UAB
Universitat Autònoma de Barcelona

Smart Traffic Control for the Era of Autonomous Driving

Jianglin Qiao

WESTERN SYDNEY UNIVERSITY
&
UNIVERSITY AUTONOMA DE BARCELONA

Smart Traffic Control for the Era of Autonomous Driving

by

Jianglin Qiao

Supervised by:

A/Prof Dongmo Zhang, Dr. Dave De Jonge

Prof Carles Sierra

Academic Tutor:

Prof Carles Sierra

A thesis submitted in partial fulfilment for the
Dual award degrees of Doctor of Philosophy in
Computer Science (UAB) & Engineering (WSU)

in the

School of Computer, Data and Mathematical Sciences
&
Artificial Intelligence Research Institute(IIIA-CSIC)

December 19, 2022

Declaration of Authorship

The work presented in this thesis is, to the best of my knowledge and belief, original except as acknowledged in the text. I hereby declare that I have not submitted this material, either in full or in part, for a degree at Western Sydney University, University Autonoma de Barcelona, Artificial Intelligence Research Institute or any other university/institution.

Signature: 

Date: 19/12/2022

Thesis co-supervised by
Dongmo Zhang (WSU)
Dave de Jonge (IIIA-CSIC)
Simeon Simoff (WSU)
Carles Sierra (IIIA-CSIC)

Authored by
Jianglin Qiao

Abstract

In the past decade, the research on autonomous vehicles (AVs) has made revolutionary progress. The advancements in Artificial Intelligence (AI), and especially machine learning, allow self-driving cars to learn how to handle complex road situations based on data from millions of accumulated driving hours, much more than any human driver could ever reach. Autonomous driving brings us hope for safer, more convenient, more efficient, and more environmentally friendly transportation. However, autonomous vehicles on roads also introduce new challenges to traffic management. New theories for a better understanding of the new era of transportation and new technologies for smart roadside infrastructures and intelligent traffic control are crucial for the development and deployment of autonomous vehicles as well as human communities.

This thesis aims to take on the challenges to address some of the key issues in traffic control and management, including intersection protocol design, congestion measurement, selfish routing and road infrastructure automation, under the assumption that all vehicles on the road are connected and self-driving.

To design and test traffic control mechanisms for AVs, we introduced a formal model to represent road networks and traffic. Based on this model, we developed a simulation system on top of an existing open-source platform (AIM4) and used it to examine a number of traffic management protocols specifically designed for traffic with fully autonomous vehicles. Simulation outcomes show that traffic management protocols for AVs can be more subtle, sensitive and variable with traffic volumes/flow rate, vehicle safe distance and road configuration. In addition, by analyzing the real-world traffic data and simulation data, we found that measuring congestion with exponential functions has considerable advantages against the traditional BPR function in certain aspects.

The deployment of autonomous vehicles provides traffic management with an opportunity of choosing either centralised control or decentralised control. The price of anarchy (PoA) of autonomous decision-making for routing gives an applicable quantitative criterion for selection between them. We extended the existing research on PoA with the

class of exponential functions as cost functions. We found an expression for the tight upper bound of the PoA for selfish routing games with exponential cost functions. Unlike existing studies, this upper bound depends on traffic demands, with which we can get a more accurate estimation of the PoA. Furthermore, by comparing the upper-bounds of PoA between the BPR function and the exponential function, we found that the exponential functions yield a smaller upper bound than the BPR functions in relatively low traffic flows.

To specify traffic management systems with autonomous roadside facilities, we propose a hybrid model of traffic assignment. This model aims to describe traffic management systems in which both vehicles and roadside controllers make autonomous decisions, therefore, are autonomous agents. We formulated a non-linear optimization problem to optimize traffic control from a macroscopic view of the road network. To avoid the complex calculations required for non-linear optimization, we proposed an approximation algorithm to calculate equilibrium routing and traffic control strategies. The simulation results show that this algorithm eventually converges to a steady state. The traffic control scheme in this steady state is an approximately optimal solution.

Acknowledgements

I have had the privilege of meeting many people who have helped me selflessly during my long overseas trip, and I give my highest respect to these lovely people.

