

Connected and Automated Mobility Modeling on Layered Transportation Networks:
Cross-Resolution Architecture of System Estimation and Optimization

by

Jiawei Lu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2022 by the
Graduate Supervisory Committee:

Xuesong (Simon) Zhou, Chair
Ram Pendyala
Guoliang Xue
Hans Mittelmann

ARIZONA STATE UNIVERSITY

December 2022

ABSTRACT

The emerging multimodal mobility as a service (MaaS) and connected and automated mobility (CAM) are expected to improve individual travel experience and entire transportation system performance in various aspects, such as convenience, safety, and reliability. There have been extensive efforts in the literature devoted to enhancing existing and developing new methodologies and tools to investigate the impacts and potentials of CAM systems. Due to the hierarchical nature of CAM systems and associated intrinsic correlated human factors and physical infrastructures from various resolutions, simply considering components across different levels into a single model may be practically infeasible and computationally prohibitive in operation and decision stages. One of the greatest challenges in existing studies is to construct a theoretically sound and computationally efficient architecture such that CAM system modeling can be performed in an inherently consistent cross-resolution manner. This research aims to contribute to the modeling of CAM systems on layered transportation networks, with a special focus on the following three aspects: (1) layered CAM system architecture with a tight network and modeling consistency, in which different levels of tasks can be efficiently performed at dedicated layers; (2) cross-resolution traffic state estimation in CAM systems using heterogeneous observations; and (3) integrated city logistics operation optimization in CAM for improving system performance.

ACKNOWLEDGMENTS

I am indebted to a great number of people who help me in both work and life during my doctoral study at Arizona State University and make the completion of this dissertation possible.

I would like to thank my supervisor Dr. Xuesong (Simon) Zhou for his patient guidance in my study, helping me develop a solid foundation and valuable research skills for my future career. He provides me with a great platform and free environment where I have the chance to perform any research of interest.

I would like to thank my committee members, Professors Ram Pendyala, Guoliang Xue, and Hans Mittelmann, for their kind support and insightful suggestions through my doctoral study, especially in improving my dissertation. I also would like to express my appreciations to the faculty and staff at Arizona State University, especially those who offered me high-quality courses and support. I would like to thank Professor Pitu Mirchandani for offering the Network Flows and Algorithms course and Professor Dimitri Bertsekas for offering the Reinforcement Learning course.

Finally, I would like to thank my family members for their unconditional support and love. Their accompany and encouragement make me confident to face any difficulties in the past, present, and future.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES.....	viii
CHAPTER	
1 INTRODUCTION.....	1
1.1 Background	1
1.2 Challenges and Motivations.....	4
1.3 Research Overview	7
1.4 Organization of the Dissertation	10
2 LITERATURE REVIEW	12
2.1 Layered CAM System Modeling Architecture	12
2.2 Cross-resolution Traffic State Estimation in CAM Systems	18
2.3 Integrated City Logistics Operation Optimization in CAM Systems	26
3 LAYERED CAM SYSTEM MODELING ARCHITECTURE	33
3.1 Introduction.....	33
3.2 Virtual-track-based Multiresolution Networks and Associated Modeling Focuses	36
3.3 CAM System Optimization on Layered Multiresolution Networks	41
3.4 CAM System Simulation on Layered Networks with Hierarchical Driving Decisions.....	50
3.5 Partially Schedulable CAM System Operation on Layered Multiresolution Networks.....	56

CHAPTER	Page
3.6 Open-source Tools for Enabling Cross-resolution Modeling	60
3.7 Experiments	66
3.8 Conclusions	70
4 CROSS-RESOLUTION TRAFFIC STATE ESTIMATION IN CAM	
SYSTEMS.....	71
4.1 Introduction	71
4.2 Problem Statement and Overall Modeling Framework	75
4.3 Derivation of Macroscopic Traffic System Dynamics Based on Fluid Queue Models	79
4.4 Formulating Cross-resolution Traffic System State Identification Problem as a Nonlinear Programming Model	85
4.5 Modeling Inconsistencies Across Different Resolutions	90
4.6 Traffic System State Identification on a Computational Graph.....	103
4.7 Numerical Experiments	112
4.8 Conclusions	128
5 INTEGRATED CITY LOGISTICS OPERATION OPTIMIZATION IN CAM	
SYSTEMS.....	129
5.1 Introduction	129
5.2 Modeling Time-dependent Travel Time and Congestion Impacts Using Fluid Queue Models with Polynomial Arrival Rates	132
5.3 Sprinkler Truck Routing: An Arc Routing Application with Rich Constraints in a Congested Traffic Network	140

CHAPTER	Page
5.4 Mathematical Formulations of the Optimization Problem	144
5.5 Exact Solution Methods	159
5.6 Computational Experiments.....	170
5.7 Conclusions.....	183
6 CONCLUSIONS AND FUTURE RESEARCH	184
6.1 Summary of the Dissertation	184
6.2 Theoretical Contributions and Broader Impacts	186
6.3 Future Research	187
REFERENCES	190

LIST OF TABLES

Table	Page
1. Comparison of Related Studies on TSE, MPE, and QPE.	26
2. Three Major Methods for Representing Time-dependent Travel Times in VRP/ARP.	29
3. Multiresolution Network Representation.	37
4. Network Size in Different Resolutions.	67
5. Model Statistics under Different Demand Levels to Highlight the Need for Developing Decomposition Methodologies.	70
6. Parameters/Functions to Be Estimated in the TSSI Problem.	78
7. Summary of the System States Based on Fluid Queue Model.	84
8. Notations for Formulating the Traffic System State Identification Problem.	85
9. Shape of the Parameters in the FCNN (Density Distribution Function).	107
10. Summary of the Highd Dataset Used in This Research.	113
11. Configurations of Virtual Traffic Detectors.	114
12. Accuracy of Speed Estimations on Six Datasets.	117
13. Evaluation of Speed Estimations with Fixed Traffic Flow Model Parameters. .	120
14. Configurations of Traffic Detectors.	123
15. Configurations of Virtual Traffic Detectors.	126
16. Notations Used in Model M1-TEN.	149
17. Additional Notations Used in the Node Routing Model.	153
18. Parameters of Polynomial Travel Time Functions.	158
19. Characteristics of the 12 Corridors.	172

Table	Page
20. Comparison Between Observed and Modeled Speed Across All Links and 15-min Interval Resolution on 12 Selected Corridors.	173
21. Characteristics of Three Benchmark Instance Sets.....	176
22. Performance Comparisons of Three Proposed Models on Small-size Instances.	177
23. Performance Comparisons of Three Proposed Models on Medium-size Instances.	178
24. Performance Comparisons of Three Proposed Models on Large-size Instances.	178
25. Optimality Gaps of Models M1-TEN and M3-CTR on Large-size Instances....	181
26. Optimal Routing Costs of Small Instances under Different Weights of System Impact ω	182

LIST OF FIGURES

Figure	Page
1. Three Classes of CAM Modeling Tools.	18
2. Multiresolution CAM System Modeling Framework on Layered Networks.	40
3. Mapping Between Spatial-continuous and Spatial-discrete Representations for a Two-lane Road Segment.	42
4. A Sample Physical Network and Its Corresponding Space-time Graph.....	43
5. Minimum Time Headway Modeling on the Space-time Network with Time Lags.	44
6. Vehicle Path Planning and Trajectory Control on Layered Transportation Networks with Boundary Consistencies at Two Ends of Each Link.....	47
7. Spatial-discrete Microscopic Road Link Representation and Corresponding Virtual Turning Layers with Meso-to-micro Intermediate Destination (MID) Nodes.	53
8. Illustration of the Consistency Between KW, CF(L) and CA(M).	54
9. Graphic Illustration of Conflict Modeling on the Proposed Virtual-track-based Network.....	60
10. Intersection Consolidation and Movement Generation.	62
11. Drivable, Bikeable, and Walkable Network near Arizona State University, Tempe Campus.	63
12. Points of Interest and Boundary Nodes Identified by osm2gmns on a Sample Network near Arizona State University, Tempe Campus.....	64
13. Entire United States Driving Network Generated by osm2gmns.	64
14. System Architecture of CAMLite.	66

Figure	Page
15. Macroscopic Network of the Research Area of Interest.	67
16. Simulation Results.	68
17. Sample Optimized Vehicle Trajectories in a Virtual-track-based Space-time Network.....	70
18. Illustration of the Fluid Queue and Continuous Space-time Modeling on Road Segments with Different Types of Traffic Detectors.....	76
19. Cross-resolution Framework Proposed in This Research.	79
20. Graphic Illustration of the Queuing System on Road Links with a Downstream Bottleneck.	81
21. Graphic Illustration of Time-dependent Travel Delay Derivations.	83
22. Physical Road Segments and Sampling Points on the Corresponding Space-time Plane.....	87
23. Illustration of the Three-detector Model.	95
24. Illustrations of Using Loop Detector Data.....	96
25. Illustrations of Using Speed Information from GPS Data.	97
26. Illustrations of Using Space-time Trajectory Information from GPS Data.	99
27. Illustrations of Using Aggregated Density Information from Video Detector Data.	100
28. Computational Graph Structure.	105
29. Detailed Representation of the fully connected Neural Network (Density Function).	107
30. Illustration of the Distributed Modeling.	110

Figure	Page
31. Changes in the Density and Speed Distribution Function in the Implementations.	115
32. Comparison Between Speed Observations and Estimations on Six Datasets.....	119
33. Loss Convergence Curves.....	119
34. Speed Estimation on Dataset 1 Without Considering the Flow Conservation Law.	121
35. Impact of GPS Sampling Rate on Estimation Quality (Dataset 1).	122
36. Layout of the Freeway Corridor on I880-N (Postmile 22-25), Adopted from the PeMS.....	123
37. Estimation Results of System-wide Measures at the Macroscopic Level.	124
38. Speed Estimations of the Proposed Method and PeMS.....	125
39. Illustration of the Hypothetical Corridor.	125
40. Comparison Between Speed Observations and Estimations on a Hypothetical Corridor.....	128
41. General Graphical Illustration of Queue Evolution for a Road Link, Adapted from Newell (2013).	133
42. Graphic Illustration of the Four-step Method for Calibrating the Key Parameters in Eq. (83).	136
43. Illustration of Road Link Marginal Delay Adapted from Ghali and Smith (1995).	140
44. An Illustrative Network and the Route of a Sprinkler Truck.....	144
45. A Simple Network and Its Corresponding Time-expanded Network.....	146

Figure	Page
46. A Simple Network and Its Corresponding Time-expanded Networks in SRP...	148
47. An Illustrative Example of Time-dependent Travel Time Modeling in Three Models, Ranging from High-fidelity Discretization M1-TEN, Semi-dynamic Slot-based Discretization M2-STD and Continuous-time Representation M3-CTR.	158
48. The Overall Framework of the BnB Module.	170
49. Corridors Used for Time-dependent Travel Time Modeling Evaluation.	172
50. Heatmap Comparison Between Observed and Modeled Speed on Corridor US29 Middle_W.	174
51. Heatmap Comparison Between Observed and Modeled Speed on Corridor I395_E.	174
52. Modeled Speed and Observed Speed Comparison on Two Links.	175
53. Pareto Curve of Operation Cost and System Impact on Instance S5 with ω Ranging from 0.0 to 1.0.	183
54. Vehicle Dynamics Modeling on a Space-time-velocity Network (Adopted from Zhou et al., 2017).	188

CHAPTER 1

INTRODUCTION

1.1 Background

As population, economic growth, and personal travel activities continue to increase, traffic congestion, air quality, and sustainability issues require systematic and innovative solutions based on a deep understanding of overall demand and supply interactions. Recent emerging trends in multimodal mobility as a service (MaaS) and connected and automated mobility (CAM) made available by public-private partnerships may create a revolutionary paradigm shift for automatic mobility applications (Jittrapirom et al., 2017). CAM technologies are expected to provide convenient and reliable travel services with seamless connections across different layers of multimodal transportation systems using individualized active traffic management. It is only a matter of time before the transportation infrastructure of freeways, roads, and traffic control systems must accommodate self-driving vehicles (SDVs) at the same time as manually driven vehicles (MDVs). In large scale systems that exist in nearly all metropolitan areas, the question is how can we efficiently, reliably and safely accomplish this?

Recognizing the differences between CAM and existing human-driver-oriented transportation system, researchers have begun to develop new methodologies and enhance existing analysis, modeling, and simulation (AMS) tools to evaluate and quantify the effects of CAM from various aspects, such as efficiency, reliability, and safety. Owing to the rapid deployment of telecommunication and vehicular technologies, it can be expected that CAM systems will be complicated hierarchical systems involving human factors and physical components from various resolutions. From a computational architecture

perspective, simply considering components across different levels in a single model may be practically infeasible and computationally prohibitive in the operation and decision stages. A layered decomposition approach needs to be adopted in the design of CAM system architectures. In the field of telecommunications, providing a typical basis for the coordination of industry standards, the open systems interconnection model has seven inter-related conceptual layers, including the physical, data link, network, transport, session, presentation, and application layers (Zimmermann, 1980). Each layer is responsible for dedicated tasks and shares necessary information with the adjacent layers. This layer-based decomposition structure dramatically reduces the modeling complexity and increases the system reliability, while, at the same time, introduces challenges in designing effective feedback and coordination mechanisms to synchronize the status of different layers.

In the field of transportation, the multiresolution modeling (MRM) framework has been applied in the analysis and simulation of multimodal transportation systems (Hadi et al., 2022; Zhou et al., 2021). A typical MRM structure includes the macroscopic, mesoscopic, and microscopic layers. The major challenge of adopting MRM methodologies is ensuring the inherent consistency between different resolutions, in terms of network representation consistency or performance measure consistency. A feedback loop is required to execute different models at different layers using a fixed-point solution to achieve a higher degree of modeling consistency. This research highlights the need of designing an inherently consistent layered CAM modeling framework, such that critical tasks such as system state estimation and operation optimization can be efficiently

performed, and layer consistency ranging from high-level trip requests to low-level vehicle motion planning or platoon can be achieved.

This dissertation aims to develop fundamental knowledge needed to design such a system. More specifically, the research will study and develop decision models and algorithms, and attendant decision-support systems to manage, in real time, large fleets of SDVs and MDVs on the current infrastructure, without the need to construct special roads or guideways. This research will assume that SDVs are cyber-connected and that, through cyber mechanisms of computing and communication, it is possible to guide these SDVs, both individually and in platoons, on our transportation infrastructure efficiently, without sacrificing comfort, safety and efficiency in mobility. The fundamental concepts in the decision-support architecture are (a) controlling directions, speeds and stops to individual SDVs in real-time, (b) grouping SDVs in platoons, (c) moving SDVs in platoons, with short and uniform headways, using the concept of cyber-enabled virtual tracks on the roads, and (d) providing traffic signals on the roads and blocking control (platooning sizing and dispatching) on the virtual tracks to maximize throughput and other desirable traffic performance measures. In addition to the tools developed for operating SDVs in real time, this research aims to help transportation planning agencies to efficiently satisfy increasing transportation demand with limited road infrastructure expansion and constrained road capacity through efficient urban logistics solutions. Finally, the research and tools will be integrated into the current and new open-source ecosystems for computer science, operations research, and transportation engineering.

This dissertation will address fundamental knowledge in networking self-driving agents at large scales to meet temporally and spatially distributed traveler demand. The

goal is to develop a set of new models for integrated traveler mobility optimization and multi-agent-based control under the new environment of shared CAM networks. It will investigate a novel cyber-track based concept and methods that optimally provide real-time guidance to meet temporally and spatially distributed traveler demand for travelling agents (from origins to destinations), possibly leading to new large scale nonlinear optimization methods that include vehicular dynamics and safety/comfort consideration.

By taking full advantage of distributed computing power associated with connected SDVs, the dispatching and operating system for SDVs will simultaneously route and control individual SDVs and platoons on existing highways and streets. Based on a space-time cyber track network modeling framework for representing physical transportation system with constraints, the dissertation will also develop real-time algorithms for proactive control of traffic supply infrastructure that optimizes delays and other performance metrics. The dissertation will integrate parallel computing and hierarchical system control, as well as a wide range of vehicle routing/scheduling algorithms, to ensure the safety, efficiency and reliability of CAM operations. The research will also study the computational tractability of large-scale deployment of SDVs using tools of cloud computing, computational graphs and parallel computation. The research will utilize standard model protocols, such as general modeling network specification (GMNS), of collecting streaming data from connected SDVs and managing, distributed computing, and effective logistics for SDV fleets.

1.2 Challenges and Motivations

Although substantial efforts have been devoted to CAM modeling with significant progress in recent years, there are still some critical research gaps need to be addressed, especially in the following three directions.

(1) Layered CAM system modeling architecture.

First, rigorously defined hierarchical multiresolution network representation schemes are required to support the optimization and simulation of CAM systems. Particularly, the commonly used spatial-continuous road link representation in low-level vehicle motion models relies on complicated nonlinear functions when describing lane-changing maneuvers and interactions between different vehicles. Second, many studies have been conducted on independent CAM simulation and optimization. However, performing simulation and optimization separately without internally consistent network representations may result in significant gaps between the results from the two modules. Third, the theoretically important aspects of layer decomposition and schedulability have not been completely exploited to recognize the hierarchical and partially schedulable nature of CAM systems, particularly in the presence of computationally intensive coordination tasks. Forth, significant progress has been made in the industry, such as NVIDIA and Qualcomm, to improve the computing capacity of individual cars by providing powerful processors (Zaveria, 2022); there is a critical need to estimate the system-level theoretical and practical computing capacity needs for network-oriented vehicle routing and movement coordination, particularly based on a well-defined MRM network structure. Finally, a viable pathway to an open-source CAV and CAM ecosystem with a standard data interface is fundamentally important, which can significantly facilitate research and cooperation in the transportation community.

(2) Cross-resolution traffic state estimation in CAM systems

Although numerous efforts have been made for different aspects of the TSSI problem, namely, cross-resolution modeling, state representation/smoothing, selection of underlying traffic flow models, utilization of heterogeneous data sources, and handling of partial differential equations in capturing traffic flow dynamics, very few studies have completely integrated all the above elements in a mathematically rigorous and computationally tractable estimation framework. Innovative efforts in this direction of model integration include Treiber and Helbing (2002) and Sun et al. (2017), highlighting the need for a deep examination of the dynamics and uncertainties of systems when a full set of coupled elements is incorporated into the model.

(3) Integrated city logistics operation optimization in CAM systems

The following modeling challenges are observed in the related vehicle routing problem (VRP) and arc routing problem (ARP) literature. First, most studies treat travel speeds on roads as constant, while the time-varying feature of transportation networks is largely simplified. A realistic, parsimonious and mathematically rigorous model with a calibration workflow for time-dependent travel time is important for VRP deployment and applications. Second, the system-wide impact of service vehicles in city logistics to the entire transportation system (with other travelers and road users) has not been systematically studied. Third, in most studies, a single model was developed for real-life rich ARP (RARP) applications, which lacks a comprehensive investigation and comparison on the effects of rich constraints. Finally, recognizing of the high complexity of RARPs, most studies developed heuristics for solving real-life problems, while exact

approaches are critically needed to offer theoretical benchmarks for quantifying the solution quality and the degree of optimality.

1.3 Research Overview

This research focuses on the modeling of emerging CAM from a layered perspective, with a special interest in the designing of inherently consistency cross-resolution system architecture and methodologies of system state estimation and operation optimization. The three main research thrusts of this dissertation are detailed below.

Research thrust 1: Layered CAM system modeling architecture

This thrust introduces a new virtual track-based framework and open-source tools for modeling partially schedulable CAM systems on layered networks. First, a coupled network representation is developed for macroscopic, mesoscopic, and microscopic CAM system modeling with tight inherent consistencies. This enables the behaviorally sound modeling of demand-supply interactions in hierarchical CAM systems from a layer decomposition perspective such that different levels of tasks can be performed in proper layers to achieve a balance between representation details and computational efficiency. A spatial-discrete virtual track-based microscopic network representation is designed for both high-fidelity vehicle dynamics modeling and maintaining consistency with high-level routing decisions in CAM applications to enable individualized active traffic management. Second, based on the proposed layered network structure, this research examines effective methods of traffic simulation, optimization, and operation of CAM systems, with a special focus on different degrees of system schedulability. Third, two open-source packages,

osm2gmns and CAMLite, are introduced to support open-source ecosystems and the research community for CAM system modeling. Representative numerical experiments are performed to demonstrate the effectiveness of the proposed methodologies and open-source tools.

Research thrust 2: Cross-resolution traffic state estimation in CAM systems

This thrust presents an integrated cross-resolution framework for the traffic system state identification (TSSI) problem by simultaneously considering traffic state estimation (TSE), traffic flow model parameter estimation (MPE), and queue profile estimation (QPE) on transportation networks using heterogeneous data sources. Systematically considering the three tasks, that is, TSE, MPE, and QPE, in an integrated modeling framework helps to fully utilize information from different components and takes advantage of larger solution spaces, which is expected to improve the reliability and accuracy of system identification results. However, potential inconsistencies between different modeling components are introduced at the same time and should be carefully dealt with to ensure model feasibility. To minimize such inconsistencies, a novel nonlinear programming model is developed to formulate the TSSI problem by considering traffic flow models and observations from different resolutions. At the macroscopic level, this research uses a fluid queue approximation to model the traffic system of interest. Based on the assumption of polynomial arrival and departure rates, critical system measures such as time-dependent delay, travel time, and queue length are analytically derived. At the mesoscopic level, with the adoption of continuous space-time distribution (CSTD) functions, a continuous traffic state representation scheme is introduced to model traffic flow variables such as traffic

volume, speed, and density. CSTD functions maintain the differentiability of traffic state variables such that partial differential equations in traffic flow models can be comprehensively considered in the proposed framework. A computational graph is constructed to represent the nonlinear programming model in a sequential propagation structure, which is then solved using a forward-backward method. Extensive numerical experiments based on real-world and hypothetical datasets are designed to demonstrate the effectiveness of the proposed framework.

Research thrust 3: Integrated city logistics operation optimization in CAM systems

City logistics, as an essential component of the city operation system, aims at managing the complex flow of goods and services from providers to customers efficiently. Delays associated with peak-period traffic congestion exists in both large and small metropolitan areas. As many of the service tasks in city logistics are needed to be performed during peak hours, operators of urban management movement should consider reducing the total trip time and delay when designing service plans. Equally important, the congestion impact of service vehicles to other road users should also be considered. This research focuses on formulating and solving RARPs in city logistics with a congested urban environment. This work highlights the needs of embedding a structurally parsimonious time-dependent travel time model in RARP for producing high-quality and practically useful solutions. A fluid queue model based analytical approach is presented for link travel time calibration in the form of polynomial arrival rate functions. Accordingly, system-wide (societal) impact of vehicles routing is analytically derived and incorporated into the RARP models which enables traffic managers to systematically consider operation costs and

societal impacts when designing routing policies in real-life city logistics applications. Additionally, this research develops two new representation schemes for time-dependent travel time modeling in RARPs, including a discretized time-expanded representation scheme and a nonlinear polynomial representation scheme. Three modeling approaches for RARPs are proposed, with different perspectives of capturing time-dependent travel time and formulating problem-specific constraints. With a real-life sprinkler truck routing problem as the representative example of RARP, two efficient exact solution algorithms, including a Lagrangian relaxation-based method and a branch-and-price based method, are developed. The latter one is embedded with an enhanced parallel branch-and-bound algorithm. Extensive numerical experiments are conducted based on real-world networks and traffic flow data to demonstrate the effectiveness of the proposed methods.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 provides a comprehensive literature review of CAM system modeling architecture, system state estimation, and system operation optimization. Chapter 3 develops new methodologies and open-source tools for modeling partially schedulable CAM systems based on an innovative construct of layered virtual track networks, with a special focus on the following aspects with the long-term goal of a fully integrated simulation and optimization modeling paradigm. Chapter 4 provides an integrated modeling framework to systematically consider different aspects of TSSI in a unified mathematical programming structure. Specifically, the proposed modeling framework simultaneously performs the critical tasks of TSE, MPE, and QPE using multi-source data based on a continuous space-time

distribution function-based traffic state representation scheme. Chapter 5 focuses on formulating and solving rich arc routing problems (RARPs) in city logistics, with highlighting the needs of embedding a structurally parsimonious time-dependent travel time model in RARP for producing high-quality and practically useful solutions. Chapter 6 concludes this dissertation with a discussion on future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Layered CAM System Modeling Architecture

2.1.1 Review on CAM Modeling Methodologies

The existing studies on CAM systems are divided and reviewed in two categories. Studies in the first category mainly focus on the investigation of lower vehicle- or platoon-level motion control to improve the stability and efficiency of a certain group of vehicles (typically lane-level applications). In the second category, studies are more concerned with higher-level vehicle coordination to improve the performance of the traffic system of interest (typically an intersection or along a corridor).

An early study of the first category for vehicle platooning on automated highways was conducted by Alvarez and Horowitz (1999). Their study focused on a single-lane scenario and designed a safe zone between two platoons considering the distance, relative speed, and maximum acceleration and deceleration rates. In recent years, studies on the connected and autonomous vehicles (CAVs) have become an emerging topic in both academia and industry owing to the fast development of advanced sensing and communication techniques. Various aspects on CAVs, including evaluating the effects on the current traffic system with human driven vehicles (Zhou and Zhu, 2021; Ma et al., 2021; Sala and Soriguera, 2021; Sun et al., 2022), efficient platoon formation strategies (Mahbub and Malikopoulos, 2021; Woo and Skabardonis, 2021; Wu et al., 2021; Wang et al., 2020; Mu et al., 2021; Bang and Ahn, 2017), platoon stability and controllability analysis (Gong et al., 2019; Ma et al., 2019; Zhou et al., 2020; Zhou et al., 2022), distributed control algorithms (Guo et al., 2020; Zhang et al., 2022; Shen et al., 2022), mixed traffic

flow modeling (Gong and Du, 2018; Mahbub and Malikopoulos, 2021; Woo and Skabardonis, 2021; Zhong et al., 2020; Sala and Soriguera, 2021; Lai et al., 2020; Feng et al., 2021), adaptive control under dynamic environments with uncertainties (Gong et al., 2019; Guo et al., 2020; Chen et al., 2018; Wang et al., 2020; Wang et al., 2022; Amirgholy et al., 2020; Ruan et al., 2022; Wang et al., 2020; Xiong et al., 2021; Ma et al., 2022; Wei et al., 2017), and multi-objective optimization (Han et al., 2020; Ma et al., 2019; Ma et al., 2021; Wang et al., 2021) have been extensively investigated. These studies primarily focus on detailed longitudinal control strategies of a certain group of vehicles, and the interest is to analyze or improve the group-level performance.

Studies in the second category aim to improve the overall traffic system performance by coordinating vehicles within a target area. These studies focus on interactions between vehicles from different lanes or even road links, whereas detailed vehicle motion dynamics modeling is usually simplified to reduce the modeling complexity. At present, CAV trajectory optimization primarily focuses on single isolated intersection applications (Yao and Li, 2021; Mohebifard and Hajbabaie, 2021; Mirheli et al., 2019; Ma et al., 2021; Ma et al., 2017; Yao and Li, 2020; Chen et al., 2021). There is increasing interest in integrated models for joint traffic signal and CAV trajectory optimization to further improve traffic system efficiency within intersections (Tajalli and Hajbabaie, 2021; Guo et al., 2019; Niroumand et al., 2020; Soleimaniamiri et al., 2020; Li and Zhou, 2017; Yu et al., 2018). Simultaneously, improving road segment throughput and merging area capacity by properly coordinating CAV trajectories on freeways has garnered substantial attention (Li et al., 2018; Yang et al., 2020; Hu and Sun, 2019; Amini et al., 2021; Sun et al., 2020).

One of the critical challenges in CAM modeling is developing theoretically rigorous, computationally tractable, and behaviorally sound models for describing interactions between vehicles from adjacent lanes, while maintaining consistency with complex route choice and link selections in a network setting. Without an integrated lane-changing and path planning model, simulation or optimization results may either generate unreasonable lane-switching vehicle trajectories or rely on over-simplistic modeling assumptions; for instance, vehicles are not allowed to change lanes in control areas. This research attempts to properly model the lane-changing maneuvers of CAVs without introducing complicated nonlinear functions.

2.1.2 Review on Problem Decomposition and Schedulability in Complex Systems

CAM systems have been recognized as complex cyber-physical systems owing to their high dynamics, stochasticity, and large number of interconnected components and decision-making agents. Across various disciplines, the key performance indices of such a complex system include scalability, reliability, and controllability. As a representative example, the communication system has been extensively studied for its system performance improvement using layered decomposition and coordination. Some examples are problem decomposition methods for network utility maximization (Palomar and Chiang, 2016), layering as optimization decomposition (Chiang et al., 2007), and cross-layer congestion, routing, and scheduling design (Chen et al., 2006). In comparison, a related MRM methodology in the transportation domain is primarily applied in simulation and control applications (Nava et al., 2012; Shelton et al., 2019; Massahi et al., 2019; Hadi et al., 2016; Rajaram et al., 2016; Li et al., 2015a; Xing et al., 2021; Gavriilidou et al.,

2019; Van Lint and Calvert, 2018). Hadi et al. (2022) and Zhou et al. (2021) offer comprehensive summaries of advanced MRM developments in the transportation AMS domain. In the transportation field, guaranteeing mathematical modeling consistency and designing theoretically sound feedback mechanisms from the perspective of system optimization needs to be profoundly studied.

The potential benefits of adopting reservation or scheduling have been investigated for various transportation systems, including freeways (Koolstra, 1999), arterial intersections (Xie et al., 2012), bike sharing (Chiariotti et al., 2018), taxis (Wang and Cheu, 2013), buses (Tong et al., 2017), and parking systems (Liu et al., 2014). In recent years, the advancement of sensing and communication techniques has enabled the incorporation of schedule elements as an integrated part of evolving CAM systems. Similar to finding optimal schedules in railway and aircraft systems, many studies have been conducted to optimize the operation of CAM systems from various perspectives, such as route guidance (Lu et al., 2016), vehicle trajectory control (Han et al., 2020; Karimi et al., 2020; Wei et al., 2017), and signal timing (Li et al., 2015b; Mohebifard and Hajbabaie, 2019). Most existing studies focus on the optimal control of fully schedulable CAM systems or real-time operations under fully automated or mixed traffic conditions, whereas robust two-stage control or real-time re-scheduling of partially schedulable systems has not been sufficiently examined.

2.1.2 Review on Open-source Tools for CAM Modeling

The development of open-source tools is essential in the CAM and MaaS communities. Open-source tools provide free access to the broad public and allow capable end users to adapt tools to their customized modeling needs.

As shown in Fig. 1, CAM modeling tools are classified into three different classes. Simulation tools in the first class target either the vehicle or aggregated flow level travel demand modeling using well-defined traffic flow models depending on the modeling fidelity. The modeling objective is to simulate or reproduce real-life traffic flow evolution to provide evaluation and decision support for effective traffic management and control strategies. The related open-source tools, to name a few, include microscopic simulator SUMO (Behrisch et al., 2011), activity based modeling framework MATSim (Horni et al., 2016), and mesoscopic dynamic traffic assignment package DTALite (Zhou and Taylor, 2014). In particular, SUMO is a highly portable, microscopic, and space-continuous multimodal traffic simulation tool, which allows the modeling of traveling agents including road vehicles, public transport, and pedestrians, and provides various APIs to enhance the ability of modeling customization. MATSim is an activity-based, extendable, multi-agent simulation framework implemented in Java. A queue-based demand-loading scheme was implemented in a parallel computing fashion to enable large-scale scenario modeling. DTALite is a lightweight dynamic network loading simulator that embeds Newell's simplified kinematic wave model (Newell, 1993).

Existing optimization tools in the first class primarily focus on optimizing vehicle routings with certain service requests to minimize total system cost, which is typically modeled as vehicle routing problems (VRPs). Open-source tools in this category include OR-Tools (<https://github.com/google/or-tools>), VROOM (<https://github.com/VROOM->

Project/vroom), jsprit (<https://github.com/graphhopper/jsprit>), and VRPLite (Zhou et al., 2018). Among the four tools, the first three tools adopt constructive heuristics, whereas VRPLite aims to find optimized space-time vehicle service/traveling paths in a Lagrangian relaxation framework.

These software tools provide specific modeling functionalities in one domain. Systematic design is still required for a fully integrated mathematical optimization, scheduling, and simulation framework, particularly based on a commonly defined shared modeling network system at different scales. OpenStreetMap, as an open map website, provides free map content to the public; however, conversion tools are still critically needed to address three challenges (Ory, 2020): (1) the original data in OpenStreetMap are not directly routable in a standard transportation planning model, (2) structure of road link attributes is not completely aligned with travel planning and multimodal simulation models, and (3) user-contributed attribute data are not sufficiently complete for high-fidelity simulation at the lane or meter-by-meter cell levels. To address the first challenge, open-source package OSMnx developed by Boeing (2017) offers rich graph analysis functionalities in modeling, projecting, visualizing, and analyzing real-world street networks from OpenStreetMap. In the context of CAM fields, open-source network preparation tools using unified network formats are critically needed to improve transparency and reproducibility in research collaboration, which motivates the building of open-source packages osm2gmns and CAMLite.

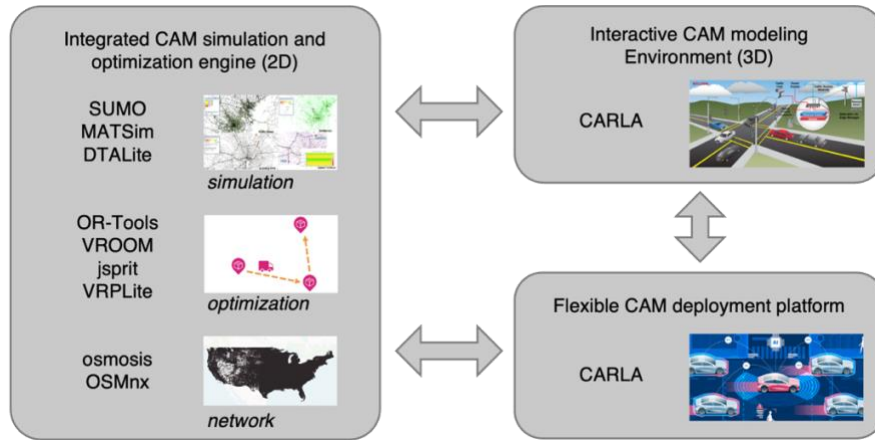


Fig. 1. Three Classes of CAM Modeling Tools.

The development of related open-source tools for use in the second and third classes relies on successful private-public partnerships that can closely connect academia and industry because of the modeling complexities. CARLA (Dosovitskiy et al., 2017) and CARMA (FHWA, 2022) are the two most popular tools in the second and third classes, respectively. CARLA provides open three-dimensional digital assets, flexible specifications of sensor suites, and environmental conditions to support the development, training, and validation of autonomous driving systems. The CARMA program, led by FHWA, aims to provide an open-source software platform with agile development practices to develop and test cooperative driving automation features associated with the infrastructure, properly equipped vehicles, and other road users.

2.2 Cross-resolution Traffic State Estimation in CAM Systems

2.2.1 Literature Review on Traffic State Estimation

Traffic states of interest include vehicle travel speed and trip travel time in early studies (Coifman, 2002; Cheu et al., 2002), while other fundamental state variables such as traffic volume, speed, and density on the two-dimensional space-time plane are more

considered in recent studies. Seo et al. (2017) offered an excellent summary of model-driven, data-driven, and streaming-data-driven methods. Model-driven methods focus on embedding interpretable traffic flow functional forms and considering tight couplings of different traffic states to provide a systematic inference at partially observable or unobservable areas. In comparison, data-driven and streaming data-driven approaches rely heavily on statistical patterns identifiable from historical and real-time streaming data. Recent developments along this line indicate that highly accurate estimations can be provided for locations with sufficient sensor data coverage (Antoniou et al., 2013; Chen et al., 2007; Tao et al., 2012; Bhaskar et al., 2014; van Erp et al., 2018; Ma and Qian, 2021). This study mainly focuses on the enhancement of model-driven methods while fully utilizing advanced computational techniques for emerging data-driven applications. Existing model-driven methods in the literature are reviewed from the following four aspects.

Traffic flow models: continuum flow models

Continuum flow models in the TSE are used to characterize complex system dynamics, particularly congestion formation, propagation, and dissipation. In the most notable first-order LWR model (Lighthill and Whitham, 1955; Richards, 1956), traffic flow dynamics can be described by three basic equations: the fundamental equation $q = kv$, the flow conservation equation $\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = u_{x,t}$, and the speed-density relationship equation, where q , k , and v are the flow, density, and speed, respectively, and $u_{x,t}$ is the net vehicle generation rate. Other higher-order continuum flow models include Payne–Whitham’s model (Payne, 1971; Whitham, 1974), Phillips’s model (Phillips, 1979), and Zhang’s

model (Zhang, 1998). The flow conservation law in continuum flow models is typically represented by a PDE. Recognizing the difficulty in obtaining closed-form solutions for PDEs, many researchers have developed a wide range of finite-difference approximations and finite element methods (Grossmann et al., 2007) to solve these equations numerically. Many existing studies approximate the original PDE using a set of linear equations. For example, Nanthawichit et al. (2003) built a discrete form of the Payne–Whitham model using the finite difference method, and Wang and Papageorgiou (2005) used the space-time discretized form of the second-order validated macroscopic traffic flow model proposed by Papageorgiou et al. (1990) in their TSE model. Work et al. (2008) formulated a Godunov discretization scheme to cast a new LWR-based PDE into a velocity-based cell transmission model. Other efforts involved the use of the cell transmission model (Daganzo, 1994), which is a discretization form of the original LWR model, in traffic state estimators (Sun et al., 2003; Tampère and Immers, 2007). An alternative shock-fitting approach aims to solve PEDs analytically (Wong and Wong, 2002; Sun et al., 2011), and these methods can potentially achieve a higher degree of accuracy. It should be noted that strong simplistic assumptions such as the linear speed-density relationship in Wong and Wong (2002) are required for the shock-fitting approach, and the corresponding applications to largescale instances are computationally intensive.

Multi-source data: mapping heterogeneous measurements to system states

Fusing different types of traffic measurements can improve system-wide observability from different perspectives; however, the theoretical challenge is to systematically establish a set of computationally tractable and inherently consistent

measurement equations. Various studies have been devoted to specific measurement categories, such as GPS and mobile phone-based data (Nanthawichit et al., 2003; Herrera and Bayen, 2010; Duret and Yuan, 2017; Cheng et al., 2006; Work 2010), Bluetooth data (Bhaskar et al., 2014), video detection data (Quiros et al., 2016), and license plate recognition data (Zhan et al., 2020). Recently, the utilization of streaming data from connected vehicles in TSE has received significant attention (Shahrbabaki et al., 2018; Seo et al., 2015; Bekiaris-Liberis et al., 2017; Luo et al., 2019). In addition to traffic flow systems, heterogeneous data sources are also widely used to estimate flow states in other systems. For example, Shang et al. (2019) integrated Lagrangian and Eulerian observations to estimate the passenger flow state in an urban rail transit network. One of the challenging issues is that, owing to detection errors, simply combining different data from various detectors may introduce inconsistencies into TSE models and the resulting state estimations. Doan et al. (1999) offered a comprehensive discussion of error sources in data observations, model structures, sensor data, and historical data.

Estimation model: formulating estimation model as optimization or online filtering problems

The ultimate challenge, once the underlying traffic flow models are selected and multi-source data are readily available, is the construction of an internally consistent and numerically stable estimation model considering the following aspects: (a) measurement equations as the mapping between traffic states of interest with traffic flow models and multi-source data and (b) assumptions about the error structure that lead to different forms of the objective function or methods of recursive state estimate updates. Ashok (1996)

discussed the equivalence and connection between the optimization problem of minimizing nonlinear generalized least squares and Kalman filtering (KF) from a state-space modeling perspective. Along the line of a generalized least square estimation framework, Deng et al. (2013) proposed ways of incorporating an extended stochastic three-detector model (Newell, 1993) to use multiple data sources. Zheng and Su (2016) built a convex optimization problem for the TSE using linearization techniques and solved the problem using the split Bregman iteration method. By adopting the Hamilton-Jacobi equation, Canepa and Claudel (2017) converted the TSE problem into a mixed integer linear programming problem. The challenge to be addressed is how to efficiently produce high-quality results, particularly for medium- or largescale instances.

In the category of filtering approaches, the original TSE problem is modeled as a recursive state learning/updating problem, and the focus is on capturing the nonlinearity in the system dynamics evolution, and various spatial and temporal correlations among the states. Several types of filters have been progressively adapted, such as particle filtering (PF) and mixture KF (Sun et al., 2003). Work et al. (2008) used the ensemble KF technique in their customized velocity cell transmission model to estimate the velocity field on a highway using data obtained from GPS devices. Using cellphone network data, Cheng et al. (2006) proposed two Bayesian framework-based traffic estimation models, both of which were implemented using PF. Using a speed-extended cell transmission model, Mihaylova et al. (2007) developed a PF framework for real-time estimation of traffic states in freeway networks.

Underlying traffic state representation scheme: discretization or continuous function-based scheme

The traffic state representation scheme determines in which form the traffic states of interest are represented. Almost all existing studies in the literature have adopted a discretization-based state representation scheme. That is, with a preset discretization resolution, the space-time regime on which the traffic states need to be estimated is discretized into a finite number of grids. With the assumption that the states on each grid are homogeneous, independent state variables can then be created and estimated on the associated grids. Although the adoption of discretization-based state representation schemes can help simplify the modeling process, issues such as state discontinuity are introduced simultaneously. Another potential impact is that discretization breaks the differentiability of traffic states, preventing traffic theories on flow dynamics represented by partial differential equations (PDEs) from being directly imposed on the TSE models.

There are a few studies in the literature that didn't adopt the discretization-based state representation scheme. Treiber and Helbing (2002) designed a nonlinear spatiotemporal low-pass filter to estimate traffic state variables based on data from stationary detectors. The estimation outputs were the velocity, flow, or other traffic variables as smooth functions of space and time. In a follow-up study by Van Lint and Hoogendoorn (2010), with the relaxation of restrictions such that data were structured in a temporal or spatial manner, the authors proposed an enhanced filter algorithm for traffic state estimations using heterogeneous data from traffic sensors on freeways.

2.2.2 Literature Review on Model Parameter Estimation

MPE refers to the process of estimating and calibrating parameters in traffic flow models using observed traffic data. Various methods have been proposed to solve the MPE problem (e.g., Cremer and Papageorgiou, 1981; Ngoduy and Hoogendoorn, 2003; Brockfeld et al., 2004; Spiliopoulou et al., 2014; Paz et al., 2015; Spiliopoulou et al., 2017; and Seo et al., 2019). It has also been recognized that traffic flow model parameters can be location-dependent and time-varying because they are affected by various factors such as traffic incidents and road or weather conditions. Therefore, directly adopting pre-calibrated traffic flow models in a specific application, especially under conditions that are different from those where the parameters are calibrated, may be problematic.

Several studies related to online dynamic traffic assignment (DTA) have focused on the adaptive calibration of traffic flow parameters and joint estimation of demand and supply parameters of DTA systems (Qin and Mahmassani, 2004; Antoniou et al., 2007). An excellent application can be found in the study for developing a weather-responsive traffic estimation and prediction system with simulation-based DTA as the core state estimator (Hou et al., 2013; Mahmassani et al., 2014). In the area of TSE, applying pre-calibrated traffic flow models in TSE is still a common practice in the literature, while the joint estimation of traffic states and traffic flow models has not yet received sufficient attention. Evidently, the joint estimation process can take advantage of a larger solution space (with both the variables of traffic states and traffic flow model parameters) to better approximate the ground truth. If reliable inferences are obtained, one can better interpret the dynamics of traffic states owing to changes in demand flow patterns or variations in supply parameters. Significant research efforts along this line in TSE include the works of

Wang and Papageorgiou (2002), Tampère and Immers (2007), Sun et al. (2017), Shi et al. (2021), and Wang et al. (2022).

2.2.3 Literature Review on Queue Profile Estimation

Limiting the maximum queue length and avoiding queue spill back are the most important tasks of signal control at oversaturated intersections during peak hours; therefore, QPE has always been a recurrent topic in urban traffic management. Various methods have been developed for QPE using different types of data sources. To name a few, Liu et al., 2009; Ban et al., 2011; Comert and Cetin, 2011; Comert, 2013; Lee et al., 2015; Tiaprasert et al., 2015; Ramezani and Geroliminis, 2015; and Zhao et al., 2019. For freeways, researchers have mainly focused on the delay and queue caused by work zones and traffic incidents (e.g., Chien et al., 2002; Jiang and Adeli, 2004; Ghosh-Dastidar and Adeli, 2006; and Li et al., 2006). In terms of queue modeling on freeway bottlenecks, Cao et al. (2015) developed a time-space discrete macroscopic model based on the shockwave theory for real-time queue estimation in uninterrupted freeway flow. Based on fluid queue approximation, a recent study by Cheng et al. (2022) analytically derived system measures, such as time-dependent delay and queue length, with the assumption of polynomial flow rates in a queuing system.

Table 1 compares related studies with this study in terms of the tasks considered, modeling approach, and solution method.

Table 1 Comparison of Related Studies on TSE, MPE, and QPE.

Publication	Tasks	Modeling approach	Solution method
Sun et al. (2017)	TSE, MPE	Nonlinear optimization	Closed-form formula, Gauss-Newton method.
Shi et al. (2021)	TSE, MPE	Nonlinear optimization	Gradient descent method
Wang et al. (2016)	TSE	State-space model	Particle filtering
Canepa and Claudel (2017)	TSE	Mixed integer linear programming	Mathematical programming solver
Liu et al. (2009)	QPE	Lighthill–Whitham–Richards shockwave theory	Numerical derivations
Duret and Yuan (2017)	TSE	Lighthill–Whitham–Richards model in Lagrangian space	Numerical solutions obtained with Godunov scheme
Jabari and Liu (2013)	TSE	State-space model	Kalman filtering
Zheng et al. (2018)	TSE	State-space model	Kalman filtering
Seo et al. (2019)	MPE	Lighthill–Whitham–Richards model	Filtering method, expectation maximization algorithm
This research	TSE, MPE, QPE	Nonlinear optimization	Gradient descent method

2.3 Integrated City Logistics Operation Optimization in CAM Systems

In this section, related existing studies are reviewed along two lines: (1) time-dependent travel time modeling in vehicle routing and arc routing problems and (2) rich arc routing problems

2.3.1 Time-dependent Travel Time Modeling in Vehicle Routing and Arc Routing Problems

The important study by Malandraki (1989) first formulated link travel time as a piecewise-constant function of the departure time. Later on, this approach was adopted by Malandraki and Daskin (1992) and Chen et al, (2006). Yet, this approach might not satisfy the First-In-First-Out (FIFO) property. The FIFO property states that, among two identical vehicles that travel along the same path, the one that departs earlier from the origin node

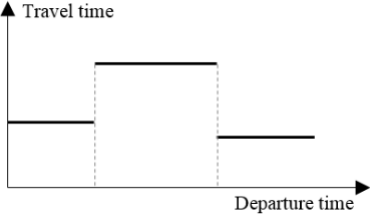
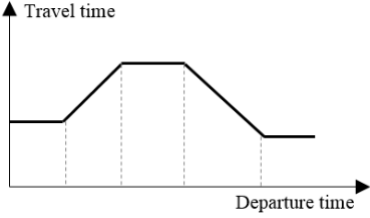
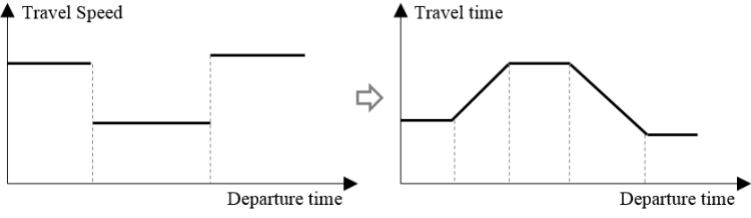
could always arrive at the destination node earlier than the other vehicle. Two major approaches are proposed to address the potential FIFO violation issue in the piecewise constant representation. First, piecewise-linear functions are considered by Ahn and Shin (1991) and improved by Fleischmann et al., (2004). In this approach, a linear transition function is built to connect travel times in two adjacent time periods such that travel time changes slowly and smoothly. It can be proved that the FIFO property holds if the slope of linear lines is less than 45° . Another method proposed by Ichoua et al., (2003) starts with time-dependent link travel speed in the form of a piecewise-constant function and derives/computes piecewise-linear travel times satisfying the FIFO property. Table 2 summarizes the above three major methods for time-dependent travel time modeling in VRP and ARP models. Xiao and Konak (2016) offers a more detailed classification and illustration.

Figliozzi (2012) offered a comprehensive set of benchmarks for modeling time-dependent travel times in VRP. The important modeling aspect of time discretization is further studied by Boland et al. (2017) for a broader class of continuous-time shortest path and service network design problems. One can find the related studies in Scherr et al. (2020), Belieres et al. (2021), He et al. (2021), Marshall et al. (2021), Vu et al. (2022), Hewitt (2022), and Lagos et al. (2022). In particular, for the fundamentally important time-dependent shortest path problem, the study by He et al. (2021) systematically considers degree 4 and degree 6 polynomial functions to approximate time-dependent travel time through the piecewise linear interpolants sampled at integer points. The polynomial function can be calibrated for an extended time period (e.g., 24 hours of a day) using real world travel time data, while the related non-linear functional form could have many local

minima and maxima, which could greatly affect the computational efficiency of dynamic discretization discovery algorithms (see Boland et al., 2017).

In fact, how to obtain reliable and accurate travel time estimates from field data has remained a challenging problem for the VRP field, while many published papers still use randomly generated or hypothetical travel time distributions for simplicity. By utilizing data from advanced traffic information systems in Berlin, Fleischmann et al. (2004) proposed a smoothing method to ensure the resulting travel time functions satisfy the FIFO property. By using traffic flow data collected from a Belgian highway, Jabali et al. (2009) created a speed profile with five periods, including two periods for morning and afternoon peak hours and three periods for the remaining non-peak hours, for each link within the research area. Kritzinger et al. (2012) adopted 15-minute link travel time information from floating car data and developed an extended version of Dijkstra's algorithm to compute the distance matrices between points with various departure times. In the paper by Gmira et al. (2021), the authors developed a discrete-event simulator that generates travel speed updates for real-time vehicle routing applications. Different from the aforementioned numerically driven approximation approaches, Van Woensel et al. (2007) first proposed an innovative queue-theoretic modeling scheme for calibrating expected travel times which was adopted in the studies by Van Woensel et al. (2008) and Lecluyse et al. (2009).

Table 2 Three Major Methods for Representing Time-dependent Travel Times in VRP/ARP.

Graphical illustration of methods	Description
	<p>Method type: piecewise-constant travel time function</p> <p>Modeling details: each link has a constant travel time within each pre-defined time period</p> <p>FIFO property: not necessarily satisfied</p>
	<p>Method type: piecewise-linear travel time function</p> <p>Modeling details: each link has a constant travel time within each pre-defined time period; a linear transition line is built between two adjacent periods</p> <p>FIFO property: satisfied</p>
	<p>Method type: piecewise-linear travel time function</p> <p>Modeling details: a piecewise-constant travel speed function is constructed for each link first; piecewise-linear travel time function is then derived based on its corresponding speed function</p> <p>FIFO property: satisfied</p>

2.3.2 Rich Arc Routing Problems

Considerable efforts have been devoted to the ARP and its variants. Interested readers are referred to a number of survey papers (Wøhlk, 2008; Corberán and Prins, 2010; Corberán and Laporte, 2015; Mourão and Pinto, 2017; Corberán et al., 2021). In this subsection, with the focus on mathematical modeling and solution method development, existing studies are reviewed along two research lines, including real-life applications of RARP on urban networks and ARP with time-dependent travel times.

Three representative applications of RARP as the urban management movement problem

Sprinkler truck routing problem: Li et al. (2008) investigated the water truck routing problem in open pit mines in which the travel time along a road for each truck is not fixed but relies on its leading truck on the same road. The authors proposed minimum cost flow and set-partitioning based heuristics solution algorithms. Huang and Lin (2014) formulated the street tree watering problem as the periodic arc routing problem with refill points where the watering frequency of each street tree may not be fixed and needs to be scheduled according to the period of watering activity. A graph transformation strategy was first adopted to convert the original problem to a VRP with an ant colony heuristic algorithm. Riquelme-Rodríguez et al. (2014) introduced periodic capacitated arc routing problem with inventory constraints that models the loss of humidity on each road with an inventory consumption function. The quantity of water delivered was considered to be fixed or variable. Two mathematical optimization models were proposed and solved by the commercial solver CPLEX.

Street sweeping problem: Street sweeping requires a special vehicle equipped with a rotating brush that can move along the roadside and sweep material into a container on the vehicle. Early studies (Bodin and Kursh, 1978; Eglese and Murdock, 1991) modeled the street sweeping problem as the capacitated Chinese postman problem and developed heuristics to find reasonable routes for the road-sweeping vehicles. Blazquez et al. (2012) considered two extra “rich” constraints in sweeper route design: the sweepers must visit each selected street exactly as many times as its number of street sides; certain types of

turns are not allowed to use. The original ARP was transformed into a VRP which was further solved by a nearest neighbor heuristic algorithm.

