most significant elements in this area.

Some findings of this study are presented below:

- This study allows us to have a comprehensive view of the past 25 years of research on applying RL to network-level traffic signal control. This view allows us to see that the community has employed classical approaches (e.g., Q-learning) in the vast majority of the investigations. Thus, we see a large avenue for extensions, especially given that Q-learning is a tabular method and, as such, it is not fully equipped to deal with continuous spaces and/or with centralized approaches, in which the state space tends to be vast.

- We note that deep learning is advancing to fill the gap left by methods that do not fully deal with huge state (and possibly, action) spaces. The number of papers employing deep learning is increasing, as demonstrated in our literature review.
• The use of non-commercial microscopic traffic simulators is on the rise, with SUMO being used more and more, especially within the computer science community. An associated positive trend is the increase in the exchange of code and experiences (e.g., SUMO has an active mailing list).

• We have noted that there is a lack of interaction between traffic and transportation engineering practitioners and researchers investigating the use of RL on traffic signal control. One of the consequences of this is that a high number of papers does not include or deal with real data, thus challenging the proper validation of the experimental results. Furthermore, no testbeds are being proposed. In fact, real-world scenarios are lacking, for which one could find at the very least a detailed map (including geometry), actual demand, fixed-time signal timings and target measurements to be used for comparison purposes. Moreover, the creation of testbeds would likely bring different communities together around common goals.

The review of the literature for the application of reinforcement learning in single isolated intersections can be considered for future practice, which is complementary to this review. The integration of the results and findings of both scales can provide useful insights.

9.3.2 Proposing a hierarchical bi-modal perimeter control by integrating proportional integral and deep RL methods

This chapter proposed a hierarchical bi-modal network-level perimeter control method to improve the network-wide performance by using a deep RL method. We proposed a PI control-based deep RL method to control the perimeter in a large-scale urban network with 400 intersections. The deep RL controller uses the PI controller as one of the state elements as well as in the reward function.

The contributions of Chapter 8 are discussed in the following:

• Designing a multivariable PI control and applying the controller as a high level input to an RL controller as a low-level controller to control the perimeter of a bi-modal network

• Applying deep RL in a hierarchical system
• Designing both high-level and low-level controllers

• Considering various factors in designing the RL controller, including providing bus priority, addressing spillback conditions, and using the output of PI controller in designing the elements of the RL controller

• Conducting microsimulation for evaluation of the proposed method

The RL method is applied to each intersection independently. This application of PI control in designing the RL method helps to provide the RL method with a coherent knowledge of the entire network facilitated by the PI controller. This also helps to reduce the state space. The results showed that the proposed method could significantly outperform the fixed time and PI control methods. In future work, the proposed method can be applied to a network with variable perimeters.
Permissions for Reusing Published Papers in PhD Thesis

The published papers in Chapter 2 is Springer copyrighted, Chapters 3-4 are IEEE copyrighted, and Chapter 5 is ASCE copyrighted material. Chapters 6-8 are journal publications, which are currently under 1st or 2nd round of reviews. This Appendix presents the copyright permissions for the papers published and used in this thesis.

**Notice of the IEEE Copyright** In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Calgary’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to [http://www.ieee.org/publications_standards/publications/rights/rights_link.html](http://www.ieee.org/publications_standards/publications/rights/rights_link.html) to learn how to obtain a License from RightsLink.
SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Jan 27, 2021

This Agreement between Mohammad Noaeen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number 4995190991063
License date Jan 24, 2021
Licensed Content Publisher Springer Nature
Licensed Content Publication Springer eBook
Licensed Content Title Transportation Engineering on Social Question and Answer Websites: An Empirical Case Study
Licensed Content Author Mohammad Noaeen, Zahra Shakeri Hossein Abad, Guenther Ruhe et al
Licensed Content Date Jan 1, 2018
Type of Use Thesis/Dissertation
Requestor type academic/university or research institute
Format print and electronic

https://s100.copyright.com/CustomerAdmin/PLF.jsp?ref=ca335b7b-653b-40dc-96fa-381ce6b0550f

267
<table>
<thead>
<tr>
<th>Portion</th>
<th>full article/chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Will you be translating?</td>
<td>no</td>
</tr>
<tr>
<td>Circulation/distribution</td>
<td>1 - 29</td>
</tr>
<tr>
<td>Author of this Springer Nature content</td>
<td>yes</td>
</tr>
<tr>
<td>Title</td>
<td>Transportation Engineering on Social Question and Answer Websites: An Empirical Case Study</td>
</tr>
<tr>
<td>Institution name</td>
<td>University of Calgary</td>
</tr>
<tr>
<td>Expected presentation date</td>
<td>Jan 2021</td>
</tr>
</tbody>
</table>

Mr. Mohammad Noaeen

Calgary, AB
Canada
Attn: Mr. Mohammad Noaeen

Total 0.00 CAD

Terms and Conditions

**Springer Nature Customer Service Centre GmbH**

**Terms and Conditions**

This agreement sets out the terms and conditions of the licence (the Licence) between you and **Springer Nature Customer Service Centre GmbH** (the Licensor). By clicking 'accept' and completing the transaction for the material (Licensed Material), you also confirm your acceptance of these terms and conditions.

1. **Grant of License**
Let's Hear it from RETTA: A Requirements Elicitation Tool for TrAffic Management Systems

Conference Proceedings:
2017 IEEE 25th International Requirements Engineering Conference (RE)

Author: Mohammad Noaeen

Publisher: IEEE

Date: Sept. 2017

Copyright © 2017, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]

2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]

2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.

3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
## American Society of Civil Engineers - License Terms and Conditions

This is a License Agreement between Mohammad Noaeen ("You") and American Society of Civil Engineers ("Publisher") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by American Society of Civil Engineers, and the CCC terms and conditions.

All payments must be made in full to CCC.

### LICENSED CONTENT

<table>
<thead>
<tr>
<th>Publication Title</th>
<th>Country</th>
<th>Rightsholder</th>
</tr>
</thead>
<tbody>
<tr>
<td>International Conference on Transportation and Development 2016 : Projects and Practices for Prosperity</td>
<td>United States of America</td>
<td>American Society of Civil Engineers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/01/2016</td>
<td>English</td>
</tr>
</tbody>
</table>

### REQUEST DETAILS

<table>
<thead>
<tr>
<th>Portion Type</th>
<th>Page range(s)</th>
<th>Total number of pages</th>
<th>Format (select all that apply)</th>
<th>Who will republish the content?</th>
<th>Duration of Use</th>
<th>Rights Requested</th>
<th>Distribution</th>
<th>Translation</th>
<th>Copies for the disabled?</th>
<th>Minor editing privileges?</th>
<th>Incidental promotional use?</th>
<th>Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>397-408</td>
<td>10</td>
<td>Print, Electronic</td>
<td>Author of requested content</td>
<td>Life of current edition</td>
<td>Main product</td>
<td>Worldwide</td>
<td>Original language of publication</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

### NEW WORK DETAILS

<table>
<thead>
<tr>
<th>Title</th>
<th>Institution name</th>
<th>Expected presentation date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic Signal Timing Optimization by Modelling the Lost Time Effect in the Shock Wave Delay Model</td>
<td>University of Calgary</td>
<td>2021-01-24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instructor name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mohammad Noaeen</td>
<td></td>
</tr>
</tbody>
</table>

### ADDITIONAL DETAILS

<table>
<thead>
<tr>
<th>Order reference number</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>