First of all, I owe my family members a debt of gratitude much more significant than anything I can express, especially to my mother, Jie Jiang and my father, Laijun Qiao. I could not have had what I now have without their unwavering support. Their constant guidance, encouragement, and support have helped me achieve my current state. To my wife Yi Li, thanks for her continuous companionship, care and encouragement in supporting my PhD trip. I am thrilled to have her in my life.

I wish to express the same tribute as my family to my principal supervisor, A/Prof. Dongmo Zhang. He has always guided me with excellent support and patience toward the right path in my life, not just in the academic field. I learned a lot from him: how to be a world-class researcher and excellent academic, what is the correct teaching attitude, how to make an excellent presentation and even how to be a great teacher. Most importantly, I am grateful for his systematic guidance and great enthusiasm in my academic training. His academic rigour and passion for all of his students have influenced my life.

I also profoundly thank my co-supervisor, Dr Dave de Jonge, for his excellent guidance and generous support during my doctoral period. Throughout, he maintained a high level of mathematical skill and sensitivity to help me sort out the theoretical difficulties that I encountered and develop correct mathematical thinking. He helped a lot in the academic field and assisted living in Spain. His helpful and positive attitude in life will continue to influence me for the rest of my life.

Furthermore, to Dongmo and Dave, I am so proud to have Prof. Simeon Simoff and Prof. Carles Sierra on my supervisor panel. They have helped me wholeheartedly with my academic career, not only with academic papers and thesis writing but also with teaching me a positive and rigorous attitude towards research and how to be an excellent researcher. With the active coordination between Western Sydney University and IIIA-CSIC, I could get this brilliant dual award doctoral program with a fee waiver. They also gave me valuable advice, not only in my field of research but also about my future academic career, which has benefited me a lot. More specifically, I would like to thank Prof Ramon Lopez de Mantaras and Prof Simeon Simoff, who initially set up the

dual award Cotutelle program in 2016 and let me as the first candidate to enrol in the program.

I thank A/Prof. Yingbin Feng for his excellent advice and referrals that led me to such a great supervisor panel. He also gave me valuable advice on life and research and motivated me to finish my studies. In addition, I am very grateful to A/Prof. Hamish Macdougall from the University of Sydney. During my undergraduate studies, he took me into the field of research and got me interested in it. On behalf of my family, I would like to thank all of my supervisors for their positive impact on my whole life.

I express my sincere gratitude to Ms Felicity Koulouris and Ms Trish Saladine. Many thanks for their help with the daily chores of school, such as airline reservations, conference registration, and funding applications. I thank Prof. Jordi Gonzalez Sabaté. Thanks for guiding enrollment in Spain and for various official documents to help me with my Spanish Visa.

To all my friends from Australia, Spain, and China. Thanks for your companionship, care, and sharing, which made my life memorable. I would not have been able to do this without their support.

Finally, thanks to Western Sydney University, IIIA-CSIC, and Universitat Autònoma de Barcelona for the Cotutela agreement, with the help of my supervisor, have provided me with an excellent learning experience.

Contents

List of Figures

List of Tables

To my dad, Laijun Qiao, my mum, Jie Jiang, my wife Yi Li and my loving families

Chapter 1

Introduction

This thesis aims to address a few key issues in traffic control and management, including intersection protocol design, congestion measurement, selfish routing and road infrastructure automation, under the assumption that all vehicles on the road are connected and self-driving. This chapter will show the general motivation for this research with a brief review of the literature. It also contains a summary of the main methodologies, major contributions as well as the overall structure of the thesis.

1.1 Motivation

Transportation is at the heart of our society. Better transportation solutions mean larger potential markets, faster supply and demand matching, more specialization, higher productivity, and more innovation. Automation promises to overcome one of the last obstacles to the almost infinite growth of the transportation industry, namely human drivers. Even if humans can be good drivers, they need breaks, make mistakes, and can be expensive. The basic concept of autonomous vehicles on the road refers to replacing part or all of the human driving labor with electronic and mechanical devices [Shladover, 2018]. Autonomous driving has the potential to reduce car accidents, alleviate traffic

congestion, and increase time and fuel efficiency [Shi and Prevedouros, 2016, Wei et al., 2017].