Waste collection problem: Mourão and Almeida (2000) and Mourão and Amado (2005) examined the waste collection problem on randomly generated networks and proposed lower bounds and a three-phase heuristic that transforms one of the lower bound solutions to a feasible one. Maniezzo, 2004 considered additional bin compatibility, forbidden turns and one-way street constraints in the model, which is solved by a local search-based heuristic. Ghiani et al. (2005) considered practical constraints in waste collection, e.g., large vehicles are not allowed to use some narrow streets, or services at some sites have to be scheduled at night to avoid traffic congestion. The problem was solved by a cluster-first route-second based heuristic. Many other constraints include traffic regulations (Bautista et al., 2008), mobile depots (Del Pia and Filippi, 2006), intermediate facilities (Ghiani et al., 2001), and trip length restrictions (Ghiani et al., 2010). Recently, Willemse and Joubert (2016a) integrated major problem features of early studies and studied the mixed capacitated arc routing problem with time window and intermediate facilities (MCARPTWIF). Four constructive heuristics were developed and comprehensively evaluated. Given splitting procedures play a key role in giant tour-based heuristics and meta-heuristics, the authors further proposed optimal and heuristic splitting procedures for the MCARPTWIF (Willemse and Joubert, 2016b). In a follow-up study (Willemse and Joubert, 2019), three acceleration mechanisms for local search meta-heuristics were developed to better cope with large-scale instances.

In general, there are three categories of approaches for solving RARPs, including constructive heuristics, meta-heuristics, and exact approach (Corberán and Laporte, 2015).

Constructive heuristics are designed based on problem features and typically provide a single final solution, while meta-heuristics are more general and can be applied to any problems. Both approaches cannot produce measurable quality of solutions. Exact optimization approaches and related model reformulations have not been studied extensively for RAPPs, especially with the consideration of time-varying traffic conditions.

ARP with time-dependent travel times

Several VRP-related studies highlight the significant impact of time-dependent travel times on vehicle routing and scheduling. To name a few, Malandraki and Daskin, 1992; Haghani and Jung, 2005; Chen et al., 2006; Donati et al., 2008; Figliozzi, 2012; Dabia et al., 2013; Spliet et al., 2018; and Sun et al., 2018. Specifically, instead of using simplified customer-based graphs, some researchers directly solve VRPs on graphs that are similar to original road networks to incorporate detailed road-network information in the modeling process (Ben Ticha et al., 2021; Huang et al., 2017; Ben Ticha et al., 2019). Ben Ticha et al. (2018) offered a comprehensive review on VRP studies using road-network information.

On the other hand, research on ARP with time-dependent travel times is still limited in the literature. Vidal et al. (2021) recently offered an extensive study for the time-dependent capacitated arc routing problem (TDCARP). Based on the piecewise-constant speed function proposed by Ichoua et al., (2003), the authors derived a closed-form representation for link arrival time functions and developed a continuous preprocessing approach for point-to-point quickest path query. A branch-cut-and-price exact algorithm and a hybrid genetic search-based metaheuristic were proposed for solving the TDCARP.

CHAPTER 3

LAYERED CAM SYSTEM MODELING ARCHITECTURE

3.1 Introduction

One of the most attractive features of CAM systems is the possibility of coordinating the activity schedule of participants in the system to a certain degree; adjusted participant schedules, in terms of departure time shift, route change, mode switching, or trip canceling, can lead to a win-win situation, in which a desirable system-level traffic performance is maintained, and essential mobility needs of users are fulfilled. This research particularly focuses on the theoretical aspect of “schedulability” in CAM systems. The actions included in a schedule may vary from travel mode choice to low-level car-following and lane-changing maneuvers depending on the modeling resolution. According to specific traffic operation targets, the general utility of a schedule may be defined as a combination of travel reliability, travel time, and environmental effects. Different degrees of connectivity and automation lead to different levels of “schedulability”.

In a schedulable transportation system, demand (trip requests) and supply (road resources) can be known in advance and various sophisticated scheduling methods are applied to seek optimal operation strategies for the system. As a typical example of a fully schedulable transportation system, rail management first determines the draft or planning timetables for different types of trains, and then dispatches online tasks. Owing to the existence of unexpected scenarios, that is, weather conditions, there is also a range of re-scheduling measures, such as re-time, re-order, re-track, and re-route, departure time changes, and trip canceling. Compared to predominately centrally managed and controlled railway systems, CAM systems in complex urban settings involve collective driving and

trip-making decisions, in which different types of road resource users, such as automobiles, transits, bikes, and pedestrians, have their own travel/routing objectives. Moreover, urban CAM systems are “partially schedulable” in the sense that “planned” schedules must be constantly adjusted to consider a wide range of stochastic and dynamic factors unfolding, ranging from microscopic lane changing disturbances to significant pattern changes (e.g., weather conditions and traffic work zones) on either demand or supply side. The use of available CAM technologies to enable the incorporation of a two-stage scheduling process, offline optimization, and online re-scheduling is critically important to study for future mobility system operations. In particular, guaranteeing the punctuality and reliability of CAM system operations calls for multiresolution approaches, which can simultaneously model detailed routes and trajectories of each type of infrastructure user, while characterizing high-level inter-correlated demand-supply interactions.

Based on a novel layered virtual track network representation, this chapter introduces new modeling methodologies and open-source computational tools for enabling the evaluation of partially schedulable CAM systems with different degrees of system schedulability and types of problem decomposition strategies. The main contribution of this chapter includes:

- (1) Inherently consistent macroscopic, mesoscopic, and microscopic layers of network structure is designed to accommodate the modeling needs of CAM systems with different resolutions. This framework can better capture hierarchical connections from travel to traffic modeling and represent different decisions at strategic and tactical levels of driving with different environmental inputs. Each modeling task can be

performed on a dedicated network layer to improve the overall system efficiency and robustness. In particular, this research introduces a virtual-track-based spatial-discrete low-level network representation to jointly model detailed vehicle routing and platoon dynamics, with which complex vehicle lane-changing maneuvers and interactions between different vehicles can be well captured.

- (2) This research defines space and time discretization schemes to construct an integrated traffic simulation and optimization platform for CAM modeling. Within this modeling scheme, this research can further adopt and incorporate different layer decomposition principles for CAM modeling, particularly from the field of electrical engineering. With a special focus on task decomposition and feedback mechanism design, layer decomposition methodologies are proposed to effectively manage and optimize large-scale hierarchical CAM systems.
- (3) In addition, recognizing the partially schedulable nature of CAM systems and new research needs for individualized active traffic management, this research introduces distributed re-scheduling methodologies to regulate online system operations such that the difference between actual system states and optimal offline schedules can be minimized. This research attempts to link methods in the existing studies and computational methods for rail and public transportation scheduling to emerging CAM applications.
- (4) Accurate and easily accessible transportation networks are the foundation of multimodal transportation demand-supply modeling. This research develops an open-source tool with standard data interfaces, `osm2gmns`, to help the CAM community easily obtain and build inherently consistent networks across different scales and enable

wide adoption of MRM methodologies. In addition, an open-source prototype for integrated CAM simulation and optimization, CAMLite, is developed to facilitate future CAM research.

3.2 Virtual-track-based Multiresolution Networks and Associated Modeling Focuses

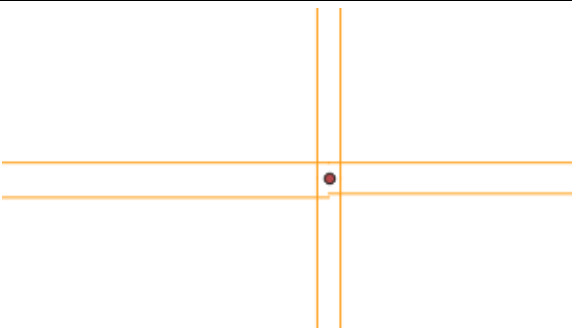
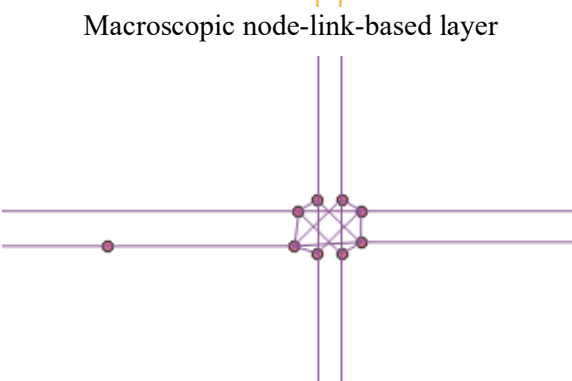
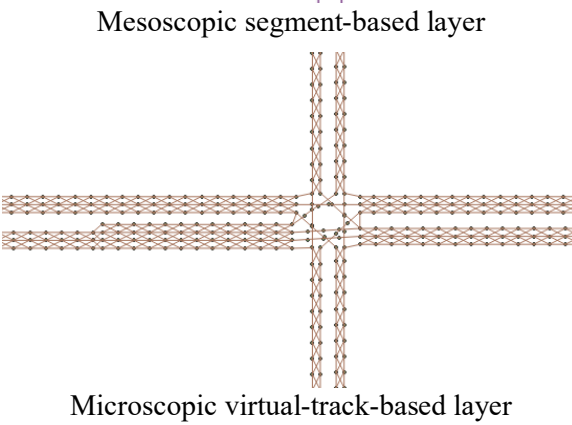
3.2.1 The Multiresolution Network Representation

This subsection introduces a consistent multiresolution network structure, including macroscopic, mesoscopic, and microscopic layers, with different modeling focuses, as listed in Table 3. The brief description below serves as a foundation for the following optimization and simulation sections.

In the macroscopic layer, each arterial intersection (or freeway merge/diverge point) and the road between two adjacent intersections are modeled as a node and directed link, respectively. In CAM systems, vehicles pick up passengers from their origins and drop them off at their destination. Traffic managers and engineers can better characterize and manage the use of limited transportation infrastructure and highway capacity and allow a wide range of real-time on-demand trip scheduling and dispatching algorithms to assign service vehicles to trip requests with specific origins, destinations, and time windows. Many studies on VRP (Psaraftis et al., 2016; Savelsbergh and Van Woensel, 2016; Hyland and Mahmassani, 2018; Ulmer et al., 2019; Liu et al., 2020) address the complexity of managing and dispatching large fleets of vehicles and platoons in real time. Each vehicle communicates with traffic information providers to receive up-to-date network traffic conditions and to share traffic data where the vehicle is traversing. The scheduling

algorithm assigns multiple trip requests to a shared vehicle. Using coordinated routing information, traffic management centers can better utilize or price limited road resources.

Table 3 Multiresolution Network Representation.

Network layer	Layer attributes
 <p data-bbox="332 745 747 777">Macroscopic node-link-based layer</p>	<ul style="list-style-type: none"> • Layer components: node and link • Intersections and roads between two adjacent intersections are modeled by nodes and links, respectively. • Network topology information for high-level demand managements.
 <p data-bbox="341 1123 738 1155">Mesoscopic segment-based layer</p>	<ul style="list-style-type: none"> • Layer components: segment • Intersections are expanded using movement segments; road links are split into several segments to ensure each segment is homogenous in terms of number of lanes, free flow speed, etc. • Network resource information for medium-level traffic operation.
 <p data-bbox="316 1501 763 1533">Microscopic virtual-track-based layer</p>	<ul style="list-style-type: none"> • Layer components: virtual track • Lane-by-lane virtual tracks, including traveling and lane-changing cells, are constructed. • Lane-specific information for high-fidelity vehicle motion planning and platooning.

In the mesoscopic layer, more network details, such as movement at intersections and lane number changes on roads, are included. Each intersection node in the macroscopic layer is expanded using movement segments such that movement-associated turning lanes, signal timings, and capacity/discharge rates can be exactly modeled. In addition, in this

layer, a macroscopic link with different attributes, such as number of lanes and free-flow speed, is split into multiple mesoscopic homogenous segments. Proactive traffic management strategies, such as intersection and bottleneck control, can be implemented in this layer to take advantage of the finer segment-based representation of transportation networks. Typical traffic assignment modules, such as DYNASMART (Mahmassani, 1992) and Dynameq (Mahut and Florian, 2010), can be applied to this layer to evaluate the benefits of active signal control in congested areas.

In the microscopic layer, to cope with the complexity of scheduling and managing vehicles on physical roads, lane-by-lane virtual tracks are constructed to jointly model detailed vehicle routing and platoon dynamics. This can be considered as a network-based cellular automata (CA) modeling scheme, which was proposed by Von Neumann (1951) and popularized by Wolfram (1983). In the work by Daganzo (2006), the equivalence between a simplified kinematic wave model and parsimonious car following model CA(M) is demonstrated. In the standard CA model of traffic flow (Nagel and Schreckenberg, 1992), a vehicle typically moves several cells in one time interval. In comparison, the CA(M) model is intended to describe vehicle motion involving only one cell at a time; thus, the boundary conditions (or potential conflict points) for merges, diverges, and lane changes can be seamlessly integrated together with high-level routing decisions. Related research surveys and developments can be found in Zheng (2014) and Laval and Daganzo (2006). In this study, without loss of generality, virtual track lanes are discretized into traveling and lane-changing cells to support the integrated network flow-based optimization for following and lane-changing maneuvers using a unified graph structure. The length of cells can be flexibly adjusted in each specific application to achieve a balance

between modeling accuracy and scalability; a thorough discussion was provided by Daganzo (2006) at different degrees of shock wave estimation accuracy.

3.2.2 Illustration of Virtual-track Representation for Vehicle Motion Planning and Overall Modeling Framework

In the microscopic layer, each lane of a freeway or urban street corresponds to a car-following track on which multiple vehicles can be coupled through virtual vehicle-vehicle and vehicle-infrastructure protocols to form a platoon. In the proposed MRM network, vehicles change lanes at dedicated specific locations to manage the complexity of lane-changing maneuvers such that a sequence of actions, such as gap acceptance, acceleration, merging, and position re-adjustment, can be better communicated between vehicles.

The introduction of cyber tracks allows the adaptation of modern automatic block signaling technologies (de Rivera and Dick, 2021 and Meng and Zhou, 2014), which are widely used in railway systems to control the movement of trains between blocks. In road cyber-track systems, vehicle lanes are considered interconnected tracks composed of multiple cells. Instead of continuously calculating the safe headways and gaps between vehicles, traveling safety is guaranteed by preventing simultaneous entrance of multiple vehicles to the same cell using a high-precision timetabling scheme (e.g., at a resolution of 0.1 s and 1 m). Traveling mobility is guaranteed by actively guiding vehicles and actively forming or breaking platoons on different tracks.

Fig. 2 presents the framework of the layered CAM system modeling on the proposed multiresolution network. A three-layer network structure is built as the

foundation of the hierarchical decomposition of CAM systems. Thereafter, different levels of tasks are cast into corresponding layers to seek a balance between modeling efficiency and fidelity in real-life deployments. Integrated simulation and optimization methodologies with a special focus on cross-layer modeling consistency and system schedulability are developed to provide operational supports for CAM systems.

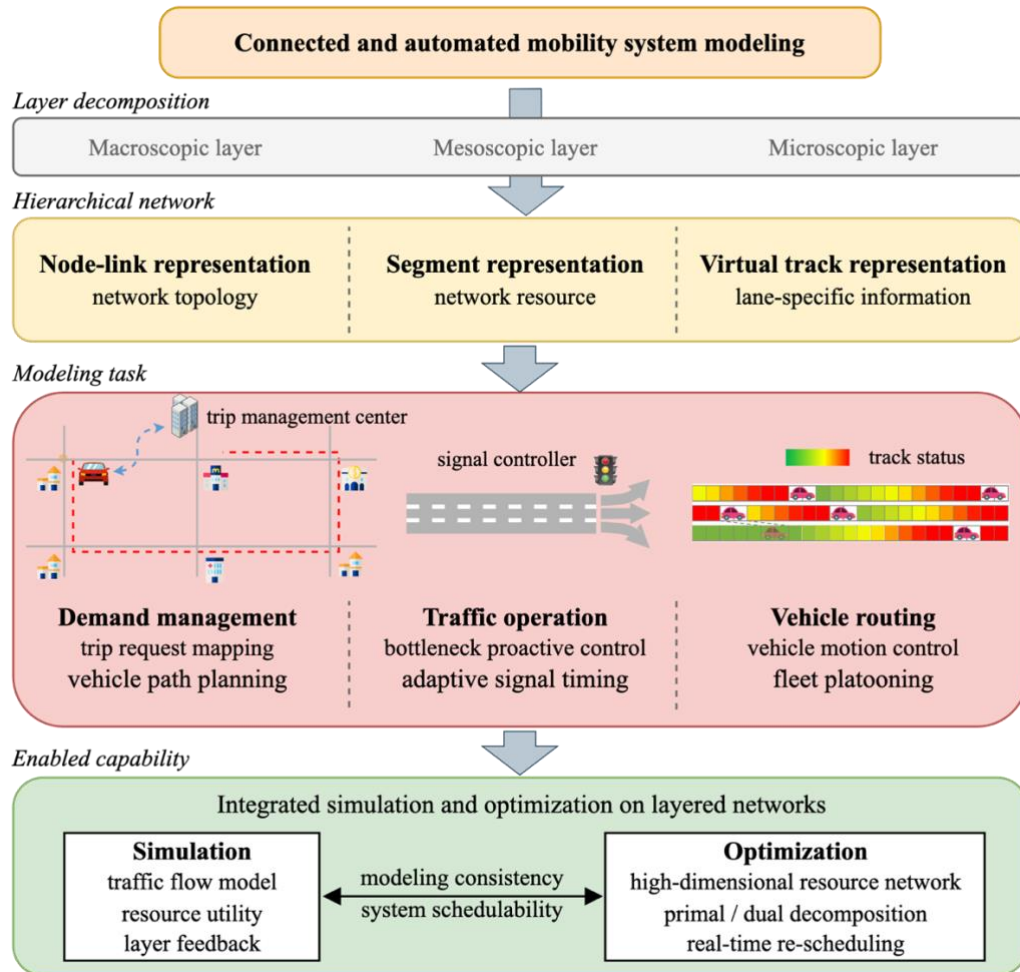


Fig. 2. Multiresolution CAM System Modeling Framework on Layered Networks.

3.3 CAM System Optimization on Layered Multiresolution Networks

3.3.1 Vehicle Trajectory Optimization on Virtual-track-based Microscopic Networks

The emergence of CAM provides the possibility of coordinating different groups of travelling agents and vehicles to effectively utilize limited infrastructure resources and improve the performance of the entire transportation system. In existing studies, extensive efforts have been devoted to vehicle motion planning and trajectory control on spatial-continuous or two-dimensional state lattice networks (Katrakazas et al., 2015) to model obstacle avoidance and smooth maneuvers. This section focuses on vehicle trajectory optimization on the proposed spatial-discrete virtual-track network. In particular, an integrated path planning and lane-changing model with the objective of minimizing total travel cost on a single road segment of interest is described, while it can be extended to general network applications.

Time-expanded graph construction for mathematical programming

Fig. 3 shows the commonly used space-continuous and the proposed spatial-discrete representation for a road segment. The spatial-discrete representation consists of intra-connected microscopic nodes and links, which can essentially be viewed as a graph for modeling. Use $G = (N, L)$ to denote the network in Fig. 3(b) for further illustrations, where sets N and L represent the set of microscopic nodes and links in graph G , respectively.

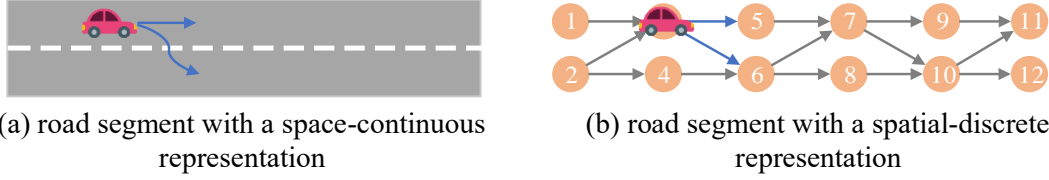


Fig. 3. Mapping Between Spatial-continuous and Spatial-discrete Representations for a Two-lane Road Segment.

To model the spatial-temporal trajectory of vehicles on graph G , a space-time graph $G^{st} = (V, A)$ is built from G , where V and A represent the vertex and arc sets, respectively. In G^{st} , vertex (i, t) is constructed from node $i \in N$, where t represents the time index; traveling arc (i, j, t, t') connecting vertex (i, t) and vertex (j, t') is constructed from link $(i, j) \in L$, representing a vehicle traveling on link (i, j) from time t to t' . $t' - t$ equals the travel time on link (i, j) . In addition to traveling arcs, waiting arcs $(i, i, t, t + 1)$ are built for each node $i \in N$, representing a vehicle does not move and wait on node i for one time interval. Fig. 4 presents the sample network in a one-dimensional space and its corresponding space-time graph. In Fig. 4(b), space-time trajectories of three vehicles are also provided, where vehicle a keeps traveling on the inner lane with odd node numbers, vehicle b is cruising on the outer lane with even node numbers, and vehicle c switches from the inner to outer lane on node 7 at time 11.

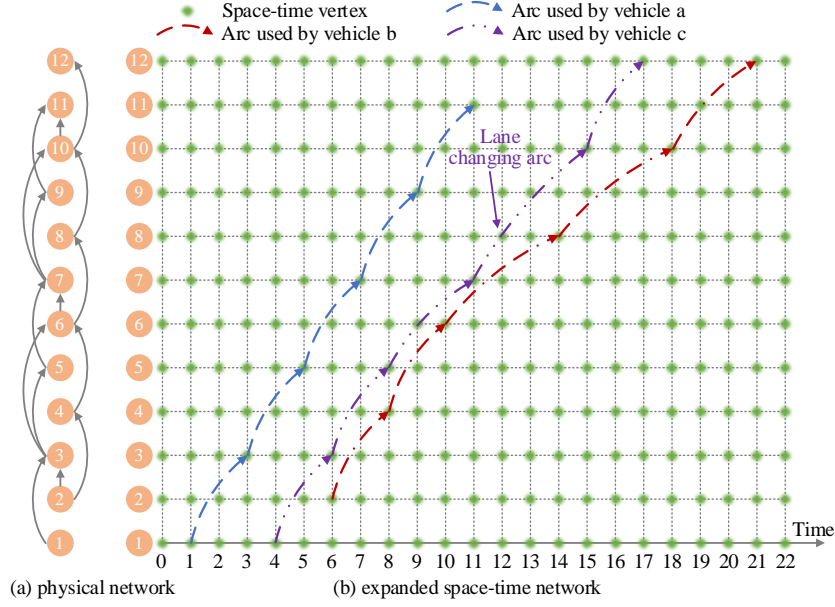


Fig. 4. A Sample Physical Network and Its Corresponding Space-time Graph

Modeling of obstacle avoiding using dynamic occupancy time lag sets

With minimum time headway rule, how safe vehicle trajectories are modeled on the proposed time-expanded virtual track-based network is illustrated below. The minimum time headway rule states that the time difference between two adjacent vehicles passing any point should be larger than or equal to a certain value. In the proposed graph, it can be expressed as, if a space-time arc (i, j, t, t') is used by one vehicle, vertices in sets $S_1 = \{(i, \tau) | t \leq \tau \leq t + h - 1\}$ and $S_2 = \{(j, \tau) | t' \leq \tau \leq t' + h - 1\}$ are also considered to be occupied by the vehicle, where h is the minimum time headway. Set $\varphi_{i,j,t,t'} = S_1 \cup S_2$ denotes the vertex set of arc (i, j, t, t') to guarantee a safe time headway, as shown in Fig. 5 with the minimum time headway of four time units.

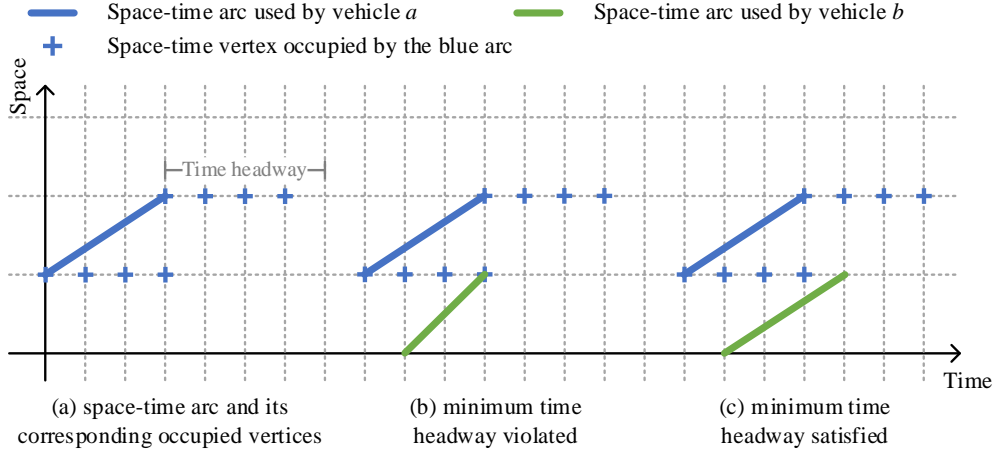


Fig. 5. Minimum Time Headway Modeling on the Space-time Network with Time Lags.

In Fig. 5(a), the vertices indicated by a blue cross are considered occupied if a vehicle uses the blue arc. In this case, these marked vertices constitute the occupancy time lag set $\varphi_{i,j,t,t'}$ of the blue arc. Fig. 5(b) and Fig. 5(c) show two scenarios where the minimum time headway is violated and satisfied when vehicle b follows leading vehicle a , respectively.

Integer linear programming formulation

Based on the introduction of space-time networks and time headway modeling, this section presents a concise optimization model for vehicle trajectory optimization. Given vehicle-based travel demand, the goal is to coordinate the space-time trajectory of each vehicle such that total travel cost is minimized while travel safety is maintained. Travel demand input data include the departure time, origin node, and destination node of each vehicle. The optimization model (M1) is as follows:

Model M1: Single-layer individualized trajectory optimization in general space-time networks

Objective function

$$\min Z = \sum_{k \in K} \sum_{(i,j,t,t') \in A} c_{i,j,t,t'} \times x_{i,j,t,t'}^k. \quad (1)$$

Subject to:

Flow balance constraint:

$$\begin{aligned} & \sum_{(i,t):(i,j,t,t') \in A} x_{i,j,t,t'}^k - \sum_{(i,t):(j,i,t',t) \in A} x_{j,i,t',t}^k = \\ & \begin{cases} -1 & j = O(k), t' = DT(k) \\ 1 & j = D(k), t' = T \\ 0 & \text{otherwise} \end{cases}, \forall k \in K, \forall (j, t') \in V. \end{aligned} \quad (2)$$

Time headway constraint:

$$x_{i,j,t,t'}^k \leq \theta_{i,t}^k, \forall k \in K, \forall (i, j, t, t') \in A, \forall (i, t) \in \varphi_{i,j,t,t'}. \quad (3)$$

Generic driving obstacle avoiding constraint:

$$\sum_{k \in K} \theta_{i,t}^k \leq 1, \forall (i, t) \in V. \quad (4)$$

Decision variables:

$$\begin{aligned} & x_{i,j,t,t'}^k \in \{0,1\}, \forall k \in K, (i, j, t, t') \in A. \\ & \theta_{i,t}^k \in \{0,1\}, \forall k \in K, (i, t) \in V. \end{aligned} \quad (5)$$

The objective function in Eq. (1) minimizes the total travel cost, where $c_{i,j,t,t'}$ is the cost of using motion arc (i, j, t, t') ; $x_{i,j,t,t'}^k$ is a binary variable indicating whether vehicle k uses arc (i, j, t, t') or not, and K denotes the vehicle set. For the two types of most-used travel costs, that is, travel distance and travel time, $c_{i,j,t,t'}$ denotes the physical length of link (i, j) and travel time of the corresponding arc $(t' - t)$, respectively. Eq. (2)

is a set of flow balance constraints in the time-expanded graph, where $O(k)$, $D(k)$, and $DT(k)$ represent the origin node, destination node, and departure time of vehicle k , respectively; T is the planning time horizon. Eq. (3) ensures that if arc (i, j, t, t') is used by vehicle k , all vertices in set $\varphi_{i,j,t,t'}$ should also be marked as “occupied” by vehicle k . $\theta_{i,t}^k$ is a binary variable. To model generic obstacle avoidance as a result of dynamic occupancy time lags, if vertex (i, t) is “occupied” by vehicle k , $\theta_{i,t}^k = 1$; otherwise, $\theta_{i,t}^k = 0$. Eq. (4) states that each vertex (i, t) can only be occupied by at most one vehicle, which guarantees a minimum time headway between different vehicles. Finally, Eq. (5) specifies the decision variables and their domains.

3.3.2 Hierarchical Modeling on Multiresolution Networks

Model M1 is a time-indexed integer programming (IP) model. However, as space-time networks rely on a discretization of time, the number of binary variables in model M1 could be extremely large, making it difficult to solve efficiently using existing IP solvers in real-life large-scale applications. Boland and Savelsbergh (2019) provided insightful discussions on various perspectives on IP for time-dependent models, focusing on a dynamic discretization discovery paradigm. On the other hand, the concise form of model M1 provides the possibility of solving it under a dual-decomposition framework. Among the three sets of constraints, the obstacle avoiding constraint in Eq. (4) is a coupling constraint. Relaxing the constraint under a Lagrangian relaxation framework results in multiple independent subproblems that can be solved using computationally efficient dynamic programming techniques without the need to explicitly create a full time-expanded network. This approach was used by Mahmoudi and Zhou (2016) to study a class

of space-time-state network within a more complex vehicle pick-up and delivery context. In addition, Lu et al. (2016), Shang et al. (2019), Yao et al. (2019), and Zhang et al. (2019) thoroughly described the application of dual decomposition and ADMM methods to solve large-scale transportation problems on time-expanded networks.

This study further builds an integrated dual-layer optimization model that enables a hierarchical decomposition and iterative feedback scheme for solving large-scale vehicle path planning and trajectory control problems. In Fig. 6, both mesoscopic and microscopic network layers are constructed for a freeway corridor of interest, on which traffic is modeled in an aggregated flow and an individual agent manner, respectively. Both networks have time-indexed variables, whereas the lower layer is associated with a finer discretization.

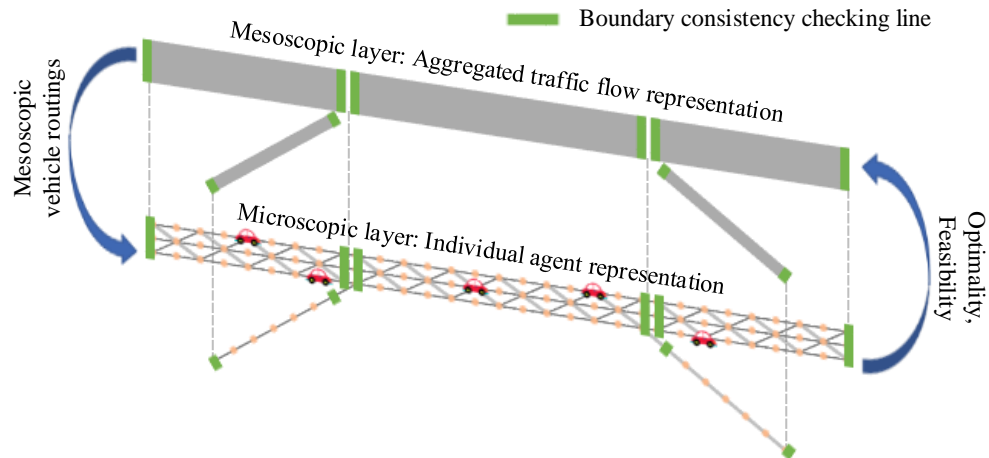


Fig. 6. Vehicle Path Planning and Trajectory Control on Layered Transportation Networks with Boundary Consistencies at Two Ends of Each Link.

Model M2 is presented as follows:

Model M2: Integrated optimization for mesoscopic flow routing and microscopic trajectory planning

Objective function

$$\min Z = CF(x_{i,j,t,t'}^k, y_{I,J,\tau}). \quad (6)$$

Subject to:

Traffic flow dynamics constraint on the microscopic layer:

$$MITD(x_{i,j,t,t'}^k) = 0, \forall k \in K, \forall (i, j, t, t') \in A. \quad (7)$$

Traffic flow dynamics constraint on the mesoscopic layer:

$$MATD(y_{I,J,\tau}) = 0, \forall (I, J) \in \mathcal{L}, \forall \tau \in \mathcal{T}. \quad (8)$$

Spatial and temporal coupling constraint between mesoscopic and microscopic layers:

$$\sum_{k \in K} \sum_{(i,j) \in \Gamma(I,J)} \sum_{t \in \Pi(\tau)} x_{i,j,t,t'}^k = y_{I,J,\tau}, \forall (I, J) \in \mathcal{L}, \forall \tau \in \mathcal{T}. \quad (9)$$

The objective function in Eq. (6) minimizes the total travel cost defined by a general cost function $CF(x_{i,j,t,t'}^k, y_{I,J,\tau})$, where variables $x_{i,j,t,t'}^k$ are used to model individualized agent movements on the microscopic layer; $y_{I,J,\tau}$ denotes the aggregated traffic flows on mesoscopic link (I, J) at time τ . The traffic flow dynamics constraint in Eq. (7) on the microscopic layer corresponds to Eqs. (2)-(5) in the single-layer model. Eq. (8) is a generalized constraint for describing mesoscopic traffic flow dynamics, where \mathcal{L} and \mathcal{T} represent the sets of links and time intervals on the mesoscopic layer, respectively. One can choose a specific mesoscopic traffic flow model, such as cell transmission model (Daganzo, 1995), or link transmission model (Yperman, 2007, Zhou and Taylor, 2014) to offer detailed formulations in Eq. (8). A recent study by Cheng et al. (2022) employed a fluid queue model and proposed a parsimonious polynomial function-based scheme for modeling time-dependent traffic system states, particularly under oversaturated conditions.

A follow-up study by Zhou et al. (2022) further investigated the macro-to-meso connection between polynomial queue model and commonly used volume-delay function. Adopting traffic flow models with a smaller number of parameters can help reduce the complexity of model M2. Eq. (9) is the coupling constraint to maintain consistent flows on space-time boundary conditions between the mesoscopic and microscopic layers, where $\Gamma(I, J)$ denotes the boundary microscopic link set of mesoscopic link (I, J) ; $\Pi(\tau)$ is the set of finer time intervals in the microscopic layer of time interval τ in the mesoscopic layer. It should be remarked that, as will be introduced in the following sections, osm2gmns package particularly guarantees the consistency in link-to-cell mapping between layers such that tightly coupled multiresolution networks are readily available for related cross-layer modeling (Chiang et al., 2007).

The embedded mesoscopic layer in model M2 naturally provides a hierarchical representation and task decomposable structure. The aggregated traffic flow representation on the mesoscopic layer is more computationally efficient than that on the microscopic layer. To model driver behavior in a multi-scale cognitive architecture, useful information that microscopic layer receives from the mesoscopic layer as boundary condition on each mesoscopic link can be utilized, enabling microscopic trajectory optimization to be decomposed into multiple sub-problems. Mathematically, each sub-problem corresponds to a mesoscopic link with given boundary conditions, which can be solved independently. The independence of sub-problems dramatically reduces the complexity of vehicle trajectory optimization on microscopic networks. Trajectory optimization results from the microscopic layer should also be able to provide information to the mesoscopic layer in terms of the optimality and feasibility of mesoscopic routing results, which can in turn

guide re-routings on the mesoscopic layer. This iterative feedback process can be repeated until a convergence between the two layers is achieved. A number of important integrated model efforts between high-level demand and low-level supply models have been proposed (Lin et al., 2008; Mahmoudi et al., 2021). From an algorithmic perspective, the iterative process can be performed under a Benders decomposition framework (Benders, 1962), which was designed to efficiently solve large-scale problems with a hierarchical structure.

3.4 CAM System Simulation on Layered Networks with Hierarchical Driving Decisions

This section introduces the CAM simulation framework on layered networks, which enables us to capture different levels of actions, namely strategic (trip planning), tactical (maneuver planning), and operational (vehicle operation). As discussed in Section 3.2, in the proposed layered modeling framework, travel request mapping is performed on the macroscopic layer, traffic management and operation are conducted on the mesoscopic layer, and high-fidelity vehicle motion planning is executed on the microscopic layer. The proposed CAM simulation framework has two major features: (1) traffic assignment and vehicle trajectory control are performed on two different layers to capture aggregated and individual behaviors, and (2) the joint routing and lane-changing decisions of a vehicle are performed on the spatial-discrete microscopic layer. Hence, this section mainly focuses on the interaction between mesoscopic and microscopic layers, and microscopic vehicle motions on spatial-discrete networks.

3.4.1 Travel Utility Modeling on the Microscopic Virtual Track Layer

In general, when a vehicle travels on a road, all maneuvers, such as acceleration, deceleration, and lane changing, performed by the driver can be considered as actions to maximize the travel utility. Travel utility may be affected by various external environmental and human factors, such as travel time, safety, and comfort. For example, a driver may switch to a lane with fewer vehicles to increase travel speed and reduce total travel time. In this regard, a general modeling framework that can conveniently characterize and calculate travel utilities is highly required for understanding and reproducing/simulating travel behaviors in real life. On the other hand, in the coming era of CAV, one of the most challenging tasks is designing effective policies and operation strategies for managing and coordinating large fleets of CAVs to improve system performance (utility) under limited infrastructure resources. An open question of meeting this requirement is how to systematically measure road resource utilities.

The lane-changing example in Fig. 3 is used to illustrate the benefits of adopting a spatial-discrete network representation for travel utility characterization. (1) Safety utility: In a spatial-discrete representation, vehicles are exactly mapped to virtual microscopic nodes; only predefined virtual tracks can be used. Therefore, interactions between different vehicles can be measured in a simplified manner, and important statistics for characterizing lane-changing safety, such as time-to-collision, can be further incorporated as enhancements. (2) Travel time utility: determining lane changing or staying in the current lane in Fig. 3(b) becomes evaluating the utilities of going to nodes 5 and 6, and their associated future utilities. In addition, because of the discrete nature of the road segment

representation, exact optimization methods, such as network flow algorithms, can be applied for utility calculations.

Considering the vehicle in Fig. 3(b) as an example, the travel time utility of node 5 consists of two parts: (1) the time required to move from the current node to node 5 and (2) time needed to travel from node 5 to the downstream stop line of the current mesoscopic link. Part (1) depends on the state of node 5, that is, whether node 5 is occupied by other vehicles. Part (2) can be viewed as predicted or experienced travel time, and it is unknown in advance. Therefore, without loss of generality, this recursive value evaluation is simplified in the proposed approximation framework by an estimated or expected travel time, using reference speed and distance to the stop line.

Benefitting from the spatial-discrete representation of microscopic networks, backward trees that store the shortest distance from each node to the stop line can be built in advance on mesoscopic links for quick queries in the simulation process. The root of each backward tree is the set of microscopic nodes on the stop line, which are called meso-to-micro intermediate destination (MID) nodes in this study. It should be noted that, for different movements on the same mesoscopic link, the set of MID nodes is different. Therefore, a backward tree must be built for each movement in a mesoscopic link. In Fig. 7, a backward tree with destination node sets of $[d_1]$, $[d_2, d_3]$, and $[d_4]$ should be built for left-turn, through, and right-turn movements, respectively. Algorithm 3 presents the backward tree construction process for movement m on a mesoscopic link.

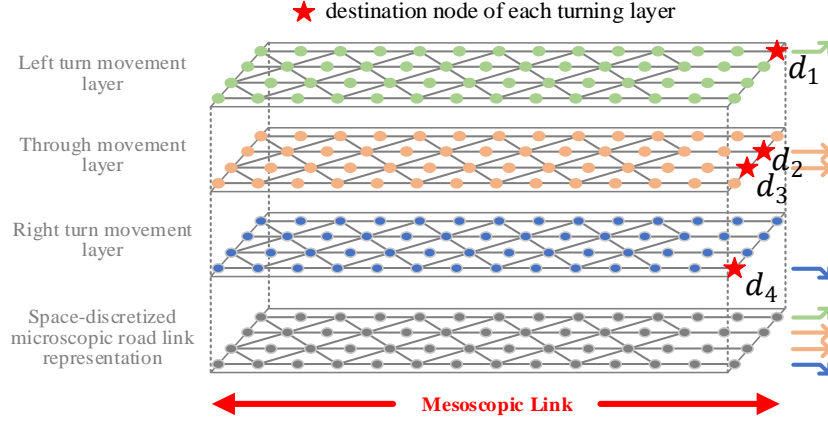


Fig. 7. Spatial-discrete Microscopic Road Link Representation and Corresponding Virtual Turning Layers with Meso-to-micro Intermediate Destination (MID) Nodes.

Algorithm 1 Backward tree construction for movement m on a mesoscopic link

Input: microscopic representation $G = (N, L)$ of the mesoscopic link, where N and L denote the microscopic node and link set, respectively; destination node set D of movement m on the mesoscopic link

Output: distance from microscopic nodes to the stop line for movement m on the mesoscopic link

- 1: $d_i^m \leftarrow +\infty, \forall i \in N$
 - 2: $d_i^m \leftarrow 0, \forall i \in D$
 - 3: $U \leftarrow D$
 - 4: **while** $U \neq \emptyset$ **do**
 - 4: move one node i out of U
 - 5: **for** incoming link $l = (j, i)$ of node i **do**
 - 6: **if** $d_i^m + \text{length of } l < d_j^m$ **then**
 - 7: $d_j^m = d_i^m + \text{length of } l$
 - 8: put j into U
 - 9: **return** $\{d_i^m | i \in N\}$
-

3.4.2 CA(M)-based Simulation for Joint Path-planning and Lane-changing Decisions

In this section, CA(M) model is used as a simple example to demonstrate the incorporation of a wide range of car-following and lane-changing models into the proposed CAM simulation framework. It should be noted that CA(M) model is adopted because of its parsimonious form and ability to describe complicated traffic flow phenomena. It has been demonstrated that CA(M) model is theoretically consistent with the linear car-following model CF(L) and kinematic wave model KW (Daganzo, 2005). The consistency

between these three models is illustrated in Fig. 8, where the bold red dashed line in each sub-plot denotes a backward wave with the same speed. Zhou et al. (2015) provided a detailed description of the impact on emission estimates.

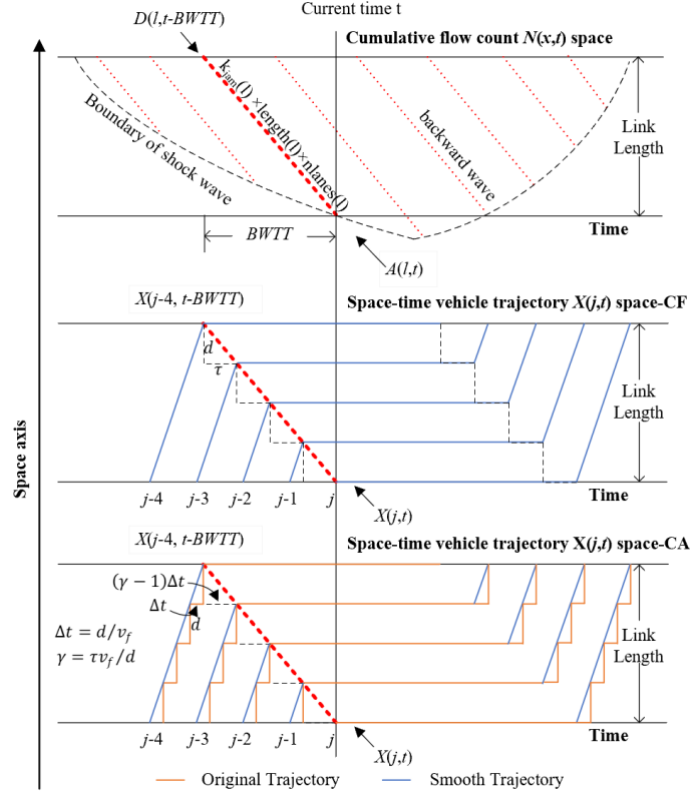


Fig. 8. Illustration of the Consistency Between KW, CF(L) and CA(M).

The car-following rule in the CA(M) model is expressed as

$$Z_{t+1}^k = \min\{Z_t^k + 1, Z_{t-\gamma+1}^{k-1} - 1\}, \quad (10)$$

where Z_t^k represents the microscopic node index in the constructed grid network used by vehicle k at time interval t . Eq. (10) can be explained as vehicles can use the next microscopic node if it has been vacant for $\gamma - 1$ time intervals; otherwise, they cannot move forward and continue to stay at where they are. The safe time headway reservation mechanism deserves future empirical and theoretical investigations, as the proposed

spatial-discrete microscopic road link representation only serves as a simplified (linear) approximation to complex multi-agent games (Yang and Wang, 2020; Huang et al., 2021; Di and Shi, 2021). A detailed illustration of the CA(M)-based simulation process is presented in Algorithm 2.

Algorithm 2 CA(M)-based microscopic simulation

Input: microscopic and mesoscopic network representations of the research area; travel demand

Output: vehicle trajectory of all travel demands

```

1: (Step 1: Traffic Assignment)
2:   perform traffic assignment on the mesoscopic network
   get mesoscopic link sequence  $P_k$  for each vehicle  $k \in K$ 

3: (Step 2: Simulation Initialization)
4:   simulation time stamp  $t =$  simulation starting time
5:    $ta_i = t, \forall i \in N$ 
6:    $K^{uld} = K, K^{act} = \emptyset$ 

7: (Step 3: Vehicle Loading)
8:   for  $k \in K^{uld}$  do
9:     if  $t_k^d == t$  then
10:       $K^{act} \leftarrow K^{act} \cup \{k\}, K^{uld} \leftarrow K^{uld} / \{k\}$ 
11:       $n_k = o_k, t_k^r = t, L_k \leftarrow$  the first link in  $P_k$ 

12: (Step 4: Vehicle Status Updating)
13:   for  $k \in K^{act}$  do
14:     if  $t_k^r == t$  then
15:       if  $n_k == d_k$  then
16:         vehicle  $k$  has finished its trip, remove  $k$  from  $K^{act}$ 
17:       Else
18:         if  $n_k$  is the last node on the mesoscopic link  $L_k$  then
19:            $L_k \leftarrow$  the link after  $L_k$  in  $P_k$ 
20:           reachable node set  $\Omega \leftarrow \phi$ 
21:           for all microscopic link  $l = (n_k, i)$  do
22:             if  $ta_i \leq t + tt_{n_k, i}$  do
23:                $\Omega \leftarrow \Omega \cup \{i\}$ 
24:             if  $\Omega == \phi$  then
25:                $t_k^r = t + 1$ 
26:             Else
27:               for  $i \in \Omega$  do
28:                  $u_i = tt_{n_k, i} + d_i^m / v_{L_k}$ 
29:                 choose  $i \in \Omega$  as the next node of vehicle  $k$  based on utilities  $\{u_i | i \in \Omega\}$ 
30:                  $t_k^r = t + tt_{n_k, i}, ta_{n_k} = t + \gamma, ta_i = inf, n_k = i$ 
31:           if  $t ==$  simulation ending time then
32:             simulation finished
33:           Else
34:              $t = t + 1$ , go back to Step 3
35:           return vehicle trajectories

```

3.5 Partially Schedulable CAM System Operation on Layered Multiresolution Networks

The optimization-oriented methodologies developed in Section 3.3 consider transportation systems that are fully schedulable, and agents in CAVs or multimodal MaaS can follow their customized schedules precisely. However, in real life, the evolution and operation of transportation systems are partially schedulable because they are affected by a range of stochastic and dynamic environmental and human factors, such as random travel demand, weather conditions, and traffic incidents.

This section mainly focuses on the optimization modeling of partially schedulable CAM system operations on layered multiresolution networks. In Subsection 3.5.1, a two-stage optimization paradigm is presented to seek optimal offline scheduling considering stochastic online scenarios. In Subsection 3.5.2, a distributed re-scheduling mechanism is designed for online vehicle control and conflict resolving.

3.5.1 Stochastic Offline Scheduling of CAM Decisions

This section presents a two-stage stochastic optimization framework to consider separate decision variables under mesoscopic level planning and microscopic level re-scheduling. The objective is to find optimal pre-trip schedules for agents with the maximum expected utilities by considering stochastic online scenarios. The conceptual model is presented as model M3, with a similar form of the commonly used two-stage stochastic programming model developed by Birge and Louveaux (2011).

In model M3, the objective function is to maximize the expected travel utilities of all agents in the mesoscopic layer by optimizing the aggregated route assignment variable y , which includes two parts. The first part, $c(y)$, is the scenario-independent utility of route

assignment $y \in \Omega$, where Ω is the feasible region of the variable y . The second part, $E_\omega[q(y, \omega)]$, denotes the expected scenario-dependent utility of route assignment y , where $q(y, \omega)$ represent the utility of y in scenario ω . The value of $q(y, \omega)$ is obtained by optimizing the vehicle trajectory variable x_ω on the microscopic layer with a given mesoscopic route assignment y in scenario ω (constraints (13)-(14)), whose feasible region is $\Phi_\omega(y)$.

Model M3: Offline scheduling with a two-stage structure

Objective function

$$\max_y Z = c(y) + E_\omega[q(y, \omega)]. \quad (11)$$

Subject to:

Mesoscopic layer flow modeling constraint:

$$y \in \Omega. \quad (12)$$

Meso-micro layer coupling constraint:

$$q(y, \omega) = \min_{x_\omega} g(y, x_\omega), \forall y, \forall \omega. \quad (13)$$

Microscopic layer agent modeling constraint:

$$x_\omega \in \Phi_\omega(y). \quad (14)$$

It is important to rigorously describe this two-stage process before tackling more complex recursive decisions in a real-world system. The first stage focuses on agent route assignment in the mesoscopic layer with a relatively simple representation. In the second stage, the system utility of a certain route assignment is evaluated in the microscopic layer

for different possible scenarios. Evaluating scenario-based route assignments in the microscopic layer provides more profound and accurate information to the first level for making route assignment decisions.

3.5.2 Distributed Online Re-scheduling of Trajectories

With the Recognition of the partial schedulability of CAM systems, this section proposes a systematic vehicle coordination scheme for CAM system operation in dynamic online environments, with the objective of minimizing deviations between actual system operation status and offline system schedules. It is easy to guide the motion of each individual vehicle based on offline schedules; while, due to potential conflicts between vehicles, it becomes complex when simultaneously coordinating multiple vehicle groups. This research introduces a systematic vehicle conflict resolving scheme in dynamic CAM environments below.

Fig. 9 presents an example of conflict resolving between two vehicles, which can be easily generalized to scenarios with multiple vehicles. In this example, vehicle a is going to leave the freeway through the exit ramp, and vehicle b keeps traveling on the mainline. There is a potential conflict between the two vehicles on point p . The convenience of modeling interactions between vehicles is one of the advantages of the proposed discretized virtual track-based modeling approach compared with the existing continuous modeling approaches.

From the perspective of resource allocation, the potential conflict between vehicles a and b can be considered a temporal resource competing at point p , which is essentially an assignment problem. In the bottom right of Fig. 9, a graphic illustration of the resource

assignment between vehicles a and b is presented, where the resource denotes the time resource on point p ; $c_{v,r}$ denotes the (schedule deviation) cost of vehicle v if resource r is assigned to vehicle v . This example can easily be extended to complicated scenarios with multiple vehicles competing for both space and time resources. The general formulation of the resource assignment modeling on the proposed discrete virtual track-based network is as follows:

Model M4: Online re-scheduling

Objective function

$$\min Z = \sum_{v \in V} \sum_{r \in R} c_{v,r} x_{v,r}. \quad (15)$$

Subject to:

Demand (vehicle) side constraint:

$$\sum_{r \in R} x_{v,r} = 1, \forall v \in V. \quad (16)$$

Supply (resource) side constraint:

$$\sum_{v \in V} x_{v,r} \leq 1, \forall r \in R. \quad (17)$$

Decision variables:

$$x_{v,r} \in \{0,1\}, \forall v \in V, r \in R. \quad (18)$$

In model M4, the binary decision variable $x_{v,r}$ denotes the assignment of resource r to vehicle v . As reviewed by Duan and Pettie (2014), many mature and efficient algorithms, such as the Hungarian algorithm, for solving model M4 have been developed. Due to the variable communication range, stability, and latency between vehicles and centralized control centers in real-time conflict-resolving applications, in addition to

algorithmic efficiency, the ability to find optimal solutions in a distributed manner is also of vital importance. The auction algorithm (Bertsekas, 1990) provides an iterative auction scheme for finding optimal solutions of assignment problems. In this scheme, bidders (vehicles) dynamically adjust their acceptable prices (dual costs) for preferred items (resources) until an equilibrium is reached (optimal solution is found), and an auctioneer (control center) is not required. The control-center-exclusive scheme enables the proposed conflict resolving methodology to be applicable in highly dynamic and distributed CAM operation environments.

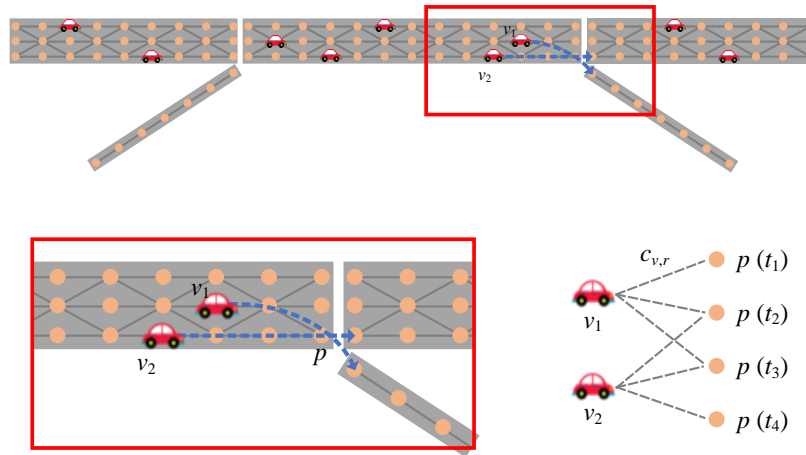


Fig. 9. Graphic Illustration of Conflict Modeling on the Proposed Virtual-track-based Network.

3.6 Open-source Tools for Enabling Cross-resolution Modeling

This section introduces two open-source tools, *osm2gmns* and *CAMLite*. In particular, *osm2gmns* helps users quickly build and manipulate transportation networks. *CAMLite* is a customizable integrated traffic simulation and optimization platform for CAM system modeling.

3.6.1 osm2gmns

As part of this research, osm2gmns is offered as an open-source package to enable users to conveniently obtain and manipulate networks from OpenStreetMap. With a single line of Python code, users can obtain and model drivable, bikeable, walkable, railway, and aeroway networks for any region in the world and output networks to CSV files in the general modeling network specification (GMNS) format for seamless data sharing and research collaboration. Here, some major features of osm2gmns pertaining to multiresolution and CAM modeling are introduced. The detailed user guide can be accessed at <https://osm2gmns.readthedocs.io>.

Standard network specification

The network specification adopted in osm2gmns is GMNS (<https://github.com/zephyr-data-specs/GMNS>; Smith et al., 2020), which enables convenient network data sharing and seamless cooperation in various network modeling applications. GMNS defines a common human- and machine-readable format for sharing routable road network files. It is designed for multimodal static and dynamic transportation planning and operation models.

Ready-to-use MRM network

The purpose of OpenStreetMap is to provide free and editable geographic map data around the world, instead of dedicated transportation modeling. To provide users with ready-to-use MRM networks for transportation modeling, osm2gmns features the following functionalities:

Network topology reconstruction. In OpenStreetMap, road links are typically represented by ways. The geometry of a way is defined by a series of reference nodes. A way may contain multiple intersections in the middle, making original networks in OpenStreetMap not routable. osm2gmns addresses this issue in the network processing stage and reconstructs topologies as needed to guarantee the network connectivity across all resolutions.

Intersection consolidation and movement generation. In OpenStreetMap, a large intersection, as shown in Fig. 10(a), is typically represented by multiple nodes. This representation scheme brings difficulties in some intersection-specific applications, such as signal control. osm2gmns automatically identifies such intersections and enables users to consolidate intersections when parsing original networks. The resulting intersection, as shown in Fig. 10(b), maintains the same geometry as the original one with a reconstructed topology. In addition, the movement generation module in osm2gmns helps users quickly generate movement information at intersections, as shown in Fig. 10(c), to enable movement-based modeling applications.

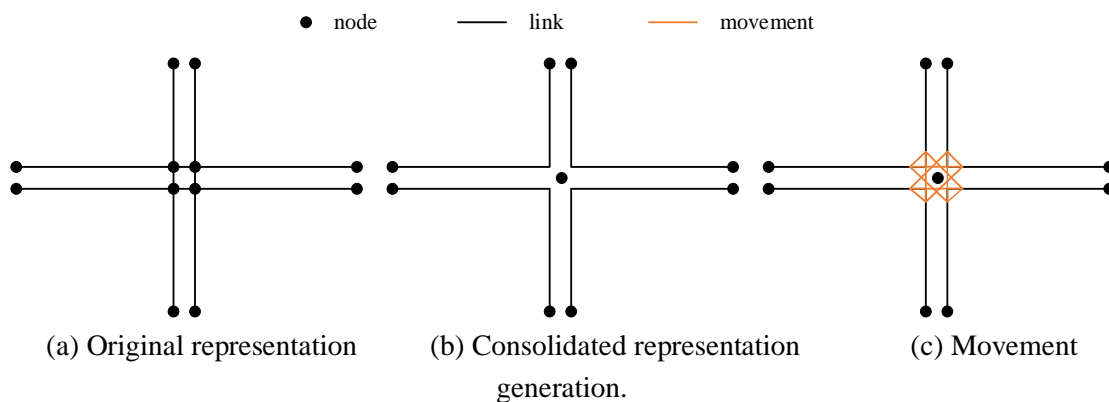


Fig. 10. Intersection Consolidation and Movement Generation.

Multiresolution network generation. As multiresolution routable networks are critically needed for CAM modeling, for any macroscopic network that meets the GMNS standard or a selected subarea from OpenStreetMap, osm2gmns can be used to build its corresponding mesoscopic and microscopic networks with consistent mapping across different layers. As an example, the multiresolution network near Arizona State University, Tempe campus, is available on the web-based transportation network visualization platform at <https://asu-trans-ai-lab.github.io/web/index.html>.

Multimodal network construction and activity generation locations for demand modeling

osm2gmns supports five different network types: auto, bike, walk, railway, and aeroway. Fig. 11 shows the drivable, bikeable, and walkable network near Arizona State University, Tempe Campus.



(a) Drivable network (b) Bikeable network (c) Walkable network
Fig. 11. Drivable, Bikeable, and Walkable Network near Arizona State University, Tempe Campus.

Travel demand data preparation, as a key part of multimodal transportation demand-supply modeling, requires considerable efforts in practical applications. osm2gmns can produce detailed point-of-interest information, including the type, location,

shape, and area within the area of interest. This information is essential to build and analyze the residential and employment characteristics of traffic zones.

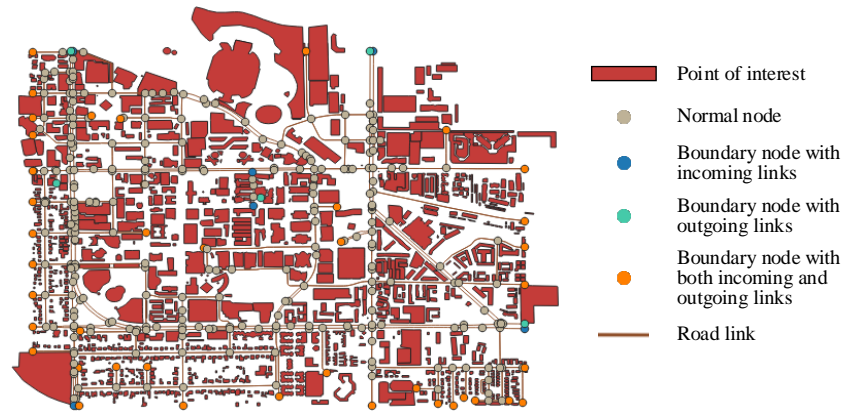


Fig. 12. Points of Interest and Boundary Nodes Identified by osm2gmns on a Sample Network near Arizona State University, Tempe Campus.

Available transportation network datasets

osm2gmns is used to generate the entire United States driving network using research computing facilities at Arizona State University, as shown in Fig. 13. A total of 1.44 TB RAM was used to generate the network. The resulting network contains 20,459,306 nodes and 49,608,229 links. State-by-state United States GMNS networks (with driving and rail modes) are shared at https://github.com/asu-trans-ai-lab/Integrated_modeling_GMNS/tree/main/examples/United_States_network to facilitate future network modeling studies.

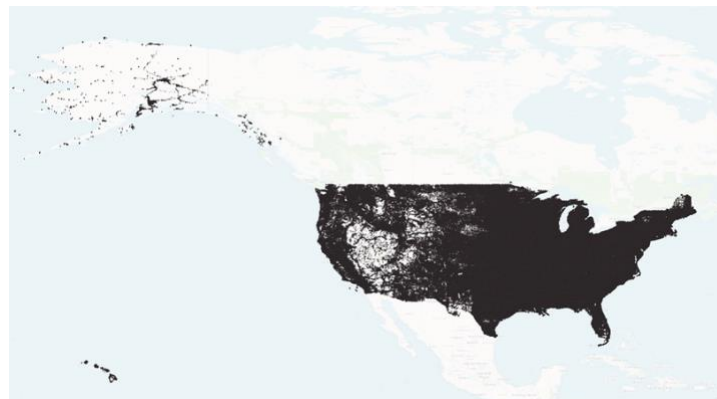


Fig. 13. Entire United States Driving Network Generated by osm2gmns.

3.6.2 CAMLite

CAMLite is an open-source platform for integrated CAM system simulation and optimization based on the proposed multiresolution network representation. One of the major differences between CAMLite and existing traffic modeling tools is the adoption of a cell-based spatial-discrete underlying network representation for tracking vehicle motion. In the era of CAV, designing and evaluating effective vehicle coordination and management strategies is the key to improve the efficiency of current traffic systems with limited infrastructure resources. Fig. 14 presents the system architecture of CAMLite, which has the following major modules:

Travel demand: Provides vehicle travel demand input in the form of an origin-destination matrix to CAMLite. Vehicle travel demand generation is performed using vehicle routing or ride-sharing algorithms on a macroscopic network with a specified passenger travel demand and service vehicle supply.

Traffic assignment: According to a specific assignment objective (user equilibrium or system optimum), traffic assignment is performed on the mesoscopic network to find a mesoscopic path for each vehicle.

Optimization API: Incorporates user-defined CAM system control algorithms (i.e., trip management and vehicle routing) into the simulation module.

Microscopic simulation: With mesoscopic vehicle paths from the traffic assignment module, the motion of human-driven vehicles and CAVs are simulated based on calibrated human-driver behaviors and user-defined CAV control algorithms, respectively.

The objective of CAMLite is to provide a highly flexible framework for CAM system simulations under different control policies and optimization strategies. Users can incorporate their own simulation rules and optimization models to simulate and evaluate CAM systems under different scenarios. The source code and release of CAMLite can be downloaded at <https://github.com/jiawlu/CAMLite>.

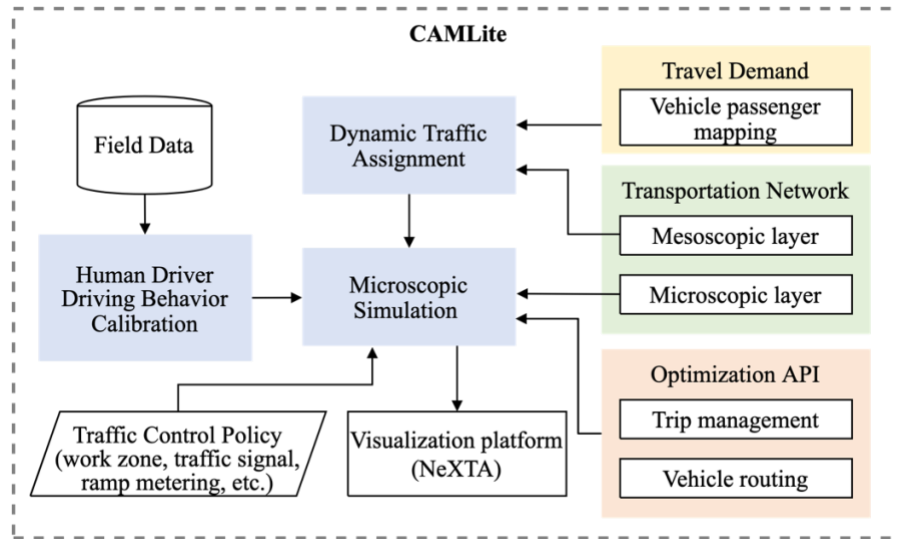


Fig. 14. System Architecture of CAMLite.

3.7 Experiments

Experiments on a freeway corridor were designed to demonstrate the effectiveness of the proposed methodology and open-source tools. As introduced in Subsection 3.7.1, the freeway network in the research area of interest was obtained using osm2gmns. In Subsections 3.7.2 and 3.7.3, the results of traffic simulation and optimization on the selected corridor are reported, respectively.

3.7.1 Network Preparation Using osm2gmns

The selected freeway corridor is on I10, Arizona, United States, and has a length of 5.8 mi. With raw map data downloaded from the OpenStreetMap website, osm2gmns was first used to produce the transportation network in GMNS format, as shown in Fig. 15. The generated network consists of node, link, and movement files. The movement file stores the lane connection information at merge and diverge points. The information is automatically generated by a built-in module in osm2gmns (generateMovements) according to the layout information in the node and link files. The corresponding mesoscopic and microscopic networks were generated using osm2gmns for MRM. Table 4 summarizes sizes of the networks at different resolutions.

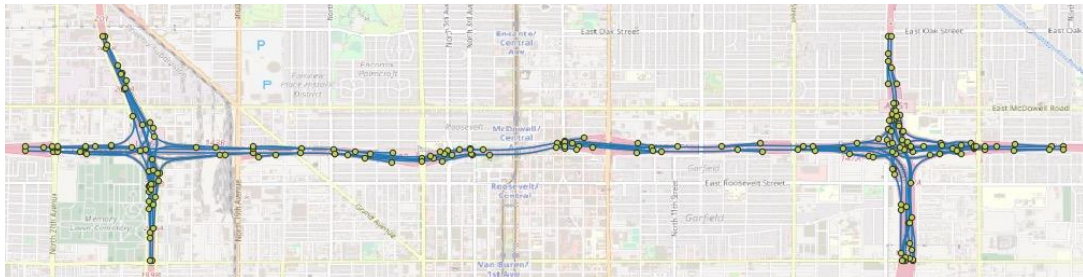


Fig. 15. Macroscopic Network of the Research Area of Interest.

Table 4 Network Size in Different Resolutions.

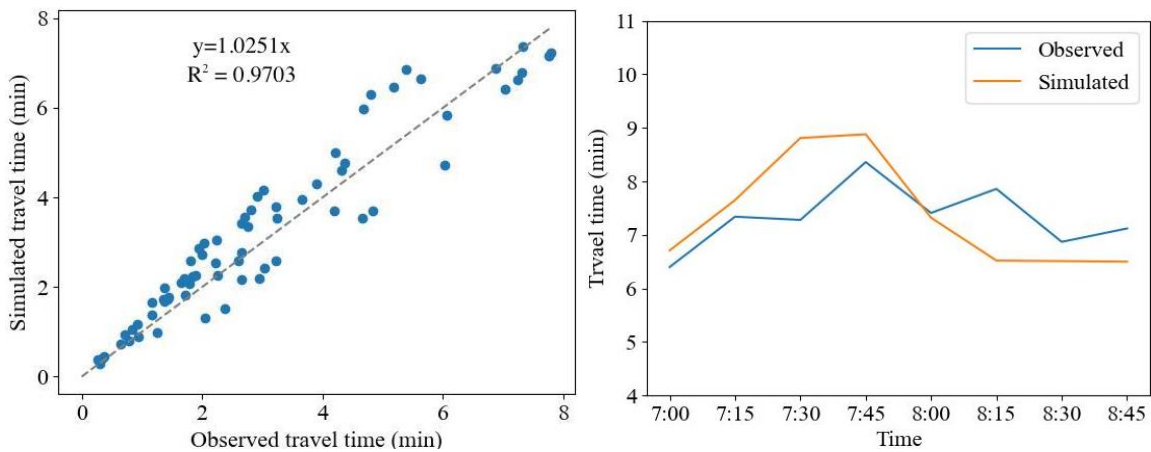
	Macroscopic network	Mesoscopic network	Microscopic network
Number of nodes	203	211	23,475
Number of links	216	222	53,288

3.7.2 Traffic Simulation Using CAMLite

First, traffic simulations on the selected freeway corridor were performed using open-source simulation package CAMLite. In addition to a detailed transportation network, another necessary input for traffic simulation is the time-dependent travel demand within

the analysis time period. The analysis time period was set from 7 to 9 am, and travel demand was obtained using the OD estimation module in DTALite (Zhou and Taylor, 2014) based on link volumes collected from loop detectors. The total number of trips during the analysis period was 62,242.

Fig. 16 presents the simulation results for the selected corridor. In particular, simulated and observed travel times were compared to examine the performance of CAMLite in modeling traffic flow evolutions. Time-dependent observed travel time was obtained using Google Map API. Fig. 16(a) compares the average simulated and observed travel times over the entire analysis period for each OD pair. The average simulated travel times satisfactorily matched with the observed values, with $R^2 = 0.9703$. Fig. 16(b) shows a comparison between time-dependent simulated and observed travel times on a major OD pair, which demonstrates the effectiveness of the proposed methodology and CAMLite in simulating dynamic traffic flows.



(a) Comparison between average simulated and observed travel times

(b) Comparison between time-dependent simulated and observed travel times on a major OD pair

Fig. 16. Simulation Results.

3.7.3 Trajectory Optimization on the Proposed Virtual-track-based Network

In this subsection, an illustrative experiment on vehicle trajectory optimization using model M1 is presented. Due to the high mathematical complexity of the problem, only small-size instances were implemented and directly solved using a commercial solver for illustration purposes.

The network used in this experiment is a merging area adopted from the freeway corridor, as shown in Fig. 15. The corresponding microscopic network is presented in Fig. 17, where the gray points and lines represent microscopic nodes and links, respectively. Four synthetic scenarios with different travel demand levels are designed. The models were solved using Gurobi 8.1 on a Dell Precision 7510 laptop with 2.9 GHz CPU and 32 GB RAM. Table 3 presents the statistics of each model. An increase in the instance size significantly increases the number of decision variables and model complexity, which highlights the need to develop efficient solution methodologies to solve large-scale instances in real life, as discussed in Sections 3.3 and 3.5. Fig. 17 presents some sample optimized vehicle trajectories in the last scenario on a virtual-track-based space-time network.

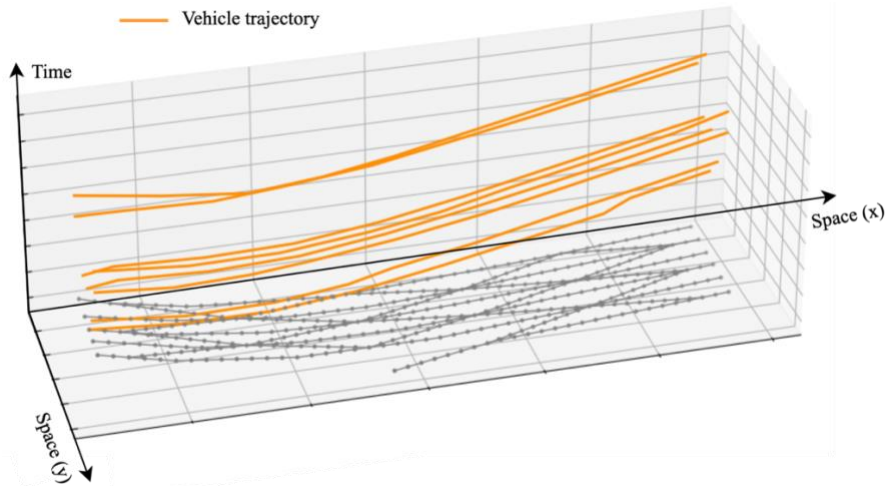


Fig. 17. Sample Optimized Vehicle Trajectories in a Virtual-track-based Space-time Network.

Table 5 Model Statistics under Different Demand Levels to Highlight the Need for Developing Decomposition Methodologies.

Number of vehicles	Number of variables in model M1	Computational time (second)
1	181 binary variables	0.02
6	333 binary variables	0.2
30	30,514 binary variables	93.16
48	259,509 binary variables	2077.09

3.8 Conclusions

This chapter introduced a new layered modeling framework for CAM systems. In the proposed layered framework, CAM systems with hierarchical structures are decomposed into strategic macroscopic, tactical mesoscopic, and operational microscopic layers such that modeling tasks with different computational and resolution requirements can be performed on dedicated layers to seek a balance between efficiency and fidelity in real-life deployments. Critical aspects including consistency and schedulability in layered CAM systems were extensively discussed. As parts of this research, two open-source tools, *osm2gmns* and *CAMLite*, were introduced to facilitate CAM modeling research and deployments.

CHAPTER 4

CROSS-RESOLUTION TRAFFIC STATE ESTIMATION IN CAM SYSTEMS

4.1 Introduction

The accurate identification of traffic system states is the foundation for the effective design and execution of control strategies. Ubiquitous sensing techniques, which enable different types of emerging mobile sensors, location-based services, and participatory sensing, can provide more reliable and richer traffic observations. Consequently, there is a need to design a system state identification framework to improve the observability of traffic systems. This brings a series of theoretically challenging and practically important modeling issues for the problem of traffic system state identification (TSSI) when utilizing heterogeneous sensor data with different degrees of uncertainty sources. Specifically, the TSSI problem under consideration aims to simultaneously estimate three sets of system state variables: (1) traffic stream states such as flow rate, density, and speed on road segments of interest; (2) fundamental diagram parameters such as free-flow speed and jam density of road links; and (3) congestion states represented by the queue profile and delays at traffic bottlenecks. In the literature, the aforementioned traffic system states are typically estimated separately in different problems. That is, (1) the traffic state estimation (TSE) problem is devoted to inferring time-varying traffic state variables; (2) the model parameter estimation (MPE) problem is dedicated to calibrating or adjusting system parameters in traffic flow models; and (3) queue profile estimation (QPE) or congestion bottleneck identification (CBI) is performed with the aim of identifying congestion duration and the resulting queue profile at signalized intersections or freeway bottlenecks.

With the recognition of their vital importance in real-life traffic management and control applications, extensive efforts have been devoted to solving these three problems (TSE, MPE, and QPE). States of traffic streams from TSE are able to provide detailed day-to-day traffic pattern evolutions and are also extremely important for identifying traffic incidents in unobservable areas (Wang et al., 2009; Kuwahara et al., 2021); well-calibrated traffic flow model parameters, especially under different road or weather conditions, enable traffic managers to understand critical attributes of traffic systems in different scenarios, contributing to the identification of traffic state regime switches and the designing of effective policies based on medium-term prediction and proactive control (Qin and Mahmassani, 2004; Geroliminis et al., 2012; Ramezani et al., 2015); and QPEs produce intuitive representations of queue evolutions at oversaturated traffic bottlenecks, supporting effective traffic managements through balancing travel demand and supply during peak hours (Ramezani and Geroliminis, 2015; Yang et al., 2018). In recent years, there has been an emerging trend of incorporating TSE and MPE into an integrated modeling structure (Wang et al., 2022) to achieve better estimations. Traffic flow models in MPE can help regulate state estimations in TSE and by utilizing states in unobserved areas produced by TSE, richer state information can be used in MPE. This study makes the first attempt to systematically perform TSE, MPE, and QPE under a unified modeling framework to take advantage of high-level queue profiles for stabilizing local estimations and in turn, improve QPEs using local estimations, which finally contributes to aggregated traffic modeling and hierarchical control. Owing to the increase in the solution space and complex correlations among different components, performing TSE, MPE, and QPE together, that is, TSSI in this study, results in a more complicated model. The challenge

was to develop a computationally tractable and mathematically rigorous model as well as an efficient solution method for the proposed TSSI problem.

From the perspective of modeling resolutions, with the tradeoff between modeling scales and levels of fidelity, there are three categories of methods for traffic flow modeling: macroscopic, mesoscopic, and microscopic modeling. Focusing on the overall system performance, macroscopic modeling provides aggregated system-wide measures for largescale networks with high computational and modeling efficiencies. On the other hand, microscopic modeling tracks the movement of individual vehicles and vehicle-to-vehicle interactions based on car-following, gap acceptance, and lane-changing theories. Thus, high-resolution modeling results can be produced, while at the same time, the size of modeling scales could be restricted owing to demanding computational requirements. As an intermediate approach, mesoscopic modeling describes traffic facilities at a higher level of resolution than macroscopic models, but the behavior and interactions of vehicles exhibit a lower level of fidelity than in microscopic models (Hadi et al., 2022). Cross-resolution modeling, as an integrated approach, aims to fully utilize the advantages and avoid the potential limitations of each type of modeling approach with a single resolution by seamlessly modeling with various resolutions. The benefits of cross-resolution modeling have been widely recognized during its applications in traffic planning, simulation, and analysis (Zhou et al., 2021). A recent study by Zhou et al. (2022) offers a cross-resolution performance approach for connecting mesoscopic polynomial arrival queue model to macroscopic volume-delay function. Nevertheless, in the field of TSSI, estimation tasks at different levels are typically performed individually or sequentially. Individual estimations may result in inconsistencies between the different modeling levels. Under a sequential

modeling framework, results from higher modeling levels serve as the input to lower modeling levels, which could be suboptimal or even infeasible at lower levels. As a result, an iterative feedback process is typically needed for communication between different levels, but issues of convergence and long computation times still exist. Recently, researchers have started to build integrated models for systematically considering the interactions between different components in multistage problems. For example, Schöbel (2017) proposed a generic model for integrating line planning, timetabling, and vehicle scheduling for public transportation. Zhang et al. (2022) built a new model for integrating line planning and train timetabling for railway systems.

This chapter aims to provide a computationally efficient and inherently consistent model-driven cross-resolution modeling framework for TSSI, utilizing multi-source heterogeneous traffic data and advanced computational techniques from machine learning communities. The main contribution of this chapter includes:

- (1) A cross-resolution modeling framework is proposed for the TSSI problem, where the critical tasks of TSE, MPE, and QPE can be simultaneously performed. Mapping equations for traffic flow models and observations at the macroscopic, mesoscopic, and microscopic levels were constructed to produce inherently consistent and numerically reliable estimations.
- (2) By modeling the traffic system of interest as a continuous-time fluid queue system, based on the assumption of quadratic traffic arrival/discharge rates, a number of macroscopic system performance evaluation measures such as time-dependent queue length, delay, and travel time, were analytically derived for congestion profile

modeling at bottlenecks.

- (3) A new continuous space-time approximation-based traffic state representation scheme was introduced to enable a differentiable structure in traffic flow dynamics modeling and for stabilizing state estimates, especially under imperfect or limited measurements. The resulting TSSI problem was then formulated as an unconstrained nonlinear optimization model, in which a set of measurement equations could be incorporated for different data sources and traffic flow models to improve the accuracy and reliability of the estimations.
- (4) A customized computational graph representation was designed to express and solve the nonlinear optimization model, where the gradient information associated with PDEs in traffic flow models can be efficiently calculated using automatic differentiation techniques. A forward-backward algorithm-based solution method was further developed to find the solution of the optimization model implemented on the computational graph in both centralized and distributed computing environments.
- (5) Extensive numerical experiments based on real-world and hypothetical datasets were designed to evaluate the effectiveness of the proposed framework. Furthermore, the proposed framework was implemented in a distributed computing architecture to demonstrate scalability and stability in largescale instances.

4.2 Problem Statement and Overall Modeling Framework

Given a set of road segments with a time period of interest, the proposed TSSI problem aims to systematically estimate traffic states and queue profiles, and calibrate traffic flow models at various resolutions by utilizing rich observations from different types

of traffic detectors while properly handling potential inconsistencies between different components and satisfying modeling principles for representing system dynamics.

As shown in Fig. 18(a), the following four major types of sensors were considered to provide observations Y in this study:

- (1) Loop detectors for providing vehicle volumes aggregated with a certain time interval (e.g., 5 min and 15 min) at fixed locations,
- (2) GPS sensors for providing semi-continuous trajectory data with timestamps of probe vehicles,
- (3) Bluetooth sensors for providing the travel time of vehicles equipped with Bluetooth devices between adjacent sensors, and
- (4) Video detectors for providing high-fidelity vehicle trajectories within the coverage area.

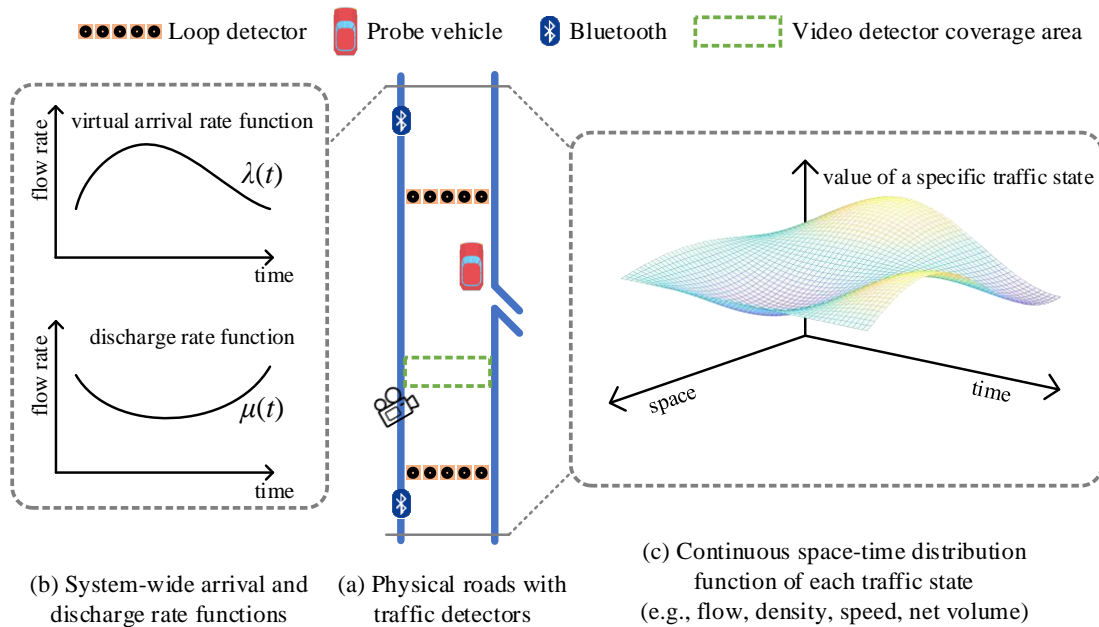


Fig. 18. Illustration of the Fluid Queue and Continuous Space-time Modeling on Road Segments with Different Types of Traffic Detectors.

Three sets of variables are to be estimated or calibrated in the proposed TSSI problem: (1) time-dependent traffic states on target road segments, (2) parameters of traffic flow models, and (3) queue evolutions in the traffic system of interest. For (2), the specific set of parameters depends on the selection of underlying traffic flow models. Traffic flow models should be carefully selected to be consistent with other modeling components. $\Pi(\phi)$ is used to denote the selected traffic flow models, where ϕ represents the model parameters to be calibrated. The variables to be estimated in (1) and (3) are illustrated below, with the introduction of the two key traffic state representation schemes adopted in this study.

Fluid queue representation at the macroscopic level. At the macroscopic level, the set of target road segments is considered as a queuing system where system states and other important measures can be modeled or derived from two fundamental states: arrival rate and service rate. In the context of traffic systems, the two fundamental states correspond to vehicle arrival rates at entrances and discharge rates at exits. As a simplified approach, the virtual arrival and discharge rates at the final downstream are also sufficient for analyzing the overall performance from a system-wide perspective. In this study, the two fundamental states are represented by two continuous functions with respect to time, that is, the virtual arrival rate function $\lambda(t)$ and the discharge rate function $\mu(t)$ [see Fig. 18(b)].

Continuous space-time representation at the mesoscopic level. At the mesoscopic level, the focus is on time-dependent traffic states, including the traffic volume, density, and speed on road segments. An important feature of the proposed framework is the

introduction of a CSTD function representation scheme for traffic states. That is, in the context of functional analysis, this research attempts to construct and calibrate a distribution function to represent each traffic state of interest over the target space-time regime [see Fig. 18(c)]. The input of each CSTD function is a space-time point (x, t) , and the output is the value of the corresponding traffic state at that point. As the traffic volume equals the product of density and speed, the CSTD functions to be estimated are associated with traffic density and speed, denoted by $f(x, t)$ and $g(x, t)$ respectively.

Table 6 summarizes the parameters and functions to be estimated for the TSSI problem. The purpose of the TSSI problem is to calibrate traffic flow models $\Pi(\phi)$ and construct two flow rate functions [i.e., $\lambda(t)$ and $\mu(t)$] and two CSTD functions [i.e., $f(x, t)$ and $g(x, t)$] such that the following inconsistencies can be minimized: (1) inconsistency between observations \mathbf{Y} and estimated states; (2) inconsistency between estimated states and underlying traffic flow models; and (3) inconsistency associated with the cross-resolution modeling structure.

Table 6 Parameters/Functions to Be Estimated in the TSSI Problem.

Parameters	
ϕ	Parameters of selected traffic flow models $\Pi(\phi)$
Functions	
$\lambda(t)$	System-wide virtual arrival rate function
$\mu(t)$	System-wide discharge rate function
$f(x, t)$	Traffic density distribution function
$g(x, t)$	Traffic speed distribution function

Fig. 19 presents the cross-resolution framework proposed in this study, with a brief introduction to each module. It should be noted that the specific traffic models and

observations in Fig. 19 are those used in this study and can be replaced with others as needed.

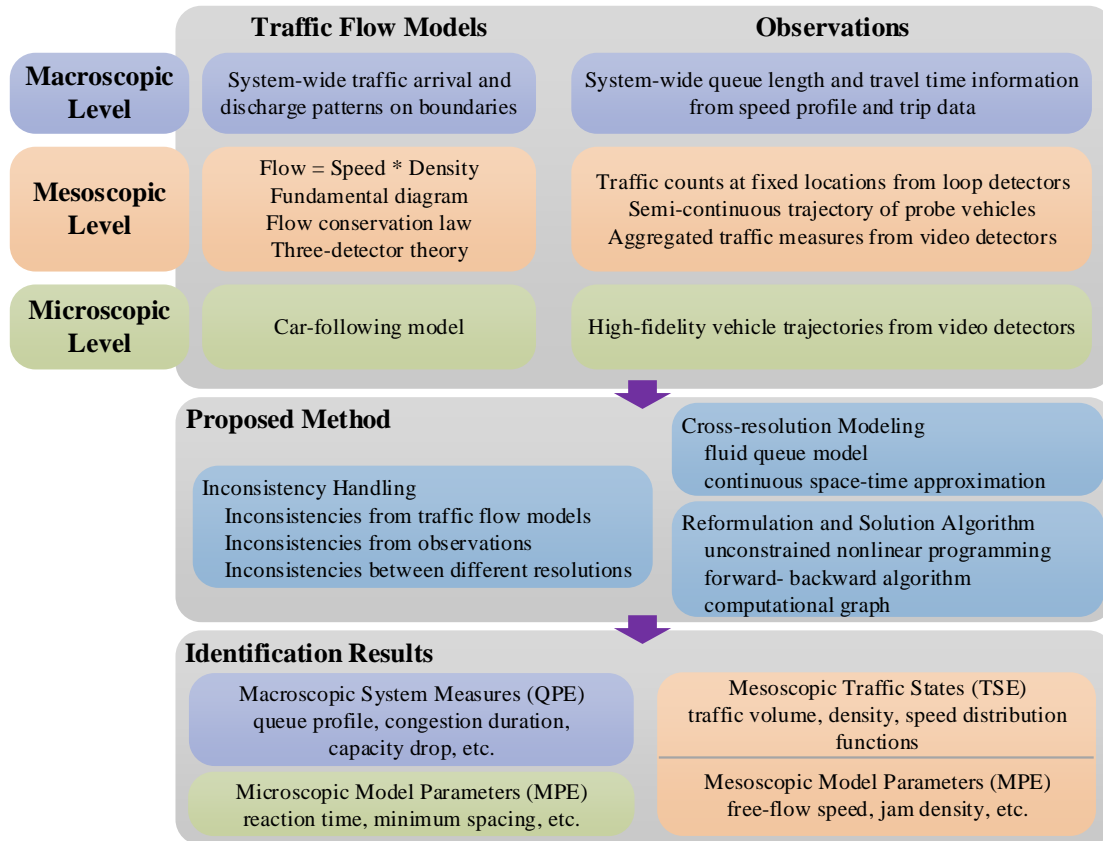


Fig. 19. Cross-resolution Framework Proposed in This Research.

4.3 Derivation of Macroscopic Traffic System Dynamics Based on Fluid Queue Models

In this section, by treating road segments or corridors with a downstream bottleneck as a single ‘server’ in the queuing system, this research extends the work by Newell (1982) and try to provide a set of analytical equations for describing traffic system dynamics, which serve as a major foundation of the subsequent macroscopic modeling.

Newell (1982) extensively investigated the application of queuing theory in traffic system dynamics modeling, analysis, and evaluation, in which useful tools such as cumulative arrival and departure diagram were introduced. With the assumption of

quadratic arrival rate and constant departure rate, Newell (1982) analytically derived critical system state (e.g., queue length, travel time, and travel delay) formulations in a closed and concise form, offering a mathematically rigorous and computationally efficient tool in real-life applications. However, owing to the simplified assumption of a constant departure rate, the derivations proposed by Newell (1982) are not able to capture the capacity (discharge rate) drop during congestion periods. This section attempts to extend Newell's work and provide a generalized modeling approach for analyzing traffic system dynamics by relaxing the assumption of constant departure rates to quadratic departure rates.

Fig. 20 presents a graphic illustration of the queuing system on road links with a downstream bottleneck. In Fig. 20(a), the blue and orange curves represent the arrival rate function $\lambda(t)$ and departure rate function $\mu(t)$ at the bottleneck, respectively, where both $\lambda(t)$ and $\mu(t)$ are approximated by quadratic functions in this study. It should be noted that as the derivations below are based on the point queue model (Vickrey, 1963), $\lambda(t)$ denotes the virtual arrival rate at the downstream bottleneck instead of the arrival rate function at the road upstream, which can be obtained by shifting the latter with link free-flow travel time t_f . t_0 , t_2 and t_3 are the times at which the queue starts to form, the queue starts to dissipate, and the queue completely dissipates, respectively, while t_1^λ and t_1^μ are the times with the highest arrival rate and lowest departure rate, respectively. Fig. 20(b) depicts the time-dependent queue length $Q(t)$, and Fig. 20(c) plots the cumulative arrival counts $A(t)$ (blue curve) and cumulative departure counts $D(t)$ (orange curve) with respect to time t . As queue starts at t_0 , the two curves in Fig. 20(c) are overlapped before t_0 . For any $t \in [t_0, t_3]$, the vertical and horizontal differences between the two curves in Fig. 20(c)

correspond to the queue length at time t [i.e., $Q(t)$] and travel delay encountered by vehicles arriving at the downstream bottleneck at time t [i.e., $w(t)$], respectively. It should be noted that the time period of interest for all derivations below is $[t_0, t_3]$, that is, $t \in [t_0, t_3]$.

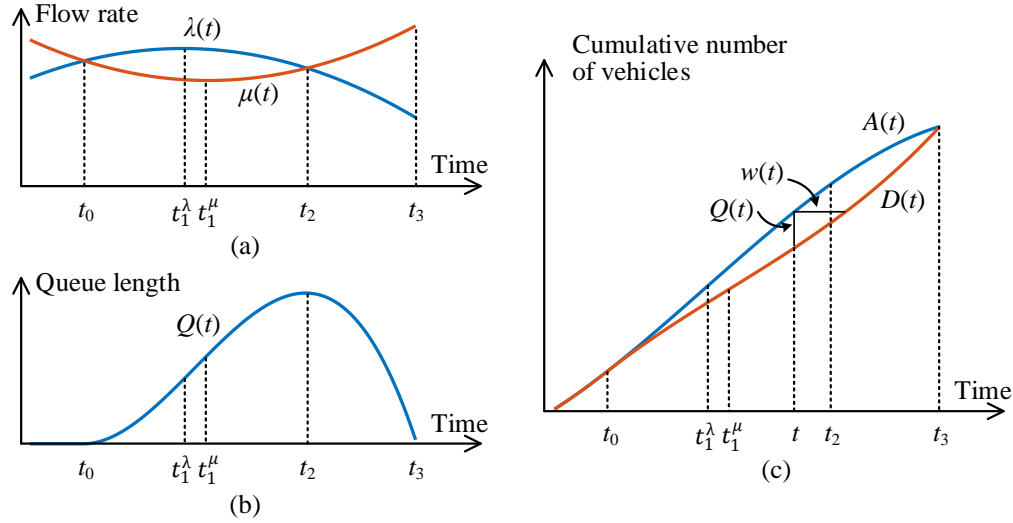


Fig. 20. Graphic Illustration of the Queuing System on Road Links with a Downstream Bottleneck.

Using the second-order Taylor approximation at time t_1^μ , the departure rate function $\mu(t)$ can be approximated by the following quadratic function:

$$\mu(t) = \mu(t_1^\mu) + \mu'(t_1^\mu)(t - t_1^\mu) + \frac{1}{2}\mu''(t_1^\mu)(t - t_1^\mu)^2. \quad (19)$$

As t_1^μ corresponds to the time at which $\mu(t)$ has the lowest value, $\mu'(t_1^\mu)$ is then equal to 0, so Eq. (19) can be simplified as

$$\mu(t) = \mu(t_1^\mu) + \gamma^\mu(t - t_1^\mu)^2, \quad (20)$$

where $\gamma^\mu = \frac{1}{2}\mu''(t_1^\mu)$. On the other hand, based on the assumption, $\lambda(t)$ and $\mu(t)$ are both approximated by quadratic functions, then net flow rate $\pi(t) = \lambda(t) - \mu(t)$ can also be

represented by a quadratic function. In addition, based on the observation that $\pi(t) = 0$ when $t = t_0$ or t_2 , the net flow function can be expressed by the factored form in Eq. (21).

$$\pi(t) = \gamma(t - t_0)(t_2 - t), \quad (21)$$

where γ is the curvature parameter. By adding Eq. (20) and Eq. (21), the arrival rate function $\lambda(t)$ can be derived as follows:

$$\lambda(t) = \mu(t) + \pi(t) = \mu(t_1^\mu) + \gamma^\mu(t - t_1^\mu)^2 + \gamma(t - t_0)(t_2 - t). \quad (22)$$

By using the factored form of the net flow function in Eq. (21), the time-dependent queue length $Q(t)$ can be derived as follows:

$$Q(t) = \int_{t_0}^t [\lambda(\tau) - \mu(\tau)] d\tau = \gamma(t - t_0)^2 \left[\frac{t_2 - t_0}{2} - \frac{t - t_0}{3} \right]. \quad (23)$$

Notice that the queue dissipates at time t_3 [i.e., $Q(t_3) = 0$], and the following relationship between critical time points can be further derived:

$$t_2 = \frac{1}{3}t_0 + \frac{2}{3}t_3. \quad (24)$$

Integrating Eqs. (23) and (24) yield a simplified expression for $Q(t)$ in Eq. (25).

$$Q(t) = \int_{t_0}^t [\lambda(\tau) - \mu(\tau)] d\tau = \frac{\gamma}{3}(t - t_0)^2(t_3 - t). \quad (25)$$

Similarly, $\pi(t)$ in Eq. (21) and $\lambda(t)$ in Eq.(22) can be rewritten as Eq. (26) and Eq. (27), respectively, to exclude the dependent parameter t_2 from the derivations.

$$\pi(t) = \gamma(t - t_0) \left(\frac{1}{3}t_0 + \frac{2}{3}t_3 - t \right), \quad (26)$$

$$\lambda(t) = \mu(t_1^\mu) + \gamma^\mu(t - t_1^\mu)^2 + \gamma(t - t_0) \left(\frac{1}{3}t_0 + \frac{2}{3}t_3 - t \right). \quad (27)$$

Next, time-dependent travel delay $w(t)$ is derived. In Newell (1982) and Cheng et al. (2022), the departure rate μ is a constant; therefore, $w(t)$ can be easily obtained by

respect to t , and one can choose an appropriate k to satisfy the approximation accuracy requirement.

$$\hat{w}^1(t) = \frac{Q(t)}{c}, \quad (29)$$

$$\hat{w}^2(t) = \frac{Q(t) - D\left(t + \frac{Q(t)}{c}\right) + D(t)}{c} + \frac{Q(t)}{c}. \quad (30)$$

$\hat{w}(t)$ denotes the final approximation of $w(t)$ using Eq. (28). In the implementation, $k = 6$, that is, $\hat{w}(t) = \hat{w}^6(t)$. With free-flow travel time t_f and travel delay $\hat{w}(t)$, time-dependent travel time $tt(t)$ can be expressed as

$$tt(t) = \hat{w}(t + t_f) + t_f. \quad (31)$$

Note that $tt(t)$ derived in Eq. (31) denotes the travel time of the vehicles entering the link upstream at time t .

Table 7 summarizes the system dynamics derivations presented in this section.

Table 7 Summary of the System States Based on Fluid Queue Model.

State	Notation	Analytical formulation
Arrival rate	$\lambda(t; \boldsymbol{\eta})$	$\lambda(t) = \mu(t_1^\mu) + \gamma^\mu(t - t_1^\mu)^2 + \gamma(t - t_0) \left(\frac{1}{3}t_0 + \frac{2}{3}t_3 - t \right)$
Discharge rate	$\mu(t; \boldsymbol{\eta})$	$\mu(t) = \mu(t_1^\mu) + \gamma^\mu(t - t_1^\mu)^2$
Net flow rate	$\pi(t; \boldsymbol{\eta})$	$\pi(t) = \gamma(t - t_0) \left(\frac{1}{3}t_0 + \frac{2}{3}t_3 - t \right)$
Time-dependent queue length	$Q(t; \boldsymbol{\eta})$	$Q(t) = \frac{\gamma}{3}(t - t_0)^2(t_3 - t)$
Time-dependent travel delay	$\hat{w}(t; \boldsymbol{\eta})$	$\hat{w}(t) = \frac{Q(t) - D\left(t + \frac{Q(t)}{c}\right) + D(t)}{c} + \frac{Q(t)}{c}$
Time-dependent travel time	$tt(t; \boldsymbol{\eta})$	$tt(t) = \frac{Q(t + t_f) - D\left(t + t_f + \frac{Q(t + t_f)}{c}\right) + D(t + t_f)}{c} + \frac{Q(t + t_f)}{c} + t_f$

Note: $w(t; \boldsymbol{\eta})$ and $tt(t; \boldsymbol{\eta})$ in the table are based on approximations of the true $w(t)$ derived in Eq. (30). For clarity of notation, a single parameter vector $\boldsymbol{\eta}$ is used in all the functions in the table, where $\boldsymbol{\eta} = [t_1^\mu \ \mu(t_1^\mu) \ \gamma^\mu \ \gamma \ t_0 \ t_3 \ t_f]^T$.

4.4 Formulating Cross-resolution Traffic System State Identification Problem as a Nonlinear Programming Model

Notations used in this section are presented in Table 8.

Parameters	
$l(\delta)$	Physical distance (time interval) between two adjacent space-time sample points
$n_x(n_t)$	Number of sample space (time) points
n_{xt}	Number of sample space-time points
$\mathbf{x}(\mathbf{t})$	Vector of sample space (time) points
\mathbf{xt}	Vector of sample space-time points
Data	
\mathbf{Y}	Observations from multi-source traffic detectors.
Variables	
Z	Total inconsistency from cross-resolution traffic flow models and observations
$\boldsymbol{\theta}$	Parameters of the density distribution function $f(x, t; \boldsymbol{\theta})$
$\boldsymbol{\varphi}$	Parameters of the speed distribution function $g(x, t; \boldsymbol{\varphi})$
$\boldsymbol{\eta}$	Parameters of system-wide measure functions in Table 7
$\boldsymbol{\phi}$	Estimated parameters of traffic flow models $\boldsymbol{\Pi}(\boldsymbol{\phi})$
\mathbf{k}	Estimated density on space-time points \mathbf{xt}
\mathbf{v}	Estimated speed on space-time points \mathbf{xt}
\mathbf{q}	Estimated volume on space-time points \mathbf{xt}
$\boldsymbol{\varepsilon}_m^{MA}(\boldsymbol{\varepsilon}_o^{MA})$	Inconsistency vector from macroscopic traffic flow models (observations)
$\boldsymbol{\varepsilon}_m^{ME}(\boldsymbol{\varepsilon}_o^{ME})$	Inconsistency vector from mesoscopic traffic flow models (observations)
$\boldsymbol{\varepsilon}_m^{MI}(\boldsymbol{\varepsilon}_o^{MI})$	Inconsistency vector from microscopic traffic flow models (observations)
Functions	
$\lambda(t; \boldsymbol{\eta})$	System-wide arrival rate function with parameter $\boldsymbol{\eta}$
$\mu(t; \boldsymbol{\eta})$	System-wide discharge rate function with parameter $\boldsymbol{\eta}$
$f(x, t; \boldsymbol{\theta})$	Density distribution function with parameter $\boldsymbol{\theta}$
$g(x, t; \boldsymbol{\varphi})$	Speed distribution function with parameter $\boldsymbol{\varphi}$
$h_x(x, t; \boldsymbol{\theta}, \boldsymbol{\varphi})$	Partial differential function of traffic volume with respect to location
$f_t(x, t; \boldsymbol{\theta})$	Partial differential function of traffic density with respect to time

As introduced in Section 2, the purpose of the TSSI problem is to simultaneously perform TSE, MPE, and QPE based on multi-source observations while minimizing the total inconsistency among the different components. Compared to existing studies, one of

the most important features of the proposed framework is the introduction of CSTD functions to represent fundamental traffic states. As a result, before proceeding to model construction, it is necessary to clarify the potential challenges and benefits associated with the CSTD representation from a modeling perspective.

(1) How to choose a proper form for each CSTD function to be calibrated?

As will be introduced in Section 4.6, in this study, a widely accepted functional form with the universal approximation property is adopted for CSTD functions. However, identification of a specific functional form requires a more comprehensive study, which is beyond the scope of this study. Without loss of generality, in the following discussions, it is assumed that the forms of the CSTD functions are given or have been well calibrated in advance. In this case, the parameters associated with the CSTD functions become variables to be estimated, that is, θ in function $f(x, t; \theta)$ and φ in function $g(x, t; \varphi)$.

(2) How to measure inconsistencies on CSTD functions?

Instead of directly describing inconsistencies associated with different functions, a set of sample points from the space-time plane of interest is selected and then accordingly, measure inconsistencies on sample points. Fig. 22 shows the scheme of the space-time point sampling used in this study. In the adopted sampling scheme, points are evenly selected from the space-time plane under consideration such that two adjacent points have a constant physical distance l or time interval δ in between. For a space-time plane with physical length L and time duration S , the set of sample space-time points is expressed as $XT = \{(x, t) | x \in X, t \in T\}$, where $X = \{x | \text{mod}(x, l) = 0, 0 \leq x \leq L\}$, $T =$

$\{t|\text{mod}(t, \delta) = 0, 0 \leq t \leq S\}$, $|XT| = n_{xt}$, $|X| = n_x$, and $|T| = n_t$ ($n_{xt} = n_x \times n_t$). Vector \mathbf{x} , \mathbf{t} , and \mathbf{xt} , with the shapes of $(n_x, 1)$, $(n_t, 1)$, and $(n_{xt}, 2)$, denote the vectorizations of sets X , T , and XT , respectively. For example, for a space-time plane with a shape of 500 m by 30 min, let $l = 10$ m and $\delta = 5$ s. Then, $X = \{x|\text{mod}(x, 10) = 0, 0 \leq x \leq 500\}$, and $T = \{t|\text{mod}(t, 5) = 0, 0 \leq t \leq 1800\}$. Accordingly, \mathbf{x} , \mathbf{t} , and \mathbf{xt} can be written as:

$$\mathbf{x} = [0 \ 10 \ 20 \ \dots \ 490 \ 500]^T, \mathbf{t} = [0 \ 5 \ 10 \ \dots \ 1795 \ 1800]^T,$$

$$\mathbf{xt} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 10 & 10 & \dots & 500 \\ 0 & 5 & 10 & \dots & 1800 & 0 & 5 & \dots & 1800 \end{bmatrix}^T.$$

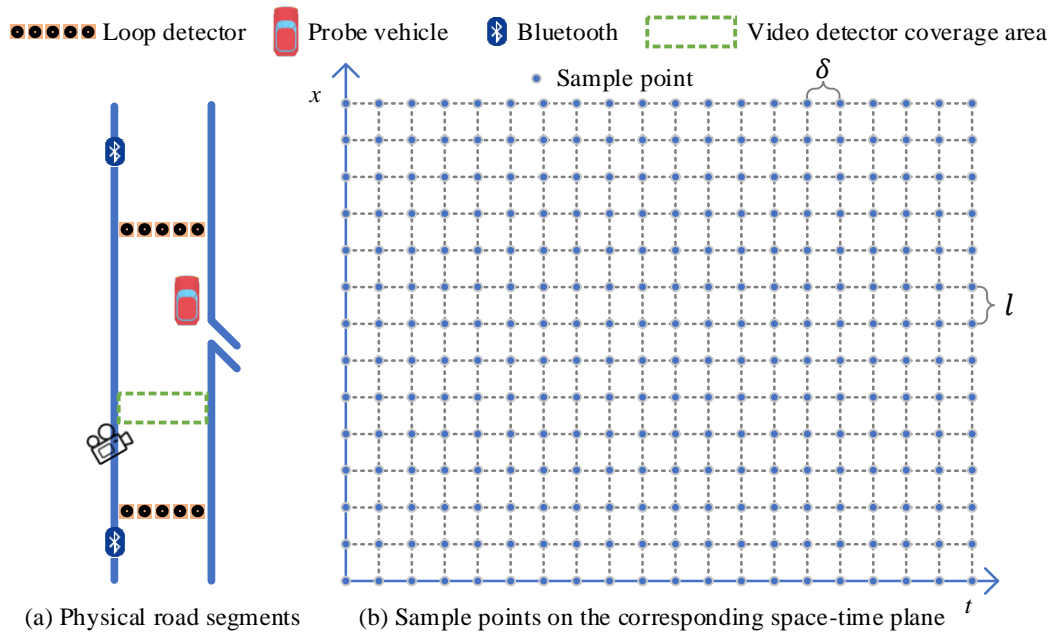


Fig. 22. Physical Road Segments and Sampling Points on the Corresponding Space-time Plane.

(3) How to handle PDEs in traffic flow models?

PDEs in traffic flow models are used to describe traffic flow dynamics and measure the evolution of traffic states in spatial and temporal domains. Eq. (32) presents the widely used first-order flow conservation law:

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = u_{x,t}, \quad (32)$$

where q and k denote traffic flow and density, respectively, and $u_{x,t}$ is the traffic flow generation rate at location x and time t . Because analytically solving PDEs is extremely difficult or even impossible in most cases, in the literature, most studies adopt a discretized scheme and approximate PDEs with a set of linear functions. Benefitting from the functional traffic state representation scheme proposed in this research, modeling PDEs is very straightforward. Because the functional forms of $f(x, t; \boldsymbol{\theta})$ and $g(x, t; \boldsymbol{\varphi})$ are known [see the discussion for Question (1)], $h_x(x, t; \boldsymbol{\theta}, \boldsymbol{\varphi}) = \frac{\partial(f(x,t;\boldsymbol{\theta}) \cdot g(x,t;\boldsymbol{\varphi}))}{\partial x}$ and $f_t(x, t; \boldsymbol{\theta}) = \frac{\partial(f(x,t;\boldsymbol{\theta}))}{\partial t}$ can also be analytically expressed. Hence, Eq. (32) can be converted to the normal equation in Eq. (33) without partial differential terms.

$$h_x(x, t; \boldsymbol{\theta}, \boldsymbol{\varphi}) + f_t(x, t; \boldsymbol{\theta}) = u_{x,t}. \quad (33)$$

With the three key questions addressed above, a nonlinear programming model M1 for the TSSI problem is presented below.

Model M1:

Objective function

$$\min Z = U(\boldsymbol{\varepsilon}_m^{MA}, \boldsymbol{\varepsilon}_o^{MA}, \boldsymbol{\varepsilon}_m^{ME}, \boldsymbol{\varepsilon}_o^{ME}, \boldsymbol{\varepsilon}_m^{MI}, \boldsymbol{\varepsilon}_o^{MI}). \quad (34)$$

Subject to

Estimation of the traffic density on sample points

$$\mathbf{k} = f(\mathbf{x}t; \boldsymbol{\theta}). \quad (35)$$

Estimation of the traffic speed on sample points

$$\mathbf{v} = g(\mathbf{x}t; \boldsymbol{\varphi}). \quad (36)$$

Estimation of the traffic volume on sample points

$$\mathbf{q} = \mathbf{k} \cdot \mathbf{v}. \quad (37)$$

Estimation of the derivative of traffic volume with respect to space on sample points

$$\mathbf{q}_x = h_x(\mathbf{x}t; \boldsymbol{\theta}, \boldsymbol{\varphi}). \quad (38)$$

Estimation of the derivative of traffic density with respect to time on sample points

$$\mathbf{k}_t = f_t(\mathbf{x}t; \boldsymbol{\theta}). \quad (39)$$

Inconsistency between traffic state estimates and macroscopic traffic flow models

$$\boldsymbol{\varepsilon}_m^{MA} = H_m^{MA}(\mathbf{q}, \mathbf{k}, \mathbf{v}, \lambda(t; \boldsymbol{\eta}), \mu(t; \boldsymbol{\eta})). \quad (40)$$

Inconsistency between traffic state estimates and macroscopic observations

$$\boldsymbol{\varepsilon}_o^{MA} = H_o^{MA}(Q(t; \boldsymbol{\eta}), tt(t; \boldsymbol{\eta}), \mathbf{Y}). \quad (41)$$

Inconsistency between traffic state estimates and mesoscopic traffic flow models

$$\boldsymbol{\varepsilon}_m^{ME} = H_m^{ME}(\mathbf{q}, \mathbf{k}, \mathbf{v}, \mathbf{q}_x, \mathbf{k}_t, \Pi(\boldsymbol{\phi})). \quad (42)$$

Inconsistency between traffic state estimates and mesoscopic observations

$$\boldsymbol{\varepsilon}_o^{ME} = H_o^{ME}(\mathbf{q}, \mathbf{k}, \mathbf{v}, \mathbf{Y}). \quad (43)$$

Inconsistency between traffic state estimates and microscopic traffic flow models

$$\boldsymbol{\varepsilon}_m^{MI} = H_m^{MI}(\mathbf{q}, \mathbf{k}, \mathbf{v}, \Pi(\boldsymbol{\phi})). \quad (44)$$

Inconsistency between traffic state estimates and microscopic observations

$$\boldsymbol{\varepsilon}_o^{MI} = H_o^{MI}(\mathbf{q}, \mathbf{k}, \mathbf{v}, \mathbf{Y}). \quad (45)$$

Decision variables

$$\boldsymbol{\theta} \in \mathbf{R}^{n_\theta}, \boldsymbol{\varphi} \in \mathbf{R}^{n_\varphi}, \boldsymbol{\eta} \in \mathbf{R}^{n_\eta}, \boldsymbol{\phi} \in \mathbf{R}^{n_\phi}. \quad (46)$$

In model M1, the objective function in Eq. (34) minimizes the overall inconsistency from both traffic flow models and observations at the three resolutions. Eqs. (35)-(37) derive the traffic state estimations on space-time points $\mathbf{x}\mathbf{t}$. Specifically, $\mathbf{k} = f(\mathbf{x}\mathbf{t}; \boldsymbol{\theta})$ in Eq. (35) is a vectorization of $k = f(x, t; \boldsymbol{\theta})$, where the input $\mathbf{x}\mathbf{t}$ is a set of sample space-time points (x, t) in vector form and the output vector \mathbf{k} represents the density estimations on $\mathbf{x}\mathbf{t}$. A similar fashion applies for the speed derivation presented in Eq. (36). Operator ‘ \cdot ’ in Eq. (37) denotes element-wise multiplication. Eqs. (38) and (39) represent the derivatives of the traffic states with respect to space and time on $\mathbf{x}\mathbf{t}$. Eqs. (40)-(45) present conceptual mapping functions for calculating the inconsistency terms. It should be noted that for simplicity, it is assumed that the inconsistency terms behave well and are mutually uncorrelated. Thus, a simplified form of the ordinary least square can be used in the function U in Eq. (34). Interested readers are referred to Deng et al. (2013) for further discussions on the possible error correlation and uncertainty propagation along this line. Finally, constraint (46) specifies the independent decision variables of model M1. Model M1 is essentially an unconstrained optimization model, leading to potentially efficient implementations in largescale real-life applications.

4.5 Modeling Inconsistencies Across Different Resolutions

Section 4.4 describes the basic structure of nonlinear optimization model M1. This section focuses on the modeling of inconsistency terms in model M1 from three different resolutions (macroscopic, mesoscopic, and microscopic), yielding a complete and practical

mathematical optimization model. For each resolution, inconsistencies from both traffic flow models and observations are discussed.

4.5.1 Macroscopic-level Modeling

At the macroscopic level, modeling the set of road segments of interest as a queuing system provides system-wide measures, contributing to easing the impact of detection errors and overfitting at local levels. By integrating the methodology proposed in Section 3, this section demonstrates how to measure the inconsistencies from a macroscopic perspective by utilizing information from both models and observations.

Traffic flow models

At the macroscopic level, the set of road segments of interest is modeled as a fluid queue with quadratic arrival/discharge rates. As mentioned previously, the arrival rate is the virtual arrival rate at the final downstream, and it is difficult to build physical mappings in cases with multiple entrances and exits. Therefore, only the quadratic discharge rate is utilized to regulate the volume estimations at downstream, as presented in Eq. (47).

$$\boldsymbol{\varepsilon}_m^{MA-dr} = \boldsymbol{\mu}(\mathbf{t}; \boldsymbol{\eta}) - \mathbf{W}_{dr} \mathbf{q}, \quad (47)$$

where $\boldsymbol{\mu}(\mathbf{t}; \boldsymbol{\eta})$ and $\mathbf{W}_{dr} \mathbf{q}$ represent the flow rate estimations at downstream sample points from the macroscopic and mesoscopic levels, respectively; \mathbf{W}_{dr} is a mapping matrix with a shape of (n_t, n_{xt}) . Matrix \mathbf{W}_{dr} is built from an identity matrix with a shape of (n_{xt}, n_{xt}) ; then, only rows that correspond to space-time points at the downstream boundary are kept, while other rows are removed.

Observations

System-wide travel time data can be collected from Bluetooth devices and probe vehicles. Let \tilde{tt} denote the travel time observations at t' , where \tilde{tt} and t' are vectors with a shape of $(n_{t'}, 1)$; and $n_{t'}$ represents the number of travel time records. Then, the inconsistency associated with the system-wide time-dependent travel time can be expressed as:

$$\boldsymbol{\varepsilon}_o^{MA-tt} = \tilde{tt} - tt(t'; \boldsymbol{\eta}), \quad (48)$$

where tt represents the analytical time-dependent travel time function of the queuing system derived in Section 3. The inconsistency associated with the time-dependent queue length can also be calculated using the analytical $Q(t)$ in Eq. (25). The observed queue length can be calibrated with a set of congestion and bottleneck identification tools, for example, the CBI tool (FHWA, 2018), from traffic speed observations. It is noteworthy that $Q(t)$ in Eq. (25) is the number of vehicles in the system rather than the physical queue length. Interested readers can refer to Lawson et al. (1997) and Cheng et al. (2022) for the conversion between these two measures.

4.5.2 Mesoscopic-level Modeling

This section focuses on the inconsistency modeling of travel flow models and observations at the mesoscopic level. Specifically, the travel flow models considered in this study consist of fundamental diagram, flow conservation law, and three-detector model. One may notice that the first two models together with $q = kv$ [which has already been considered in Eq.(37)] constitutes the LWR model. It should be noted that although

the proposed framework has no limitations on traffic flow model selection, theoretically compatible models should be carefully selected to avoid inherent inconsistencies.

Traffic flow models

(1) Fundamental diagram

A fundamental diagram describes the equilibrium relationship between traffic flow volume and density. This study uses the triangular fundamental diagram as an example to illustrate how to utilize analytical fundamental diagram models to regulate state estimations. Eq. (49) presents the volume-density (q - k) relationship of the triangular fundamental diagram, where v_f , w_b and k_j denote the free-flow speed, backward wave speed, and jam density, respectively. Accordingly, the inconsistency associated with the fundamental diagram can be calculated using Eq. (50).

$$q = \min[v_f k, -w_b(k - k_j)], \quad (49)$$

$$\boldsymbol{\varepsilon}_m^{ME-fd} = \min[v_f \mathbf{k}, -w_b(\mathbf{k} - k_j \mathbf{1}_{n_{xt}})] - \mathbf{q}, \quad (50)$$

where $\mathbf{1}_{n_{xt}}$ denotes an all-one vector with the shape of $(n_{xt}, 1)$.

(2) Flow conservation law

With the discussion of PDEs handling in Section 4.4, the inconsistency associated with the flow conservation law is calculated in Eq. (51). Note that $\mathbf{q}_x - \mathbf{k}_t$ denotes the traffic flow generation rates at all sample space-time points $\mathbf{x}\mathbf{t}$, while generation rates at space-time points that correspond to physical entrances and exits are not necessarily zero.

\mathbf{W}_{FC} is a mapping matrix built from an identity matrix with a shape of (n_{xt}, n_{xt}) , with

elements in rows that correspond to space-time points having non-zero traffic flow generation rates set as zero.

$$\boldsymbol{\varepsilon}_m^{ME-fc} = \mathbf{W}_{FC}(\mathbf{q}_x - \mathbf{k}_t). \quad (51)$$

(3) Three-detector model

According to Newell's three-detector model (Newell, 1993), the traffic state on a point is governed by one of the two waves, that is, a forward wave with free flow speed v_f from the upstream and a backward wave with backward wave speed w_b from the downstream. In the following discussion, traffic density is used as the state of interest to illustrate the modeling process, while volume and speed can be modeled in the same manner. As shown in Fig. 23, the density at point $A(x, t)$ should be the same as the density along the forward wave (green line) or the density along the backward wave (red line).

Choose one point from each wave, say, point B from the forward wave and point C from the backward wave. Then, the density at point A is either close to the density at point B or the density at point C . As a result, Eq. (52) can be adopted to calculate the inconsistency associated with the three-detector theory at point A :

$$\varepsilon_m^{ME-td} = \min[(k_A - k_B)^2, (k_A - k_C)^2], \quad (52)$$

where k_A , k_B , and k_C represent the densities at points A , B , and C , respectively. Note that points B and C may not belong to the sample points in $\mathbf{x}\mathbf{t}$, but their states can be linearly expressed using interpolation methods. To simplify the calculation process, points B and C are selected on the grid in this study. Eqs. (53) and (54) present the formulations for calculating k_B and k_C from the density vector \mathbf{k} :

$$k_B = \frac{l/v_f}{\delta} k_{B_1} + \frac{\delta - l/v_f}{\delta} k_{B_2} = \mathbf{w}_B^T \mathbf{k}, \quad (53)$$

$$k_C = \frac{l/w_b}{\delta} k_{C_1} + \frac{\delta - l/w_b}{\delta} k_{C_2} = \mathbf{w}_C^T \mathbf{k}, \quad (54)$$

where \mathbf{w}_B and \mathbf{w}_C are two weight vectors with the shape of $(n_{xt}, 1)$. Substituting k_B and k_C into Eq. (52) using Eqs. (53) and (54), Eq. (52) can be rewritten as follows:

$$\varepsilon_m^{ME-td} = \min[(\mathbf{w}_A^T \mathbf{k} - \mathbf{w}_B^T \mathbf{k})^2, (\mathbf{w}_A^T \mathbf{k} - \mathbf{w}_C^T \mathbf{k})^2], \quad (55)$$

where \mathbf{w}_A is a weight vector built from an all-zero vector with a shape of $(n_{xt}, 1)$ while the weight of point A is set as 1.

Eq. (56) represents the inconsistency associated with the three-detector model for all the sample points $\mathbf{x}t$:

$$\varepsilon_m^{ME-td} = \min[(\mathbf{I}\mathbf{k} - \mathbf{W}_F \mathbf{k})^2, (\mathbf{I}\mathbf{k} - \mathbf{W}_B \mathbf{k})^2], \quad (56)$$

where each row is constructed for a specific sample point in $\mathbf{x}t$ using Eq. (55).

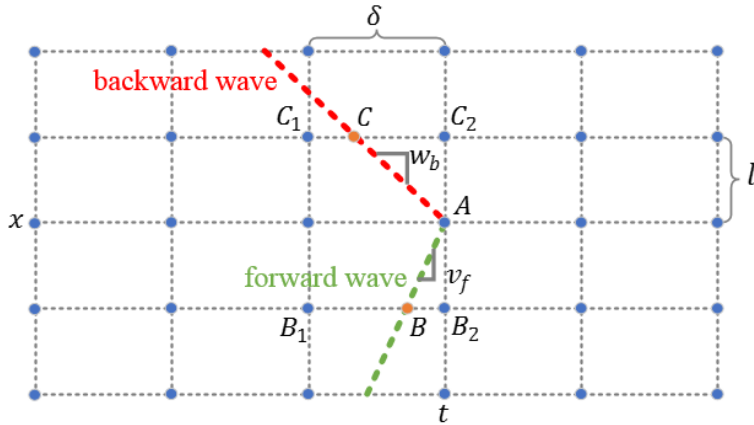


Fig. 23. Illustration of the Three-detector Model.

Observations

(1) Using loop detector data

Loop detectors provide aggregated traffic counts within a certain time interval at fixed locations. As illustrated in Fig. 24, for each traffic count record \tilde{c}_i collected from time

t_s to t_e at location x_l , its corresponding estimated value can always be expressed in the form of $\mathbf{w}_i^T \mathbf{q}$, where \mathbf{w}_i is a weight vector with the shape of $(n_{xt}, 1)$. Accordingly, the inconsistency associated with all the records from the loop detector data $\tilde{\mathbf{c}}$ can be expressed as

$$\boldsymbol{\varepsilon}_o^{ME-l} = \tilde{\mathbf{c}} - \mathbf{W}_L \mathbf{q}, \quad (57)$$

where \mathbf{W}_L is a mapping matrix with a shape of (m_L, n_{xt}) , and m_L is the number of traffic count records from the loop detectors. Each row in \mathbf{W}_L is a weight vector for the corresponding traffic count record, such as \mathbf{w}_i derived in Fig. 24.

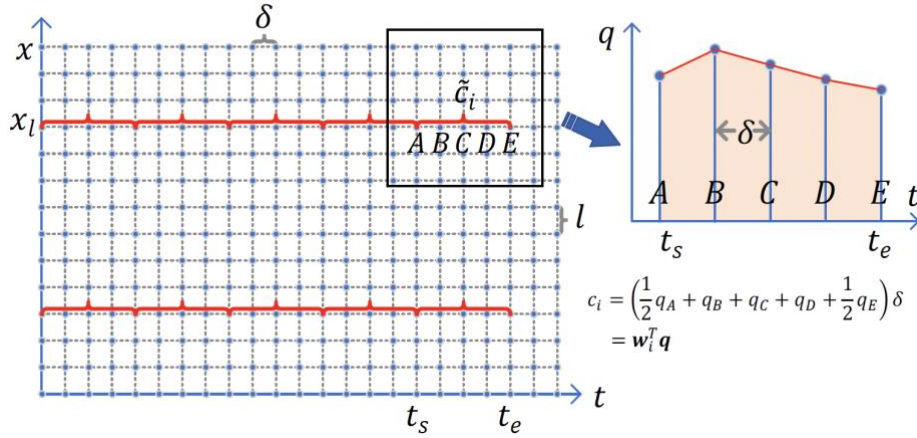


Fig. 24. Illustrations of Using Loop Detector Data.

(2) Using GPS data

Probe vehicles equipped with GPS report their locations and instant speeds within a short time interval (e.g., every 10 s). Tuple (t_i, x_i, s_i) represents one record from a GPS-equipped vehicle, where t_i denotes the timestamp, and x_i and s_i represent the location and speed of the vehicle at time t_i . As illustrated in Fig. 25, for each record (t_i, x_i, s_i) , the corresponding speed estimation can always be expressed in the form of $\mathbf{w}_i^T \mathbf{v}$, where \mathbf{w}_i is

a weight vector with a shape of $(n_{xt}, 1)$. Accordingly, the inconsistency associated with all the records from the GPS speed data $\tilde{\mathbf{s}}$ can be expressed as

$$\boldsymbol{\varepsilon}_o^{ME-gs} = \tilde{\mathbf{s}} - \mathbf{W}_G \mathbf{v}, \quad (58)$$

where \mathbf{W}_G is a mapping matrix with the shape of (m_G, n_{xt}) , and m_G is the number of GPS point records. Each row in \mathbf{W}_G is a weight vector for the corresponding GPS point record, such as \mathbf{w}_i derived in Fig. 25.

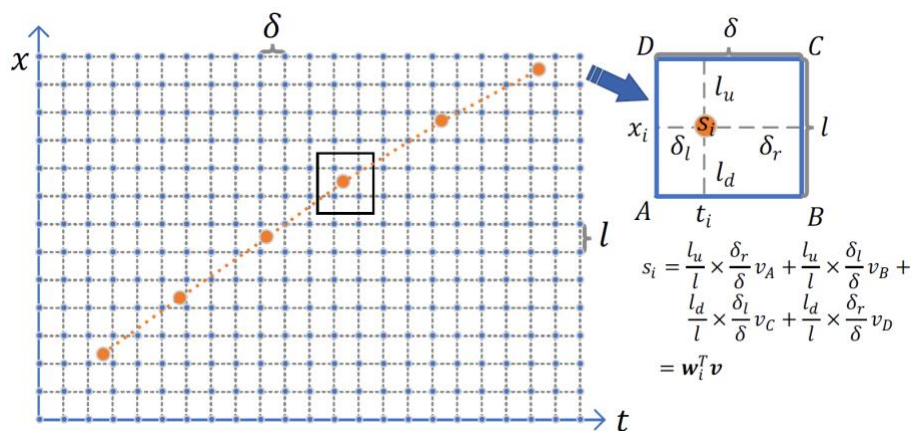


Fig. 25. Illustrations of Using Speed Information from GPS Data.

In addition to speed information, GPS data also provide space-time trajectory information of probe vehicles. As shown in Fig. 26, points $A(x_A, t_A)$ and $B(x_B, t_B)$ are two adjacent points in a trajectory point sequence, indicating that the probe vehicle entered segment $[x_A, x_B]$ at time t_A and left it at time t_B . Under the assumption of first-in-first-out (FIFO), vehicles in segment $[x_A, x_B]$ at time t_A all leave the segment within the time window $[t_A, t_B]$. In other words, the number of vehicles in segment $[x_A, x_B]$ at time t_A (denoted by z_k) is equal to the number of vehicles passing location x_B within time window $[t_A, t_B]$ (denoted by z_q). Note that the flow conservation approximation is valid only if there is no entrance or exit between points A and B . To facilitate the derivation of z_k and

z_q , point $C(x_C, t_C)$ is constructed, where $x_C = x_B$ and $t_C = t_A$. Subsequently, z_k is calculated as follows:

$$z_k = \bar{k}_{AC} \times (x_B - x_A) = \mathbf{w}'_k \mathbf{k} \times (x_B - x_A) = \mathbf{w}_k^T \mathbf{k}, \quad (59)$$

where \bar{k}_{AC} denotes the average density of the segment $[x_A, x_B]$ at time t_A , which can be expressed by a weighted average of densities on $\mathbf{x}t$, i.e., $\mathbf{w}'_k \mathbf{k}$. Similarly, z_q can be calculated using Eq. (60).

$$z_q = \bar{q}_{CB} \times (t_C - t_B) = \mathbf{w}'_q \mathbf{q} \times (t_C - t_B) = \mathbf{w}_q^T \mathbf{q}. \quad (60)$$

Then, the inconsistency associated with the FIFO assumption on points $A(x_A, t_A)$ and $B(x_B, t_B)$ can be expressed as

$$\varepsilon_o^{ME-gf} = z_k - z_q = \mathbf{w}_k^T \mathbf{k} - \mathbf{w}_q^T \mathbf{q}. \quad (61)$$

Finally, the inconsistency associated with the FIFO assumption on all pairs of adjacent points from the GPS data can be calculated as:

$$\boldsymbol{\varepsilon}_o^{ME-gf} = \mathbf{W}_{Gk} \mathbf{k} - \mathbf{W}_{Gq} \mathbf{q}, \quad (62)$$

where \mathbf{W}_{Gk} and \mathbf{W}_{Gq} are two weight matrices with the shape of (n_{gf}, n_{xt}) , and n_{gf} is the number of adjacent point pairs in GPS data. Each row in Eq. (62) represents the inconsistency measure for a specific pair of points, similar to Eq. (61).

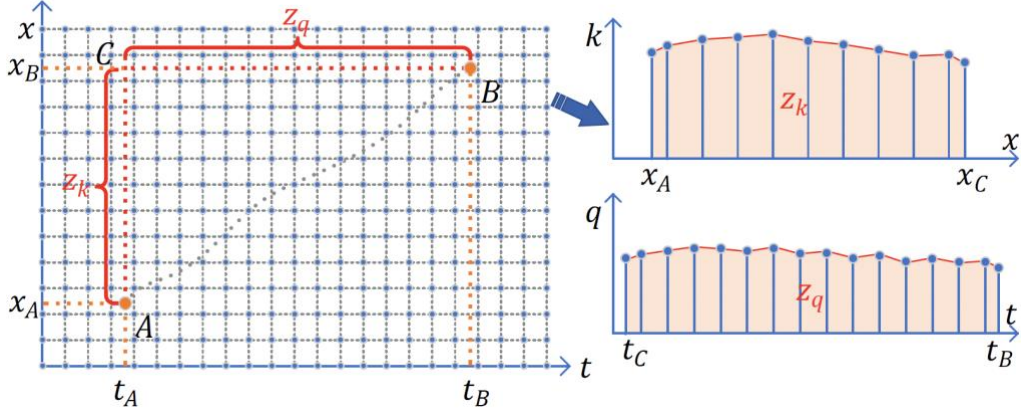


Fig. 26. Illustrations of Using Space-time Trajectory Information from GPS Data.

(3) Using video detector data

With computer vision technologies, high-fidelity vehicle trajectories and aggregated traffic measures (e.g., volume, density, and speed) can be obtained from video detector data (Coifman and Li, 2022). Aggregated traffic density data is used for illustrating inconsistency measurement, while speed and volume data can be processed similarly. As shown in Fig. 27, the video detection area is divided into multiple rectangles with evenly distributed sample points $\mathbf{x}\mathbf{t}$. In the example of the rectangle with a red boundary, \tilde{r}_i denotes the observed density in the rectangle. Using a linear interpolation method, the corresponding estimated density can be represented as the weighted sum of the density estimations at the four corners. Therefore, the inconsistency associated with the traffic density in that rectangle can be expressed as

$$\varepsilon_o^{ME-vd} = \tilde{r}_i - \mathbf{w}_i^T \mathbf{k}, \quad (63)$$

where \mathbf{w}_i is a weight vector with the shape of $(n_{xt}, 1)$. Accordingly, the inconsistency associated with all the aggregated traffic density records $\tilde{\mathbf{r}}$ can be calculated as follows:

$$\varepsilon_o^{ME-vd} = \tilde{\mathbf{r}} - \mathbf{W}_v \mathbf{k}, \quad (64)$$

where \mathbf{W}_V is a mapping matrix with the shape of (m_V, n_{xt}) , and m_V is the number of density data records from the video detectors. Each row in \mathbf{W}_V is a weight vector that corresponds to a specific density record.

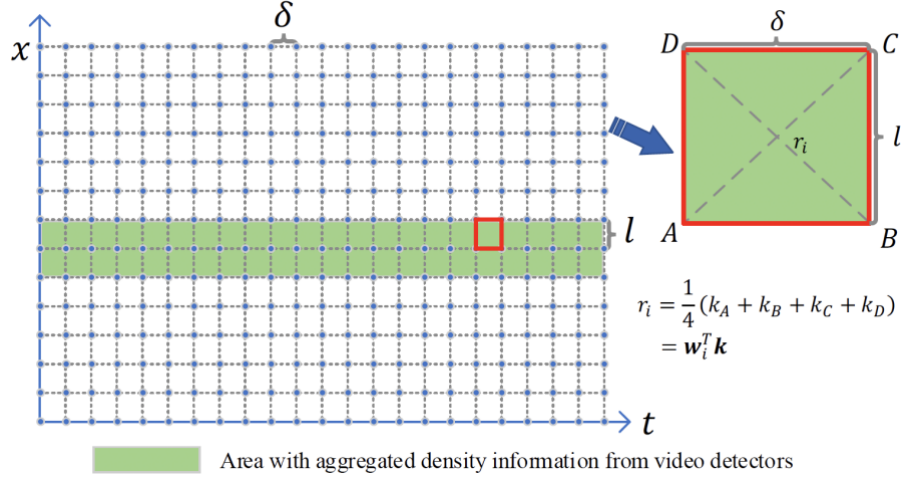


Fig. 27. Illustrations of Using Aggregated Density Information from Video Detector Data.

4.5.3 Microscopic-level Modeling

This section discusses inconsistency modeling at the microscopic level, with a focus on car-following models and observations. The car-following model used in this study is Newell's simplified linear car-following model (Newell, 2002), with a concise form presented in Eq. (65).

$$x_n(t + \tau) = x_{n-1}(t) - d, \quad (65)$$

where $x_n(t)$ represents the position of vehicle n at time t , τ denotes the reaction time of drivers, and d is the minimum safety spacing between two adjacent vehicles. Note that τ and d in Eq. (65) should be vehicle-dependent, whereas in this study, it is assumed that vehicle groups have similar deriving behaviors and share the same set of parameters.

Traffic flow models

Newell's simplified car-following model used in this study is consistent with the triangular fundamental diagram adopted in mesoscopic-level modeling: the minimum safety spacing d is the inverse of the jam density k_j , and the minimum safety spacing d divided by the reaction time τ equals the backward wave speed w_b . Consequently, the potential inconsistency associated with these two layers of traffic flow model can be expressed as

$$\boldsymbol{\varepsilon}_m^{MI-cf} = \left[d - \frac{1}{k_j} \quad \frac{d}{\tau} - w_b \right]^T, \quad (66)$$

where $\boldsymbol{\varepsilon}_m^{MI-cf}$ is the inconsistency vector with a shape of (2,1).

Observations

Reaction time τ and minimum safety spacing d are the only two parameters of Newell's simplified linear car-following models, and there have been extensive studies on parameter calibration in the literature. This study adopts the approach proposed by Taylor et al. (2015), in which the dynamic time warping algorithm is utilized to perform a point-to-point match on two adjacent trajectories, and τ and d are in turn calibrated. Let $\tilde{\tau}$ and \tilde{d} denote the calibrated reaction time and minimum safety spacing from the trajectory dataset. The inconsistency associated with the car-following model parameters can be calculated as

$$\boldsymbol{\varepsilon}_o^{MI-cf} = [\tilde{\tau} - \tau \quad \tilde{d} - d]^T, \quad (67)$$

where $\boldsymbol{\varepsilon}_o^{MI-cf}$ is the inconsistency vector with a shape of (2,1).

4.5.4 The Complete Model

With the discussion on inconsistency modeling across the three different resolutions, the complete model is presented as follows.

Model M2:

Objective function

$$\begin{aligned}
 \min Z = & \alpha_m^{MA-dr} \|\boldsymbol{\varepsilon}_m^{MA-dr}\|^2 + \alpha_o^{MA-tt} \|\boldsymbol{\varepsilon}_o^{MA-tt}\|^2 + \alpha_m^{ME-fd} \|\boldsymbol{\varepsilon}_m^{ME-fd}\|^2 \\
 & + \alpha_m^{ME-fc} \|\boldsymbol{\varepsilon}_m^{ME-fc}\|^2 + \alpha_m^{ME-td} \|\boldsymbol{\varepsilon}_m^{ME-td}\|^2 \\
 & + \alpha_o^{ME-l} \|\boldsymbol{\varepsilon}_o^{ME-l}\|^2 + \alpha_o^{ME-gs} \|\boldsymbol{\varepsilon}_o^{ME-gs}\|^2 \\
 & + \alpha_o^{ME-gf} \|\boldsymbol{\varepsilon}_o^{ME-gf}\|^2 + \alpha_o^{ME-vd} \|\boldsymbol{\varepsilon}_o^{ME-vd}\|^2 \\
 & + \alpha_m^{MI-cf} \|\boldsymbol{\varepsilon}_m^{MI-cf}\|^2 + \alpha_o^{MI-cf} \|\boldsymbol{\varepsilon}_o^{MI-cf}\|^2,
 \end{aligned} \tag{68}$$

Subject to

State estimations on sample points: (35)-(39),

Consistency constraints at the macroscopic level: (47) and (48),

Consistency constraints at the macroscopic level: (50), (51), (56), (57), (58), (62)

and (64),

Consistency constraints at the macroscopic level: (66) and (67),

Decision variables

$$\boldsymbol{\theta} \in \mathbf{R}^{n_\theta}, \boldsymbol{\varphi} \in \mathbf{R}^{n_\varphi}, \boldsymbol{\eta} \in \mathbf{R}^{n_\eta}, \boldsymbol{\phi} \in \mathbf{R}^{n_\phi}. \tag{69}$$

The objective function in Eq. (68) minimizes the weighted sum of all the inconsistency terms. The weight of a specific inconsistency term can be determined based

on the reliability of the corresponding traffic flow models or observations. Constraint (69) specifies all the independent decision variables, including θ , φ , η and ϕ . Specifically, according to the traffic flow models used in this study, $\phi = [\tau \quad d \quad v_f \quad k_j \quad w_b]^T$.

4.6 Traffic System State Identification on a Computational Graph

This section first introduces a customized computational graph constructed to represent the nonlinear programming model M2 described in Section 4.5. The second part of this section illustrates the process of solving the optimization model on the computational graph using the forward-backward algorithm. Furthermore, a distributed computing framework of the computational graph is presented to handle largescale instances in real-life applications.

4.6.1 Computational Graph Structure

Model M2 is a standard nonlinear programming model that can be solved using existing nonlinear solvers such as Ipopt (Wächter and Biegler, 2006), BARON (Tawarmalani and Sahinidis, 2004), and Knitro (Waltz et al., 2006). However, there are three challenges in doing so. First, nonlinear solvers can only model scalar variables, indicating that gradient calculations and value updating are independently performed for each scalar variable in each optimization iteration, whereas the vector-based representation in model M2 is not fully utilized to accelerate the optimization process. Second, the partial differential functions $h_x(x, t; \theta, \varphi)$ and $f_t(x, t; \theta)$ in Eq. (33) must be manually derived each time the form of the traffic state distribution functions (i.e., f and g) changes, which may affect the applicability in real-life instances. Third, the functional forms of f and g

must be properly determined for each instance, which could be extremely difficult for cases with complicated time-varying traffic conditions.

To address these three challenges, this research casts and solves model M2 on a customized computational graph. The computational graph approach has been successfully applied to solve complex and largescale traffic problems. For example, Wu et al. (2018) cast the four-step method on a computational graph for travel demand estimation, Ma et al. (2020) estimated multiclass dynamic origin-destination demand on computational graphs using a forward-backward algorithm, and Kim et al. (2021) introduced computational graph-based frameworks to integrate the strengths of econometric models and machine learning algorithms in discrete choice modeling applications. There is also a new trend of integrating traffic flow modeling with machine learning to improve model performance while keeping the interpretability (e.g., Yuan et al., 2021 and Thodi et al., 2022). Solving model M2 on a computational graph enables the use of various powerful computational techniques, such as vectorization, parallel computing, and automatic differentiation, from the deep learning field.

The structure of the computational graph constructed in this study is shown in Fig. 28 and denoted as graph G . The dataflow of graph G starts with sample space-time points $\mathbf{x}t$, and in turn, calculates various state estimations on $\mathbf{x}t$ (i.e., \mathbf{k} , \mathbf{v} , \mathbf{q} , \mathbf{q}_x , and \mathbf{k}_t), which correspond to Eqs. (35)-(39) in model M2. Next, by comparing the state estimations with observations and traffic flow models, the total loss (i.e., inconsistency across different layers) can be measured. The objective of training the graph G is to minimize the total loss by optimizing the value of trainable variables (i.e., $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$, $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$).

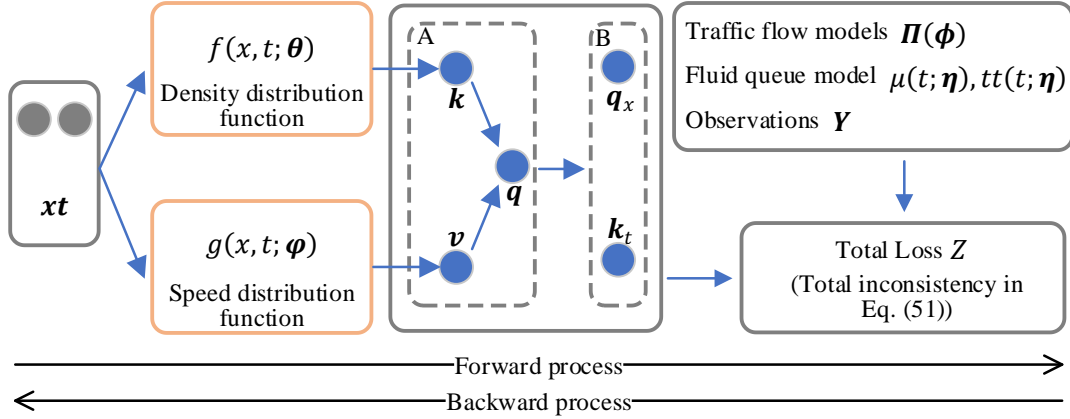


Fig. 28. Computational Graph Structure.

How the proposed computational graph can address the above three challenges is illustrated below.

The first two challenges are addressed by the data structure used in the computational graphs, that is, the tensor. Tensors are matrices. Consequently, all calculations and value updating are performed in a vectorization scheme, resulting in significant efficiency improvements. In addition, compared to normal matrices, tensors have the ability to remember what operations happen in what order on them, which is the core of calculating derivatives using the chain rule. Modern deep learning frameworks such as TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019) are capable of performing automatic differentiation based on the ability of tensors. Every time data flow passes to a tensor, the derivatives of the tensor with respect to all upstream tensors are ready to use. In Fig. 28, the values of q_x and k_t in rectangle B are available as soon as the values of k and q in rectangle A are updated. With the automatic differentiation functionality of computational graphs, partial differential functions $h_x(x, t; \theta, \varphi)$ and $f_t(x, t; \theta)$ are readily available when the functions f and g are differentiable. .

In terms of the third challenge, the form of the distribution functions can be determined in two ways. The first is to derive analytical forms for distribution functions based on prior knowledge, but this is only feasible for simple functional forms. The second approach uses form-free functions. For example, a polynomial function can approximate a complex function with a certain degree of estimation errors. This research uses a powerful tool in the context of computational graphs to represent the state distribution functions f and g : fully-connected neural networks (FCNNs). Compared with polynomial functions, an FCNN can approximate functions using fewer parameters. The ability that an FCNN is capable of approximating any function with high accuracy is known as “universal approximation” in the field of deep learning. By using FCNNs to represent functions f and g , the function parameters θ and φ are actually the weight matrices and bias matrices of FCNNs.

A three-layer FCNN is used as an illustrative example to show how to calculate the density estimation \mathbf{k} with an input $\mathbf{x}\mathbf{t}$ (see Fig. 29), while the speed estimation \mathbf{v} can be obtained in a similar manner. Assume that layer i has m_i neurons, $i = 1, 2, 3$. The shapes of weight matrix \mathbf{W}_i and bias vector \mathbf{b}_i are listed in Table 9. The density estimation \mathbf{k} can be calculated as follows:

$$\mathbf{k} = \Gamma\{\Gamma[\Gamma(\mathbf{x}\mathbf{t}\mathbf{W}_1 + \mathbf{1}_{m_1}\mathbf{b}_1)\mathbf{W}_2 + \mathbf{1}_{m_2}\mathbf{b}_2]\mathbf{W}_3 + \mathbf{1}_{m_3}\mathbf{b}_3\}\mathbf{W}_4 + \mathbf{1}_{m_4}\mathbf{b}_4, \quad (70)$$

where $\mathbf{1}_m$ denotes an all-one vector with the shape of $(m, 1)$; and Γ is an activation function. The role of the activation function is to add nonlinearity to FCNNs, which can significantly improve their approximation ability. This research uses ReLU as the activation function, which can be expressed as

$$\Gamma(\sigma) = \max(0, \sigma). \quad (71)$$

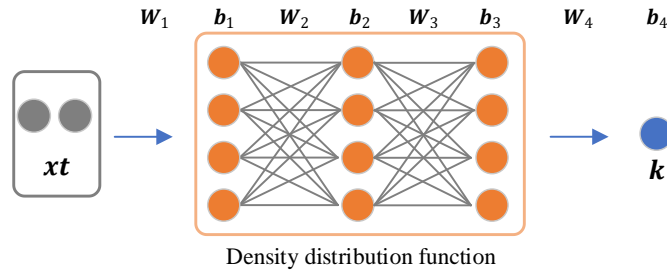


Fig. 29 Detailed Representation of the fully connected Neural Network (Density Function).

Table 9 Shape of the Parameters in the FCNN (Density Distribution Function).

Parameter	W_1	b_1	W_2	b_2	W_3	b_3	W_4	b_4
Shape	$(2, m_1)$	$(1, m_1)$	(m_1, m_2)	$(1, m_2)$	(m_2, m_3)	$(1, m_3)$	$(m_4, 1)$	$(1, 1)$

4.6.2 Estimation Using Forward-backward Algorithm

This subsection introduces how to train graph G using the forward-backward algorithm. The forward-backward algorithm includes two processes: forward and backward processes. In the forward process, the value of the total loss Z is calculated based on the forward data flow, shown in Fig. 28. In the backward process, the gradients of the total loss Z with respect to trainable variables θ , φ , η and ϕ are calculated using the automatic differentiation technique. Subsequently, a gradient-based optimization algorithm, such as stochastic gradient descent (SGD) (Robbins and Monro, 1951) and Adam (Kingma and Ba, 2014), is used to update the value of trainable variables based on the gradient information in the backward direction. The pseudocode of the forward-backward algorithm used in this study is shown in Algorithm 3.

Algorithm 3 Forward-backward Algorithm

Input: graph G ; max number of training iterations N

Output: trained graph G with minimum loss

- 1: set initial values for trainable variables (i.e., θ , φ , η , and ϕ) of graph G
 - 2: **for** $iter := 1, 2, \dots, N$ **do**
 - 3: (forward process)
 - 4: *state estimations:* $\mathbf{k} = f(\mathbf{x}_t; \theta)$, $\mathbf{v} = g(\mathbf{x}_t; \varphi)$, $\mathbf{q} = \mathbf{k} \cdot \mathbf{v}$
 - 5: *partial differential terms:* calculate \mathbf{q}_x and \mathbf{k}_t using automatic differentiation
 Calculate inconsistencies from observations and traffic flow models:
 - 6: $\boldsymbol{\varepsilon}_m^{MA-dr}$, $\boldsymbol{\varepsilon}_o^{MA-tt}$, $\boldsymbol{\varepsilon}_m^{ME-fd}$, $\boldsymbol{\varepsilon}_m^{ME-fc}$, $\boldsymbol{\varepsilon}_m^{ME-td}$, $\boldsymbol{\varepsilon}_o^{ME-l}$, $\boldsymbol{\varepsilon}_o^{ME-gs}$, $\boldsymbol{\varepsilon}_o^{ME-gf}$, $\boldsymbol{\varepsilon}_o^{ME-vd}$,
 $\boldsymbol{\varepsilon}_m^{MI-cf}$ and $\boldsymbol{\varepsilon}_o^{MI-cf}$ (Eqs. (35)-(39), (47), (48), (50), (51), (56), (57), (58),
 (62), (64), (66) and (67))
 - 7: *total loss:* calculate total loss Z using Eq. (68)
 - 8: (backward process)
 - 9: *gradient:* calculate gradients of Z with respect to trainable variables (i.e.,
 $\frac{\partial Z}{\partial \theta}$, $\frac{\partial Z}{\partial \varphi}$, $\frac{\partial Z}{\partial \eta}$, $\frac{\partial Z}{\partial \phi}$)
 - 10: *update variables:* $\theta = \theta + \lambda_\theta^T \frac{\partial Z}{\partial \theta}$, $\varphi = \varphi + \lambda_\varphi^T \frac{\partial Z}{\partial \varphi}$, $\eta = \eta + \lambda_\eta^T \frac{\partial Z}{\partial \eta}$, $\phi = \phi +$
 $\lambda_\phi^T \frac{\partial Z}{\partial \phi}$
 - 11: **return** graph G
-

In line 10, λ_θ , λ_φ , λ_η and λ_ϕ represent the step sizes used to update the corresponding variables. In this study, as suggested by the comprehensive numerical study by Ruder (2016), Adam is adopted for the variable updating in line 10, during which the step sizes are adaptively determined based on first-order and second-order moments. Another improvement made in the forward-backward process is training with small batches. Instead of calculating inconsistencies on all sample points, this research randomly selects parts of the sample points for inconsistency calculation and use the resulting derivatives for subsequent variable updating. Compared with training with the entire dataset, training with small batches involves more randomness in the process and is expected to have more chances to jump out of local minimums and saddle points. This

training approach with small batches has been suggested in many other studies (Keskar et al., 2016).

The forward and backward processes are iteratively performed until a preset stop criterion has been reached, for example, a maximum number of iterations or a target loss value. This research adopts the maximum number of iterations as the stopping criterion.

4.6.3 Distributed Computing

In addition to the vectorization-based computing introduced above, this section presents a distributed training framework for the proposed computational graph to improve computational efficiency, which is of vital importance in real-time largescale applications. In addition, a distributed implementation does not have privacy concerns and has higher reliability than a centralized implementation (Nedić and Liu, 2018). The reason is that under a distributed computing framework, computations are performed independently on local servers, and only a small amount of necessary information is shared with other servers. Therefore, there is no need to upload all the data to a central server, which could render the entire system fragile and unsafe. A long corridor AD (see Fig. 30) is used as an example to illustrate the distributed implementation of the proposed computational graph method, while the methodology presented below can be applied to more complicated networks.

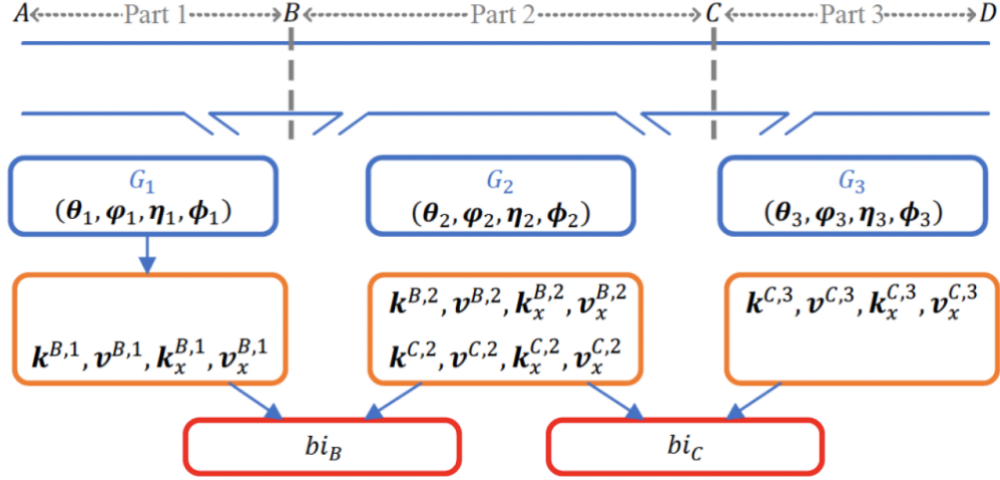


Fig. 30. Illustration of the Distributed Modeling.

In Fig. 30, the long corridor AD is split into three parts: Part 1 (AB), Part 2 (BC), and Part 3 (CD), with a computational graph built for each part, that is, graphs G_1 , G_2 and G_3 . These three computational graphs have the same structure as that introduced in Section 4.6.1. For each computational graph G_i , in addition to the existing outputs, additional state outputs on its boundaries are added. For example, the additional state outputs of graph G_2 include $\mathbf{k}^{B,2}, \mathbf{v}^{B,2}, \mathbf{k}_x^{B,2}, \mathbf{v}_x^{B,2}$ for boundaries B and $\mathbf{k}^{C,2}, \mathbf{v}^{C,2}, \mathbf{k}_x^{C,2}, \mathbf{v}_x^{C,2}$ for boundary C . These additional state outputs are used to measure the inconsistencies of the state estimations on the boundary from different computational graphs. The state inconsistency on boundary B is expressed as:

$$\begin{aligned}
 bi_B = & \|\mathbf{k}^{B,1} - \mathbf{k}^{B,2}\|^2 + \|\mathbf{v}^{B,1} - \mathbf{v}^{B,2}\|^2 + \|\mathbf{k}_x^{B,1} - \mathbf{k}_x^{B,2}\|^2 + \|\mathbf{v}_x^{B,1} - \\
 & \mathbf{v}_x^{B,2}\|^2.
 \end{aligned} \tag{72}$$

The right-hand side of Eq. (72) comprises two groups: (1) $\|\mathbf{k}^{B,1} - \mathbf{k}^{B,2}\|^2 + \|\mathbf{v}^{B,1} - \mathbf{v}^{B,2}\|^2$ and (2) $\|\mathbf{k}_x^{B,1} - \mathbf{k}_x^{B,2}\|^2 + \|\mathbf{v}_x^{B,1} - \mathbf{v}_x^{B,2}\|^2$. The first group measures the inconsistency in state values, whereas the second measures the inconsistency in the

derivative of the state values. Minimizing bi_B results in consistent state values from G_1 and G_2 on boundary B and smooth state changes in the area beside boundary B . Similarly, the state inconsistency on boundary C can be expressed as

$$bi_C = \|\mathbf{k}^{C,2} - \mathbf{k}^{C,3}\|^2 + \|\mathbf{v}^{C,2} - \mathbf{v}^{C,3}\|^2 + \|\mathbf{k}_x^{C,2} - \mathbf{k}_x^{C,3}\|^2 + \|\mathbf{v}_x^{C,2} - \mathbf{v}_x^{C,3}\|^2. \quad (73)$$

The forward-backward algorithm is used to minimize the boundary inconsistencies bi_B and bi_C .

This research uses the forward-backward algorithm to minimize the consistency loss bi_B and bi_C . After obtaining bi_B and bi_C through a forward process, the gradients of bi_B and bi_C with respect to trainable variables in the graphs are calculated using automatic differentiation. For simplicity, vectors $\mathbf{V}_i = [\boldsymbol{\theta}_i^T, \boldsymbol{\varphi}_i^T, \boldsymbol{\eta}_i^T, \boldsymbol{\phi}_i^T]^T$ are used to represent all the trainable variables in the graph G_i . Then, $\frac{\partial bi_B}{\partial \mathbf{V}_1}$ and $\frac{\partial bi_B}{\partial \mathbf{V}_2}$ are calculated for bi_B , while $\frac{\partial bi_C}{\partial \mathbf{V}_2}$ and $\frac{\partial bi_C}{\partial \mathbf{V}_3}$ are calculated for bi_C . Finally, using the gradient descent method, the values of the trainable variables are updated using Eqs. (74)-(76), where $\boldsymbol{\lambda}$ is the step size vector.

$$\mathbf{V}_1 = \mathbf{V}_1 + \boldsymbol{\lambda}_{\mathbf{V}_1}^T \frac{\partial bi_B}{\partial \mathbf{V}_1}, \quad (74)$$

$$\mathbf{V}_2 = \mathbf{V}_2 + \boldsymbol{\lambda}_{\mathbf{V}_2}^T \left(\frac{\partial bi_B}{\partial \mathbf{V}_2} + \frac{\partial bi_C}{\partial \mathbf{V}_2} \right) / 2, \quad (75)$$

$$\mathbf{V}_3 = \mathbf{V}_3 + \boldsymbol{\lambda}_{\mathbf{V}_3}^T \frac{\partial bi_C}{\partial \mathbf{V}_3}. \quad (76)$$

As variable \mathbf{V}_2 contributes to both bi_B and bi_C in the forward process, the average gradient from bi_B and bi_C is used to update variable \mathbf{V}_2 in Eq. (75). The training process of the distributed computational graph is summarized in Algorithm 4.

Algorithm 4 Training process of the distributed computational graph

Input: computational graphs $G_i, i = 1,2,3$; max number of training iterations N

Output: trained computational graphs $G_i, i = 1,2,3$

- 1: set initial values for trainable variables \mathbf{V}_i of graph $G_i, i = 1,2,3$
 - 2: **for** epoch $iter := 1,2, \dots, N$ **do**
 - 3: Perform the forward process and backward process in Algorithm 3 for each graph $G_i, i = 1,2,3$
 - 4: Calculate the boundary inconsistency bi_j for each boundary $j, j = B, C$
 - 5: Calculate the gradients of boundary inconsistency bi_j for each boundary $j, j = B, C$
 - 6: Update trainable variables using Eqs. (74)-(76)
 - 7: **return** graphs $G_i, i = 1,2,3$
-

Under the distributed modeling framework, local graph training is performed (line 3), and only necessary information is shared with the adjacent graphs to reach consistency on boundaries (line 4), which makes the resulting model more efficient and robust.

4.7 Numerical Experiments

This section examines the performance of the proposed TSSI framework by using both real-world and hypothetical datasets. In Section 4.7.1, extensive experiments performed on six freeway segments under various traffic conditions are presented, and Section 4.7.2 applies the proposed framework to a freeway corridor with ramps. In Section 4.7.3, a hypothetical freeway corridor designed to demonstrate the effectiveness of the proposed framework in a distributed computing environment is presented. The computational graphs constructed in this study are implemented using the open-source machine learning framework TensorFlow (Abadi et al., 2016). The source code and dataset used in this study are publicly available at <https://github.com/jiawlu/Traffic-State-Estimation-Computational-Graph>.

4.7.1 Real-world Freeway Segments

The Highd dataset (Krajewski et al., 2018) is adopted as the first dataset to evaluate the performance of the proposed framework. The dataset provides detailed vehicle trajectories extracted from high-resolution videos captured by drones at different locations on German freeways (each location contains two directions). To investigate the model performance under complicated traffic conditions, six segments are selected from the Highd dataset, where both light and heavy traffic conditions and transitions between the two states are also included. Detailed information regarding the dataset used in this study is presented in Table 10.

Table 10 Summary of the Highd Dataset Used in This Research.

Dataset	Highd ID	Direction	Month	Weekday	Start time	End time
1	12	1	201709	Thu	17:21	17:36
2	25	1	201710	Mon	8:55	9:14
3	26	1	201710	Mon	9:20	9:38
4	25	2	201710	Mon	8:55	9:14
5	26	2	201710	Mon	9:20	9:38
6	46	2	201711	Wed	8:47	9:06

The adopted freeway segments are approximately 420-meter long, and no ramp is involved. To reduce the impacts of vehicle identification errors on segment boundaries, for all datasets, data processing and subsequent estimations are performed within the range of 30 meters to 410 meters. In addition, depending on the battery consumption of drones, the time span of each dataset is not constant and varies around 1,000 seconds. The first 900-seconds vehicle trajectory data is used to maintain the same duration across the six datasets. It should be noted that as the time duration of these datasets is relatively short and does not cover a complete congestion duration, the components of the macroscopic modeling are not included in the estimation model in this subsection.

Benefitting from complete vehicle trajectories, the traffic state ground truth can be easily obtained using simple aggregation methods. In addition, various virtual detectors can be designed, similar to real-world cases, to collect traffic flow data. The configurations of the virtual traffic detectors used in the following experiments are summarized in Table 11.

Table 11 Configurations of Virtual Traffic Detectors.

Detector name	Configurations
Loop detector	Location: 120 meters and 320 meters from the segment upstream Aggregation time interval: 1 minute
GPS	Sampling rate: 10% Reporting frequency: 5 seconds
Bluetooth detector	Location: 40 meters and 400 meters from the segment upstream Sampling rate: 5%
Video detector	Location: 220 meters to 230 meters from the segment upstream

In Section 4.6, density distribution function f and speed distribution function g are modeled by two different and independent FCNNs, while the structure is modified to improve the training efficiency of the resulting network. As shown in Fig. 31, a shared FCNN (module S) is added before modules A and B. In general, for a neural network, front layers are designed to extract features from inputs, whereas subsequent back layers are responsible for performing regressions and producing final outputs based on the features from the front layers. In the TSSI problem, the front layers in modules A and B actually conduct the same task, that is, extracting high-dimension features from a given space-time regime \mathbf{xt} . Therefore, instead of constructing and training two feature extracting layers, a shared feature extracting network is built for modules A and B to reduce the total number of variables to be trained, which will help speed up the training process. The structures of the three modules used for the six real-world datasets are as follows:

Module S: two fully-connected hidden layers with 125 neurons at each layer; and
 Modules A and B: five fully-connected hidden layers with 125 neurons in each
 layer.

Other solution algorithm related settings:

Number of iterations: 30,000

Learning rate: 0.001

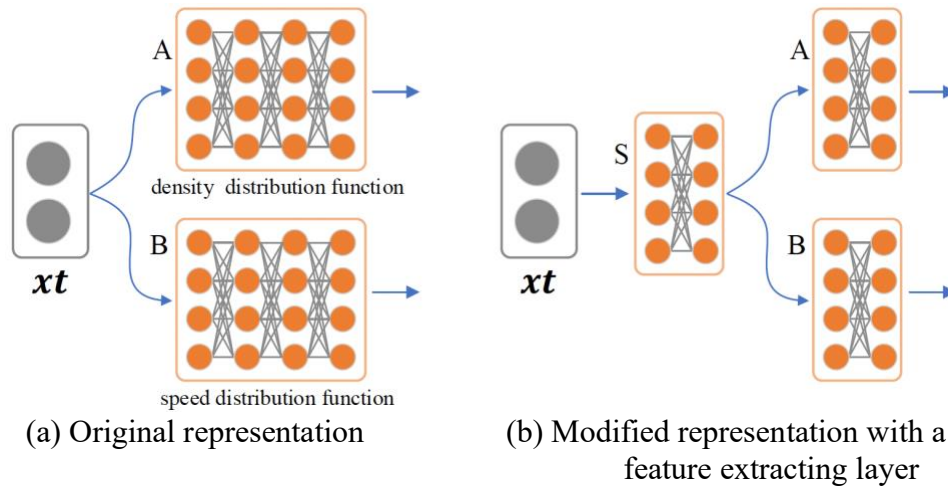


Fig. 31. Changes in the Density and Speed Distribution Function in the Implementations.

Estimation results

As illustrated in Algorithm 3, the first step in training a computational graph using the forward-backward algorithm is to set the initial values for the variables to be optimized. Owing to the high complexity and non-convexity of the model built on the proposed computational graph, gradient descent-based algorithms may get stuck in local optimal or saddle points during the training process. Therefore, different initial values may result in different final outputs. To evaluate the average performance and stability of the proposed framework, the solution algorithm is executed five times with different starting points for each dataset.

First, the results of speed estimations are presented. Table 12 lists the accuracy of the speed estimations on the six datasets, and Fig. 32 depicts the observed and estimated speed profiles for each dataset. The following findings can be observed from Table 12 and Fig. 32.

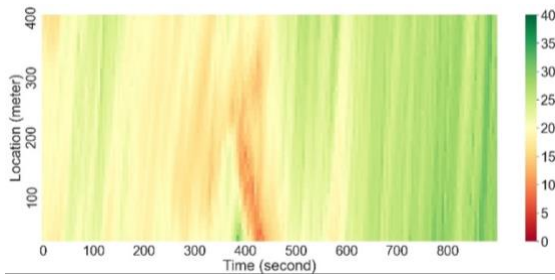
- (1) Accurate speed estimations are obtained for all the datasets. Dataset 3 has the highest mean absolute error (MAE) (2.31 m/s), and dataset 5 has the lowest MAE (0.98 m/s), while the MAEs of the other datasets vary around 1.23 m/s. In terms of the mean absolute percentage error (MAPE), except for datasets 2 and 3, the average MAPEs of the other datasets are all less than 6%. Datasets 2 and 3 have relatively high MAPEs because the traffic speeds in these two datasets are very low owing to traffic congestions, in which case a small absolute error would result in a large percentage error.
- (2) Compared to datasets with congested conditions (i.e., datasets 2 and 3), better estimation results are observed on datasets with light traffic conditions (i.e., datasets 1, 4, 5, and 6). A possible reason may be related to the higher randomness of traffic flows and frequent stop-and-go waves (Stern et al., 2018) under congested conditions, whereas the traffic flow models adopted in this study are based on deterministic settings. Researchers also found that the flow-density relationship in the congested regime may depend on vehicle length (Coifman, 2015). Integrating traffic flow models that consider stochasticity and have great performance under congested traffic conditions into the proposed framework is expected to improve the estimation performance under congested traffic conditions, which can be investigated in a future study.

- (3) The results of the different runs on each dataset are very close, indicating that the proposed model is not sensitive to the initial values of the decision variables and is capable of producing reliable and accurate results with different starting points. Owing to the adoption of the Adam algorithm (Kingma and Ba, 2014) for updating the variables, the current TSSI implementation can help jump out of some local minima and saddle points using adaptive estimates of lower-order moments in the training process.
- (4) The traffic flow dynamics are captured well under different traffic conditions. As shown in Fig. 32, the estimated speed profile is very close to the corresponding observed speed profile for each dataset. The propagation of forward waves in non-congested regimes and backward waves in congested regimes have been precisely reproduced.
- (5) It is also observed that compared to the observed speed profiles, the estimated speed profiles appear much smoother, and some sharp speed changes cannot be reproduced very well. This may be because of the use of CSTD functions. This issue can be addressed by using more hidden layers in the FCNNs. More hidden layers in neural networks help improve the fitting ability at the expense of a longer computing time.

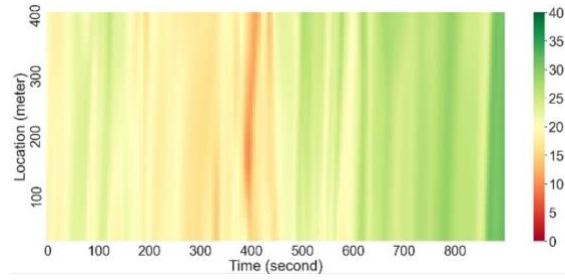
Table 12 Accuracy of Speed Estimations on Six Datasets.

Dataset	1st run	2nd run	3rd run	4th run	5th run	Average
1	1.22/5.95	1.24/6.07	1.18/5.76	1.15/5.60	1.23/6.01	1.20/5.88
2	1.24/14.09	1.22/13.74	1.27/14.46	1.30/14.61	1.25/14.35	1.26/14.25
3	2.25/18.15	2.34/18.91	2.33/18.92	2.33/18.69	2.30/18.60	2.31/18.65
4	1.28/5.26	1.19/4.88	1.21/5.00	1.22/5.05	1.20/4.95	1.22/5.03
5	1.00/4.65	1.01/4.69	0.94/4.41	0.98/4.59	0.99/4.66	0.98/4.60
6	1.24/5.94	1.26/5.90	1.21/5.66	1.27/6.14	1.22/5.88	1.24/5.90

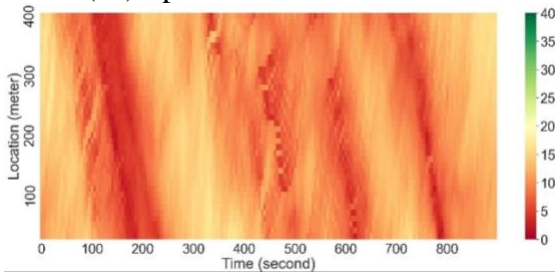
Note: Values in each cell denote MAE/ MAPE.



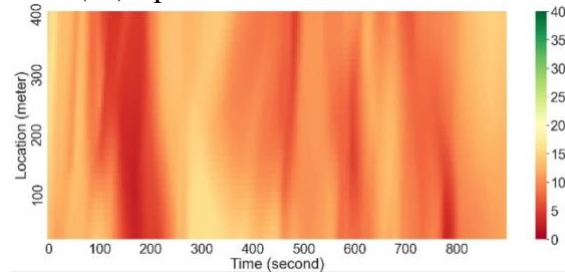
(a1) Speed observations for DS 1



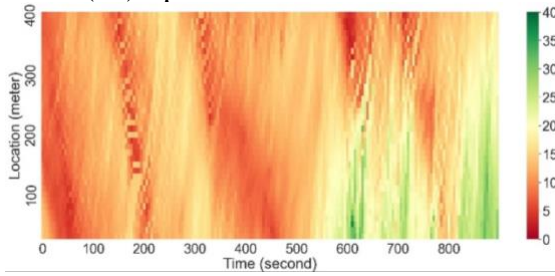
(a2) Speed estimations for DS 1



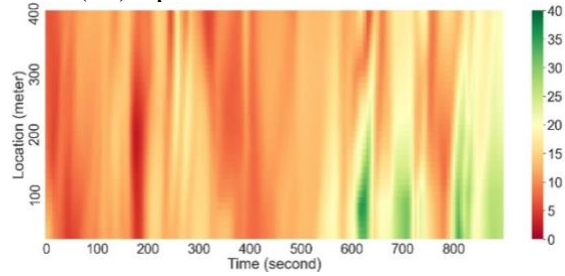
(b1) Speed observations for DS 2



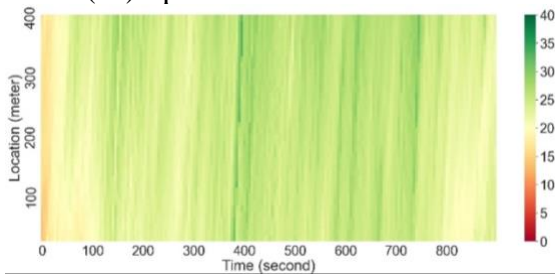
(b2) Speed estimations for DS 2



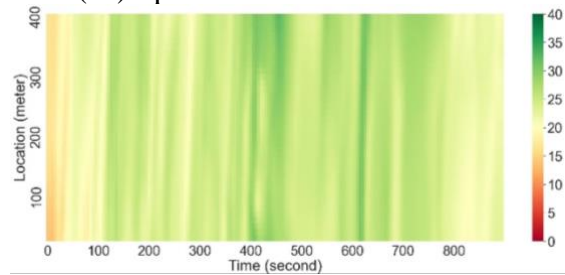
(c1) Speed observations for DS 3



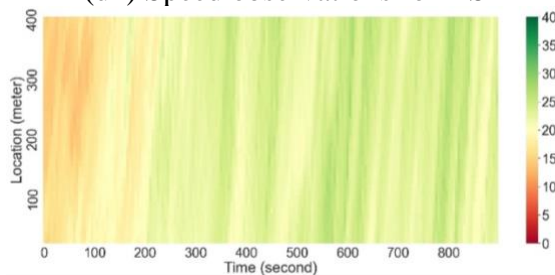
(c2) Speed estimations for DS 3



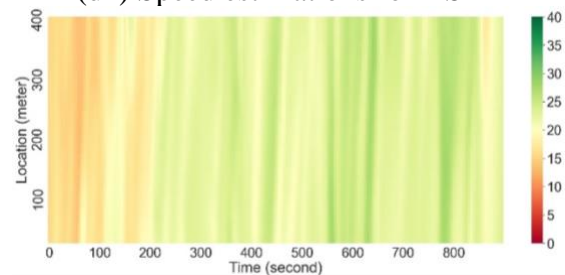
(d1) Speed observations for DS 4



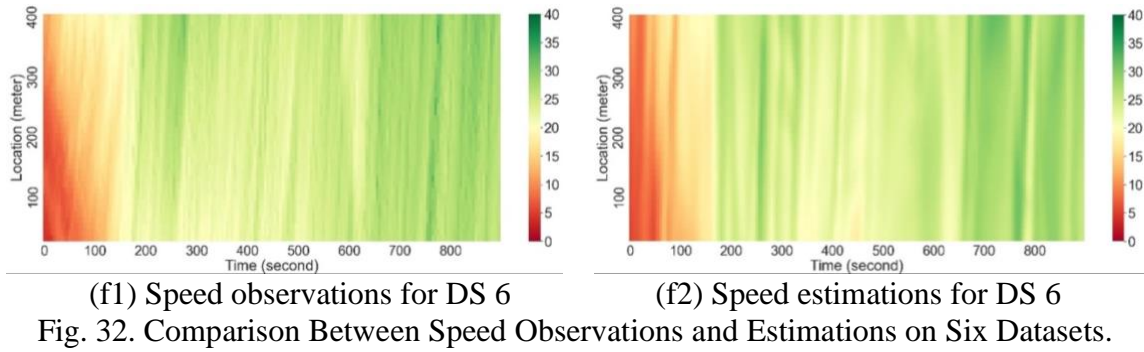
(d2) Speed estimations for DS 4



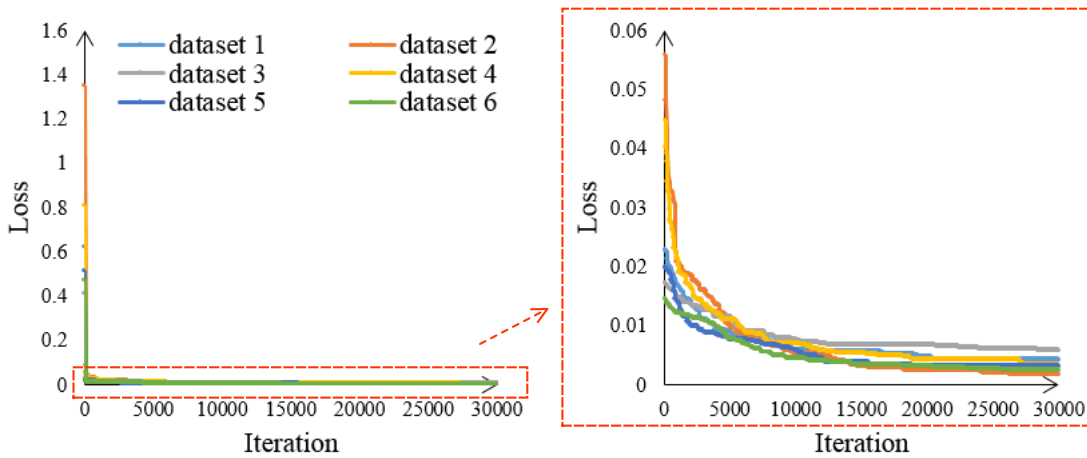
(e1) Speed observations for DS 5



(e2) Speed estimations for DS 5



To demonstrate the convergence of the proposed method, Fig. 33 depicts the loss convergence curve of the first run for each dataset. As can be seen from the figure, the loss decreases very quickly in early iterations (approximately the first 5000 iterations), and then slowly decreases in later iterations. The right part of Fig. 33 shows that the losses become stable after 20,000 iterations on all the datasets, indicating the reasonable stability and good convergence characteristics of the proposed method.



Fixing traffic flow model parameters to show the value of joint estimation

After evaluating the performance of the proposed TSSI modeling framework, this research tries to investigate, compared with using pre-calibrated traffic flow models in TSE, how much a joint estimation framework can help improve the accuracy of state

estimations. By using the entire dataset from Highd, this research first calibrates the parameters of fundamental diagrams offline and then treat them as constants in the state estimation model. Table 13 lists the speed estimations with fixed traffic flow model parameters. Compared with Table 12, the estimation error increases for all six datasets, which means that simply using pre-calibrated traffic flow models in the TSSI could significantly affect the state estimation accuracy.

Table 13 Evaluation of Speed Estimations with Fixed Traffic Flow Model Parameters.

Case ID	1st run	2nd run	3rd run	4th run	5th run	Average
1	1.22/5.95	1.24/6.07	1.18/5.76	1.15/5.60	1.23/6.01	1.20/5.88
2	1.24/14.09	1.22/13.74	1.27/14.46	1.30/14.61	1.25/14.35	1.26/14.25
3	2.25/18.15	2.34/18.91	2.33/18.92	2.33/18.69	2.30/18.60	2.31/18.65
4	1.28/5.26	1.19/4.88	1.21/5.00	1.22/5.05	1.20/4.95	1.22/5.03
5	1.00/4.65	1.01/4.69	0.94/4.41	0.98/4.59	0.99/4.66	0.98/4.60
6	1.24/5.94	1.26/5.90	1.21/5.66	1.27/6.14	1.22/5.88	1.24/5.90

Note: Values in each cell denote MAE/ MAPE.

Removing flow conservation law

One of the major advantages of adopting the proposed CSTD representation for traffic flow variables is the convenience of modeling PDEs. This experiment attempts to identify how the proposed model behaves without considering the flow conservation law expressed by a PDE. For dataset 1, the conservation law is removed from the proposed model while keeping the other settings and parameters unchanged. Fig. 34 shows the speed estimation produced by the modified model. The total loss of the final estimation is 0.0041, whereas it's 0.0045 under the original settings. Although the total loss decreases without considering the flow conservation law, the speed estimation is worse than that in Fig. 32(a2). The traffic flow shown in Fig. 34 is no longer continuous. Forward waves in the non-congest regimes and backward waves in the congested regimes cannot be reproduced.

This is caused by overfitting, that is, without the regulation of the flow conservation law, the optimization model would try to fit the estimations with local observations as much as possible while neglecting the reasonableness of state distributions on the space-time plane. This experiment demonstrates the necessity of considering the flow conservation law in TSSI and the advantages of the proposed CSTD representation for modeling PDEs.

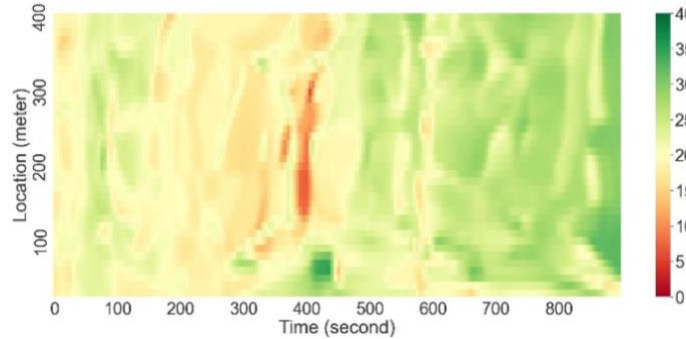


Fig. 34. Speed Estimation on Dataset 1 Without Considering the Flow Conservation Law.

Sensitivity analysis of the value of different types of measurement

The results reported above are obtained based on the four types of traffic detectors, as listed in Table 15; however, in many real-world applications, data availability varies from site to site, especially the data availability from automatic vehicle identification devices (AVI), for example, loop detectors, Bluetooth detectors, and video detectors. On the other hand, massive amounts of GPS data are produced by taxis, map companies, and ridesharing companies every day. The coverage of GPS data is significantly larger than that of data from AVI devices, and can be collected at almost no expense. Therefore, it is necessary to investigate the performance of the proposed framework using only GPS data.

As an example, Fig. 35 shows the impact of the GPS sampling rate on state estimations using dataset 1, where the base case denotes performing estimations using four types of detectors, as listed in Table 15. MAPE of the speed estimations is 8.43% after all

AVI data were removed from the base case. The speed estimation error decrease with the increase in the GPS sampling rate. When the GPS sampling rate reaches 80%, MAPE of the speed estimation is 7.26%, which is still higher than that in the base case. Based on these results, it can be concluded that the proposed solution framework can produce good results even with limited GPS data, and the performance can be further improved by utilizing multiple data sources.

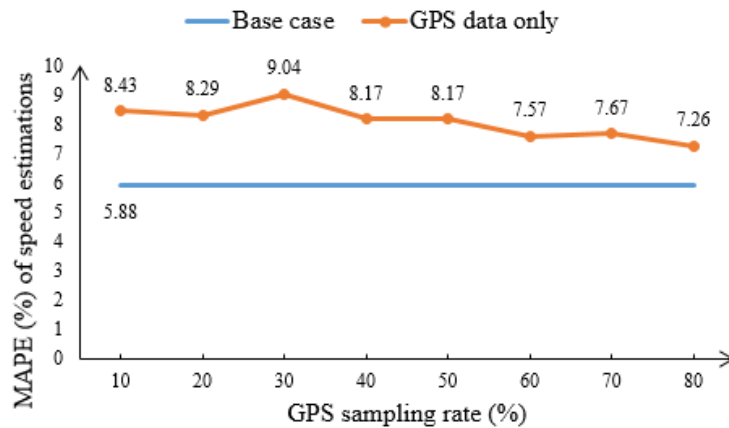


Fig. 35. Impact of GPS Sampling Rate on Estimation Quality (Dataset 1).

4.7.2 A Real-world Freeway Corridor with a Downstream Bottleneck

In this section, examination of the proposed cross-resolution TSSI framework on a 3-mile long freeway corridor with a downstream bottleneck is presented. As shown in Fig. 36, the corridor of interest is within the absolute postmile 22 to 25 on freeway I880-N in Alameda County, California. The analysis time horizon is 10:00 am – 12:00 pm on February 8th, 2008. As illustrated in Table 14, two types of data collected in the Mobile Century experiment (Herrera et al., 2010) in 2008 are used, where the loop detector data is from the Caltrans Performance Measurement System (PeMS). The travel time of the probe vehicles along the entire corridor is also extracted from the GPS data to serve as a system-

wide macroscopic observation. The estimation model settings in this section are the same as those used in Section 7.1, except for the involvement of macroscopic modeling. The analytical time-dependent travel time function used in the experiment is based on the fifth approximation of the waiting time derived in Eq. (28).

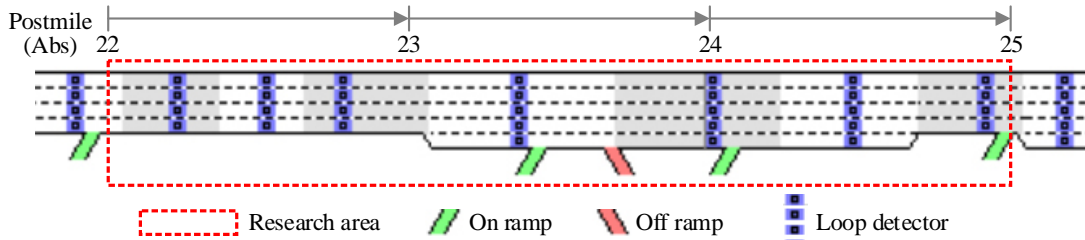


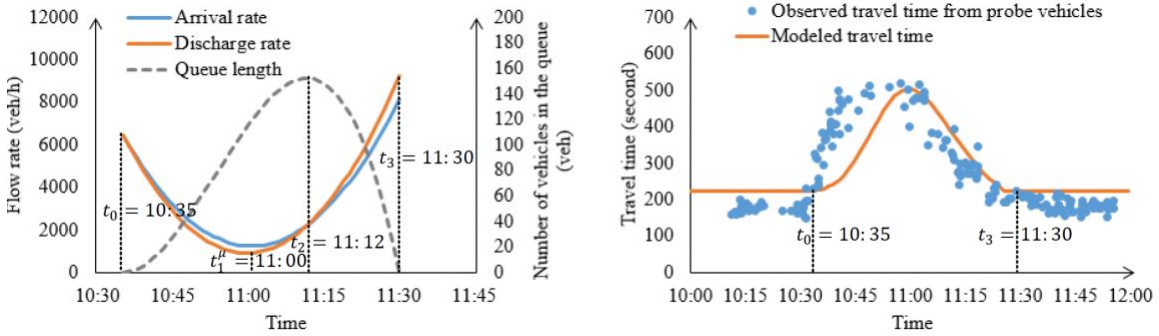
Fig. 36. Layout of the Freeway Corridor on I880-N (Postmile 22-25), Adopted from the PeMS.

Table 14 Configurations of Traffic Detectors.

Detector name	Configurations
Loop detector	Location (postmile): 22.23, 22.53, 22.78, 23.37, 24.01, 24.48, and 24.92
GPS	Aggregation time interval: 5 minutes Sampling rate: 1.74% (192 probe vehicles) Reporting frequency: 3.5 seconds on average

Fig. 37 shows the estimation results at the macroscopic level. Specifically, Fig. 37(a) plots the calibrated arrival rate, discharge rate, and queue length curves, with an emphasis on the values of the critical time points, that is, $t_0 = 10:35$, $t_1^\mu = 11:00$, $t_2 = 11:12$, and $t_3 = 11:30$. The values of the other essential parameters are $\mu(t_1^\mu) = 913$ (veh/h), $\gamma^\mu = 7.0 \times 10^{-7}$, and $\gamma = 8.6 \times 10^{-8}$. From Fig. 37(a), it can be seen that, at $t_0 = 10:35$, the traffic flow arrival rate equals the discharge rate, and a queue starts to form; at $t_1^\mu = 11:00$, the discharge rate reaches its minimum; at $t_2 = 11:12$, the arrival rate again equals the discharge rate, at which the maximum queue length is observed and the queue starts to dissipate; and at $t_3 = 11:30$, the queue disappears and the congestion

period ends. Fig. 37(b) depicts the corresponding modeled time-dependent travel time curve, as well as the observed travel time from probe vehicles. Travel time is relatively stable outside the congestion period, whereas it is time-dependent between t_0 and t_3 . In addition, the modeled travel time curve closely matches the observed data, demonstrating the capability of the proposed method in modeling and calibrating the macroscopic system dynamics. Calibrated system-wide demand and supply curves are vital for traffic control and management.

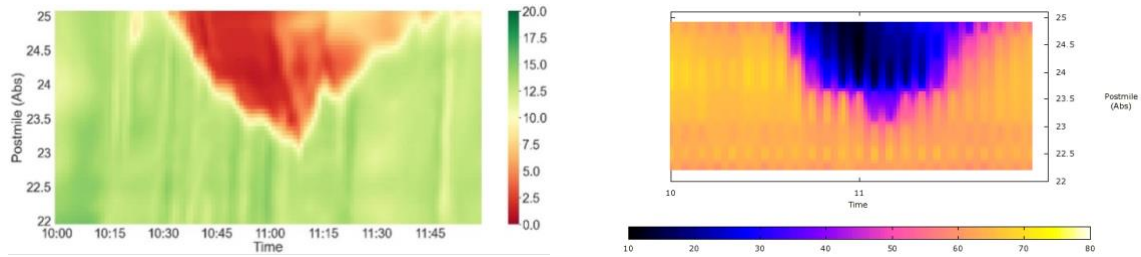


(a) Estimated arrival rate, discharge rate and queue length curves

(b) Estimated travel time curve and observed travel time from probe vehicles

Fig. 37. Estimation Results of System-wide Measures at the Macroscopic Level.

Fig. 38(a) shows the estimated speed profile at the mesoscopic level. In terms of the congestion starting and ending times, one can easily verify the consistency between the estimation results at the macroscopic and mesoscopic levels. As for the accuracy of the state estimations, owing to the lack of ground truth, the speed estimations obtained by smoothing methods from PeMS in Fig. 38(b) is used for comparison. It can be seen that the speed estimation produced by the proposed method clearly shows forward waves in free-flow regimes and backward waves in congested regimes. Moreover, state changes within the congested regime have also been successfully reconstructed.



(a) Speed estimations of the proposed method

(b) Speed estimations from PeMS

Fig. 38. Speed Estimations of the Proposed Method and PeMS.

4.7.3 Applying Distributed Computing on a Hypothetical Freeway Corridor

This section focuses on the distributed implementation of the proposed framework. To compare the estimation results with the ground truth, a 3 km-long hypothetical corridor with four ramps (see Fig. 39) was built in a microscopic traffic simulator, SUMO (Lopez et al., 2018), and simulated with assumed travel demands to provide complete observations. As summarized in Table 15, virtual traffic detectors were created to collect the traffic flow data. The traffic simulation is ran for 65 minutes, where the first 5 minutes are used for simulation warm up, and data collected in the remaining 60 minutes are used for estimation evaluation purposes.

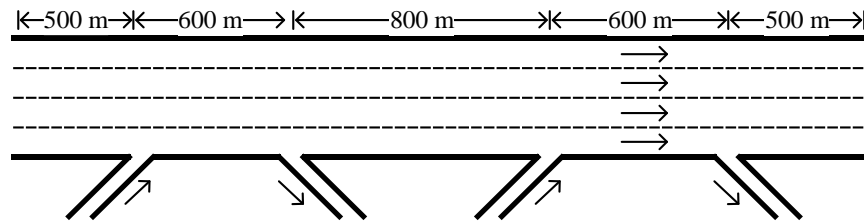


Fig. 39. Illustration of the Hypothetical Corridor.

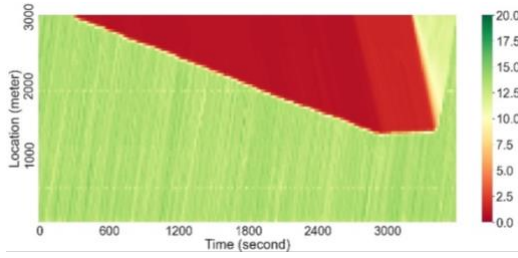
Table 15 Configurations of Virtual Traffic Detectors.

Detector name	Configurations
Loop detector	Location: 800 meters and 2200 meters from the corridor upstream
GPS	Aggregation time interval: 1 minute Sampling rate: 6% Reporting frequency: 30 seconds
Bluetooth detector	Location: 1200 meters and 1800 meters from the corridor upstream
Video detector	Sampling rate: 5% Location: 200 meters to 300 meters and 2700 meters to 2800 meters from the corridor upstream

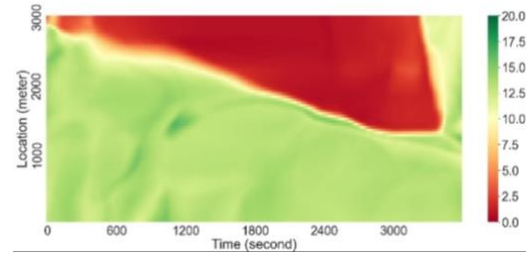
To perform distributed computing, the corridor is evenly split into three parts. As introduced in Section 4.6.3, a computational graph is built for each part to estimate the traffic states on that part, and only the necessary information is shared between two adjacent corridor parts to reach state consistencies on the boundary. As a result, each graph training can be conducted in parallel and independently, which could help improve the training efficiency and system robustness. The three computational graphs used in this section have the same settings as those described in Section 4.7.1.

Fig. 40 shows the speed observation and estimation on the hypothetical corridor. Fig. 40(a) shows the observed speed profile, which serves as the ground truth. Fig. 40(b) and Fig. 40(c) depict the estimated speed profiles obtained using centralized computing and distributed computing, respectively. As shown in Fig. 40(b), the queue formation and dissipation have been successfully reproduced. In terms of the overall accuracy of the speed estimation, MAE and MAPE of Fig. 40(b) are 0.75 m/s and 13.9%, respectively. The following findings can be concluded when comparing Fig. 40(b) and Fig. 40(c). First, the overall speed estimation profiles produced using centralized computing and distributed computing are very close, indicating the effectiveness of the proposed distributed

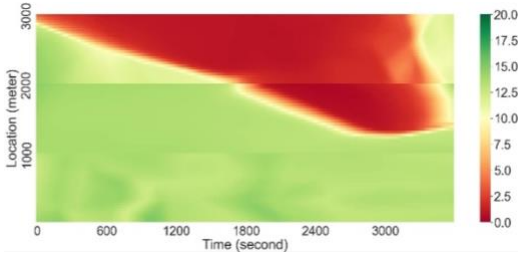
computing framework. Second, in the bottom left of Fig. 40(b), unsmooth speed evolutions can be observed, whereas it is better in Fig. 40(c), especially within the range of 1–2 km. Under a centralized computing framework, state distribution functions need to fit traffic states on the entire space-time regime of interest, which results in high complexity if the target regime is large. On the other hand, under a distributed computing framework, state distribution functions will be built and fit on each separate smaller regime, and the complexity of the distribution functions can be significantly reduced. Therefore, better state estimations on local regimes can be expected from a distributed computing framework than from a centralized computing framework. Third, in Fig. 40(c), minor speed discontinuities at the boundaries (i.e., 1 and 2 km) can still be observed. This issue can be addressed by increasing the weight of consistency loss on the boundaries while training the computational graph. However, this may affect the estimation accuracy for each local regime. Therefore, there is a tradeoff between local estimation accuracy and state consistency on boundaries, and balancing this tradeoff is also a research direction for future work.



(a) Observed speed profile



(b) Estimated speed profile using centralized computing



(a) Estimated speed profile using distributed computing

Fig. 40. Comparison Between Speed Observations and Estimations on a Hypothetical Corridor.

4.8 Conclusions

Focusing on traffic system state estimation (TSSI), this chapter presented an integrated framework for simultaneous traffic state estimation, model parameter estimation, and queue profile estimation in connected and automated mobility systems. Based on the fluid queue approximation at the macroscopic level and the continuous space-time distribution function representation scheme at the mesoscopic level, the TSSI problem was formulated with a nonlinear optimization model, which was then solved on a layered computational graph using the forward-backward algorithm. Numerical experiments based on real-world and hypothetical datasets were designed to demonstrate the effectiveness of the proposed estimation framework.

CHAPTER 5

INTEGRATED CITY LOGISTICS OPERATION OPTIMIZATION IN CAM SYSTEMS

5.1 Introduction

In a broader sense, city logistics refers to the management of the flow of goods and services from providers to customers in urban areas. The urban management movement (UMM) problem (Cattaruzza et al., 2017), as an example, aims to find an optimal set of routes for a fleet of vehicles to satisfy requests for the development, public maintenance, and other functional needs in a city. An efficient city logistics system helps to reduce operation cost, mitigate traffic congestion impact, protect the environment, respond to climate change, connect underserved communities, and support economic vitality. Considering the traffic congestion experienced in cities, this type of problems needs to be formulated as rich arc routing problems (RARPs) under congested traffic conditions.

The word “rich” is used in RARP because besides normal constraints in the variants of the standard ARP (e.g., capacity and time window constraint), some other problem-specific constraints are also considered. For example, in the winter gritting problem, service requests on road links change with time and weather conditions (Eglese and Li, 1992; Eglese, 1994; Li and Eglese, 1996; Tagmouti et al., 2007; Tagmouti et al., 2010; Tagmouti et al., 2011), or in the snow plowing problem, truck routes are specified at the lane level (Perrier et al., 2007a; Perrier et al., 2007b; Salazar-Aguilar et al., 2012; Dussault et al., 2013; Dussault et al., 2014; Quirion-Blais et al., 2017; Castro Campos et al., 2020). Rich constraints considered in RARPs have great practical significance and values while, at the same time, they bring additional challenges, especially for large-scale instances. By

fully recognizing rich features in transportation networks, this research aims to develop a modeling framework and solution approaches for RARPs that can systematically examine traffic-oriented characteristics.

One of the most important “rich” features of transportation networks is time-varying traffic conditions. Service vehicles may experience time-dependent travel times on roads when serving customers (Liu et al., 2020; Yao et al., 2021). In early research on both vehicle routing problems (VRPs) and ARPs, travel times on links are treated as constant or time-independent; however, in a congested urban environment, solutions obtained with the constant travel time assumption may significantly underestimate the delay and could even lead to infeasibility under tight schedules in real-life applications. Although many recent VRP studies considered piecewise travel time functions to capture time-varying traffic conditions in a more realistic fashion, as discussed by Vidal et al. (2021), additional efforts are still critically needed to offer more precise approximations/representations of reality. On the other hand, in the ARP literature, time-dependent travel times have been largely simplified or ignored (Gendreau et al., 2015). Vidal et al. (2021) first conducted extensive studies on the ARP with time-dependent travel times at a network level and proposed methods for quick travel and service time queries as well as quickest path queries, based on the travel speed function definition given by Ichoua et al. (2003).

Another important goal of RARPs on transportation networks is how to reduce the system-wide (societal) congestion impact of service vehicle routings. RARP applications in city logistics are typically fulfilled by large trucks, e.g., freight trucks and street sweeping trucks. A service truck with a much lower driving speed could affect traffics on multiple lanes which leads to another important class of moving bottleneck problems

studied in the literature (see e.g., Li et al., 2020). Thus, RARP applications in city logistics should minimize not only the total operating cost for meeting customer requests but also the potential negative effects to the background transportation system.

The main contribution of this chapter includes:

- (1) recognizing the potential negative impacts of service vehicles to background traffics when providing services, this research systematically considers service vehicle operation cost and system (societal) impact of vehicle routings in city logistics so as to reduce the system-wide congestion impact.
- (2) based on the fluid queue model, a novel time-dependent travel time representation with the form of nonlinear function is introduced. Compared to the widely adopted piecewise linear functions, the proposed nonlinear travel time function has advantages on parsimonious form, easy calibration, and differentiability. This also leads to the analytical derivations of time-dependent system (societal) impact of service vehicles to background traffics.
- (3) three optimization models are developed from different perspectives for modeling RARPs in city logistics, in which the impacts of problem-specific rich constraints on modeling complexity are comprehensively investigated.
- (4) with a real-life sprinkler truck routing problem (SRP) as the representative example of RARP, this research develops two exact solution algorithms, namely a Lagrangian relaxation based method and a branch-and-price based method which are embedded with an enhanced parallel branch-and-bound algorithm.

5.2 Modeling Time-dependent Travel Time and Congestion Impacts Using Fluid Queue Models with Polynomial Arrival Rates

To calculate the travel time of a specific link, one can divide link length by travel speed, which is in turn derived from simulated/estimated flow/density based on empirically calibrated fundamental diagrams (Greenshields et al., 1935). However, as pointed out by Van Woensel et al., (2007), in the context of vehicle routing problems, instantaneous speed and density have to be embedded in another set of continuum flow models (Kuhne and Michalopoulos, 1997) to characterize congestion evolution. As an alternative method, queuing models hold the promise for modeling travel flows with the capability of offering analytical evaluation and sensitivity analysis (Heidemann, 1996). However, the queueing-based approach is mainly based on the stochastic queueing principle with under-saturated conditions such as M/M/1 or G/G/1 (Van Woensel et al., 2007). By adapting and extending the fluid queue model with quadratic arrival rates proposed by Newell (2013) to represent time-dependent travel times at both link and path levels, this section introduces analytical forms that satisfy FIFO conditions and offer precise congestion impact measures during a period of oversaturation. Compared to the widely used piecewise linear functions, the proposed method has the following three advantages: (1) a parsimonious form, (2) an analytical expression with FIFO property, and (3) differentiable.

5.2.1 Time-dependent Travel Time

Based on free-flow speed or cut-off speed, which is more precisely defined in congestion bottleneck identification (Hale et al., 2016), traffic states on transportation networks can be classified into two distinct classes: uncongested state and congested state.

Under uncongested states or non-peak hours, aggregated vehicle speed on each road link is relatively stable and is approximated using free-flow speed in this research. This assumption is consistent with the constant free-flow speed in the uncongested regime of triangular fundamental diagram, which is adopted in the widely used cell transmission model (Daganzo, 1994). Under congested states, vehicle travels are constrained by road capacity. Queues form when total inflow travel demand exceeds road capacity. In this case, travel time of vehicles includes two parts: (1) free-flow travel time, and (2) time spent in the queue (travel delay). Due to the dynamics of travel inflow demands, queue length evolves for an extended congestion period, resulting in time-dependent travel time of vehicle trips. This study mainly focuses on time-dependent vehicle travel time calibration during each single congested period and use constant or piecewise-linear travel time functions during uncongested periods.

Without loss of generality, each congested road link on transportation networks is modeled as a single queuing system with a constant service rate that equals to the maximum link discharge rate. Fig. 41 shows a graphical illustration of queue evolution for a road link in the classical work by Newell (2013).

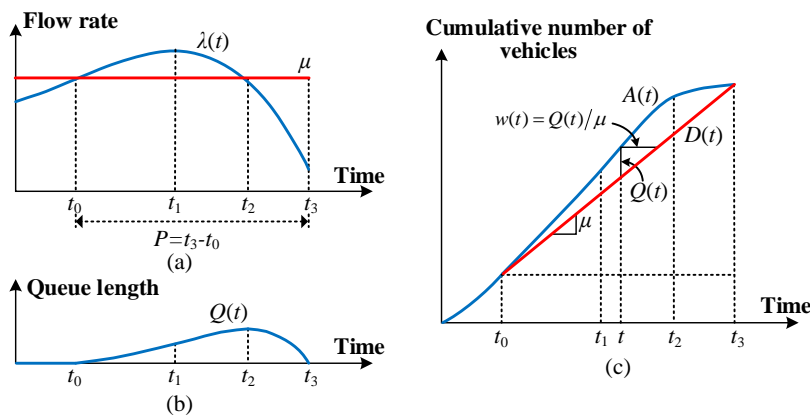


Fig. 41. General Graphical Illustration of Queue Evolution for a Road Link, Adapted from Newell (2013).

In Fig. 41(a), the blue curve and red straight line represent the arrival rate $\lambda(t)$ and departure (discharge) rate μ , respectively. t_0 , t_2 , and t_3 denote the time at which queue starts to form, queue starts to dissipate, and queue disappears; t_1 is the time with the highest arrival rate. By using the second-order Taylor approximation at time t_1 , arrival rate $\lambda(t)$ can be approximated by the following quadratic function:

$$\lambda(t) = \lambda(t_1) + \lambda'(t_1)(t - t_1) + \frac{1}{2}\lambda''(t_1)(t - t_1)^2. \quad (77)$$

With the observation that $\lambda'(t_1) = 0$, Eq. (77) can be simplified as

$$\lambda(t) = \lambda(t_1) - \gamma(t - t_1)^2, \quad (78)$$

where $\gamma = -\frac{1}{2}\lambda''(t_1)$ ($\gamma > 0$). Notice that $\lambda(t)$ passes two points (t_0, μ) and (t_2, μ) , arrival rate $\lambda(t)$ can also be expressed by the following factored form:

$$\lambda(t) = \gamma(t - t_0)(t_2 - t) + \mu. \quad (79)$$

With Eq. (79), time-dependent queue length $Q(t)$ can be derived as follows:

$$Q(t) = A(t) - D(t) = \int_{t_0}^t [\lambda(\tau) - \mu] d\tau = \gamma(t - t_0)^2 \left[\frac{t_2 - t_0}{2} - \frac{t - t_0}{3} \right], \quad (80)$$

where $A(t)$ and $D(t)$ denote the cumulative arrival and departure at time t , respectively. By introducing the queue clearance time t_3 (i.e., $Q(t_3) = 0$), the following relationship between critical time points can be derived from Eq. (80). The detailed derivation process can be found in Newell (2013) for the quadratic arrival rates and cubic arrival rates by Cheng et al. (2022).

$$t_3 = t_0 + \frac{3}{2}(t_2 - t_0). \quad (81)$$

Then, Eq. (80) can also be written as

$$Q(t) = A(t) - D(t) = \int_{t_0}^t [\lambda(\tau) - \mu] d\tau = \frac{\gamma}{3}(t - t_0)^2(t_3 - t). \quad (82)$$

The discharge rate μ , queue forming time t_0 , and queue clearance t_3 can be directly observed from field data. Thus, time-dependent queue length $Q(t)$ in Eq. (82) only has one inflow demand curvature parameter γ that needs to be calibrated from observed spatial queue length or link travel times. Interested readers can refer to a recent paper by Cheng et al. (2022) for the detailed calibration process that connects the above queuing model with the observations from a spatial queue representation. Finally, by integrating $w(t) = Q(t)/\mu$, time-dependent delay $w(t)$ can be expressed as

$$w(t) = \frac{\gamma}{3\mu} (t - t_0)^2 (t_3 - t). \quad (83)$$

In many city logistics applications, only a single data source of observed speed v_t^{obs} is available, where observation time interval t can be 5 minutes or 15 minutes. The following paragraphs describe four steps for calibrating the key parameters of μ and γ in Eq. (83), and the corresponding graphic illustration is provided in Fig. 42.

- (1) Even without the flow count observations, an estimate of the ultimate road capacity c can still be obtained according to the facility types and speed limit. The cutoff speed can be determined based on the well-established traffic fundamental diagram between flow, density and speed, such as Greenshields model (Greenshields et al., 1935).
- (2) One can determine the congestion duration P , while t_0 and t_3 correspond to the timestamps at which speed is dropping from or recovering to the cutoff speed.
- (3) The average discharge rate μ can be estimated from the volume-speed curve and the observed speed during the congestion duration. A default value of μ for undersaturated links can be ultimate road capacity c .
- (4) The space-mean speed v_t^{obs} can be converted to the virtual waiting time $w(t)$ in

Eq. (83), and one can use nonlinear regression methods to calibrate parameter γ accordingly. Alternatively, as $w(t_2) = \frac{\gamma}{6\mu}(t_2 - t_0)^3$, parameter γ can also be quickly estimated based on the lowest speed and converted highest waiting time.

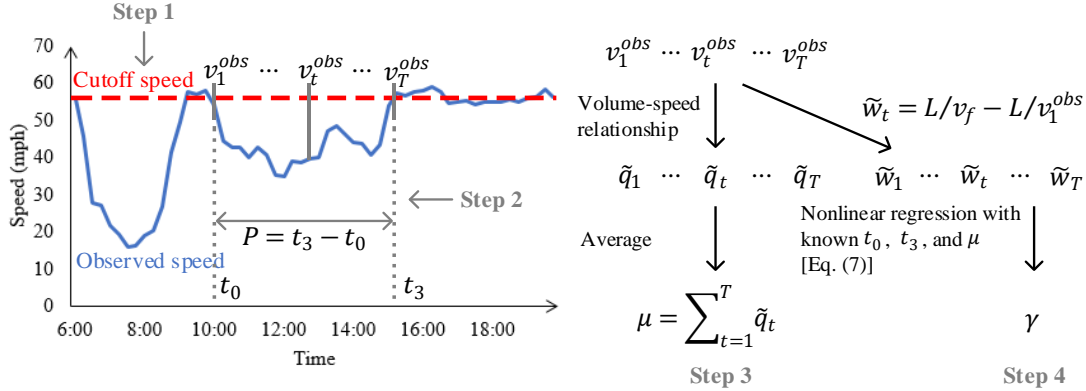


Fig. 42. Graphic Illustration of the Four-step Method for Calibrating the Key Parameters in Eq. (83).

It should be noted that, as derivations are based on the point queue model (Vickrey, 1963), $w(t)$ in Eq. (83) represents the delay of a link for vehicles arriving the downstream node (stop line) of the link at time t . With link free-flow travel time $FFTT$, time-dependent travel time of a link during congested periods (denoted by $LT(t)$) can be calculated using Eq. (84) for vehicle leaving from the upstream node of a link at time t .

$$LT(t) = w(t + FFTT) + FFTT = \frac{\gamma}{3\mu}(t + FFTT - t_0)^2(t_3 - t - FFTT) + FFTT. \quad (84)$$

The range of feasibility for parameter γ in Eq. (84) is derived as follows. For a queuing system on transportation networks, the arrival and departure rates must be positive. As departure rate μ is a constant parameter, it is also required that arrival rate $\lambda(t)$ during the analysis period be nonnegative for any t between t_0 and t_3 . The quadratic function $\lambda(t)$ has the lowest value at time t_3 in the congestion period. That is, $\lambda(t_3) \geq 0$ ensures $\lambda(t) \geq 0, \forall t \in [t_0, t_3]$. Integrating Eq. (79) and Eq. (81) yields Eq. (9):

$$\lambda(t_3) = \gamma(t_3 - t_0)(t_2 - t_3) + \mu = \gamma(t_3 - t_0) \left(\frac{2}{3}t_3 + \frac{1}{3}t_0 - t_3 \right) + \mu = -\frac{1}{3}\gamma P^2 + \mu \geq 0, \quad (85)$$

$$\mu \geq 0,$$

where $P = t_3 - t_0$ represents the congestion duration.

Proposition 1. The time-dependent link travel time function $LT(t)$ in Eq. (84) satisfies the FIFO property within the time period of interest $[t_0, t_3]$.

Proof. The FIFO property can be proved if $t + LT(t) \leq t' + LT(t')$ holds for any $t \leq t', t \in [t_0, t_3], t' \in [t_0, t_3]$. Alternatively, it can be proved if $\frac{dLT(t)}{dt} \geq -1$ holds for any $t \in [t_0, t_3]$ (Carey et al., 2014). Based on the derivation of $LT(t)$ given in Eq. (84), $\frac{dLT(t)}{dt}$ can be expressed as

$$\frac{dLT(t)}{dt} = \frac{d\left[\frac{\gamma}{3\mu}(t-t_0)^2(t_3-t) + FFFT\right]}{dt} = \frac{d\left[\frac{\gamma}{3\mu}(t-t_0)^2(t_3-t)\right]}{dt}. \quad (86)$$

Let $h = t - t_0, h \in [0, P]$, Eq. (86) can be rewritten as

$$\frac{dLT(t)}{dt} = \frac{dLT(h+t_0)}{dh} \frac{dh}{dt} = \frac{d\left[\frac{\gamma}{3\mu}h^2(P-h)\right]}{dh} = \frac{\gamma}{3\mu}(2hP - 3h^2). \quad (87)$$

It is easy to observe that $\frac{\gamma}{3\mu}(2hP - 3h^2)$ is a quadratic function of $h \in [0, P]$ and reaches its minimum at $h = P$. That is,

$$\frac{dLT(t)}{dt} = \frac{\gamma}{3\mu}(2hP - 3h^2) \geq \frac{\gamma}{3\mu}(2PP - 3P^2) = -\frac{\gamma}{3\mu}P^2. \quad (88)$$

By utilizing the range of feasibility for γ derived in Eq. (85), it is obvious that $-\frac{\gamma}{3\mu}P^2 \geq -1$, indicating $\frac{dLT(t)}{dt} \geq -1, \forall t \in [t_0, t_3]$. Thus, function $LT(t)$ satisfying the FIFO property is proved.

On the basis of the time-dependent link travel time function, travel time derivations and FIFO property proof are further performed on a path level. Consider two arbitrary nodes i and j on a transportation network, and there are φ paths from i to j . Take path $\rho \in \{1, 2, \dots, \varphi\}$ as an example, the travel time of path ρ at time t ($PT_\rho(t)$) can be calculated as

$$PT_\rho(t) = \sum_{l=1}^L LT_l(a_l^t), \quad (89)$$

where $l \in \{1, 2, \dots, L\}$ denotes the index of links in path ρ ; L is the total number of links in path ρ ; LT_l represents the link travel time function of the l th link in path ρ ; a_l^t is the arrival time at the upstream node of the l th link if departure from node i at time t . $a_1^t = t$, $a_l^t = a_{l-1}^t + LT_{l-1}(a_{l-1}^t)$ when $2 \leq l \leq L$.

Proposition 2. Path travel time function $PT_\rho(t)$ in Eq. (89) satisfies the FIFO property.

Proof. Consider two vehicles, v^1 and v^2 , departing along the same path ρ at time t and time t' respectively, and $t \leq t'$. That is, $a_1^t \leq a_1^{t'}$. As the travel time function of link 1 satisfies the FIFO property, $a_2^t = a_1^t + LT_1(a_1^t) \leq a_2^{t'} = a_1^{t'} + LT_1(a_1^{t'})$. Note that all links along path ρ satisfy the FIFO property, it means $a_3^t = a_2^t + LT_2(a_2^t) \leq a_3^{t'} = a_2^{t'} + LT_2(a_2^{t'})$, and apply the process recursively till the last link L , i.e., $a_L^t = a_{L-1}^t + LT_{L-1}(a_{L-1}^t) \leq a_L^{t'} = a_{L-1}^{t'} + LT_{L-1}(a_{L-1}^{t'})$. Thanks to the FIFO property on link L , it can be easily concluded that vehicle v^1 will arrive at node j earlier than vehicle v^2 . Thus, path travel time $PT_\rho(t)$ satisfying the FIFO property is proved.

It should be noted that the derived path travel time might still have multiple local minima and maxima, but its structurally parsimonious form could balance the tradeoff

between computational tractability and the required level of details in representing real-world traffic congestion.

5.2.2 Analytical Form of Modeling System-wide Congestion Impacts to Background Traffic

As mentioned earlier, service vehicles used in RARP applications are typically slow-moving trucks, thus may bring significant impacts to background traffics, especially during peak hours. This subsection further utilizes calibrated queuing profile to analytically measure the system impact of service vehicles, by following the approach proposed by Ghali and Smith (1995).

In Fig. 43, the blue and red solid lines denote the cumulative arrival and departure on a road link, respectively. Let us assume that there is a service truck entering the link at time t . Note that, similar to Section 5.2.1, this subsection still focuses on the congested period, i.e., $t_0 \leq t \leq t_3$. The marginal delay arising from the service truck is the blue dash area, which also equals to the grey area. The marginal delay includes two parts $w(t)$ and $SI(t)$. $w(t)$ is the delay experienced by the service truck, while $SI(t)$ denotes the additional delay experienced on that link by every vehicle arriving between time t and t_3 , due to the arrival of the service truck at time t , which is called system-wide (societal) congestion impact in this research. Therefore, with the derivation of $w(t)$ in Section 5.2.1, $SI(t)$ can be calculated as follows:

$$SI(t) = t_3 - t - w(t) = t_3 - t - \frac{\gamma}{3\mu}(t - t_0)^2(t_3 - t). \quad (90)$$

With the consideration that service trucks typically move slower than passenger cars and use more road resources, $SI(t)$ obtained in Eq. (90) further is multiplied by a

passenger car equivalent (PCE), i.e., $SI(t) \times PCE$, to measure the system impact of a service truck entering a road link at time t .

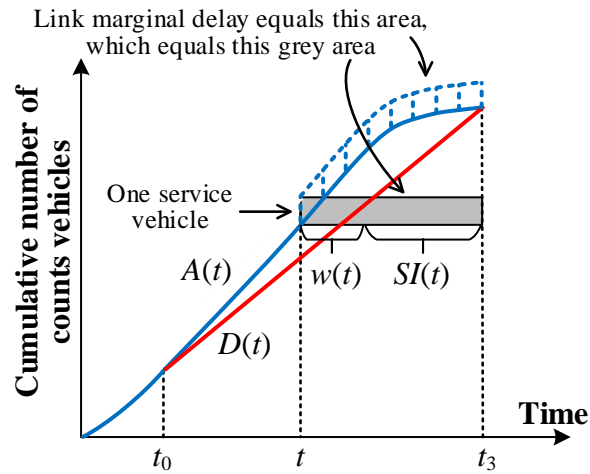


Fig. 43. Illustration of Road Link Marginal Delay Adapted from Ghali and Smith (1995).

With the link-level marginal cost formulation in Eq. (90), system-wide impact along a given path can also be recursively derived similar to travel time derivations in Section 5.2.1.

5.3 Sprinkler Truck Routing: An Arc Routing Application with Rich Constraints in a Congested Traffic Network

Vehicle travels as well as the action of wind could incur serious airborne particulate matters from roads (Li et al., 2008), causing dust emissions that bring substantial negative effects to surrounding workers and pedestrians. The dust in the air also reduces the visibility of roads and is therefore likely to cause traffic accidents (Bhattachan et al., 2019). Street watering constitutes one of the most common and essential services provided by municipal departments (Gambatese et al., 2001). In such a service, sprinkler trucks, also known as water carts or water trucks, are assembled into a fleet to spray water and wash

the road surface alongside streets in urban networks. As a crucial component in street watering operation systems, sprinkler truck route design aims to determine a set of optimal routes for a fleet of sprinkler trucks such that total cost is minimized while road cleaning tasks can be completed as required.

This study uses the SRP as an example to illustrate the modeling framework and solution approaches for RARPs on urban networks. The SRP studied in this work is essentially a capacitated arc routing problem with time window (CARPTW), with the consideration of following additional rich features:

- (1) time-varying traffic conditions on urban transportation networks,
- (2) turn delays at intersections,
- (3) repeated cleaning services on certain links, and
- (4) water refilling at water refilling stations.

It should be noted that the rich constraints listed above can also be generalized to many other city logistics applications. For example, for the emerging electric vehicle routing problem in green logistics, the rich constraint (4) can be changed to charging vehicles at charging stations, without changing the essence of the resulting problem.

For a given transportation network $G = (N, L)$, where N and L denote the set of nodes and road links respectively, the SRP studied in this research is to find a set of routes for sprinkler trucks such that all cleaning tasks can be fulfilled as required and the total cost is minimized. Each link $(i, j) \in L$ is associated with a cleaning task. That is, link (i, j) must be cleaned $m_{i,j} \in Z$ times within its time window $[s_{i,j}, e_{i,j}]$, where $s_{i,j}$ and $e_{i,j}$ denote the earliest and latest service starting time, respectively. To clean link (i, j) once, a

sprinkler truck will consume $w_{i,j}$ unit of water. Some nodes belonging to N also serve as water refilling stations. For simplicity, this research considers the following three assumptions on water consumption and refilling: (a) once a sprinkler truck starts to clean a road link, it must clean the whole link. In other words, cleaning part of a link then going to refill water is not allowed; (b) a sprinkler truck is always refilled to its maximum water tank capacity when visiting a water refilling station, and the time used for refilling is fixed, e.g., 5 minutes, no matter how much water left before visiting a water refilling station; (c) sprinkler trucks are full of water when departing from their origin depot. Due to the existence of assumption (a), the water consumption $w_{i,j}$ of each link (i,j) should not exceed the maximum water capacity C of sprinkler trucks. In the case that $w_{i,j}$ is larger than C , link (i,j) will be split into multiple short links, with each of which meeting the requirements mentioned above.

The total cost to be optimized is calculated by $TOC + \omega \times TSIC$, where TOC and $TSIC$ represent total operating cost and total system-wide impact cost respectively; parameter ω is a user-defined weight of $TSIC$ to measure the importance of societal cost in routing solutions. The societal impact of using a specific link has been derived in Eq. (90). The total operation cost TOC consists of sprinkler truck acquisition cost and total travel time cost. In this research, all sprinkler trucks used are identical, and acquisition cost h will be applied for each used sprinkler truck. Besides, the number of available sprinkler trucks is unlimited. The travel time cost of each sprinkler truck equals to the time difference between leaving depot and returning back to the depot, which further consists of four parts: cleaning time, deadheading time, waiting time, and water refilling time. Deadheading means a sprinkler truck traverses a road link without cleaning the link. It may occur in two

situations: (a) the link does not need to clean or has been cleaned; (b) a sprinkler has used up water and is heading to a water refilling station. For each road link, its time-dependent deadheading time $DT(t)$ and cleaning time $CT(t)$ are calculated as follows:

$$DT(t) = w(t) + LL/v_d, \quad (91)$$

$$CT(t) = \max[DT(t), LL/v_c], \quad (92)$$

where $w(t)$ is time-dependent link delay derived in Eq. (83); LL represent the length of the link, v_d and v_c denote the maximum speed of sprinkler trucks in the deadheading mode and cleaning mode respectively. Waiting time represents the time gap between the arrival time and service starting time of a sprinkler on links (due to time window), rather than the waiting time at intersections caused by control delay. Note that sprinkler trucks are not allowed to wait if they do not have to, e.g., keep staying at a link after cleaning service is complete.

Fig. 44 shows an illustrative example with a simple network and a route of a sprinkler truck. The network consists of 17 nodes (intersections) indexed from 0 to 16. The depot is located at node 0, and two water refilling stations are located at node 11 and 12 respectively. The orange line denotes an example route, which is composed of solid lines (truck in cleaning mode) and dash lines (truck in deadheading mode). In Fig. 44, the sprinkler truck starts from depot node 0, cleans links (0,4), (4,1), (2,5), (5,8), refills water at node 11, cleans links (16,15), (15,14), (14,13), (13,7), (7,4), (4,0), and returns back to depot node 0.

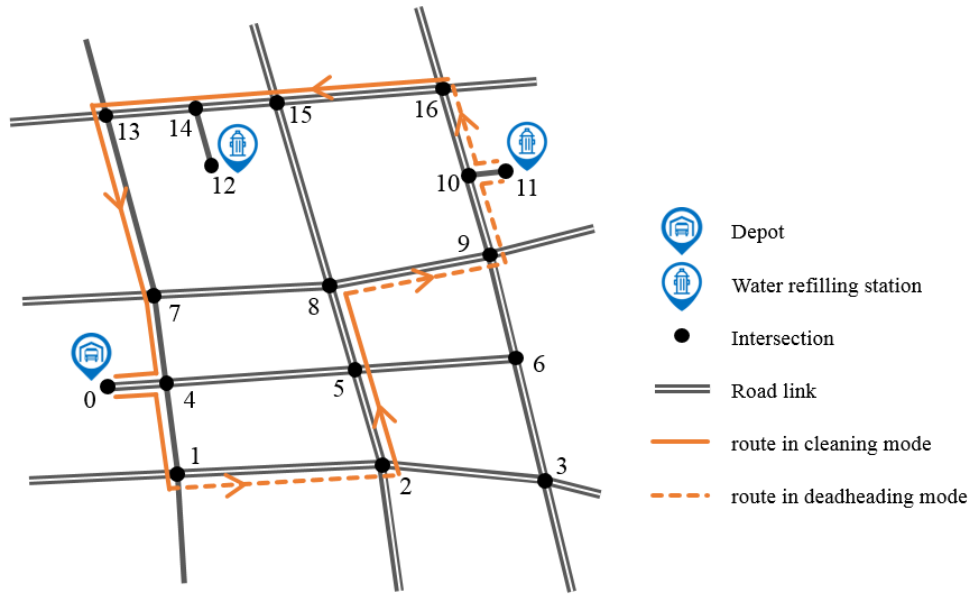


Fig. 44. An Illustrative Network and the Route of a Sprinkler Truck.

5.4 Mathematical Formulations of the Optimization Problem

In this section, three models are developed for the proposed SRP from different perspectives. Specifically, a discretized time-expanded network based arc routing model M1 in Section 5.4.1, an arc-based node routing model M2 in Section 5.4.2, and a path-based node routing model M3 in Section 5.4.3. Section 5.4.4 offers a comprehensive comparison.

5.4.1 An Arc Routing Model Based on Time-expanded Network (Model M1-TEN)

A standard VRP or ARP typically assumes, each customer must be served once and exactly once. Benefitting from the assumption, vehicle states (arrival time, cumulative loads) can be associated with customers, thus a concise physical network based model can be built (Cordeau,2006). However, in the SRP considered in this study, some road links may be required to be cleaned multiple times, implying a road link may be serviced

multiple times by sprinkler trucks, which makes vehicle states intractable if associating them with physical networks. As a result, this research adopts a time-expanded network based modeling approach. In the literature, the time-expanded network based modeling approach has been successfully applied in solving a wide range of transportation supply-side optimization problems, e.g., dynamic traffic assignment (Lu et al., 2016), VRP (Yao et al., 2019), passenger flow state estimation (Shang et al., 2019), and train timetabling problem (Zhang et al., 2019). By extending a physical network to a time-expanded network, nodes and road links in the original physical network are extended to vertexes and arcs with an extra time dimension. With the extended time dimension, the multiple-services requirements can be systematically modelled as well as time-dependent travel times. To enable model M1 to accommodate the need of turn delay modeling, with the intersection expansion process similar to the approach discussed by Kirby and Potts (1969), Ziliaskopoulos and Mahmassani (1996), and Pallottino and Scutella (1998), a new network $G_j = (N_j, L_j)$ is first constructed from the network G described in Section 5.3.

For a time-expanded *network* G_{ST} from G_j , vertex $(i, t) \in V$ is constructed from physical node $i \in N_j$, where t denotes time. Arc $(i, j, t, t') \in A$ represents a space-time traveling activity from vertex (i, t) to vertex (j, t') . As the time dimension is continuous, the entire planning horizon $[0, T]$ is evenly discretized into short time intervals, e.g., 10 seconds, so that the number of vertexes and arcs is finite. As a result, t and t' both represent the indices of discretized time intervals. $\chi = \{0, 1, 2, \dots, s\}$ denotes the set of indices of discretized time intervals, where s is the index of the last time interval. Arcs in a time-expanded network consist of two categories: traveling arc (i, j, t, t') and waiting arc $(i, i, t, t + 1)$. A traveling arc (i, j, t, t') means a vehicle enters link $(i, j) \in L_j$ at time t

and leaves at time t' , and $t' - t$ equals the travel time of the link at time t . Note that link (i, j) can either be a road link or a movement link in network G_j . A waiting arc $(i, i, t, t + 1)$ corresponds to the waiting activity of a vehicle at node $i \in N_j$ for one time interval. Waiting arcs are used when a sprinkler truck arrives at a road link earlier than the link's earliest service starting time. Fig. 45 depicts a simple three-node network with its corresponding time-expanded network, where the travel time of link $(1,2)$ and link $(2,3)$ are 1 and 2, respectively.

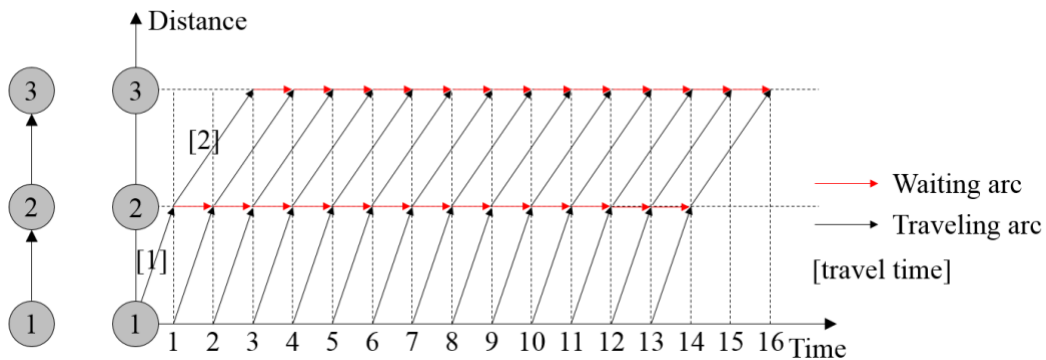


Fig. 45. A Simple Network and Its Corresponding Time-expanded Network.

With the basic concepts introduced above, the time-expanded network built for the SRP is presented in Fig. 46, with the following problem-specific highlights:

- (1) Cleaning arc and deadheading arc: When a sprinkler moves on a road link, it has two possible modes, i.e., cleaning mode and deadheading mode. In the cleaning mode, the sprinkler truck cleans the link when traveling on it, but not in the deadheading mode. Note that sprinkler trucks typically have different speeds in the cleaning mode and deadheading mode. Therefore, it is necessary to build cleaning arcs (Fig. 46(a)) and deadheading arcs (Fig. 46(b)) with different travel times for each physical link. For a cleaning arc (i, j, t, t') in Fig. 46(a), $t' - t$ equals to cleaning time on link (i, j) ; for a

deadheading arc (i, j, t, t') in Fig. 46(b), $t' - t$ equals to deadheading time on link (i, j) . Another difference between cleaning arcs and deadheading arcs is about water consumption. $w_{i,j,t,s}$ denotes the water consumption of arc (i, j, t, t') , then $w_{i,j,t,t'} = w_{i,j}$ if the arc is a cleaning arc; otherwise, $w_{i,j,t,t'} = 0$.

(2) Time-dependent travel time: For a deadheading arc (i, j, t, t') , $t' - t$ equals to $DT(t)$ in Eq. (91); for a cleaning arc (i, j, t, t') , $t' - t$ equals to $CT(t)$ in Eq. (92). One can observe that each traveling arc (including deadheading arc and cleaning arc) in Fig. 46 is associated with its own travel time, which can be viewed as a fine discretized approximation of continuous functions in Eqs. (91) and (92).

(3) Service time window: In the SRP, each link is associated with a service time window, within which cleaning services must be started. In other words, for a specific link, cleaning arcs outside its service time window are not allowed to use. This constraint can be easily imposed by deleting cleaning arcs outside service time windows when building a time-expanded network. For example, in Fig. 46(a), the service time window of link (1,2) is [2,6], then the arcs with a red cross will be deleted.

(4) Waiting arc: According to the problem description in Section 3, waiting is only allowed when a sprinkler truck arrives earlier than link service starting time. Therefore, compared with the illustrative example in Fig. 45, tighter restrictions are considered in building waiting arcs for the SRP to avoid invalid waiting. First, waiting arcs are only built for inbound nodes of road links that need cleaning services. Second, waiting arcs with entering time later than the corresponding link's service starting time will not be generated. For road link (1,2) with time window [2,6] in Fig. 46, waiting arcs $(1,1, t, t + 1)$ are only constructed for $t = 0, 1$.

(5) Water refilling arc: Water refilling arcs $(i, i, t, t + r)$ are built on each water refilling station i , where r denotes the time required to refill a sprinkler truck.

(6) Origin and destination: In the newly generated time-expanded network, vertex $o_{ST}(o, 0)$ and $d_{ST}(d, s)$ will serve as the origin and destination vertex of sprinkler trucks, where o and d denote the origin node and destination node in network G_T ; s is the index of the last discretized time interval in the whole planning horizon. Waiting arcs on destination d are constructed to ensure that sprinkler trucks are able to return back to and keep staying at the destination depot after finishing cleaning tasks.

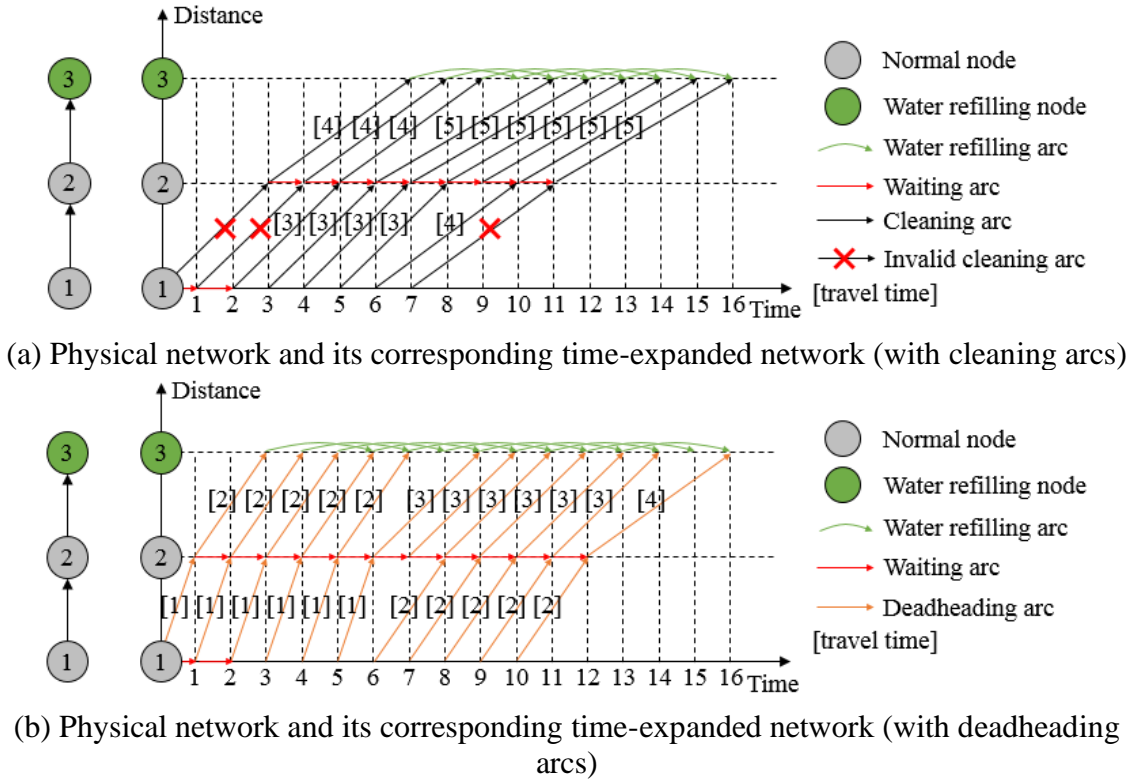


Fig. 46. A Simple Network and Its Corresponding Time-expanded Networks in SRP.

To reduce the size of time-expanded networks and simplify the subsequent optimization model, vertexes and arcs outside the space-time prism between the origin vertex o_{ST} and destination vertex d_{ST} can be future removed. A space-time prism is an

envelope that covers all possible paths between two space-time vertexes. Interested readers are referred to Miller (2005) and Tong et al. (2015).

With the preparation of intersection expansion and time-expanded network construction, the arc routing formulation on time-expanded network (model M1-TEN) for the proposed SRP is presented below. Notations are summarized in Table 16.

Table 16 Notations Used in Model M1-TEN.

Symbols	Definition
Indices	
i, j	Index of nodes in graph G_j
t, t'	Index of time intervals
f	Index of sprinkler trucks
Sets	
L_j	Set of links in graph G_j
F	Set of sprinkler trucks
V	Set of vertexes in graph G_{ST}
A	Set of arcs in graph G_{ST}
$CA_{i,j}$	Cleaning arc set associated with link $(i, j) \in L_j$
RA	Set of water refilling arcs in graph G_{ST}
Parameters	
$o_{ST}(d_{ST})$	Origin (destination) vertex of sprinkler trucks in graph G_{ST}
$ct_{i,j,t,t'}$	Travel time cost of arc $(i, j, t, t') \in A$
$cg_{i,j,t,t'}$	System impact cost of arc $(i, j, t, t') \in A$
$c_{i,j,t,t'}$	General cost of arc $(i, j, t, t') \in A$
$m_{i,j}$	Number of cleaning requests of link (i, j) in graph G_j
$w_{i,j,t,t'}$	Water consumption of arc $(i, j, t, t') \in A$
C	Water tank capacity of sprinkler trucks
r	Water refilling time
h	Sprinkler truck acquisition cost
Variables	
$x_{i,j,t,t'}^f$	Binary variable. $x_{i,j,t,t'}^f = 1$ if sprinkler f uses arc (i, j, t, t') ; otherwise, $x_{i,j,t,t'}^f = 0$
$q_{i,t}^f$	Water level of sprinkler f at vertex (i, t)

Model M1-TEN:

Objective function

$$\begin{aligned} \min Z_1 = & \sum_{f \in F} \sum_{(i,j,t,t') \in A} ct_{i,j,t,t'} x_{i,j,t,t'}^f + h \sum_{f \in F} \sum_{(j,t'):(o,j,0,t') \in A} x_{o,j,0,t'}^f \\ & + \omega \sum_{f \in F} \sum_{(i,j,t,t') \in A} cg_{i,j,t,t'} x_{i,j,t,t'}^f \end{aligned} \quad (93)$$

Subject to:

Flow balance constraint:

$$\begin{aligned} \sum_{(j,t'):(i,j,t,t') \in A} x_{i,j,t,t'}^f &= \sum_{(j,t'):(j,t',t) \in A} x_{j,t',t}^f, \\ \forall f \in F, (i,t) \in V / \{o_{ST}, d_{ST}\} \end{aligned} \quad (94)$$

Cleaning request satisfaction constraint:

$$\sum_{f \in F} \sum_{(i,j,t,t') \in CA_{i,j}} x_{i,j,t,t'}^f = m_{i,j}, \quad \forall (i,j) \in L_j \quad (95)$$

Sprinkler truck water level updating constraint:

$$\begin{aligned} q_{j,t'}^f &\leq q_{i,t}^f - w_{i,j,t,t'} x_{i,j,t,t'}^f + C(1 - x_{i,j,t,t'}^f), \\ \forall f \in F, (i,j,t,t') \in A/RA \end{aligned} \quad (96)$$

Decision variables:

$$\begin{aligned} x_{i,j,t,t'}^f &\in \{0,1\}, \quad \forall f \in F, (i,j,t,t') \in A \\ 0 &\leq q_{i,t}^f \leq C, \quad \forall f \in F, (i,t) \in V \end{aligned} \quad (97)$$

The objective function in Eq. (93) minimizes the total cost to complete cleaning tasks, which includes three parts: travel time cost $\sum_{f \in F} \sum_{(i,j,t,t') \in A} ct_{i,j,t,t'} x_{i,j,t,t'}^f$, vehicle acquisition cost $h \sum_{f \in F} \sum_{(j,t'):(o,j,0,t') \in A} x_{i,j,t,t'}^f$, and system impact cost $\omega \sum_{f \in F} \sum_{(i,j,t,t') \in A} cg_{i,j,t,t'} x_{i,j,t,t'}^f$. The travel time cost $ct_{i,j,t,t'}$ of arc (i,j,t,t') equals to

$t' - t$ for all arcs in set A , except waiting arcs built at sprinkler destination node d . The cost of waiting arcs at node d is set as 0 (i.e., $c_{i,i,t,t+1} = 0$ if $i = d$). Expression $\sum_{f \in F} \sum_{(j,t'):(o,j,0,t') \in A} x_{i,j,t,t'}^f$ denotes the number of sprinkler trucks used. By integrating three types of coefficient of each $x_{i,j,t,t'}^f$, Eq. (93) can be simplified as

$$\min Z_1 = \sum_{f \in F} \sum_{(i,j,t,t') \in A} c_{i,j,t,t'} x_{i,j,t,t'}^f, \quad (98)$$

where $c_{i,j,t,t'}$ represents the general cost of arc (i,j,t,t') , consisting of travel time cost, vehicle acquisition cost, and system impact cost. Constraint (94) guarantees that the incoming flow equals to outgoing flow on vertexes. Note that this constraint is not imposed on the origin vertex $o_{\mathcal{S}\mathcal{T}}$ and destination vertex $d_{\mathcal{S}\mathcal{T}}$. Constraint (95) makes sure that links are cleaned as requested, while constraint (96) is used to update the water level of sprinkler trucks. Finally, constraint (97) specifies decision variables with their domains. Specifically, $x_{i,j,t,t'}^f$ are binary variables, and $q_{i,t}^f$ are positive continuous variables with an upper bound of C , where C represents the water tank capacity of sprinkler trucks.

5.4.2 Graph Transformation and a Slot-based Time Discretization Node Routing Model (Model M2-STD)

The adoption of intersection expansion and time-expanded networks can allow the consideration of various rich constraints into the generated network, contributing to a concise form of model M1-TEN. Yet, additional movement links created at intersections and high-dimension time-expanded networks significantly increase the size of model M1-TEN, making it extremely challenging to solve on large-scale instances. This section further proposes a node routing model by converting the original ARP to a node routing

problem (NRP). Solving an ARP by converting it to an NRP was first proposed by Pearn et al. (1987) and was adopted and improved by Longo et al., (2006). The core idea behind the conversion process is treating arcs to be served as activity nodes in the NRP and building virtual edges to connect each pair of the resulting nodes based on the shortest paths in the original network.

In the proposed SRP, due to the consideration of additional rich features, there are three major challenges during the conversion process. First, on a congested urban transportation network with time-varying traffic conditions, the shortest path (in terms of travel time) between two nodes changes during the day. Second, cost of a path consists of travel time cost and system impact cost, therefore a shortest path with the least travel time between two activity nodes does not necessarily correspond to the best path with the least total cost. Third, due to the existence of service time window, even the best path with the least total cost does not guarantee an optimal solution. Consider two paths (e.g., path 1 and path 2) between a pair of activity nodes, where path 1 has a lower total cost than path 2. However, path 1 takes longer travel time, causing some future service nodes not to be served from path 1 due to time windows. Adopting path 1 with a lower total cost may need additional vehicles to visit these unserved service nodes, which could result in a suboptimal routing solution. To address this issue, φ paths are kept between each pair of activity nodes in the resulting NRP.

Due to the adoption of nonlinear functions for representing time-varying traffic conditions in the converted graph $G_{\mathcal{N}}$, i.e., $\tau_i(t)$, $\tau_{i,j,\rho}(t)$, $cg_i(t)$, and $cg_{i,j,\rho}(t)$, one can expect that an optimization model built on $G_{\mathcal{N}}$ is highly nonconvex and extremely hard to solve. Therefore, in model M2-STD, nonlinear functions are approximated by piecewise-

constant functions. That is, the entire planning time horizon $[0, T]$ is split into multiple time slots with the same duration, e.g., 10 minutes, and nonlinear functions are approximated by constants within each slot. Model M2-STD as well as additional notations is presented as follows.

Table 17 Additional Notations Used in the Node Routing Model.

Symbols	Definition
Indices	
i, j	Index of nodes in graph $G_{\mathcal{N}}$
ρ	Index of edges between each pair of nodes in graph $G_{\mathcal{N}}$
p	Index of time slots
Sets	
$N_{\mathcal{N}}$	Set of nodes in graph $G_{\mathcal{N}}$
$N_{\mathcal{N}}^s$	Set of service nodes in graph $G_{\mathcal{N}}$
$N_{\mathcal{N}}^w$	Set of water refilling stations in graph $G_{\mathcal{N}}$
$E_{\mathcal{N}}$	Set of edges in graph $G_{\mathcal{N}}$
$P_{\mathcal{N}}$	Set of time slots
Parameters	
φ	Number of edges between each pair of nodes in graph $G_{\mathcal{N}}$
$\tau_{i,p}$	Service time of node $i \in N_{\mathcal{N}}$ in slot p
$\tau_{i,j,\rho,p}$	Travel time of edge $(i, j, \rho) \in E_{\mathcal{N}}$ in slot p
$c\mathcal{g}_{i,p}$	Congestion impact of serving node $i \in N_{\mathcal{N}}$ in slot p
$c\mathcal{g}_{i,j,\rho,p}$	Congestion impact of using edge $(i, j, \rho) \in E_{\mathcal{N}}$ in slot p
w_i	Water consumption of node $i \in N_{\mathcal{N}}$
$s_i(e_i)$	The earliest (latest) service starting time of node $i \in N_{\mathcal{N}}$
$T_{i,p}$	Ending time of slot p for node i
Variables	
$x_{i,j,\rho,p}$	Binary variable. $x_{i,j,\rho,p} = 1$ if a sprinkler uses edge (i, j, ρ) in slot p ; otherwise, $x_{i,j,\rho,p} = 0$
$y_{i,p}$	Binary variable. $y_{i,p} = 1$ if cleaning service starts on node i in slot p ; otherwise, $y_{i,p} = 0$
q_i	Water level of a sprinkler at node i
t_i^d	Departure time of a sprinkler from node i
t_i^s	Service starting time of node i

Model M2-STD:

Objective function

$$\begin{aligned} \min Z_2 = & \sum_{f \in F} t_{d_N^f}^d + h \sum_{(j,\rho):(o_N,j,\rho) \in E_N, p \in P_N} x_{o_N,j,\rho,p} \\ & + \omega \left(\sum_{(i,j,\rho) \in E_N, p \in P_N} c g_{i,j,\rho,p} x_{i,j,\rho,p} + \sum_{i \in N_N, p \in P_N} c g_{i,p} y_{i,p} \right) \end{aligned} \quad (99)$$

Subject to:

Cleaning request satisfaction constraint:

$$\sum_{(j,\rho):(j,i,\rho) \in E_N} \sum_{p \in P_N} x_{j,i,\rho,p} = 1, \quad \forall i \in N_N^S \quad (100)$$

Sprinkler truck spatial route constraint:

$$\sum_{(j,\rho):(j,i,\rho) \in E_N} \sum_{p \in P_N} x_{j,i,\rho,p} = \sum_{(j,\rho):(i,j,\rho) \in E_N} \sum_{p \in P_N} x_{i,j,\rho,p}, \quad \forall i \in N_N^S \cup N_N^W \quad (101)$$

$$\sum_{(j,\rho):(j,i,\rho) \in E_N} \sum_{p \in P_N} x_{j,i,\rho,p} \leq 1, \quad \forall i \in N_N^W \cup \{d_N^f | f \in F\} \quad (102)$$

Sprinkler truck temporal route constraint:

$$t_{o_N}^d = 0 \quad (103)$$

$$t_j^s \geq t_i^d + \tau_{i,j,\rho,p} + M(x_{i,j,\rho,p} - 1), \quad \forall (i,j,\rho) \in E_N, \forall p \in P_N \quad (104)$$

$$t_i^d \geq t_i^s + \tau_{i,p} y_{i,p}, \quad \forall i \in N_N, \forall p \in P_N \quad (105)$$

$$\sum_{p \in P_N} y_{i,p} = 1, \quad \forall i \in N_N \quad (106)$$

$$T_{p-1} x_{i,j,\rho,p} \leq t_i^d < T_p + M(1 - x_{i,j,\rho,p}), \quad \forall (i,j,\rho) \in E_N, \forall p \in P_N \quad (107)$$

$$T_{p-1} y_{i,p} \leq t_i^s < T_p + M(1 - y_{i,p}), \quad \forall i \in N_N, \forall p \in P_N \quad (108)$$

Time window constraint:

$$s_i \leq t_i^s \leq e_i, \quad \forall i \in N_N \quad (109)$$

Sprinkler truck water level updating constraint:

$$q_{o_N} = C \quad (110)$$

$$q_j \leq q_i - x_{i,j,\rho,p} w_j + C(1 - x_{i,j,\rho,p}), \quad (111)$$

$$\forall (i, j, \rho) \in \{E_N | i \in N_N^s\}, \forall p \in P_N$$

$$q_j \leq C - x_{i,j,\rho,p} w_j, \quad \forall (i, j, \rho) \in \{E_N | i \in N_N^w \cup \{o_N\}\}, \forall p \in P_N \quad (112)$$

Decision variables:

$$x_{i,j,\rho,p} \in \{0,1\}, \quad \forall (i, j, \rho) \in E_N, \forall p \in P_N$$

$$y_{i,p} \in \{0,1\}, \quad \forall i \in N_N, \forall p \in P_N \quad (113)$$

$$0 \leq q_i \leq C, t_i^d \geq 0, t_i^s \geq 0, \quad \forall i \in N_N$$

The objective function in Eq. (99) aims to minimize the total cost, where o_N represents the origin node of sprinkler trucks in graph G_N , and d_N^f denotes the copy of destination node d_N for sprinkler f in graph G_N . By creating a dummy copy d_N^f of the destination depot for each sprinkler truck f , the arrival time of sprinkler truck f at the destination depot d_N^f is the time used by sprinkler truck f . Note that since service time at destination depots is zero, arrival times and departure times at destination depots are the same, thus departure time $t_{d_N^f}^d$ can be used in the objective function to avoid defining extra variables. Constraint (100) ensures that each service node is serviced exactly once. Constraints (101)-(102) and (103)-(108) formulate the spatial and temporal route constraints of sprinkler trucks, respectively. In constraint (105), $\tau_{i,p}$ denotes the service time of node i if service starts in time slot p . For service nodes, $\tau_{i,p}$ equals to the cleaning time of corresponding road links in slot p . For water refilling stations and depot nodes, $\tau_{i,p}$

equals to the water refilling time and zero, respectively. Constraint (109) makes sure that cleaning services start within preset time windows. Constraints (110)-(112) are used to track the water level updating of sprinkler trucks. Finally, constraint (113) specifies decision variables and their domains used in model M2-STD.

In addition to the essential constraints (100)-(113), two sets of constraints are further constructed below to tighten the linear relaxation of model M2-STD. Specifically, constraint (114) ensures that at least one sprinkler is needed to complete cleaning tasks. Note that, as sprinkler trucks are allowed to refill water during their trips, the minimum number of sprinkler trucks cannot be calculated using total water needed divided by water tank capacity of sprinkler trucks. Although constraint (114) seems to be very loose, it does serve as the lower bound of many instances in the numerical experiment section. Constraint (115) is used to break the symmetry of model M2-STD.

$$\sum_{(j,\rho):(o_{\mathcal{N}},j,\rho)\in E_{\mathcal{N}},p\in P_{\mathcal{N}}} x_{o_{\mathcal{N}},j,\rho,p} \geq 1 \quad (114)$$

$$t_{d_{\mathcal{N}}^1}^d \leq t_{d_{\mathcal{N}}^2}^d \leq \dots \leq t_{d_{\mathcal{N}}^{|F|-1}}^d \leq t_{d_{\mathcal{N}}^{|F|}}^d \quad (115)$$

5.4.3 Path-based Set Partition Formulation with Continuous-time Representation (Model M3-CTR)

This section further proposes a path-based model (set partition formulation) on the graph $G_{\mathcal{N}}$. Instead of time discretization, the original continuous polynomial form of service time function and travel time function is adopted. As pointed by out Boland et al. (2017), the granularity of the discretization has an impact on both candidate solutions and

the computational tractability. Let $r \in \Omega$ denote a feasible sprinkler truck route, where Ω represents the set of all feasible routes. The set partition formulation can be expressed as follows:

Model M3-CTR:

Objective function

$$\min Z_3 = \sum_{r \in \Omega} c_r \theta_r \quad (116)$$

Subject to:

Cleaning request satisfaction constraint:

$$\sum_{r \in \Omega} \alpha_{i,r} \theta_r = 1, \quad \forall i \in N_{\mathcal{N}}^s \quad (117)$$

Decision variables:

$$\theta_r \in \{0,1\}, \quad \forall r \in \Omega \quad (118)$$

The objective function (116) minimizes the cost of all selected routes, where θ_r is a binary variable indicating whether route r is chosen in a solution or not; c_r is the cost of route r , including operation cost and system impact. Constraint (117) states that all service nodes are serviced once, where $\alpha_{i,r}$ is a mapping coefficient between service node i and route r (number of times that route r passes activity node i).

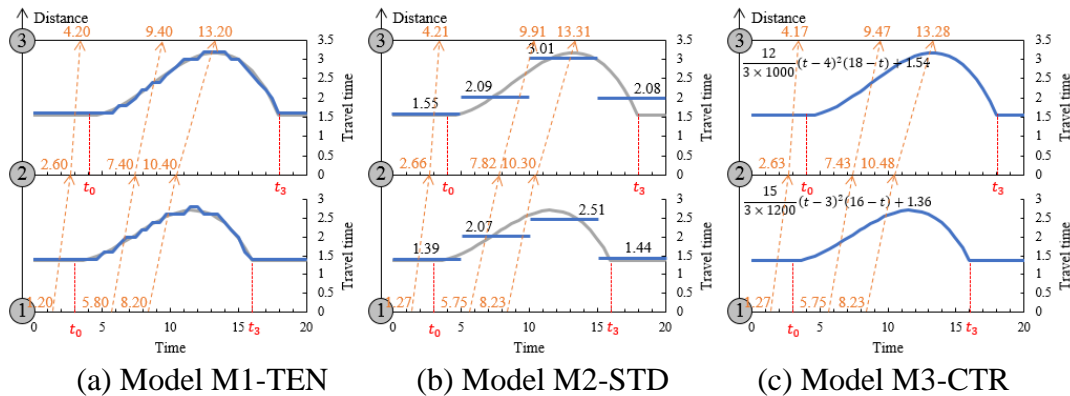
5.4.4 Model Comparison in Terms of Time-varying Travel Time Representation and Model Formulation

An illustrative example with two consecutive links is first used to show the difference in time-dependent travel time modeling among the three models. Table 18 presents the sample values for the parameters of polynomial travel time functions proposed in Section 2. The time unit in this example is minute.

Table 18 Parameters of Polynomial Travel Time Functions.

Link	t_0	t_3	γ	μ	$FFTT$	Link travel time function
1-2	3	16	15	1200	1.36	$LT(t) = \frac{15}{3 \times 1200} (t - 3)^2 (16 - t) + 1.36$
2-3	4	18	12	1000	1.54	$LT(t) = \frac{12}{3 \times 1000} (t - 4)^2 (18 - t) + 1.54$

Fig. 47 shows the travel time functions (blue lines) used in the three models, among which piecewise-constant functions depicted in Fig. 47(b) correspond to the first method in Table 2, while functions in Fig. 47(a) and Fig. 47(c) are the two proposed approaches in this study. For model M1-TEN, the time dimension is discretized into 0.2-minute time intervals. Therefore, both link travel times and node arrival (departure) times are approximated with a 0.2-minute resolution. For model M2-STD, the entire time horizon is split into four 5-minute slots, and travel times remain constant in each slot. For model M3-CTR, the original polynomial form of travel time functions is used.



(a) Model M1-TEN (b) Model M2-STD (c) Model M3-CTR
 Fig. 47. An Illustrative Example of Time-dependent Travel Time Modeling in Three Models, Ranging from High-fidelity Discretization M1-TEN, Semi-dynamic Slot-based Discretization M2-STD and Continuous-time Representation M3-CTR.

5.5 Exact Solution Methods

The models developed in Section 5.4 are all (mixed) integrated linear programming models and theoretically solvable using existing solvers such as Gurobi and Cplex. While, in real-life applications, the size of resulting models could be extremely large, making it hard to obtain desirable solutions from solvers within a reasonable time. In this section, with the recognition of the mathematical characteristics of each proposed model, two exact solution methods are developed for efficiently solving these models. Specifically, a Lagrangian relaxation (LR) based method is proposed for solving model M1-TEN, while a branch-and-price (BnP) based approach is proposed for solving model M3-CTR.

5.5.1 Lagrangian Relaxation

The concise form of model M1-TEN provides the possibility of using decomposition techniques to obtain good solutions quickly. Model M1-TEN consists of three sets of side constraints, among which only cleaning request satisfaction constraint (95) couples different sprinkler trucks. If constraint (95) is relaxed, model M1 can be decomposed into multiple easier sprinkler-specific sub-problems, which can be solved independently. In this research, a LR-based method is proposed for solving model M1-TEN.

Under the LR framework, coupling constraint (95) is relaxed, and the violation is penalized in objective function (93) using Lagrangian multipliers $\lambda_{i,j}$. The relaxed problem is

Objective function

$$\min Z_4 = Z_1 + \sum_{(i,j) \in L_j} \lambda_{i,j} \left(\sum_{f \in F} \sum_{(i,j,t,t') \in CA_{i,j}} x_{i,j,t,t'}^f - m_{i,j} \right) \quad (119)$$

Subject to:

$$(94), (96), (97).$$

By reorganizing the coefficients of decision variable $x_{i,j,t,t'}^f$, the objective function (119) can be further simplified to Eq. (120). Given the value of $\lambda_{i,j}$, the relaxed problem needs to find the least cost shortest path on the discretized time-expanded network with modified arc cost $c'_{i,j,t,t'}$ for each sprinkler truck, which can be exactly solved by the dynamic programming approach (Mahmoudi and Zhou, 2016; Yao et al., 2019).

$$\min Z_4 = \sum_{f \in F} \sum_{(i,j,t,t') \in A} c'_{i,j,t,t'} x_{i,j,t,t'}^f - \sum_{(i,j) \in L_j} \lambda_{i,j} m_{i,j} \quad (120)$$

This research adopts the classical sub-gradient method to update the values of Lagrangian multipliers $\lambda_{i,j}$ using Eq. (121) and Eq. (122), where $\lambda_{i,j}^k$ and α^k denote the value of $\lambda_{i,j}$ and step length at iteration k , respectively. It should be noted that, as constraint (95) is an equality constraint, Lagrangian multipliers can be positive, negative, or 0. A positive Lagrangian multiplier means an additional penalty when serving the corresponding arc, while a negative value means an additional bonus. Lagrangian iteration stops when constraint (95) is satisfied on all links, i.e., no Lagrangian multipliers updating in Eq. (121).

$$\lambda_{i,j}^{k+1} = \lambda_{i,j}^k + \alpha^k \left(\sum_{f \in F} \sum_{(i,j,t,t') \in CA_{i,j}} x_{i,j,t,t'}^f - m_{i,j} \right) \quad (121)$$

$$\alpha^k = 1/(k + 1) \quad (122)$$

5.5.2 Branch and Price Algorithm

For model M3-CTR, the number of feasible routes in set Ω can be extremely large in large-scale problems, making it nearly impossible to enumerate all routes and solve the resulting model using existing solvers. This section further proposes a Branch and Price (BnP)-based exact solution approach to solve model M3-CTR. The proposed BnP algorithm consists of two modules, i.e., column generation (CG) and bound-and-bound (BnB). Model M3-CTR is denoted as the master problem (MP), then CG module is used to solve the linear master problem (LMP), while the BnB module is needed to obtain integer solutions based on the results from the CG module. The LMP is the same as MP except the domain of decision variables. In LMP, the integer requirement of decision variable θ_r , i.e., $0 \leq \theta_r \leq 1$, is relaxed. Due to the existence of constraint (117), the domain constraint $0 \leq \theta_r \leq 1$ can be further simplified to $\theta_r \geq 0$.

Column generation

The LMP is still hard to solve due to the large size of route set Ω . Therefore, instead of directly solving the LMP, its corresponding restricted linear master problem (RLMP) is iteratively solved, whose route set Ω' only contains part of feasible routes and is extended with new routes as needed. The RLMP is presented as follows:

Objective function

$$\min Z_5 = \sum_{r \in \Omega'} c_r \theta_r \quad (123)$$

Subject to:

Cleaning request satisfaction constraint:

$$\sum_{r \in \Omega'} \alpha_{i,r} \theta_r = 1, \quad \forall i \in N_N^S \quad (124)$$

Decision variables:

$$\theta_r \geq 0, \quad \forall r \in \Omega' \quad (125)$$

In this research, the initial routes in set Ω' are generated by assigning one sprinkler for each service node. Note that, when generating initial routes, if a node cannot be serviced due to the violation of time window constraint or water consumption constraint, the original problem is then infeasible. The route set Ω' in RLMP is extended by solving the so-called pricing problem, during which routes with negative reduced cost are added into set Ω' .

The pricing problem is formulated as follows:

Objective function

$$\begin{aligned} \min Z_5 = & h + t_{d_N}^d + \omega \left(\sum_{(i,j,\rho) \in E_N} c g_{i,j,\rho} (t_i^d) x_{i,j,\rho} + \sum_{(i,j,\rho) \in E_N} c g_j (t_j^s) x_{i,j,\rho} \right) \\ & - \sum_{(i,j,\rho) \in E_N: j \in N_N^S} x_{i,j,\rho} \pi_j \end{aligned} \quad (126)$$

Subject to:

Feasible route constraints.

Objective function (126) minimizes the reduced cost of a feasible route, where π_j is the reduced cost associated with node j . Essentially, the optimization problem presented above is an elementary shortest path problem with resource constraint (ESPPRC).

Finding an elementary shortest path with resource constraint in a network is an NP-hard problem. Therefore, finding the exact solution for the aforementioned pricing problem may be extremely time-consuming in large-scale instances. In the literature, to reduce the computational complexity of the pricing problem, following the pioneering work by Christofides et al. (1981), researchers started to develop efficient algorithms for finding non-elementary shortest paths with resource constraint through relaxing the elementary requirement, i.e., allowing visiting a node multiple times (Irnich and Villeneuve, 2006; Baldacci et al., 2011; Martinelli et al., 2014). The use of non-elementary shortest paths will not affect finding the optimal solution of the original problem but will weaken its linear relaxations. This research adopts the dynamic programming ng-route algorithm proposed by Martinelli et al. (2014) with some modifications. Specifically, in Martinelli et al. (2014), a vehicle only needs to collect goods during its trip, therefore the load of a vehicle keeps increasing along with its trip. As a result, a two-dimension matrix with vehicle load as one axis can be created, and the dynamic programming process can be performed in the order of vehicle load increasing. However, in the SRP considered in this work, sprinkler trucks are allowed to refill water during their trips, hence the water level changes on sprinkler trucks are not monotonous anymore. Another challenge is that demands are assumed to be discrete in Martinelli et al. (2014), while it is continuous in the proposed SRP, thus creating a matrix with sprinkler water level as an axis is impossible.

The three key components of a dynamic programming algorithm, including state (label) definition, extension rule, and dominance rule, are introduced below, followed by the modified ng-route algorithm developed in this research.

Label definition. Labels used in the pricing problem have the following form: $l = \{node, R_c, R_t, R_w, prev, \Phi\}$, where *node* means current node label l is on; R_c , R_t , R_w denotes the reduced cost, the actual service starting time, and the water level of the sprinkler at current node, respectively; *prev* stands for the previous label of label l ; Φ represents the set of forbidden nodes that are not allowed to extend from label l , which will be introduced in the extension rule below.

Extension rule. The process of extending current label l to node j via edge ρ is presented in Algorithm 5. In Algorithm 5, line 2-4 check if node j is reachable from current node i ; line 7 checks if the destination depot is reachable after visiting node j . Line 12 creates a new label l' on node j from label l via edge ρ . Specifically, the ng-set \mathcal{V}_j of node j is used when updating forbidden set Φ' . For each node i , a ng-set \mathcal{V}_i is predefined which includes a certain number of nearest nodes of i (including itself). A label on node i only includes the intersection of visited nodes and \mathcal{V}_i into its forbidden set Φ , resulting in the possibility of forming cycles. Generally, the larger the ng-sets are, the less likely to contain cycles in routes, and also closer to the original ESPPRC (harder to solve). Water refilling stations will also not be put into forbidden sets as they are always allowed to revisit in the SRP settings.

Algorithm 5 Label Extension Procedure.

Input: label $l = \{node, R_c, R_t, R_w, prev, \Phi\}$, node service time function $\tau_i(t)$, edge travel time function $\tau_{i,j,\rho}(t)$, system impact functions $cg_i(t)$ and $cg_{i,j,\rho}(t)$

Output: new label l' on node j

```

1:  label  $l' = null$ 
2:  if  $j \notin \Phi$  and  $R_w - w_j \geq 0$  then
3:      arrival time at node  $j$ :  $at_j = R_t + \tau_i(R_t) + \tau_{i,j,\rho}(R_t + \tau_i(R_t))$ 
4:      if  $at_j \leq e_j$  then
5:          the earliest service starting time at node  $j$ :  $R'_t = \max(at_j, s_j)$ 
6:          the earliest time to return back to the depot after visiting node  $j$ :
           $at_{d_N} = R'_t + \tau_j(R'_t) + \min_{\rho'} \{ \tau_{j,d_N,\rho'}(R'_t + \tau_j(R'_t)) \}$ 
7:          if  $at_{d_N} \leq e_{d_N}$  then
8:               $R'_c = \begin{cases} R_c + R'_t - R_t + cg_{i,j,\rho}(R_t + \tau_i(R_t)) + cg_i(R_t) - \pi_j & \text{if } j \in N_N^s \\ R_c + R'_t - R_t + cg_{i,j,\rho}(R_t + \tau_i(R_t)) + cg_i(R_t) & \text{if } j \notin N_N^s \end{cases}$ 
9:               $R'_w = \begin{cases} C & \text{if } j \in N_N^w \\ R_w - w_j & \text{if } j \notin N_N^w \end{cases}$ 
10:              $prev' = l$ 
11:              $\Phi' = \begin{cases} \Phi \cap \mathcal{V}_j & \text{if } j \in N_N^w \\ \Phi \cap \mathcal{V}_j \cup \{j\} & \text{if } j \notin N_N^w \end{cases}$ 
12:             label  $l' = \{j, R'_c, R'_t, R'_w, prev', \Phi'\}$ 
13:  return label  $l'$ 

```

Dominance rule. For two labels l and l' on the same node, label l' is dominated by

label l if the following conditions are satisfied and at least one of them is not equal:

(i) $l.R_c \leq l'.R_c - (l'.R_t - l.R_t)$,

(ii) $l.R_t \leq l'.R_t$,

(iii) $l.R_w \leq l'.R_w$,

(iv) $l.\Phi \subseteq l'.\Phi$.

It should be noted that condition (i) is much tighter than the condition that is commonly used in VRPs with the objective of minimizing total travel distance, i.e., $l.R_c \leq$

$l'.R_c$. A stricter dominance rule means less labels can be dominated, thus there are more labels to be processed (the pricing problem is harder to solve). If label l' is dominated by other labels, label l' can then be safely discarded without affecting the final optimal results.

The dynamic programming ng-route algorithm is presented in Algorithm 6. Algorithm 6 finds the best route with given graph G_N and ng-sets \mathcal{V}_i , which may be very time-consuming in large instances. When used under CG, to save computing time, Algorithm 6 can be terminated as soon as a certain number of paths with negative reduced cost have been found.

Algorithm 6 Dynamic Programming ng-route Algorithm.

Input: node-based graph G_N , ng-set \mathcal{V}_i for each node $i \in N_N$

Output: the best ng-route

```

1: create a root label  $l_0 = \{o_N, h, 0, C, null, \{o_N\}\}$ 
2:  $\mathcal{U} \leftarrow \{l_0\}$ 
3: while  $\mathcal{U} \neq \emptyset$  do
4:     choose a label  $l$  with the minimum service starting time from  $\mathcal{U}$ , and delete  $l$ 
       from  $\mathcal{U}$ 
5:     for  $j \in N_N$  do
6:         for  $\rho := 1, 2, \dots, \varphi$  do
7:             if label  $l$  can be extended to node  $j$  via edge  $\rho$  (feasibility checking
               in Algorithm 5) then
8:                 create a new label  $l'$  on node  $j$  from label  $l$ 
9:                  $insertLabel \leftarrow \mathbf{true}$ 
10:                for label  $l'' \in \mathcal{U}$  on node  $j$  do
11:                    if label  $l''$  dominates  $l'$  then
12:                         $insertLabel \leftarrow \mathbf{false}$ 
13:                        break
14:                    else if label  $l'$  dominates  $l''$  then
15:                        delete  $l''$ 
16:                if  $insertLabel$  then
17:                     $\mathcal{U} \leftarrow \mathcal{U} \cup \{l'\}$ 
18: best label  $l^* \leftarrow$  the label with the least reduced cost on the destination depot node
19: retrieving the best route from label  $l^*$ 
20: return the best route

```

Although finding non-elementary shortest paths using Algorithm 6 is faster than finding the exact solution for elementary shortest path problem, the pricing problem is still hard to solve, especially in large-scale instances. This research further utilizes the following three techniques to speed up Algorithm 6.

Decremental state-space relaxation. It is observed that the larger ng-set of nodes are, the harder it is to solve the pricing problem. Therefore, instead of directly using the complete ng-set \mathcal{V}_i , Algorithm 6 can start with a subset of ng-set \mathcal{V}_i , i.e., \mathcal{V}'_i , for each node i and iteratively add nodes from \mathcal{V}_i to \mathcal{V}'_i until a valid ng-route is found. A node to be added to \mathcal{V}'_i should satisfy the following requirement: the absence of that node in \mathcal{V}'_i leads to a cycle in the final route that prevents the route from being an ng-route.

Heuristic domination. The first condition in the dominance rule presented above is by far stricter than the condition used in VRPs with the objective of minimizing total travel distance, which is $l.R_c \leq l'.R_c$. Using the condition $l.R_c \leq l'.R_c$ is of course helpful to dominate more labels, then accelerates the pricing process. However, some valid labels will also be discarded, which may affect the final solution. In preliminary experiments, it is found that only a small portion of valid labels are discarded due to the use of the loose domination condition. In other words, a relatively good solution can still be obtained with condition $l.R_c \leq l'.R_c$ in a much shorter time. Therefore, before using the exact dominance condition, i.e., $l.R_c \leq l'.R_c - (l'.R_t - l.R_t)$, the loose condition will be used until no valid route can be found.

Truncated labels. As in Yao et al. (2021), a truncated version of dynamic programming is used before calling the exact dynamic programming presented in Algorithm 6. In the truncated version, each node only maintains a limited number of

promising labels, e.g., 100 labels with the least reduced cost. Note that, as the number of labels in the truncated version is limited, running one iteration of the truncated version is very fast, therefore node revisiting is not allowed and heuristic domination is not used so that routes with higher quality can be obtained.

Branch and bound

An optimal solution obtained from the CG module may contain fractional path usages, making it infeasible for the MP (model M3-CTR). Therefore, BnB module is further utilized to produce an integer optimal solution based on results from the CG module.

If an optimal solution obtained from the CG module is fractional, the BnB module will perform branching on an edge with a fractional usage. The branching process is presented in Algorithm 7.

Algorithm 7 Edge-based Branching Procedure.

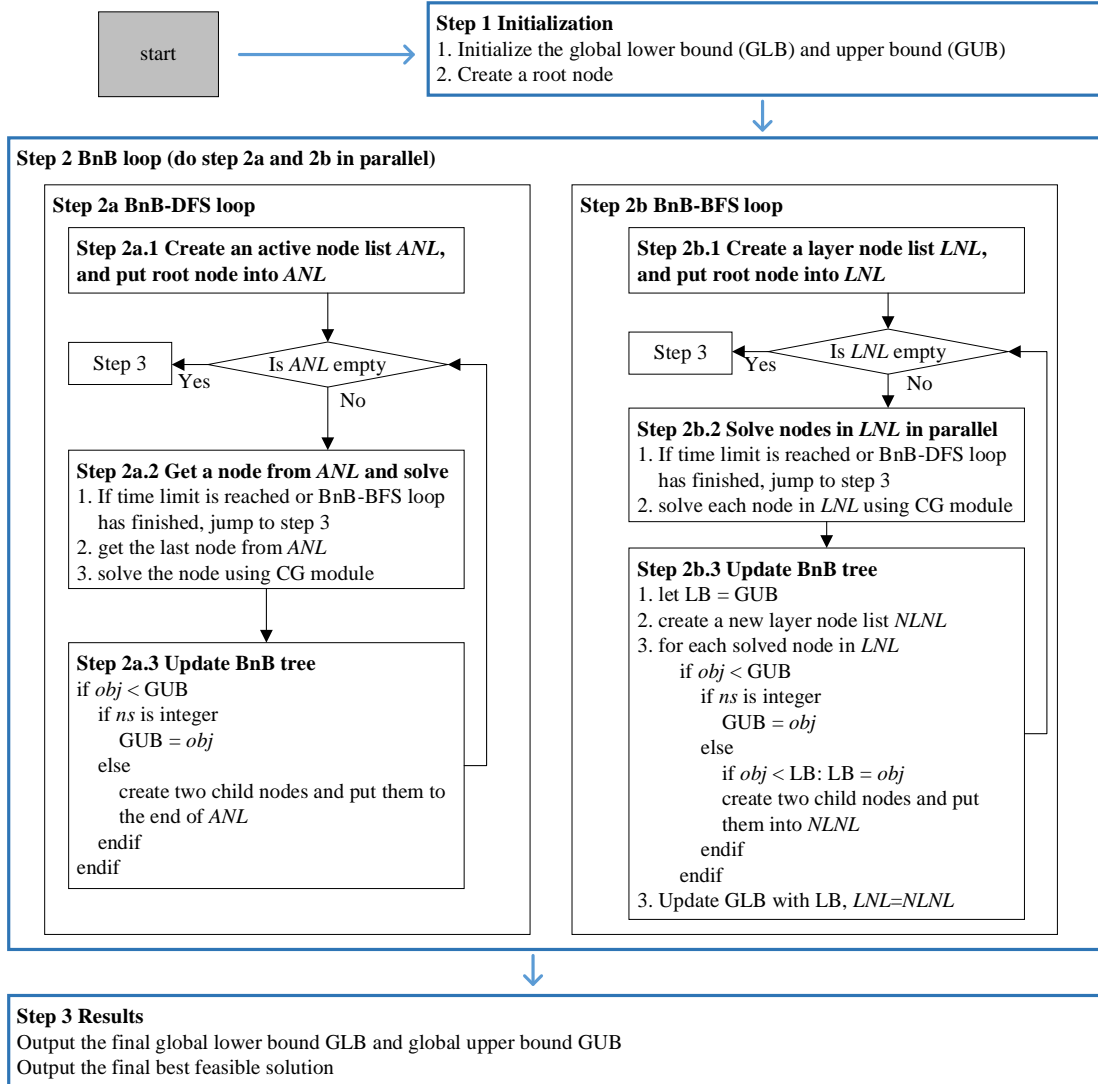
Input: path usages from a CG solution

Output: two child nodes of the current node

- 1: Calculate edge usages based on path usages from a CG solution.
- 2: Select multiple candidate edges (10 edges in this study) with their usages closed to 0.5. Evaluate the impact of performing branching on these candidate edges as in Dabia et al. (2013). The quick pricing heuristic used in this study is the dynamic programming algorithm using heuristic domination and truncated labels introduced in the aforementioned acceleration techniques. Select an edge, say (i, j, ρ) , from all candidate edges such that branching on it results in the tightest estimated lower bound.
- 3: (Child node one) remove edge (i, j, ρ) from the graph used in the pricing problem.
- 4: (Child node two) remove edge set $\{(i', j', \rho') | (i', j', \rho') \in E_N, i' = i\}$, edge set $\{(i', j', \rho') | (i', j', \rho') \in E_N, j' = j\}$, and edge set $\{(i', j', \rho') | (i', j', \rho') \in E_N, i' = j, j' = i\}$ from the graph used in the pricing problem, but keep edge (i, j, ρ) .
- 5: **return** child node one, child node two

Child node one forces sprinkler trucks not to use edge (i, j, ρ) , while child node two forces sprinkler trucks to use edge (i, j, ρ) if service node i is in their routes.

A searching strategy determines the sequence of solving active BnB nodes, and different strategies may affect the performance of the BnB module for a specific problem. Breadth-first search (BFS) and depth-first search (DFS) are the two widely used searching strategies. Generally, BFS is helpful to improve the global lower bound (GLB) of a problem, while DFS is useful to quickly find a feasible solution thus contributes to improving the global upper bound (GUB) of the problem. Benefitting from mature parallel computing technologies, both BFS and DFS are implemented and run in parallel in this work. Moreover, as nodes in the same layer in BnB-BFS are independent from each other, solving nodes in Step 2b.1 is also conducted in a parallel manner. The overall framework of the BnB module designed in this research is presented in Fig. 48.



Note: *obj* means the objective value of a BnB node; *ns* represents the solution state of a BnB node, it can be integer or fractional

Fig. 48. The Overall Framework of the BnB Module.

5.6 Computational Experiments

In this section, extensive numerical experiments are conducted to evaluate the methods developed in this research. Specifically, 12 corridors in the Washington DC metropolitan area are first selected to demonstrate the suitability of the proposed time-dependent travel time modeling method. Based on calibrated travel time functions, three sets of SRP instances with different sizes are designed to examine the performance of the

three models, followed by an analysis on the system impact of vehicle routings. Finally, sensitivity analysis of solution methods is performed.

5.6.1 Modeling of Time-dependent Travel Time

Fig. 49 presents the 12 corridors selected for evaluating the proposed time-dependent travel time modeling method, including 1 expressway corridor, 3 freeway corridors, and 8 arterial corridors that experience different levels of traffic congestion. Table 19 summarizes the characteristics of the selected corridors. The time horizon of interest is set as 6:00 – 20:00. Note that transportation networks typically experience two or three congestion periods across a day (morning peak hours, afternoon peak hours, and possibly midday hours), while the method proposed in Section 5.2 is suitable for a single congestion period analysis. Hence, the analysis time horizon is split into three periods, including am (6:00-10:00), md (10:00-14:00), and pm (14:00-20:00), then sequentially apply the method for each period. The dataset used for method evaluation was collected from the Regional Integrated Transportation Information System (RITIS). The raw dataset contains 5-minute space-mean traffic speed data from loop detectors and probe vehicles for a majority of links along each corridor. Congestion duration P , average discharge rate μ , and inflow demand curvature parameter γ are first calibrated using the four-step method proposed in Section 5.2. Then smoothed time-dependent travel times are estimated based on Eq. (84) for each link along corridors.

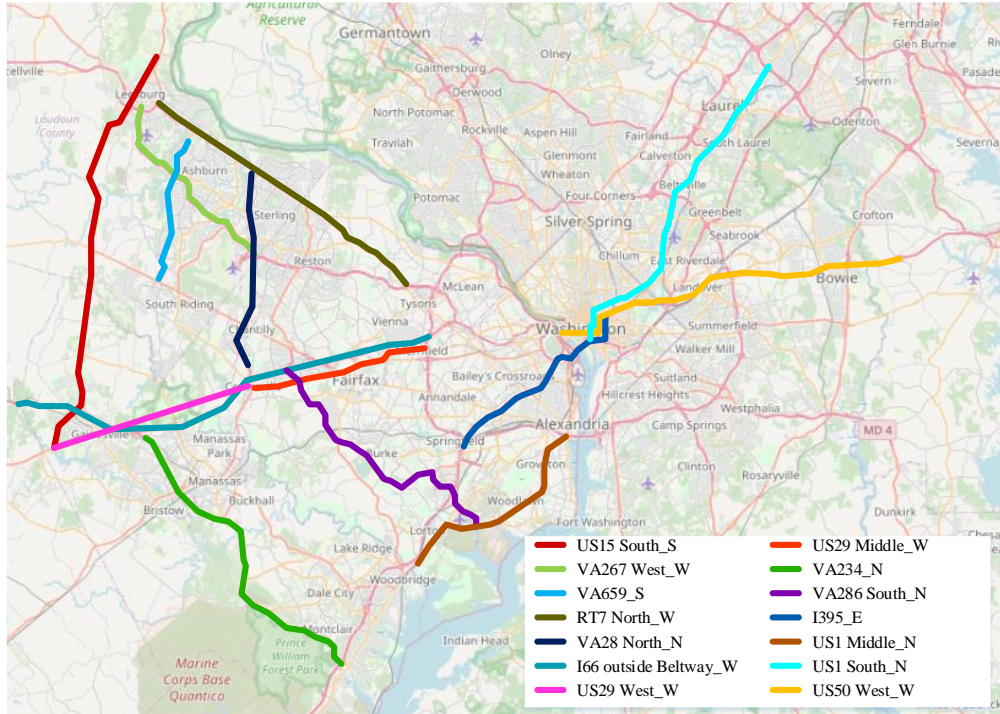


Fig. 49 Corridors Used for Time-dependent Travel Time Modeling Evaluation.

Table 19 Characteristics of the 12 Corridors.

Corridor name	Type	# of links	Corridor length (mile)	LLS (mph)	HLS (mph)	ALS (mph)	ALCD (hour)
US15 South_S	Arterial	39	28.6	8.5	60.8	39.41	2.73
VA267 West_W	Freeway	20	13.0	16.6	74.7	63.80	0.49
VA659_S	Arterial	17	10.3	8.9	45.1	31.18	2.39
RT7 North_W	Arterial	36	19.7	14.0	62.1	44.76	3.68
VA28 North_N	Expressway	28	13.8	19.2	68.1	60.59	1.10
I66 outside Beltway_W	Freeway	50	27.9	15.0	74.7	58.47	2.93
US29 West_W	Arterial	27	13.1	14.0	55.9	42.19	2.82
US29 Middle_W	Arterial	17	11.8	11.5	51.2	26.45	6.76
VA234_N	Arterial	25	21.8	20.1	57.2	43.40	2.60
VA286 South_N	Arterial	27	19.1	14.4	58.1	47.15	1.40
I395_E	Freeway	28	9.7	6.1	65.8	43.26	5.63
US1 Middle_N	Arterial	14	13.7	8.8	52.7	36.88	4.74

Notes: HLS – highest link speed, LLS – lowest link speed, ALS – average link speed, ALCD – average link congestion duration for an entire day

Table 20 compares the observed speed and modelled speed calibrated from the fluid-queue model with polynomial arrival rates. The measure in terms of travel speed is selected as it is more intuitive than link travel time depending on a specific link length. First, reasonable modeling accuracy was achieved on all corridors with varied degrees of

congestion. The heavily congested corridor I395_E has the highest mean link MAE (7.55 mph) and mean link MAPE (27.27%), as it has a relatively long congestion duration (5.63 hours for all three congestion periods during the entire day in Table 19). It should be remarked that, under a low average traffic speed associated with a heavy congestion, a typical deviation in absolute speed values could still result in a relatively large percentage difference, due to the smaller value in the denominator. In this regard, the mean link-based MAPE of corridor I395_E is still explainable. Second, for each corridor, max link MAE are not significantly worse than corresponding mean link MAE, indicating that the approximation error is reasonably bounded along each corridor. In short, better calibration results are observed on the corridors with a shorter congestion duration, whereas the corridors with severer congestion have larger modeling errors (e.g., corridor US29 Middle_W and I395_E).

Table 20 Comparison Between Observed and Modeled Speed Across All Links and 15-min Interval Resolution on 12 Selected Corridors.

Corridor name	Mean link MAE (mph)	Mean link MAPE (%)	Max link MAE (mph)
US15 South_S	2.77	8.83	4.77
VA267 West_W	2.86	4.52	3.22
VA659_S	2.38	8.48	3.62
RT7 North_W	4.09	11.04	5.91
VA28 North_N	2.47	4.55	5.03
I66 outside Beltway_W	4.35	9.10	7.52
US29 West_W	2.91	7.75	6.34
US29 Middle_W	4.02	16.92	7.64
VA234_N	3.20	8.14	7.53
VA286 South_N	3.05	7.47	5.94
I395_E	7.55	27.27	12.50
US1 Middle_N	3.41	11.08	6.99

Notes: MAE – mean absolute error, MAPE – mean absolute percentage error

To further examine the travel time modeling accuracy, Fig. 50 and Fig. 51 present the heatmap comparison between the observed and modeled travel speed on two corridors with a relatively poor modeling accuracy in Table 20, i.e., corridor US29 Middle_W and I395_E. On both corridors, the overall modeled speed pattern is closed to the corresponding observed one. Specifically, areas affected by congestion propagation (grey dash rectangles) have been realistically captured.

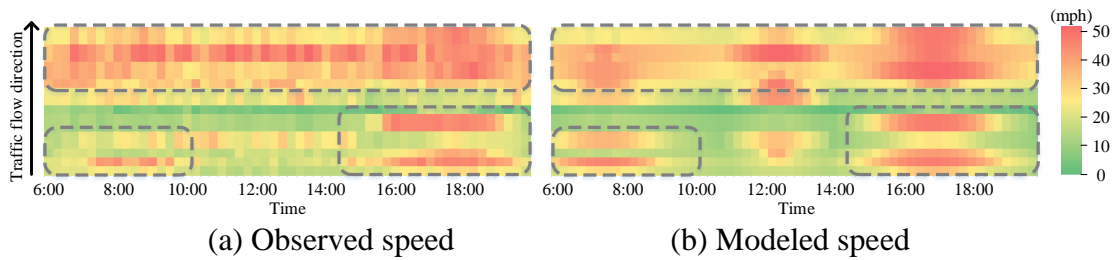


Fig. 50. Heatmap Comparison Between Observed and Modeled Speed on Corridor US29 Middle_W.

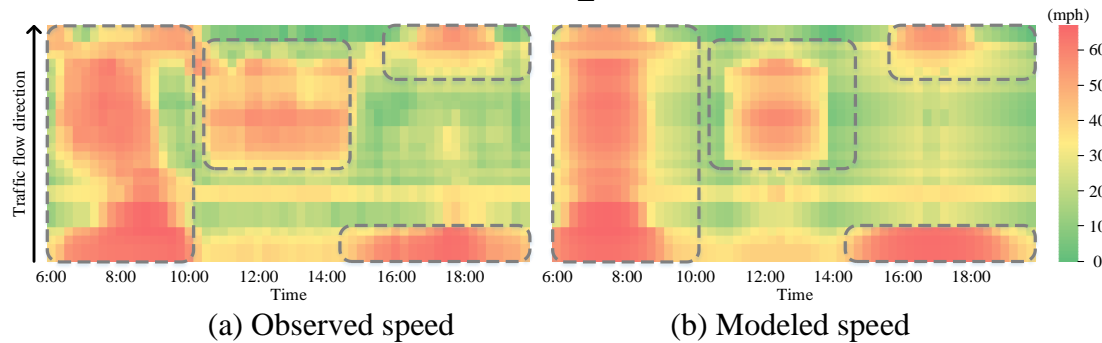
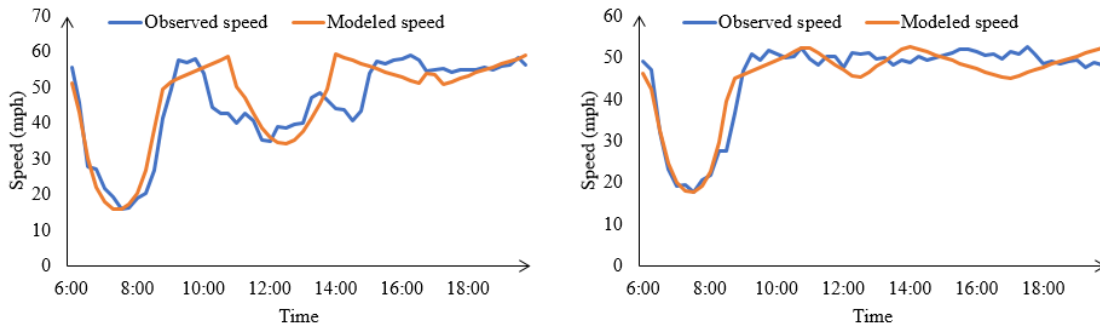


Fig. 51. Heatmap Comparison Between Observed and Modeled Speed on Corridor I395_E.

Focusing on two links with different traffic congestion patterns from corridor I395_E and corridor US1 Middle_N, Fig. 52 provides the time-dependent observed and modeled speed across the analysis time horizon. The congestion occurs in the periods of am and md on link 32609, whereas only am experiences significant delay on link 31948. One can observe that, for the link with a single congestion period (link 31948), the modeled speed closely matches the observed speed. However, for the link with multiple congestion

periods (link 32609), there are still obvious deviations between two lines, which highlights the challenges in modeling complex and heavy traffic congestion.



(a) Link 32609 on corridor I395_E (b) Link 31948 on corridor US1 Middle_N
 Fig. 52. Modeled Speed and Observed Speed Comparison on Two Links.

5.6.2 Performance Evaluation of the Solution Methods

As summarized in Table 21, three sets of instances with different sizes are designed to evaluate the three optimization models developed in this research. Instance networks consist of expressway, freeway and arterial link. Travel time functions calibrated on the 12 representative corridors are applied to links in benchmark instances according to their road types.

The first instance set includes 15 small-size instances. The average number of road links to be cleaned is 16. The length of links in the first set ranges from 0.07 to 0.74 mile. The second set contains 15 medium-size instances, where the number of road links ranges from 28 to 38. The last set includes 13 large-size instances, with 62 links to be cleaned on average for each instance network. For all of the instances, the planning horizon is 3 hours. The vehicle capacity is 1200 gallons of water, and it is assumed that cleaning 1 mile of road consumes 400 gallons of water. The speed limit of sprinkler trucks in the cleaning mode and deadheading mode is set as 5 mph and 14 mph, respectively. This research assumes that there is no limit on sprinkler fleet size, but each sprinkler has an acquisition

cost of equivalent 10 units of travel time. Note that, as this subsection mainly focuses on efficiency evaluation on the three proposed models, the weight of system impact cost ω is set as 0. The tradeoff between societal impact and private operating cost will be performed in Section 5.6.3.

Table 21 Characteristics of Three Benchmark Instance Sets.

Small-size benchmark instance set				Medium-size benchmark instance set				Large-size benchmark instance set			
Instance	Links	MinLL	MaxLL	Instance	Links	MinLL	MaxLL	Instance	Links	MinLL	MaxLL
S1	16	0.19	0.53	M1	28	0.14	0.68	L1	72	0.04	0.68
S2	14	0.22	0.28	M2	36	0.11	0.65	L2	66	0.11	0.65
S3	16	0.09	0.29	M3	28	0.09	0.31	L3	60	0.09	0.54
S4	16	0.07	0.24	M4	30	0.14	0.42	L4	54	0.08	0.42
S5	16	0.13	0.50	M5	30	0.15	0.5	L5	60	0.13	0.5
S6	14	0.10	0.46	M6	38	0.13	0.44	L6	74	0.13	0.54
S7	18	0.18	0.50	M7	32	0.18	0.5	L7	62	0.18	0.63
S8	14	0.07	0.18	M8	34	0.07	0.49	L8	62	0.07	0.49
S9	16	0.13	0.37	M9	30	0.13	0.56	L9	50	0.13	0.56
S10	16	0.08	0.31	M10	28	0.15	0.33	L10	62	0.09	0.62
S11	15	0.09	0.68	M11	32	0.07	0.23	L11	60	0.07	0.7
S12	14	0.10	0.44	M12	30	0.09	0.41	L12	60	0.09	0.38
S13	18	0.09	0.23	M13	28	0.07	0.7	L13	58	0.07	1.43
S14	18	0.09	0.27	M14	32	0.09	0.37				
S15	16	0.15	0.74	M15	28	0.1	0.74				
Average	16	0.12	0.40	Average	31	0.11	0.49	Average	62	0.10	0.63

Notes: MinLL and MaxLL represent minimum link length and maximum link length respectively, with the unit of mile

In the results presented below, model M1-TEN is solved by the LR method presented in Section 5.5.1, M2-STD is directly solved by a commercial MILP solver (CPLEX), M3-CTR is solved by the BnP algorithm presented in Section 5.5.2. Both LR method and BnP algorithm proposed in this research are coded in C++. The RMLP in BnP is solved with CPLEX by calling its built-in C++ API. CPLEX solver version 12.10 is used throughout the experiments. All numerical experiments conducted in this study are evaluated on a 64-bit Linux server with Intel Xeon Gold 6230R processor @ 2.10 GHz and 180 GB RAM.

The results of the proposed models on the three sets of real-life instances (small-size, medium-size, large-size) are shown in Table 22, Table 23, and Table 24, respectively. For each instance, Table 22 - Table 24 provide lower bound (LB), upper bound (UB), Gap, solution time (ST), and memory usage (MU) of results obtained from the proposed models on the three sets of instances, respectively. The Gaps in tables are relative gaps, which are calculated by $\text{Gap} = (\text{UB} - \text{LB})/\text{UB} \times 100$. The solution time limit is set as 15 minutes for small-size instances, 30 minutes for medium-size instances, 60 minutes for large-size instances. The time discretization resolution of model M1-TEN is set as 0.2 min. For model M3-CTR, the size of ng-set is set as 2.

Table 22 Performance Comparisons of Three Proposed Models on Small-size Instances.

Instance	M1-TEN (LR)					M2-STD (MILP Solver)					M3-CTR (BnP)				
	LB	UB	Gap (%)	ST (s)	MU (GB)	LB	UB	Gap (%)	ST (s)	MU (GB)	LB	UB	Gap (%)	ST (s)	MU (GB)
S1	108.01	112.6	4.08	136	0.14	23.34	112.38	79.23	900	22.33	112.38	112.38	0.0	1	0.0
S2	70.73	76.0	6.94	135	0.17	15.15	78.07	80.59	900	21.74	75.65	75.65	0.0	3	0.0
S3	65.96	73.4	10.13	540	0.54	10.0	73.03	86.31	900	22.46	73.03	73.03	0.0	20	0.0
S4	48.16	51.6	6.66	591	0.55	11.28	51.87	78.25	900	26.97	51.85	51.85	0.0	212	1.15
S5	77.64	82.6	6.01	315	0.28	12.59	83.24	84.87	900	22.68	82.36	82.36	0.0	5	0.0
S6	71.91	77.0	6.61	274	0.37	21.09	77.05	72.63	900	20.71	77.05	77.05	0.0	4	0.0
S7	120.59	132.0	8.64	284	0.29	15.32	131.87	88.38	900	21.72	131.87	131.87	0.0	6	0.0
S8	39.31	45.6	13.8	899	1.18	11.76	45.84	74.35	900	17.79	45.84	45.84	0.0	264	3.65
S9	74.91	82.6	9.31	293	0.24	13.24	80.28	83.5	900	13.67	80.02	80.02	0.0	4	0.0
S10	63.25	71.4	11.42	333	0.33	12.44	71.12	82.5	900	20.15	71.12	71.12	0.0	16	0.0
S11	121.22	130.8	7.32	900	0.94	62.99	130.69	51.8	900	19.88	130.69	130.69	0.0	1	0.0
S12	59.30	59.4	0.16	143	0.17	15.4	72.27	78.69	900	19.02	59.48	59.48	0.0	1	0.0
S13	68.94	78.0	11.62	227	0.26	11.32	78.9	85.65	900	27.19	78.53	78.53	0.0	62	6.83
S14	53.81	56.0	3.91	434	0.64	10.0	56.35	82.25	900	4.1	54.49	56.29	3.2	900	0.47
S15	109.57	111.4	1.65	273	0.37	26.66	113.35	76.48	900	26.75	111.4	111.4	0.0	1	0.0
Average			7.22	385	0.43			79.03	900	20.48			0.21	100	0.81

Table 23 Performance Comparisons of Three Proposed Models on Medium-size Instances.

Instance	M1-TEN (LR)					M2-STD (MILP Solver)					M3-CTR (BnP)				
	LB	UB	Gap (%)	ST (s)	MU (GB)	LB	UB	Gap (%)	ST (s)	MU (GB)	LB	UB	Gap (%)	ST (s)	MU (GB)
M1	199.29	226.2	11.9	1265	0.98	151.12	212.76	28.97	1800	77.36	211.4	211.4	0.0	758	106.24
M2	215.73	245.8	12.23	1800	1.58	38.14	-	-	1800	17.23	221.28	221.28	0.0	27	0.0
M3	125.77	147.4	14.68	1182	1.08	10.0	146.64	93.18	1800	62.52	134.04	146.64	8.59	1800	156.71
M4	185.04	193.2	4.22	1800	1.11	53.32	193.07	72.39	1800	24.98	190.07	190.07	0.0	14	0.0
M5	217.37	232.4	6.47	1800	1.25	232.48	232.48	0.0	1186	9.19	232.48	232.48	0.0	4	0.0
M6	198.56	227.2	12.61	1800	1.35	10.0	253.17	96.05	1800	11.4	202.72	215.2	5.8	1800	169.99
M7	215.99	237.6	9.1	1800	1.09	15.2	-	-	1800	14.71	225.52	225.52	0.0	20	0.0
M8	132.51	160.4	17.39	1800	1.6	10.03	153.4	93.46	1800	28.81	136.06	150.12	9.37	1800	1.45
M9	185.2	201.2	7.95	1800	1.11	22.0	194.04	88.66	1800	35.02	189.89	192.76	1.49	1800	179.87
M10	148.68	156.0	4.69	1223	0.79	25.78	155.64	83.44	1800	24.82	153.03	153.03	0.0	22	0.0
M11	98.67	116.4	15.23	1800	1.7	10.0	112.2	91.09	1800	93.41	106.22	112.92	5.93	1800	4.62
M12	199.69	218.6	8.65	1245	0.92	13.29	222.67	94.03	1800	23.09	205.63	215.16	4.43	1800	132.7
M13	167.8	189.0	11.22	1800	1.27	10.02	186.37	94.62	1800	118.91	173.61	184.68	5.99	1800	175.27
M14	192.94	225.6	14.48	1800	1.1	10.0	197.49	94.94	1800	74.56	197.49	197.49	0.0	14	0.0
M15	198.52	225.0	11.77	1800	0.8	31.31	222.55	85.93	1800	30.59	204.3	222.55	8.2	1800	132.2
Average			10.84	1648	1.18			81.12	1759	43.11			3.32	1017	70.6

Table 24 Performance Comparisons of Three Proposed Models on Large-size Instances.

Instance	M1-TEN (LR)					M2-STD (MILP Solver)					M3-CTR (BnP)				
	LB	UB	Gap (%)	ST (s)	MU (GB)	LB	UB	Gap (%)	ST (s)	MU (GB)	LB	UB	Gap (%)	ST (s)	MU (GB)
L1	203.71	632.0	67.77	3600	9.61	10.0	-	-	3600	25.26	531.0	543.87	2.37	3600	8.39
L2	159.6	540.4	70.47	3600	8.59	10.0	-	-	3600	21.36	450.57	467.53	3.63	3600	22.02
L3	112.4	629.6	82.15	3600	6.15	10.0	-	-	3600	13.5	553.44	558.19	0.85	3600	3.39
L4	10.0	346.8	97.12	3600	9.27	10.0	431.64	97.68	3600	19.71	330.26	341.46	3.28	3600	12.14
L5	10.0	789.8	98.73	3600	10.08	10.0	-	-	3600	15.8	371.99	403.93	7.91	3600	4.19
L6	10.0	647.0	98.45	3600	9.86	10.0	-	-	3600	17.99	465.71	504.8	7.74	3600	10.46
L7	69.64	1111.2	93.73	3600	8.29	186.99	-	-	3600	14.08	503.75	519.44	3.02	3600	19.36
L8	10.0	625.8	98.40	3600	9.50	10.0	-	-	3600	22.12	303.62	320.1	5.15	3600	7.55
L9	396.03	772.4	48.73	3600	5.32	105.18	-	-	3600	13.59	602.83	602.83	0.0	12	0.71
L10	27.06	637.4	95.76	3600	8.36	10.0	-	-	3600	16.8	382.14	399.41	4.32	3600	4.24
L11	10.0	694.0	98.56	3600	8.45	10.0	-	-	3600	17.55	320.05	338.0	5.31	3600	4.57
L12	162.28	1022.2	84.12	3600	7.55	10.0	-	-	3600	15.8	514.32	545.44	5.7	3600	23.17
L13	313.87	935.8	66.46	3600	5.32	10.0	-	-	3600	15.32	639.36	678.18	5.72	3600	15.72
Average			84.65	3600	8.18			99.82	3600	-			4.23	3324	10.45

Overall, the two models (M1-TEN and M3-CTR) solved by customized solution algorithms outperform the model (M2-STD) solved by a MILP solver in the three sets of

instances; model M3-CTR has the best performance in terms of solution quality and solution time among the three proposed models.

For small-size instances, the average relative gaps of models M1-TEN, M2-STD, and M3-CTR are 7.22%, 79.03%, and 0.21%, respectively. Specifically, among the 15 small-size instances, model M3-CTR is able to produce optimal solutions on 14 instances within the time limit, while solution optimality cannot be proven on all 15 instances for models M1-TEN and M2-STD. On the other hand, although the relative gaps of model M1-TEN and M2-STD are higher than that of model M3-CTR, the upper bounds provided by the first two models are always closed to optimal solutions, indicating that these two models are still able to produce acceptable routing solutions for practical use in small-size instances. In terms of the computational speed, the average solution times of the three models on small-size instances are 385s, 900s, and 100s, respectively.

For medium-size instances, due to the increase of instance complexity, larger relative gaps and longer solution times are observed on three models. In terms of solution quality, similar to the results on small-size instances, model M3-CTR still has the best performance, with an average relative gap of 3.32%. For model M1-TEN, the average relative gap is 10.84%, while it is over 80% for model M2-STD. The large gap of model M2-STD is mainly due to its weak lower bounds. The model M2-STD is directly solved by CPLEX with built-in linear relaxation and branch-and-bound capability. Compared to model M3-CTR, it can be found that linearly relaxing a path-based model (M3-CTR) provides tighter lower bounds than simply relaxing an arc-based model (M2-STD).

For large-size instances, model M2-STD is only able to produce feasible solutions on one instance (L4), and weak lower bounds are provided on all instances, resulting in an

average relative gap of 99.82%. This highlights the need of designing customized solution algorithms for solving large-scale RAPPs in real life. Compared with model M3-CTR with an average gap of 4.23%, model M1-TEN provides relatively poor lower and upper bounds, resulting in an average gap of 84.65%. Its weak bounds are mainly due to the symmetry issue, as discussed by Niu et al., (2018) and Yao et al., (2019). As all sprinkler trucks are assumed to be identical, trucks tend to select the same least-cost path when serving customers, resulting in weaker lower bounds and additional efforts for generating feasible solutions for upper bounds.

In Table 25, both Gap and BestGap are further provided to comprehensively measure the quality of solutions obtained from model M1-TEN and M3-CTR. BestGap represents the relative difference between upper bound and the best lower bound of the two models. The Gaps and BestGaps of model M3-CTR are exactly the same on all instances, while, for model M1-TEN, BestGaps are significantly smaller than the corresponding Gaps. The average BestGap of model M1 is 33.37%. This means that although the solution gaps of model M1-TEN are relatively large, the upper bounds can actually serve as good feasible solutions in practice.

Another important finding from this comparison is the memory usage of each model, especially for real-time applications. Overall, model M1-TEN consumes much less memory than models M2-STD and M3-CTR. Even for large-size instances, the average memory usage of model M1-TEN is 8.18 GB, which highlight the applicability of model M1-TEN in on-line computing. Model M3-CTR consumes more memory than model M1-TEN. This is due to the use of parallel computing in the branch and bound module, which involves a large number of branch and bound nodes being processed simultaneously. For

model M2-STD, the average memory usage on medium-size instances is 43.11 GB. Note that the memory statistics for model M2-STD was not reported for large-size instances, as it is still in the pre-solving stage when reaching the time limit. Another interesting finding is that the average memory usage of model M3-CTR on large-scale instances is less than that on medium-scale instances. This is due to the high complexity of large-scale instances, and only a few number of branch-and-bound nodes are explored within the pre-set time limit.

Table 25 Optimality Gaps of Models M1-TEN and M3-CTR on Large-size Instances.

Instance	M1-TEN (LR)				M3-CTR (BnP)			
	LB	UB	Gap (%)	BestGap (%)	LB	UB	Gap (%)	BestGap (%)
L1	203.71	632.0	67.77	15.98	531.0	543.87	2.37	2.37
L2	159.6	540.4	70.47	16.62	450.57	467.53	3.63	3.63
L3	112.4	629.6	82.15	12.10	553.44	558.19	0.85	0.85
L4	10.0	346.8	97.12	4.77	330.26	341.46	3.28	3.28
L5	10.0	789.8	98.73	52.90	371.99	403.93	7.91	7.91
L6	10.0	647.0	98.45	28.02	465.71	504.8	7.74	7.74
L7	69.64	1111.2	93.73	54.67	503.75	519.44	3.02	3.02
L8	10.0	625.8	98.40	51.48	303.62	320.1	5.15	5.15
L9	396.03	772.4	48.73	21.95	602.83	602.83	0.0	0.0
L10	27.06	637.4	95.76	40.05	382.14	399.41	4.32	4.32
L11	10.0	694.0	98.56	53.88	320.05	338.0	5.31	5.31
L12	162.28	1022.2	84.12	49.68	514.32	545.44	5.7	5.7
L13	313.87	935.8	66.46	31.68	639.36	678.18	5.72	5.72
Average			84.65	33.37			4.23	4.23

5.6.3 System Impact and Private Cost of Vehicle Routing

Table 26 presents the optimal solution costs of small-size instances for $\omega = 0.0$ and 1.0. Note that, to exactly measure the changes of optimal operation cost and system impact under different weights of system impact, only instances that can be solved to optimality are selected, thus instance S14 is not included in Table 26. Overall, after putting more priorities on the marginal cost of vehicle routing, the system impact obviously

decreases while the private operation cost increases. For each instance, OCPI (SIPD) denotes the percentage increase (decrease) of the instance's optimal operation cost (system impact) when varying ω from 0.0 to 1.0. Due to the diversity of selected instances, OCPI and SIPD may vary from one instance to another. Specifically, for instance S1, its operation cost increases by 4.23% when changing $\omega = 0.0$ from 0.0 to 1.0, while it increases by 69.35% in instance S4.

Table 26 Optimal Routing Costs of Small Instances under Different Weights of System Impact ω .

Instance	Weight of system impact $\omega = 0.0$			Weight of system impact $\omega = 1.0$			OCPI (%)	SIPD (%)
	Total cost	Operation cost	System impact	Total cost	Operation cost	System impact		
S1	112.38	112.38	297.55	391.51	117.13	274.38	4.23	7.79
S2	75.65	75.65	320.44	335.83	88.63	247.20	17.16	22.85
S3	73.03	73.03	350.95	348.17	88.26	259.91	20.85	25.94
S4	51.85	51.85	328.06	328.10	87.81	240.29	69.35	26.76
S5	82.36	82.36	347.14	350.75	118.10	232.65	43.43	32.98
S6	77.05	77.05	289.92	318.01	89.25	228.76	15.83	21.09
S7	131.87	131.87	396.73	449.78	148.25	301.53	12.42	24.00
S8	45.84	45.84	318.53	292.49	58.60	233.89	27.84	26.57
S9	80.02	80.02	339.51	362.46	89.43	273.03	11.76	19.58
S10	71.12	71.12	343.32	339.80	88.73	251.07	24.76	26.87
S11	130.69	130.69	259.40	360.36	146.78	213.58	12.31	17.66
S12	59.48	59.48	278.47	297.54	89.42	208.12	50.34	25.26
S13	78.53	78.53	385.28	389.30	89.67	299.63	14.19	22.23
S14	-	-	-	-	-	-	-	-
S15	111.40	111.40	343.32	396.13	147.65	248.48	32.54	27.63

Notes: OCPI – operation cost percentage increase, SIPD – system impact percentage decrease

As an example, Fig. 53 presents the Pareto frontier of operation cost and system impact on instance S5 with ω ranging from 0.0 to 1.0. Thus, a pareto-optimal solution should be systematically selected by decision makers and planners to balance these two important criteria.

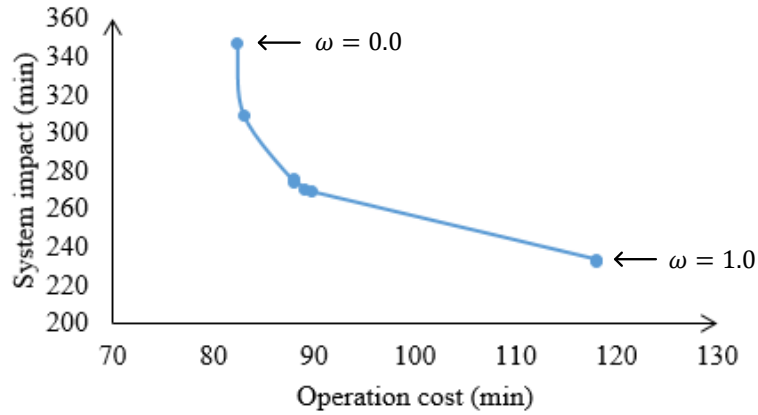


Fig. 53. Pareto Curve of Operation Cost and System Impact on Instance S5 with ω Ranging from 0.0 to 1.0.

5.7 Conclusions

This chapter focuses on formulating and solving RARPs in city logistics with a congested urban environment. Based on the fluid queuing model with a polynomial functional assumption for arrival flow rates, time-dependent link travel time as well as system-wide (societal) congestion impact was analytically derived. With the sprinkler truck routing problem as an example, there different mixed integer linear programming models were constructed. Two solution methods including Lagrangian relaxation and branch-and-price algorithm were developed for efficiently solving the models. Numerical experiments based on real-world traffic flow data were designed to investigate the effectiveness of the proposed models and solution methods.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

6.1 Summary of the Dissertation

This dissertation investigates the modeling of CAM systems on layered transportation networks, with a special focus on the following three research thrusts.

Research thrust 1: Layered CAM system modeling architecture. In Chapter 3, a new modeling framework with preliminary experiments for modeling CAM systems from a layered decomposition perspective was introduced. CAM systems with hierarchical structures are decomposed into strategic macroscopic, tactical mesoscopic, and operational microscopic layers. System modeling tasks with different computational and resolution requirements were performed on dedicated layers to seek a balance between the efficiency and fidelity in real-life deployments. The methodologies for CAM system simulation, optimization, and integrated simulation and optimization were comprehensively investigated based on the layered decomposition structure. Two open-source tools were developed to support CAM system modeling. `osm2gmns` aims to help users quickly prepare multiresolution transportation networks. `CAMLite` is intended to provide an integrated traffic simulation and optimization platform for modeling CAM systems. Numerical experiments were designed to illustrate the effectiveness of the proposed methodologies and open-source tools.

Research thrust 2: Cross-resolution traffic state estimation in CAM systems. In Chapter 4, the traffic system state estimation (TSSI) problem was introduced to systematically consider traffic state estimation, model parameter estimation, and queue profile estimation problems under a unified framework. Based on the fluid queue

approximation at the macroscopic level and the continuous space-time distribution function representation scheme at the mesoscopic level, the TSSI problem was formulated with a nonlinear optimization model in which traffic flow models and observations from different levels of resolution are systematically considered. The optimization model was cast in a customized computational graph and solved by a forward-backward algorithmic method, which take advantage of the computational efficiency and automatic differentiation techniques offered by the deep learning community for complex nonlinear but differentiable programming problems. Numerical experiments based on real-world and hypothetical datasets were designed to demonstrate the effectiveness of the proposed framework. Specifically, traffic states on road segments can be well reproduced using partially observed traffic data; the integrated modeling framework can help increase the accuracy of estimations; the analytically derived macroscopic system dynamic measures fit well with field observations; reliable traffic system state identification results can be obtained using the proposed joint estimation framework in real-life applications with limited observations; and the proposed modeling framework can be easily extended to a distributed version in largescale applications.

Research thrust 3: Integrated city logistics operation optimization in CAM systems. In Chapter 5, focusing on formulating and solving RARPs in city logistics with a congested urban environment, this research introduced a comprehensive modeling framework and exact solution algorithms. Specifically, based on the fluid queuing model with a polynomial functional assumption for arrival flow rates, time-dependent link travel time as well as system-wide (societal) congestion impact was analytically derived. Two new time-dependent travel time representation schemes were investigated. With the SRP as an

example, three optimization models, including a time-expanded network-based arc routing formulation, an arc-based node routing formulation, and a path-based node routing formulation, were constructed from different perspectives of capturing time-dependent travel time and formulating problem-specific constraints. In addition, two exact solution methods, i.e., Lagrangian relaxation and branch-and-price algorithm, were developed for efficiently solving the proposed models. Numerical experiments based on real-world traffic flow data were designed to demonstrate the effectiveness of the proposed models and solution methods.

6.2 Theoretical Contributions and Broader Impacts

The proposed research addresses several fundamental research issues in traffic monitoring and control systems. (1) This research offers a set of novel techniques on holistic traveler mobility optimization, agent-based sensing and control under the new environment of shared CAM networks. (2) A new class of differentiable computing-based algorithms on space-time traffic networks, including routing and scheduling problems for operations, is studied. The proposed algorithms could overcome the typical computational difficulties with a single-core CPU platform. (3) This study provides an innovative virtual-track resource management mechanism for dynamically scheduling a large number of SDVs. (4) A multi-resolution network system will allow next-generation traffic system operators to utilize internally consistent views to manage potential hotspots of traffic congestions. (5) This research has provided a set of open-source software packages for improving both research and education effectiveness on agent-based system optimization.

The theoretical methodologies, insights and open-source tools developed from this dissertation will be invaluable for modeling and optimizing new autonomous vehicle operation and control methods for metropolitan regions. This research will help transportation agencies to efficiently satisfy increasing transportation demand with a limited road expansion budget and constrained road capacity. Education-oriented software tools, and real-world case studies that can contribute to the training of future computer scientists, operations researchers and transportation engineers.

6.3 Future Research

Future studies on the CAM system architecture design include (1) integrating the proposed methodologies with a differentiable programming framework (Hu et al., 2019, Lu et al., 2022), and (2) comprehensive computational experiments on large-scale instances. In addition, for trajectory controls with a higher fidelity, the extension of the space-time approach for modeling high-dimensional vehicle dynamics (e.g., acceleration) without introducing nonlinearity for solving differential equations of motion is briefly discussed below. Fig. 54 demonstrates the addition of another velocity axis to the basic space-time network, where vehicle dynamics can be exactly represented by used space-time-velocity (STV) vertices. The green and gray vertices in the left part of Fig. 54 represent feasible STV vertices and infeasible vertices due to speed limits on links or acceleration restrictions of vehicles, respectively. In the right part of Fig. 54, the projection of a vehicle's STV path on the velocity-space coordinate plane shows the speed profile of the vehicle, and the projection on the space-time coordinate plane indicates the spatial trajectory. This STV network framework clearly demonstrates the location- and time-

dependent vehicle dynamics, such as the speed, acceleration, and deceleration. Additional constraints, including speed limits and acceleration restrictions associated with different types of vehicles, can be easily added to this representation.

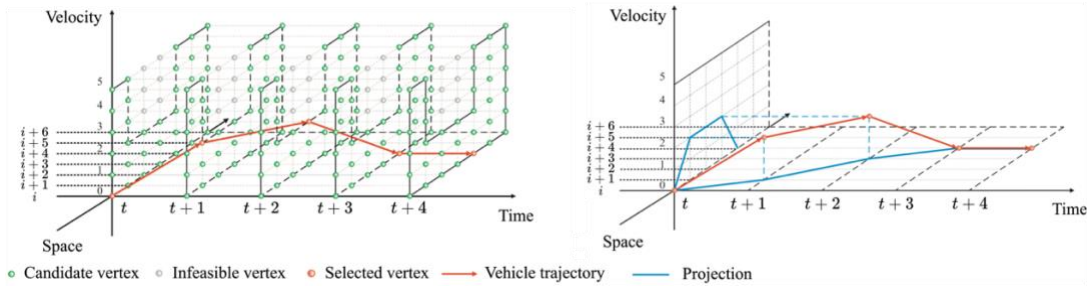


Fig. 54. Vehicle Dynamics Modeling on a Space-time-velocity Network (Adopted from Zhou et al., 2017).

The future studies on CAM system state estimation include (1) incorporating stochastic traffic flow models in the proposed framework to improve its performance under congested traffic conditions with high randomness and (2) extending the current framework to an online version to support real-time traffic management and control.

In terms of system optimization, this study can be extended along the following directions in the future. First, this research considers fixed departure time of service vehicles at the origin depot. Future studies could relax this restriction to allow the selection of proper departure times/schedules so as to minimize the overall cost. One interesting yet challenging topic along this line is how to produce optimal continuous-time path solutions without explicitly performing time discretization. This theoretically important and practically useful question was discussed in a recent paper by Boland et al., (2017) for the service network design problem. Second, with simplified fluid queue models, the system-wide (societal) congestion impact of routing solutions is considered in this research. The tradeoff among additional costs such as energy use, emissions reduction, and location

congestion reduction (Lam and Hentenryck, 2016) could be further systematically investigated under more realistic traffic flow modeling frameworks to accommodate different practical needs in real-life applications.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
- Ahn, B.H., Shin, J.Y., 1991. Vehicle-routing with time windows and time-varying congestion. *Journal of the Operational Research Society* 42(5), 393-400.
- Alvarez, L., Horowitz, R., 1999. Safe platooning in automated highway systems part I: Safety regions design. *Vehicle System Dynamics*, 32(1), 23-55.
- Amini, E., Omidvar, A., Elefteriadou, L., 2021. Optimizing operations at freeway weaves with connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 126, 103072.
- Amirgholy, M., Shahabi, M., Gao, H. O., 2020. Traffic automation and lane management for communicant, autonomous, and human-driven vehicles. *Transportation research part C: emerging technologies*, 111, 477-495.
- Antoniou, C., Ben-Akiva, M., Koutsopoulos, H. N., 2007. Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. *IEEE Transactions on Intelligent Transportation Systems*, 8(4), 661-670.
- Antoniou, C., Koutsopoulos, H. N., Yannis, G., 2013. Dynamic data-driven local traffic state estimation and prediction. *Transportation Research Part C: Emerging Technologies*, 34, 89-107.
- Baldacci, R., Mingozzi, A., Roberti, R., 2011. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research* 59(5), 1269-1283.
- Ban, X. J., Hao, P., Sun, Z., 2011. Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transportation Research Part C: Emerging Technologies*, 19(6), 1133-1156.
- Bang, S., Ahn, S., 2017. Platooning strategy for connected and autonomous vehicles: transition from light traffic. *Transportation Research Record*, 2623(1), 73-81.
- Bautista, J., Fernández, E., Pereira, J., 2008. Solving an urban waste collection problem using ants heuristics. *Computers & Operations Research* 35(9), 3020-3033.
- Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D., 2011. SUMO—simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.

- Bekiaris-Liberis, N., Roncoli, C., Papageorgiou, M., 2017. Highway traffic state estimation per lane in the presence of connected vehicles. *Transportation research part B: methodological*, 106, 1-28.
- Belieres, S., Hewitt, M., Jozefowicz, N., Semet, F., 2021. A time-expanded network reduction heuristic for the logistics service network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 147, 102203.
- Ben Ticha, H., Absi, N., Feillet, D., Quilliot, A., 2018. Vehicle routing problems with road-network information: State of the art. *Networks*, 72(3), 393-406.
- Ben Ticha, H., Absi, N., Feillet, D., Quilliot, A., Van Woensel, T., 2019. A branch-and-price algorithm for the vehicle routing problem with time windows on a road network. *Networks*, 73(4), 401-417.
- Ben Ticha, H., Absi, N., Feillet, D., Quilliot, A., Van Woensel, T., 2021, March. The Time-Dependent Vehicle Routing Problem with Time Windows and Road-Network Information. In *Operations Research Forum* (Vol. 2, No. 1, pp. 1-25). Springer International Publishing.
- Benders, J., 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerischeathemata*, 4(1), 238-252.
- Bertsekas, D. P., 1990. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4), 133-149.
- Bhaskar, A., Tsubota, T., Chung, E., 2014. Urban traffic state estimation: Fusing point and zone based data. *Transportation Research Part C: Emerging Technologies*, 48, 120-142.
- Bhattachan, A., Okin, G.S., Zhang, J., Vimal, S., Lettenmaier, D.P., 2019. Characterizing the role of wind and dust in traffic accidents in California. *GeoHealth* 3(10), 328-336.
- Birge, J. R., Louveaux, F., 2011. *Introduction to stochastic programming*. Springer Science & Business Media.
- Blazquez, C.A., Beghelli, A., Meneses, V.P., 2012. A novel methodology for determining low-cost fine particulate matter street sweeping routes. *Journal of the Air & Waste Management Association* 62(2), 242-251.
- Bodin, L.D., Kursh, S.J., 1978. A computer-assisted system for the routing and scheduling of street sweepers. *Operations Research* 26(4), 525-537.
- Boeing, G., 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65, 126-139.

- Boland, N. L., Savelsbergh, M. W., 2019. Perspectives on integer programming for time-dependent models. *Top*, 27(2), 147-173.
- Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. *Operations research*, 65(5), 1303-1321.
- Brockfeld, E., Kühne, R. D., Wagner, P., 2004. Calibration and validation of microscopic traffic flow models. *Transportation Research Record*, 1876(1), 62-70.
- Canepa, E. S., Claudel, C. G., 2017. Networked traffic state estimation involving mixed fixed-mobile sensor data using Hamilton-Jacobi equations. *Transportation Research Part B: Methodological*, 104, 686-709.
- Carey, M., Humphreys, P., McHugh, M., McIvor, R., 2014. Extending travel-time based models for dynamic network loading and assignment, to achieve adherence to first-in-first-out and link capacities. *Transportation Research Part B: Methodological*, 65, 90-104.
- Castro Campos, R.A., Rodríguez Villalobos, C.A., Zaragoza Martínez, F.J., 2020. Plowing with precedence in polynomial time. *Networks* 76(4), 451-466.
- Cattaruzza, D., Absi, N., Feillet, D., González-Feliu, J., 2017. Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics*, 6(1), 51-79.
- Chen, C., Wang, J., Xu, Q., Wang, J., Li, K., 2021. Mixed platoon control of automated and human-driven vehicles at a signalized intersection: dynamical analysis and optimal control. *Transportation Research Part C: Emerging Technologies*, 127, 103138.
- Chen, H. K., Hsueh, C. F., Chang, M. S., 2006. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(5), 383-408.
- Chen, L., Low, S. H., Chiang, M., Doyle, J. C., 2006. Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks.
- Chen, N., Wang, M., Alkim, T., Van Arem, B., 2018. A robust longitudinal control strategy of platoons under model uncertainties and time delays. *Journal of Advanced Transportation*, 2018.
- Chen, Y., Gao, L., Li, Z. P., Liu, Y. C., 2007, October. A new method for urban traffic state estimation based on vehicle tracking algorithm. In *2007 IEEE Intelligent Transportation Systems Conference* (pp. 1097-1101). IEEE.
- Cheng, P., Qiu, Z., Ran, B., 2006, September. Particle filter based traffic state estimation using cell phone network data. In *2006 IEEE Intelligent Transportation Systems Conference* (pp. 1047-1052). IEEE.

- Cheng, Q., Liu, Z., Guo, J., Wu, X., Pendyala, R., Belezamo, B., Zhou, X. S., 2022. Estimating key traffic state parameters through parsimonious spatial queue models. *Transportation Research Part C: Emerging Technologies*, 137, 103596.
- Cheng, Q., Liu, Z., Guo, J., Wu, X., Pendyala, R., Belezamo, B., Zhou, X. S., 2022. Estimating key traffic state parameters through parsimonious spatial queue models. *Transportation Research Part C: Emerging Technologies*, 137, 103596.
- Cheng, Q., Liu, Z., Guo, J., Wu, X., Pendyala, R., Belezamo, B., Zhou, X. S., 2022. Estimating key traffic state parameters through parsimonious spatial queue models. *Transportation Research Part C: Emerging Technologies*, 137, 103596.
- Cheu, R.L., Xie, C., Lee, D. H., 2002. Probe vehicle population and sample size for arterial speed estimation. *Computer-Aided Civil and Infrastructure Engineering*, 17(1), 53-60.
- Chiang, M., Low, S. H., Calderbank, A. R., Doyle, J. C., 2007. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1), 255-312.
- Chiariotti, F., Pielli, C., Zanella, A., Zorzi, M., 2018. A dynamic approach to rebalancing bike-sharing systems. *Sensors*, 18(2), 512.
- Chien, S. I. J., Goulias, D. G., Yahalom, S., Chowdhury, S. M., 2002. Simulation-based estimates of delays at freeway work zones. *Journal of Advanced Transportation*, 36(2), 131-156.
- Christofides, N., Mingozzi, A., Toth, P., 1981. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1), 255-282.
- Coifman, B., 2002. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. *Transportation Research Part A: Policy and Practice*, 36(4), 351-364.
- Coifman, B., 2015. Empirical flow-density and speed-spacing relationships: Evidence of vehicle length dependency. *Transportation Research Part B: Methodological*, 78, 54-65.
- Coifman, B., Li, L., 2022. A New Method for Validating and Generating Vehicle Trajectories From Stationary Video Cameras. *IEEE Transactions on Intelligent Transportation Systems*.
- Comert, G., 2013. Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals. *Transportation Research Part B: Methodological*, 55, 59-74.

- Comert, G., Cetin, M., 2011. Analytical evaluation of the error in queue length estimation at traffic signals from probe vehicle data. *IEEE Transactions on Intelligent Transportation Systems*, 12(2), 563-573.
- Corberán, Á., Eglese, R., Hasle, G., Plana, I., Sanchis, J.M., 2021. Arc routing problems: a review of the past, present, and future. *Networks* 77(1), 88-115.
- Corberán, Á., Laporte, G., 2015. *Arc routing: problems, methods, and applications*. SIAM, Philadelphia.
- Corberán, A., Prins, C., 2010. Recent results on arc routing problems: an annotated bibliography. *Networks* 56(1), 50-69.
- Cordeau, J. F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54(3), 573-586.
- Dabia, S., Ropke, S., Van Woensel, T., De Kok, T., 2013. Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation science*, 47(3), 380-396.
- Daganzo, C. F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4), 269-287.
- Daganzo, C. F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4), 269-287.
- Daganzo, C. F., 1995. The cell transmission model, part II: network traffic. *Transportation Research Part B: Methodological*, 29(2), 79-93.
- Daganzo, C. F., 2006. In traffic flow, cellular automata= kinematic waves. *Transportation Research Part B: Methodological*, 40(5), 396-403.
- de Rivera, A. D., Dick, C. T., 2021. Illustrating the implications of moving blocks on railway traffic flow behavior with fundamental diagrams. *Transportation Research Part C: Emerging Technologies*, 123, 102982.
- Del Pia, A., Filippi, C., 2006. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2), 125-141.
- Deng, W., Lei, H., Zhou, X., 2013. Traffic state estimation and uncertainty quantification based on heterogeneous data sources: A three detector approach. *Transportation Research Part B: Methodological*, 57, 132-157.

- Di, X., Shi, R., 2021. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transportation research part C: emerging technologies*, 125, 103008.
- Doan, D. L., Ziliaskopoulos, A., Mahmassani, H., 1999. On-line monitoring system for real-time traffic management applications. *Transportation Research Record*, 1678(1), 142-149.
- Donati, A. V., Montemanni, R., Casagrande, N., Rizzoli, A. E., Gambardella, L. M., 2008. Time dependent vehicle routing problem with a multi ant colony system. *European journal of operational research*, 185(3), 1174-1191.
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017, October. CARLA: An open urban driving simulator. In *Conference on robot learning* (pp. 1-16). PMLR.
- Duan, R., Pettie, S., 2014. Linear-time approximation for maximum weight matching. *Journal of the ACM (JACM)*, 61(1), 1-23.
- Duret, A., Yuan, Y., 2017. Traffic state estimation based on Eulerian and Lagrangian observations in a mesoscopic modeling framework. *Transportation research part B: methodological*, 101, 51-71.
- Dussault, B., Golden, B., Groër, C., Wasil, E., 2013. Plowing with precedence: a variant of the windy postman problem. *Computers & Operations Research* 40(4), 1047-1059.
- Dussault, B., Golden, B., Wasil, E., 2014. The downhill plow problem with multiple plows. *Journal of the Operational Research Society* 65(10), 1465-1474.
- Eglese, R.W., 1994. Routeing winter gritting vehicles. *Discrete Applied Mathematics* 48(3), 231-244.
- Eglese, R.W., Li, L.Y.O., 1992. Efficient routeing for winter gritting. *Journal of the Operational Research Society* 43(11), 1031-1034.
- Eglese, R.W., Murdock, H., 1991. Routeing road sweepers in a rural area. *Journal of the Operational Research Society* 42(4), 281-288.
- Feng, S., Yan, X., Sun, H., Feng, Y., Liu, H. X., 2021. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1), 1-14.
- FHWA, 2018. Congestion and Bottleneck Identification (CBI) Software Tool User's Guide (NO. FHWA-HRT-18-071).
- FHWA., 2022. CARMA Program. URL. <https://highways.dot.gov/research/operations/CARMA>.

- Figliozzi, M. A., 2012. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, 48(3), 616-636.
- Fleischmann, B., Gietz, M., Gnutzmann, S., 2004. Time-varying travel times in vehicle routing. *Transportation science*, 38(2), 160-173.
- Fountoulakis, M., Bekiaris-Liberis, N., Roncoli, C., Papamichail, I., Papageorgiou, M., 2017. Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing. *Transportation Research Part C: Emerging Technologies*, 78, 13-33.
- Gambatese, J.A., James, D.E., 2001. Dust suppression using truck-mounted water spray system. *Journal of Construction Engineering and Management* 127(1), 53-59.
- Gavriilidou, A., Daamen, W., Yuan, Y., Hoogendoorn, S. P., 2019. Modelling cyclist queue formation using a two-layer framework for operational cycling behaviour. *Transportation research part C: emerging technologies*, 105, 468-484.
- Gendreau, M., Ghiani, G., Guerriero, E., 2015. Time-dependent routing problems: A review. *Computers & operations research*, 64, 189-197.
- Geroliminis, N., Haddad, J., Ramezani, M., 2012. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(1), 348-359.
- Ghali, M. O., Smith, M. J., 1995. A model for the dynamic system optimum traffic assignment problem. *Transportation Research Part B: Methodological*, 29(3), 155-170.
- Ghiani, G., Guerriero, F., Improta, G., Musmanno, R., 2005. Waste collection in Southern Italy: solution of a real - life arc routing problem. *International Transactions in Operational Research* 12(2), 135-144.
- Ghiani, G., Improta, G., Laporte, G., 2001. The capacitated arc routing problem with intermediate facilities. *Networks* 37(3), 134-143.
- Ghiani, G., Laganà, D., Laporte, G., Mari, F., 2010. Ant colony optimization for the arc routing problem with intermediate facilities under capacity and length restrictions. *Journal of heuristics* 16(2), 211-233.
- Ghosh-Dastidar, S., Adeli, H., 2006. Neural network-wavelet microsimulation model for delay and queue length estimation at freeway work zones. *Journal of Transportation Engineering*, 132(4), 331-341.

- Gmira, M., Gendreau, M., Lodi, A., Potvin, J. Y., 2021. Managing in real-time a vehicle routing plan with time-dependent travel times on a road network. *Transportation Research Part C: Emerging Technologies*, 132, 103379.
- Gong, S., Du, L., 2018. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation research part B: methodological*, 116, 25-61.
- Gong, S., Zhou, A., Peeta, S., 2019. Cooperative adaptive cruise control for a platoon of connected and autonomous vehicles considering dynamic information flow topology. *Transportation research record*, 2673(10), 185-198.
- Greenshields, B. D., Bibbins, J. R., Channing, W. S., Miller, H. H., 1935. A study of traffic capacity. In *Highway research board proceedings (Vol. 1935)*. National Research Council (USA), Highway Research Board.
- Grossmann, C., Roos, H. G., Stynes, M., 2007. Numerical treatment of partial differential equations (Vol. 154). Berlin: Springer.
- Guo, H., Liu, J., Dai, Q., Chen, H., Wang, Y., Zhao, W., 2020. A distributed adaptive triple-step nonlinear control for a connected automated vehicle platoon with dynamic uncertainty. *IEEE Internet of Things Journal*, 7(5), 3861-3871.
- Guo, Y., Ma, J., Xiong, C., Li, X., Zhou, F., Hao, W., 2019. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation research part C: emerging technologies*, 98, 54-72.
- Hadi, M., Xiao, Y., Wang, T., Qom, S. F., Azizi, L., Iqbal, M. S., ... Massahi, A., 2016. Framework for multi-resolution analyses of advanced traffic management strategies.
- Hadi, M., Zhou, X., Hale, D., 2022. *Multiresolution Modeling for Traffic Analysis: Guidebook (No. FHWA-HRT-22-055)*. United States. Federal Highway Administration.
- Hadi, M., Zhou, X., Hale, D., 2022. *Multiresolution Modeling for Traffic Analysis: Guidebook (No. FHWA-HRT-22-055)*. United States. Federal Highway Administration.
- Haghani, A., Jung, S., 2005. A dynamic vehicle routing problem with time-dependent travel times. *Computers & operations research*, 32(11), 2959-2986.
- Hale, D., Jagannathan, R., Xyntarakis, M., Su, P., Jiang, X., Ma, J., Hu, J., Krause, C., 2016. *Traffic bottlenecks: identification and Solutions (No. FHWA-HRT-16-064)*. United States. Federal Highway Administration. Office of Operations Research and Development.

- Han, X., Ma, R., Zhang, H. M., 2020. Energy-aware trajectory optimization of CAV platoons through a signalized intersection. *Transportation Research Part C: Emerging Technologies*, 118, 102652.
- He, E. Y., Boland, N., Nemhauser, G., Savelsbergh, M., 2021. Dynamic discretization discovery algorithms for time-dependent shortest path problems. *INFORMS Journal on Computing*.
- Heidemann, D., 1996. A queueing theory approach to speed-flow-density relationships. In *transportation and traffic theory. Proceedings of the 13th international symposium on transportation and traffic theory*, Lyon, France, 24-26 July 1996.
- Herrera, J. C., Bayen, A. M., 2010. Incorporation of Lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B: Methodological*, 44(4), 460-481.
- Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., Bayen, A. M., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4), 568-583.
- Hewitt, M., 2022. The Flexible Scheduled Service Network Design Problem. *Transportation Science*.
- Horni, A., Nagel, K., Axhausen, K. W., 2016. Introducing matsim. In *The multi-agent transport simulation MATSim* (pp. 3-7). Ubiquity Press.
- Hou, T., Mahmassani, H. S., Alfelor, R. M., Kim, J., Saberi, M., 2013. Calibration of traffic flow models under adverse weather and application in mesoscopic network simulation. *Transportation research record*, 2391(1), 92-104.
- Hu, X., Sun, J., 2019. Trajectory optimization of connected and autonomous vehicles at a multilane freeway merging area. *Transportation Research Part C: Emerging Technologies*, 101, 111-125.
- Hu, Y., Anderson, L., Li, T. M., Sun, Q., Carr, N., Ragan-Kelley, J., Durand, F., 2019. DiffTaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*.
- Huang, K., Chen, X., Di, X., Du, Q., 2021. Dynamic driving and routing games for autonomous vehicles on networks: A mean field game approach. *Transportation Research Part C: Emerging Technologies*, 128, 103189.
- Huang, S.H., Lin, T.H., 2014. Using ant colony optimization to solve periodic arc routing problem with refill points. *Journal of Industrial and Production Engineering* 31(7), 441-451.

- Huang, Y., Zhao, L., Van Woensel, T., Gross, J. P., 2017. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95, 169-195.
- Hyland, M., Mahmassani, H. S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies*, 92, 278-297.
- Ichoua, S., Gendreau, M., Potvin, J. Y., 2003. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2), 379-396.
- Irnich, S., Villeneuve, D., 2006. The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$. *INFORMS Journal on Computing*, 18(3), 391-406.
- Jabali, O., Van Woensel, T., De Kok, A. G., Lecluyse, C., Peremans, H., 2009. Time-dependent vehicle routing subject to time delay perturbations. *Iie Transactions*, 41(12), 1049-1066.
- Jabari, S. E., Liu, H. X., 2013. A stochastic model of traffic flow: Gaussian approximation and estimation. *Transportation Research Part B: Methodological*, 47, 15-41.
- Jiang, X., Adeli, H., 2004. Object-oriented model for freeway work zone capacity and queue delay estimation. *Computer-Aided Civil and Infrastructure Engineering*, 19(2), 144-156.
- Jittrapirom, P., Caiati, V., Feneri, A. M., Ebrahimigharehbaghi, S., Alonso González, M. J., Narayan, J., 2017. Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges.
- Karimi, M., Roncoli, C., Alecsandru, C., Papageorgiou, M., 2020. Cooperative merging control via trajectory optimization in mixed vehicular traffic. *Transportation Research Part C: Emerging Technologies*, 116, 102663.
- Katrakazas, C., Quddus, M., Chen, W. H., Deka, L., 2015. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60, 416-442.
- Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P. T. P., 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.
- Kim, T., Zhou, X., Pendyala, R. M., 2021. Computational graph-based framework for integrating econometric models and machine learning algorithms in emerging data-driven analytical environments. *Transportmetrica A: Transport Science*, 1-30.

- Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Kirby, R. F., Potts, R. B., 1969. The minimum route problem for networks with turn penalties and prohibitions. *Transportation Research*, 3(3), 397-408.
- Koolstra, K., 1999, June. Potential benefits of a freeway slot-reservation system: Queuing costs versus scheduling costs. In *Proc. Urban Transport Systems Conference*.
- Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018, November. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2118-2125). IEEE.
- Kritzing, S., Doerner, K. F., Hartl, R. F., Kiechle, G. Ě., Stadler, H., Manohar, S. S., 2012. Using traffic information for time-dependent vehicle routing. *Procedia-Social and Behavioral Sciences*, 39, 217-229.
- Kuhne, R., Michalopoulos, P., 1997. Continuum flow models. *Traffic flow theory: A state of the art report* revised monograph on traffic flow theory, 18.
- Kuwahara, M., Takenouchi, A., Kawai, K., 2021. Traffic state estimation by backward moving observers: an application and validation under an incident. *Transportation research part C: emerging technologies*, 127, 103158.
- Lagos, F., Boland, N., Savelsbergh, M., 2022. Dynamic discretization discovery for solving the Continuous Time Inventory Routing Problem with Out-and-Back Routes. *Computers & Operations Research*, 105686.
- Lai, J., Hu, J., Cui, L., Chen, Z., Yang, X., 2020. A generic simulation platform for cooperative adaptive cruise control under partially connected and automated environment. *Transportation Research Part C: Emerging Technologies*, 121, 102874.
- Lam, E., Hentenryck, P. V., 2016. A branch-and-price-and-check model for the vehicle routing problem with location congestion. *Constraints*, 21(3), 394-412.
- Laval, J. A., Daganzo, C. F., 2006. Lane-changing in traffic streams. *Transportation Research Part B: Methodological*, 40(3), 251-264.
- Lawson, T. W., Lovell, D. J., Daganzo, C. F., 1997. Using input-output diagram to determine spatial and temporal extents of a queue upstream of a bottleneck. *Transportation Research Record*, 1572(1), 140-147.
- Lecluyse, C., Van Woensel, T., Peremans, H., 2009. Vehicle routing with stochastic time-dependent travel times. *4OR*, 7(4), 363-377.

- Lee, S., Wong, S. C., Li, Y. C., 2015. Real-time estimation of lane-based queue lengths at isolated signalized junctions. *Transportation Research Part C: Emerging Technologies*, 56, 1-17.
- Li, J., Lan, C. J., Gu, X., 2006. Estimation of incident delay and its uncertainty on freeway networks. *Transportation research record*, 1959(1), 37-45.
- Li, J.Q., Mirchandani, P.B., Knights, P., 2008. Water truck routing and location of refilling stations in open pit mines. In: *Proceedings of 2008 Australian Mining Technology Conference*, Sunshine Coast, Australia.
- Li, P. T., Zhou, X., 2017. Recasting and optimizing intersection automation as a connected-and-automated-vehicle (CAV) scheduling problem: A sequential branch-and-bound search approach in phase-time-traffic hypernetwork. *Transportation Research Part B: Methodological*, 105, 479-506.
- Li, P., Mirchandani, P., Zhou, X., 2015a. Hierarchical multiresolution traffic simulator for metropolitan areas: architecture, challenges, and solutions. *Transportation Research Record*, 2497(1), 63-72.
- Li, P., Mirchandani, P., Zhou, X., 2015b. Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork. *Transportation Research Part B: Methodological*, 81, 103-130.
- Li, X., Ghiasi, A., Xu, Z., Qu, X., 2018. A piecewise trajectory optimization model for connected automated vehicles: Exact optimization algorithm and queue propagation analysis. *Transportation Research Part B: Methodological*, 118, 429-456.
- Li, Z. C., Huang, H. J., Yang, H., 2020. Fifty years of the bottleneck model: A bibliometric review and future research directions. *Transportation research part B: methodological*, 139, 311-342.
- Lighthill, M. J., Whitham, G. B., 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178), 317-345.
- Lin, D.Y., Eluru, N., Waller, S.T. and Bhat, C.R., 2008. Integration of activity-based modeling and dynamic traffic assignment. *Transportation Research Record*, 2076(1), pp.52-61.
- Liu, H. X., Wu, X., Ma, W., Hu, H., 2009. Real-time queue length estimation for congested signalized intersections. *Transportation research part C: emerging technologies*, 17(4), 412-427.
- Liu, J., Mirchandani, P., Zhou, X., 2020. Integrated vehicle assignment and routing for system-optimal shared mobility planning with endogenous road congestion. *Transportation Research Part C: Emerging Technologies* 117, p.102675.

- Liu, J., Mirchandani, P., Zhou, X., 2020. Integrated vehicle assignment and routing for system-optimal shared mobility planning with endogenous road congestion. *Transportation Research Part C: Emerging Technologies*, 117, 102675.
- Liu, W., Yang, H., Yin, Y., 2014. Expirable parking reservations for managing morning commute with parking space constraints. *Transportation Research Part C: Emerging Technologies*, 44, 185-201.
- Longo, H., De Aragao, M. P., Uchoa, E., 2006. Solving capacitated arc routing problems using a transformation to the CVRP. *Computers & Operations Research*, 33(6), 1823-1837.
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y. P., Hilbrich, R., ... Wießner, E., 2018, November. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2575-2582). IEEE.
- Lu, C. C., Liu, J., Qu, Y., Peeta, S., Roupail, N. M., Zhou, X., 2016. Eco-system optimal time-dependent flow assignment in a congested network. *Transportation Research Part B: Methodological* 94, 217-239.
- Lu, C. C., Liu, J., Qu, Y., Peeta, S., Roupail, N. M., Zhou, X., 2016. Eco-system optimal time-dependent flow assignment in a congested network. *Transportation Research Part B: Methodological*, 94, 217-239.
- Lu, J., Chen, X., Nie, Q., Hong, R., Xia, J., 2018. K Shortest Path Searching for Time-Dependent Road Networks. In *CICTP 2017: Transportation Reform and Change—Equity, Inclusiveness, Sharing, and Innovation* (pp. 793-804). Reston, VA: American Society of Civil Engineers.
- Lu, J., Li, C., Wu, X. B., Zhou, X. S., 2022. Traffic System State Identification with Integrated Traffic State, Model Parameter and Queue Profile Estimation: Nonlinear Programming Reformulation with Differentiable Traffic State Variables Across Resolutions. Working Paper. <https://ssrn.com/abstract=4149585>.
- Luo, X., Liu, B., Jin, P. J., Cao, Y., Hu, W., 2019. Arterial traffic flow estimation based on vehicle-to-cloud vehicle trajectory data considering multi-intersection interaction and coordination. *Transportation Research Record*, 2673(6), 68-83.
- Ma, C., Yu, C., Yang, X., 2021. Trajectory planning for connected and automated vehicles at isolated signalized intersections under mixed traffic environment. *Transportation research part C: emerging technologies*, 130, 103309.
- Ma, F., Yang, Y., Wang, J., Liu, Z., Li, J., Nie, J., ..., Wu, L., 2019. Predictive energy-saving optimization based on nonlinear model predictive control for cooperative connected vehicles platoon with V2V communication. *Energy*, 189, 116120.

- Ma, G., Wang, B., Ge, S. S., 2022. Robust optimal control of connected and automated vehicle platoons through improved particle swarm optimization. *Transportation research part C: emerging technologies*, 135, 103488.
- Ma, J., Li, X., Zhou, F., Hu, J., Park, B. B., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part II: Computational issues and optimization. *Transportation Research Part B: Methodological*, 95, 421-441.
- Ma, K., Wang, H., Ruan, T., 2021. Analysis of road capacity and pollutant emissions: Impacts of Connected and automated vehicle platoons on traffic flow. *Physica A: Statistical Mechanics and its Applications*, 583, 126301.
- Ma, W., Pi, X., Qian, S., 2020. Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs. *Transportation Research Part C: Emerging Technologies*, 119, 102747.
- Ma, W., Qian, S., 2021. High-resolution traffic sensing with probe autonomous vehicles: A data-driven approach. *Sensors*, 21(2), 464.
- Mahbub, A. I., Malikopoulos, A. A., 2021. A platoon formation framework in a mixed traffic environment. *IEEE Control Systems Letters*, 6, 1370-1375.
- Mahmassani, H. S., 1992. Dynamic traffic assignment and simulation for advanced network informatics (DYNASMART). In the 2nd International Seminar on Urban Traffic Networks, 1992.
- Mahmassani, H. S., 1992. Dynamic traffic assignment and simulation for advanced network informatics (DYNASMART). In the 2nd International Seminar on Urban Traffic Networks, 1992.
- Mahmassani, H. S., Hou, T., Kim, J., Chen, Y., Hong, Z., Halat, H., Haas, R., 2014. Implementation of a weather responsive traffic estimation and prediction system (TrEPS) for signal timing at Utah DOT (No. FHWA-JPO-14-140). United States. Department of Transportation. Intelligent Transportation Systems Joint Program Office.
- Mahmoudi, M., Tong, L. C., Garikapati, V. M., Pendyala, R. M., Zhou, X., 2021. How many trip requests could we support? An activity-travel based vehicle scheduling approach. *Transportation Research Part C: Emerging Technologies*, 128, 103222.
- Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. *Transportation Research Part B: Methodological*, 89, 19-42.
- Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach

based on state–space–time network representations. *Transportation Research Part B: Methodological*, 89, 19-42.

Malandraki, C., 1989. Time-dependent vehicle routing problems: Formulations, solution algorithms and computational experiments (Doctoral dissertation, Northwestern University).

Malandraki, C., Daskin, M. S., 1992. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science* 26(3), 185-200.

Maniezzo, V., 2004. Algorithms for large directed CARP instances: urban solid waste collection operational support. UBLCS Technical Report Series, Bolonha, Italy: University of Bolonha, 27.

Marshall, L., Boland, N., Savelsbergh, M., Hewitt, M., 2021. Interval-based dynamic discretization discovery for solving the continuous-time service network design problem. *Transportation science*, 55(1), 29-51.

Martinelli, R., Pecin, D., Poggi, M., 2014. Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research* 239(1), 102-111.

Massahi, A., Hadi, M., Shams, K., Baqersad, M., 2019. Evaluating Incident Responsive Signal Control Plans using Multi-Resolution Modeling. *Transportation Research Record*, 2673(10), 804-813.

Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B: Methodological*, 67, 208-234.

Michalopoulos, P. G., Beskos, D. E., Lin, J. K., 1984. Analysis of interrupted traffic flow by finite difference methods. *Transportation Research Part B: Methodological*, 18(4-5), 409-421.

Michon, J. A., 1985. A critical view of driver behavior models: what do we know, what should we do?. In *Human behavior and traffic safety* (pp. 485-524). Springer, Boston, MA.

Mihaylova, L., Boel, R., Hegyi, A., 2007. Freeway traffic estimation within particle filtering framework. *Automatica*, 43(2), 290-300.

Miller, H. J., 2005. A measurement theory for time geography. *Geographical Analysis* 37(1), 17-45.

Mirheli, A., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2019. A consensus-based distributed trajectory control in a signal-free intersection. *Transportation research part C: emerging technologies*, 100, 161-176.

- Mohebifard, R., Hajbabaie, A., 2019. Optimal network-level traffic signal control: A benders decomposition-based solution algorithm. *Transportation Research Part B: Methodological*, 121, 252-274.
- Mohebifard, R., Hajbabaie, A., 2021. Trajectory control in roundabouts with a mixed fleet of automated and human-driven vehicles. *Computer-Aided Civil and Infrastructure Engineering*.
- Mourão, M. C., Almeida, M. T., 2000. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of operational research*, 121(2), 420-434.
- Mourão, M. C., Pinto, L. S., 2017. An updated annotated bibliography on arc routing problems. *Networks*, 70(3), 144-194.
- Mourão, M.C., Amado, L., 2005. Heuristic method for a mixed capacitated arc routing problem: a refuse collection application. *European Journal of Operational Research* 160(1), 139-153.
- Mu, C., Du, L., Zhao, X., 2021. Event triggered rolling horizon based systematical trajectory planning for merging platoons at mainline-ramp intersection. *Transportation research part C: emerging technologies*, 125, 103006.
- Nagel, K., Schreckenberg, M., 1992. A cellular automaton model for freeway traffic. *Journal de physique I*, 2(12), 2221-2229.
- Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., Chung, E., 2016. Real-time traffic state estimation in urban corridors from heterogeneous data. *Transportation Research Part C: Emerging Technologies*, 66, 99-118.
- Nanthawichit, C., Nakatsuji, T., Suzuki, H., 2003. Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. *Transportation research record*, 1855(1), 49-59.
- Nava, E., Shelton, J., Chiu, Y. C., 2012. Analyzing impacts of dynamic reversible lane systems using a multiresolution modeling approach (No. 12-4672).
- Nedić, A., Liu, J., 2018. Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1, 77-103.
- Newell, C., 2013. *Applications of queueing theory (Vol. 4)*. Springer Science & Business Media.
- Newell, G. F., 1993. A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B: Methodological*, 27(4), 281-287.

- Newell, G. F., 1993. A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B: Methodological*, 27(4), 281-287.
- Newell, G. F., 2002. A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, 36(3), 195-205.
- Newell, G.F., 1982. *Applications of queueing theory*, second ed. Chapman and Hall Ltd, New York.
- Ngoduy, D., Hoogendoorn, S. P., 2003. An automated calibration procedure for macroscopic traffic flow models. *IFAC Proceedings Volumes*, 36(14), 263-268.
- Niroumand, R., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2020. Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. *Transportation research part C: emerging technologies*, 116, 102659.
- Niu, H., Zhou, X., Tian, X., 2018. Coordinating assignment and routing decisions in transit vehicle schedules: A variable-splitting Lagrangian decomposition approach for solution symmetry breaking. *Transportation Research Part B: Methodological*, 107, 70-101.
- Ory, D., 2020. OSMnx Software Badge. URL. <https://medium.com/zephyrfoundation/osmnx-software-badge-3e206db65825>.
- Pallottino, S., Scutella, M. G., 1998. Shortest path algorithms in transportation models: classical and innovative aspects. In *Equilibrium and advanced transportation modelling* (pp. 245-281). Springer, Boston, MA.
- Palomar, D. P., Chiang, M., 2006. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8), 1439-1451.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.
- Payne, H. J., 1971. Models of freeway traffic and control. *Mathematical Models of Public Systems*, 1 (1), 51-61.
- Paz, A., Molano, V., Martinez, E., Gaviria, C., Arteaga, C., 2015. Calibration of traffic flow models using a memetic algorithm. *Transportation Research Part C: Emerging Technologies*, 55, 432-443.
- Pearn, W. L., Assad, A., Golden, B. L., 1987. Transforming arc routing into node routing problems. *Computers & operations research*, 14(4), 285-288.

- Perrier, N., Langevin, A., Campbell, J.F., 2007a. A survey of models and algorithms for winter road maintenance. Part III: Vehicle routing and depot location for spreading. *Computers & Operations Research* 34(1), 211-257.
- Perrier, N., Langevin, A., Campbell, J.F., 2007b. A survey of models and algorithms for winter road maintenance. Part IV: Vehicle routing and fleet sizing for plowing and snow disposal. *Computers & Operations Research* 34(1), 258-294.
- Phillips, W. F., 1979. A kinetic model for traffic flow with continuum implications. *Transportation Planning and Technology*, 5(3), 131-138.
- Psaraftis, H. N., Wen, M., Kontovas, C. A., 2016. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1), 3-31.
- Qin, X., Mahmassani, H. S., 2004. Adaptive calibration of dynamic speed-density relations for online network traffic estimation and prediction applications. *Transportation research record*, 1876(1), 82-89.
- Qu, Y., Zhou, X., 2017. Large-scale dynamic transportation network simulation: A space-time-event parallel computing approach. *Transportation research part c: Emerging technologies*, 75, 1-16.
- Quirion-Blais, O., Langevin, A., Trépanier, M., 2017. A case study of combined winter road snow plowing and de-icer spreading. *Canadian Journal of Civil Engineering* 44(12), 1005-1013.
- Quiros, A. R. F., Bedruz, R. A., Uy, A. C., Abad, A., Bandala, A., Dadios, E. P., 2016, November. Machine vision of traffic state estimation using fuzzy logic. In *2016 IEEE Region 10 Conference (TENCON)* (pp. 2104-2109). IEEE.
- Rajaram, R. N., Ohn-Bar, E., Trivedi, M. M., 2016. Looking at pedestrians at different scales: A multiresolution approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 17(12), 3565-3576.
- Ramezani, M., Geroliminis, N., 2015. Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civil and Infrastructure Engineering*, 30(6), 414-432.
- Ramezani, M., Haddad, J., Geroliminis, N., 2015. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. *Transportation Research Part B: Methodological*, 74, 1-19.
- Richards, P. I., 1956. Shock waves on the highway. *Operations research*, 4(1), 42-51.
- Riquelme-Rodríguez, J.P., Gamache, M., Langevin, A., 2014. Periodic capacitated arc-routing problem with inventory constraints. *Journal of the Operational Research Society* 65(12), 1840-1852.

- Robbins, H., Monro, S., 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400-407.
- Ruan, T., Wang, H., Zhou, L., Zhang, Y., Dong, C., Zuo, Z., 2022. Impacts of Information Flow Topology on Traffic Dynamics of CAV-MV Heterogeneous Flow. *IEEE Transactions on Intelligent Transportation Systems*.
- Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
- Sala, M., Soriguera, F., 2021. Capacity of a freeway lane with platoons of autonomous vehicles mixed with regular traffic. *Transportation research part B: methodological*, 147, 116-131.
- Salazar-Aguilar, M.A., Langevin, A., Laporte, G., 2012. Synchronized arc routing for snow plowing operations. *Computers & Operations Research* 39(7), 1432-1440.
- Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2), 579-590.
- Scano, G., Huguet, M. J., Ngueveu, S. U., 2015. Adaptations of k-shortest path algorithms for transportation networks. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 663-669). IEEE.
- Scherr, Y. O., Hewitt, M., Saavedra, B. A. N., Mattfeld, D. C., 2020. Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141, 164-195.
- Schöbel, A., 2017. An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies*, 74, 348-365.
- Seo, T., Bayen, A. M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. *Annual reviews in control*, 43, 128-151.
- Seo, T., Kawasaki, Y., Kusakabe, T., Asakura, Y., 2019. Fundamental diagram estimation by using trajectories of probe vehicles. *Transportation Research Part B: Methodological*, 122, 40-56.
- Seo, T., Kusakabe, T., Asakura, Y., 2015. Estimation of flow and density using probe vehicles with spacing measurement equipment. *Transportation Research Part C: Emerging Technologies*, 53, 134-150.
- Shahrbabaki, M. R., Safavi, A. A., Papageorgiou, M., Papamichail, I., 2018. A data fusion approach for real-time traffic state estimation in urban signalized links. *Transportation research part C: emerging technologies*, 92, 525-548.

- Shang, P., Li, R., Guo, J., Xian, K., Zhou, X., 2019. Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: a space-time-state hyper network-based assignment approach. *Transportation Research Part B: Methodological* 121, 135-167.
- Shang, P., Li, R., Guo, J., Xian, K., Zhou, X., 2019. Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: a space-time-state hyper network-based assignment approach. *Transportation Research Part B: Methodological*, 121, 135-167.
- Shang, P., Li, R., Guo, J., Xian, K., Zhou, X., 2019. Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: a space-time-state hyper network-based assignment approach. *Transportation Research Part B: Methodological*, 121, 135-167.
- Shelton, J., Wagner, J., Samant, S., Goodin, G., Lomax, T., Seymour, E., 2019. Impacts of connected vehicles in a complex, congested urban freeway setting using multi-resolution modeling methods. *International Journal of Transportation Science and Technology*, 8(1), 25-34.
- Shen, J., Kammara, E. K. H., Du, L., 2022. Fully distributed optimization-based CAV platooning control under linear vehicle dynamics. *Transportation Science*, 56(2), 381-403.
- Shi, R., Mo, Z., Huang, K., Di, X., Du, Q., 2021. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. *IEEE Transactions on Intelligent Transportation Systems*.
- Smith, S., Berg, I., Yang, C., 2020. *General Modeling Network Specification: documentation, software, and data*.
- Soleimaniamiri, S., Ghiasi, A., Li, X., Huang, Z., 2020. An analytical optimization approach to the joint trajectory and signal optimization problem for connected automated vehicles. *Transportation Research Part C: Emerging Technologies*, 120, 102759.
- Spiliopoulou, A., Kontorinaki, M., Papageorgiou, M., Kopelias, P., 2014. Macroscopic traffic flow model validation at congested freeway off-ramp areas. *Transportation Research Part C: Emerging Technologies*, 41, 18-29.
- Spiliopoulou, A., Papamichail, I., Papageorgiou, M., Tyrinopoulos, Y., Chrysoulakis, J., 2017. Macroscopic traffic flow model calibration using different optimization algorithms. *Operational Research*, 17(1), 145-164.
- Spliet, R., Dabia, S., Van Woensel, T., 2018. The time window assignment vehicle routing problem with time-dependent travel times. *Transportation Science*, 52(2), 261-276.

- Stern, R. E., Cui, S., Delle Monache, M. L., Bhadani, R., Bunting, M., Churchill, M., ... Work, D. B., 2018. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89, 205-221.
- Sun, P., Veelenturf, L. P., Hewitt, M., Van Woensel, T., 2018. The time-dependent pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 116, 1-24.
- Sun, W., Wang, S., Shao, Y., Sun, Z., Levin, M. W., 2022. Energy and mobility impacts of connected autonomous vehicles with co-optimization of speed and powertrain on mixed vehicle platoons. *Transportation Research Part C: Emerging Technologies*, 142, 103764.
- Sun, W., Wong, S. C., Zhang, P., Shu, C. W., 2011. A shock-fitting algorithm for the Lighthill–Whitham–Richards model on inhomogeneous highways. *Transportmetrica*, 7(2), 163-180.
- Sun, X., Muñoz, L., Horowitz, R., 2003, December. Highway traffic state estimation using improved mixture Kalman filters for effective ramp metering control. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475) (Vol. 6, pp. 6333-6338)*. IEEE.
- Sun, Z., Huang, T., Zhang, P., 2020. Cooperative decision-making for mixed traffic: A ramp merging example. *Transportation research part C: emerging technologies*, 120, 102764.
- Sun, Z., Jin, W. L., Ritchie, S. G., 2017. Simultaneous estimation of states and parameters in Newell’s simplified kinematic wave model with Eulerian and Lagrangian traffic data. *Transportation research part B: methodological*, 104, 106-122.
- Tagmouti, M., Gendreau, M., Potvin, J.Y., 2007. Arc routing problems with time-dependent service costs. *European Journal of Operational Research* 181(1), 30-39.
- Tagmouti, M., Gendreau, M., Potvin, J.Y., 2010. A variable neighborhood descent heuristic for arc routing problems with time-dependent service costs. *Computers & Industrial Engineering* 59(4), 954-963.
- Tagmouti, M., Gendreau, M., Potvin, J.Y., 2011. A dynamic capacitated arc routing problem with time-dependent service costs. *Transportation Research Part C: Emerging Technologies* 19(1), 20-28.
- Tajalli, M., Hajbabaie, A., 2021. Traffic signal timing and trajectory optimization in a mixed autonomy traffic stream. *IEEE Transactions on Intelligent Transportation Systems*.

- Tampère, C. M., Immers, L. H., 2007, September. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. In 2007 IEEE Intelligent Transportation Systems Conference (pp. 209-216). IEEE.
- Tao, S., Manolopoulos, V., Rodriguez Duenas, S., Rusu, A., 2012. Real-time urban traffic state estimation with A-GPS mobile phones as probes. *Journal of Transportation Technologies*, 2(1), 22-31.
- Tawarmalani, M., Sahinidis, N. V., 2004. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical programming*, 99(3), 563-591.
- Taylor, J., Zhou, X., Roupail, N. M., Porter, R. J., 2015. Method for investigating intradriver heterogeneity using vehicle trajectory data: A dynamic time warping approach. *Transportation Research Part B: Methodological*, 73, 59-80.
- Thodi, B. T., Khan, Z. S., Jabari, S. E., Menéndez, M., 2022. Incorporating kinematic wave theory into a deep learning method for high-resolution traffic speed estimation. *IEEE Transactions on Intelligent Transportation Systems*.
- Tiapraserit, K., Zhang, Y., Wang, X. B., Zeng, X., 2015. Queue length estimation using connected vehicle technology for adaptive signal control. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 2129-2140.
- Tong, L. C., Zhou, L., Liu, J., Zhou, X., 2017. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transportation Research Part C: Emerging Technologies*, 85, 451-475.
- Tong, L., Zhou, X., Miller, H. J., 2015. Transportation network design for maximizing space–time accessibility. *Transportation Research Part B: Methodological* 81, 555-576.
- Treiber, M., Helbing, D., 2002. Reconstructing the spatio-temporal traffic dynamics from stationary detector data. *Cooperative Transportation Dynamics*, 1(3), 3-1.
- Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., Hennig, M., 2019. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53(1), 185-202.
- van Erp, P. B., Knoop, V. L., Hoogendoorn, S. P., 2018. Macroscopic traffic state estimation using relative flows from stationary and moving observers. *Transportation Research Part B: Methodological*, 114, 281-299.
- Van Lint, J. W. C., Calvert, S. C., 2018. A generic multi-level framework for microscopic traffic simulation—Theory and an example case in modelling driver distraction. *Transportation Research Part B: Methodological*, 117, 63-86.

- Van Lint, J. W. C., Hoogendoorn, S. P., 2010. A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways. *Computer-Aided Civil and Infrastructure Engineering*, 25(8), 596-612.
- Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N., 2007. A queueing framework for routing problems with time-dependent travel times. *Journal of Mathematical Modelling and Algorithms*, 6(1), 151-173.
- Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N., 2008. Vehicle routing with dynamic travel times: A queueing approach. *European journal of operational research*, 186(3), 990-1007.
- Vickrey, W. S., 1963. Pricing in urban and suburban transport. *The American Economic Review*, 53(2), 452-465.
- Vickrey, W. S., 1963. Pricing in urban and suburban transport. *The American Economic Review*, 53(2), 452-465.
- Vidal, T., Martinelli, R., Pham, T. A., Hà, M. H., 2021. Arc Routing with Time-Dependent Travel Times and Paths. *Transportation Science*, 55(3), 706-724.
- Von Neumann, J., 1951. The general and logical theory of automata. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior*. New York: Wiley, 1-41.
- Vu, D. M., Hewitt, M., Vu, D. D., 2022. Solving the time dependent minimum tour duration and delivery man problems with dynamic discretization discovery. *European Journal of Operational Research*.
- Wächter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25-57.
- Waltz, R. A., Morales, J. L., Nocedal, J., Orban, D., 2006. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical programming*, 107(3), 391-408.
- Wang, C., Gong, S., Zhou, A., Li, T., Peeta, S., 2020. Cooperative adaptive cruise control for connected autonomous vehicles by factoring communication-related constraints. *Transportation Research Part C: Emerging Technologies*, 113, 124-145.
- Wang, C., Xie, Y., Huang, H., Liu, P., 2021. A review of surrogate safety measures and their applications in connected and automated vehicles safety modeling. *Accident Analysis & Prevention*, 157, 106157.
- Wang, H., Cheu, R. L., 2013. Operations of a taxi fleet for advance reservations using electric vehicles and charging stations. *Transportation research record*, 2352(1), 1-10.

- Wang, J., Gong, S., Peeta, S., Lu, L., 2019. A real-time deployable model predictive control-based cooperative platooning approach for connected and autonomous vehicles. *Transportation Research Part B: Methodological*, 128, 271-301.
- Wang, J., Lu, L., Peeta, S., 2022. Real-time deployable and robust cooperative control strategy for a platoon of connected and autonomous vehicles by factoring uncertain vehicle dynamics. *Transportation Research Part B: Methodological*, 163, 88-118.
- Wang, P., Wu, X., He, X., 2020. Modeling and analyzing cyberattack effects on connected automated vehicular platoons. *Transportation research part C: emerging technologies*, 115, 102625.
- Wang, Q., Yang, X., Huang, Z., Yuan, Y., 2020. Multi-vehicle trajectory design during cooperative adaptive cruise control platoon formation. *Transportation research record*, 2674(4), 30-41.
- Wang, R., Fan, S., Work, D. B., 2016. Efficient multiple model particle filtering for joint traffic state estimation and incident detection. *Transportation Research Part C: Emerging Technologies*, 71, 521-537.
- Wang, Y., Papageorgiou, M., 2005. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39(2), 141-167.
- Wang, Y., Papageorgiou, M., Messmer, A., 2006. RENAISSANCE—A unified macroscopic model-based approach to real-time freeway network traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 14(3), 190-212.
- Wang, Y., Papageorgiou, M., Messmer, A., Coppola, P., Tzimitsi, A., Nuzzolo, A., 2009. An adaptive freeway traffic state estimator. *Automatica*, 45(1), 10-24.
- Wang, Y., Zhao, M., Yu, X., Hu, Y., Zheng, P., Hua, W., ... Guo, J., 2022. Real-time joint traffic state and model parameter estimation on freeways with fixed sensors and connected vehicles: State-of-the-art overview, methods, and case studies. *Transportation Research Part C: Emerging Technologies*, 134, 103444.
- Wei, Y., Avci, C., Liu, J., Belezamo, B., Aydın, N., Li, P. T., Zhou, X., 2017. Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. *Transportation research part B: methodological*, 106, 102-129.
- Whitham, G. B., 1974. *Linear and nonlinear waves*. John Wiley & Sons.
- Willemse, E.J., Joubert, J.W., 2016a. Constructive heuristics for the mixed capacity arc routing problem under time restrictions with intermediate facilities. *Computers & Operations Research* 68, 30-62.

- Willemse, E.J., Joubert, J.W., 2016b. Splitting procedures for the mixed capacitated arc routing problem under time restrictions with intermediate facilities. *Operations Research Letters* 44(5), 569-574.
- Willemse, E.J., Joubert, J.W., 2019. Efficient local search strategies for the mixed capacitated arc routing problems under time restrictions with intermediate facilities. *Computers & Operations Research* 105, 203-225.
- Wøhlk, S., 2008. A decade of capacitated arc routing. In *The vehicle routing problem: latest advances and new challenges* (pp. 29-48). Springer, Boston, MA.
- Wolfram, S., 1983. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3), 601.
- Wong, S. C., Wong, G. C. K., 2002. An analytical shock-fitting algorithm for LWR kinematic wave model embedded with linear speed–density relationship. *Transportation Research Part B: Methodological*, 36(8), 683-706.
- Woo, S., Skabardonis, A., 2021. Flow-aware platoon formation of Connected Automated Vehicles in a mixed traffic with human-driven vehicles. *Transportation research part C: emerging technologies*, 133, 103442.
- Work, D. B., Blandin, S., Tossavainen, O. P., Piccoli, B., Bayen, A. M., 2010. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010(1), 1-35.
- Work, D. B., Tossavainen, O. P., Blandin, S., Bayen, A. M., Iwuchukwu, T., Tracton, K., 2008, December. An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In *2008 47th IEEE Conference on Decision and Control* (pp. 5062-5068). IEEE.
- Wu, J., Ahn, S., Zhou, Y., Liu, P., Qu, X., 2021. The cooperative sorting strategy for connected and automated vehicle platoons. *Transportation Research Part C: Emerging Technologies*, 123, 102986.
- Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. *Transportation Research Part C: Emerging Technologies*, 96, 321-346.
- Xiao, Y., Konak, A., 2016. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E: Logistics and Transportation Review* 88, 146-166.
- Xie, X. F., Smith, S. F., Lu, L., Barlow, G. J., 2012. Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies*, 24, 168-189.

- Xing, Y., Lv, C., Cao, D., Velenis, E., 2021. Multi-scale driver behavior modeling based on deep spatial-temporal representation for intelligent vehicles. *Transportation research part C: emerging technologies*, 130, 103288.
- Xiong, X., Sha, J., Jin, L., 2021. Optimizing coordinated vehicle platooning: An analytical approach based on stochastic dynamic programming. *Transportation Research Part B: Methodological*, 150, 482-502.
- Yang, G., Tian, Z., Xu, H., Wang, Z., Wang, D., 2018. Impacts of traffic flow arrival pattern on the necessary queue storage space at metered on-ramps. *Transportmetrica A: Transport Science*, 14(7), 543-561.
- Yang, X. T., Huang, K., Zhang, Z., Zhang, Z. A., Lin, F., 2020. Eco-driving system for connected automated vehicles: multi-objective trajectory optimization. *IEEE Transactions on Intelligent Transportation Systems*, 22(12), 7837-7849.
- Yang, Y., Wang, J., 2020. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*.
- Yao, H., Li, X., 2020. Decentralized control of connected automated vehicle trajectories in mixed traffic at an isolated signalized intersection. *Transportation research part C: emerging technologies*, 121, 102846.
- Yao, H., Li, X., 2021. Lane-change-aware connected automated vehicle trajectory optimization at a signalized intersection with multi-lane roads. *Transportation research part C: emerging technologies*, 129, 103182.
- Yao, Y., Van Woensel, T., Veelenturf, L.P., Mo, P., 2021. The consistent vehicle routing problem considering path consistency in a road network. *Transportation Research Part B: Methodological*, 153, 21-44.
- Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Tong, L. C., Zhou, X., 2019. ADMM-based problem decomposition scheme for vehicle routing problem with time windows. *Transportation Research Part B: Methodological*, 129, 156-174.
- Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Tong, L.C., Zhou, X., 2019. ADMM-based problem decomposition scheme for vehicle routing problem with time windows. *Transportation Research Part B: Methodological* 129, 156-174.
- Yperman, I., 2007. The link transmission model for dynamic network loading.
- Yu, C., Feng, Y., Liu, H. X., Ma, W., Yang, X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological*, 112, 89-112.

- Yuan, Y., Van Lint, J. W. C., Wilson, R. E., van Wageningen-Kessels, F., Hoogendoorn, S. P., 2012. Real-time Lagrangian traffic state estimator for freeways. *IEEE Transactions on Intelligent Transportation Systems*, 13(1), 59-70.
- Yuan, Y., Zhang, Z., Yang, X. T., Zhe, S., 2021. Macroscopic traffic flow modeling with physics regularized Gaussian process: A new insight into machine learning applications in transportation. *Transportation Research Part B: Methodological*, 146, 88-110.
- Zaveria, K., 2022. Nvidia wants to be the brains of your self-driving cars: Drive thor. URL. <https://www.analyticsinsight.net/nvidia-wants-to-be-the-brains-of-your-self-driving-cars-drive-thor>.
- Zhan, X., Li, R., Ukkusuri, S. V., 2020. Link-based traffic state estimation and prediction for arterial networks using license-plate recognition data. *Transportation Research Part C: Emerging Technologies*, 117, 102660.
- Zhang, H. M., 1998. A theory of nonequilibrium traffic flow. *Transportation Research Part B: Methodological*, 32(7), 485-498.
- Zhang, H., Du, L., Shen, J., 2022. Hybrid MPC system for platoon based cooperative lane change control using machine learning aided distributed optimization. *Transportation Research Part B: Methodological*, 159, 104-142.
- Zhang, Y., Peng, Q., Lu, G., Zhong, Q., Yan, X., Zhou, X., 2022. Integrated line planning and train timetabling through price-based cross-resolution feedback mechanism. *Transportation Research Part B: Methodological*, 155, 240-277.
- Zhang, Y., Peng, Q., Yao, Y., Zhang, X., Zhou, X., 2019. Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and Alternating Direction Method of Multipliers methods. *Transportation Research Part B: Methodological* 128, 344-379.
- Zhang, Y., Peng, Q., Yao, Y., Zhang, X., Zhou, X., 2019. Solving cyclic train timetabling problem through model reformulation: extended time-space network construct and alternating direction method of multipliers methods. *Transportation Research Part B: Methodological*, 128, 344-379.
- Zhao, Y., Zheng, J., Wong, W., Wang, X., Meng, Y., Liu, H. X., 2019. Various methods for queue length and traffic volume estimation using probe vehicle trajectories. *Transportation Research Part C: Emerging Technologies*, 107, 70-91.
- Zheng, F., Jabari, S. E., Liu, H. X., Lin, D., 2018. Traffic state estimation using stochastic Lagrangian dynamics. *Transportation Research Part B: Methodological*, 115, 143-165.

- Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. *Transportation research part B: methodological*, 60, 16-32.
- Zheng, Z., Su, D., 2016. Traffic state estimation through compressed sensing and Markov random field. *Transportation Research Part B: Methodological*, 91, 525-554.
- Zhong, Z., Lee, E. E., Nejad, M., Lee, J., 2020. Influence of CAV clustering strategies on mixed traffic flow characteristics: An analysis of vehicle trajectory data. *Transportation Research Part C: Emerging Technologies*, 115, 102611.
- Zhou, H., Zhou, A., Li, T., Chen, D., Peeta, S., Laval, J., 2022. Congestion-mitigating MPC design for adaptive cruise control based on Newell's car following model: History outperforms prediction. *Transportation Research Part C: Emerging Technologies*, 142, 103801.
- Zhou, J., Zhu, F., 2021. Analytical analysis of the effect of maximum platoon size of connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 122, 102882.
- Zhou, L., Tong, L. C., Chen, J., Tang, J., Zhou, X., 2017. Joint optimization of high-speed train timetables and speed profiles: A unified modeling approach using space-time-speed grid networks. *Transportation Research Part B: Methodological*, 97, 157-181.
- Zhou, X., Cheng, Q., Wu, X., Li, P., Belezamo, B., Lu, J., Abbasi, M., 2022. A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio. *Multimodal Transportation*, 1(2), 100017.
- Zhou, X., Hadi, M., Hale, D. K., 2021. *Multiresolution Modeling for Traffic Analysis: State-of-Practice and Gap Analysis Report (No. FHWA-HRT-21-082)*. United States. Federal Highway Administration.
- Zhou, X., Hadi, M., Hale, D. K., 2021. *Multiresolution Modeling for Traffic Analysis: State-of-Practice and Gap Analysis Report (No. FHWA-HRT-21-082)*. United States. Federal Highway Administration.
- Zhou, X., Tanvir, S., Lei, H., Taylor, J., Liu, B., Rouphail, N. M., Frey, H. C., 2015. Integrating a simplified emission estimation model and mesoscopic dynamic traffic simulator to efficiently evaluate emission impacts of traffic management strategies. *Transportation Research Part D: Transport and Environment*, 37, 123-136.
- Zhou, X., Taylor, J., 2014. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration.
- Zhou, X., Tong, L., Mahmoudi, M., Zhuge, L., Yao, Y., Zhang, Y., ..., Shi, T., 2018. Open-source VRPLite package for vehicle routing with pickup and delivery: a path finding engine for scheduled transportation systems. *Urban Rail Transit*, 4(2), 68-85.

- Zhou, X.S., Cheng, Q., Wu, X., Li, P., Belezamo, B., Lu, J., Abbasi, M., 2022. A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio. *Multimodal Transportation*, 1(2), 100017.
- Zhou, Y., Ahn, S., Wang, M., Hoogendoorn, S., 2020. Stabilizing mixed vehicular platoons with connected automated vehicles: An H-infinity approach. *Transportation Research Part B: Methodological*, 132, 152-170.
- Ziliaskopoulos, A. K., Mahmassani, H. S., 1996. A note on least time path computation considering delays and prohibitions for intersection movements. *Transportation Research Part B: Methodological*, 30(5), 359-367.
- Zimmermann, H., 1980. OSI reference model-the ISO model of architecture for open systems interconnection. *IEEE Transactions on communications*, 28(4), 425-432.