The idea of driving automation was conceived as early as 1918, and General Motors exhibited the first concept of self-driving vehicles in 1939 [Pendleton et al., 2017, Shladover, 2018]. Since then, research and development (R& D) efforts have continued apace in academia and industry. The initial phase of R& D was initiated in the 1950s by General Motors and Sarnoff Laboratories of the Radio Corporation of America from 1964 to 2003, under separate and joint initiatives of different government agencies and academics. In 2004, the U.S. accelerated the research on self-driving vehicles through the Department of Defense Advanced Research Projects Agency's (DARPA) Grand Challenges program. These challenges led to AVs capable of traversing desert terrain in 2005 and 2007. The researchers also succeeded in putting AVs on urban roads through DARPA's Urban Challenge program. For example, auto parking is a common function in a normal human-driven car. It can provide safety in parking and is easy to use by pushing one button. Some governments give a great deal of support to it to develop safe and reliable autonomous vehicles. For example, the German government helps automobile manufacturers, such as Audi, Mercedes Benz, and BMW, test their AVs in a real environment to improve the reliability of autonomous driving. Research and development efforts have accelerated to make the concept of autonomous cars a reality. Furthermore, the global automobile sector invests around $77 billion in research & development to promote innovation and maintain competitiveness [Association et al., 2015, Nieuwenhuijsen, 2015].

With the rapid development of sensors, computing and artificial intelligence technologies, autonomous driving research has become extremely active. An autonomous vehicle system integrates many technologies, including computer vision, graphical processing, navigation, sensor technologies, etc. Most significantly, the recent advance in machine learning technologies enables a self-driving car to learn to drive in any complex road situation with millions of accumulated driving hours, which are much higher than any experienced human driver can reach. However, driving is not a purely technical job

but involves complicated social activities, which could be hard to reproduce purely with machine learning algorithms. For example, if two cars meet on a narrow road or a long bridge through which only one car can go through, how do the cars decide which one should reverse to give way to the other? Many such situations require direct interaction between vehicles and infrastructure or between vehicles and authorities [Cui et al., 2017, Gruel and Stanford, 2016]. Such demands push AV research in a different direction from machine learning concerning communication, negotiation, and cooperation among autonomous vehicles. However, coordinating autonomous vehicles on the road remains a daunting task for vehicles that usually travel at relatively high speeds. Therefore, conducting in-depth research on the traffic management of autonomous vehicles is necessary. In the future, fully autonomous driving will be considered the ultimate goal of driver assistance systems.

When an autonomous vehicle appears on a real-world road, it must be capable of handling various complex situations, such as lane keeping, vehicle following, lane changing, lane merging, avoiding collisions with dynamic or static objects, and interacting with other related facilities and vehicles, following the same traffic rules. Developing an autonomous vehicle with these functionalities requires highly intelligent algorithms to support and integrate the latest hardware technologies in perception, localization, decision-making, trajectory planning, communication, and control. On the other hand, the emergence of self-driving vehicles poses a serious challenge to traffic management. New theories for a better understanding of the new era of transportation and new technologies for smart roadside infrastructures and intelligent traffic control are crucial for the development and deployment of autonomous vehicles as well as human communities. This thesis aims to address the following research questions:

1. *How to formally represent road networks and traffic so that AVs can easily understand and reason about them?*

   A self-driving vehicle needs to understand the road it travels. A formal model of road networks and traffic is essential. Such a model should be able to specify

the road configuration, connections, potential collision, right of road and traffic situations.

2. *How to design traffic control protocols to manage traffic with AVs?*

   The advent of self-driving cars has brought new challenges to traffic management. Traditional traffic management is based on traffic signals or signs, such as traffic lights, traffic circles, and the like. Although self-driving vehicles can continue using existing traffic management approaches through vision or sensors, this is inefficient. With V2V, V2I and V2X connections, the approaches to control and manage AV traffic can be totally different. Many traffic management protocols that are too ideal for human-driven vehicles, such as first-come first-serve, can apply to autonomous vehicles. Furthermore, autonomous vehicles can negotiate with each other or negotiate with roadside facilities for better traffic throughput or travel efficiency.

3. *How to model interaction between AVs as well as their interaction with roadside infrastructures?*

   An autonomous vehicle is an intelligent agent, or named a mobile robot. Traffic with fully autonomous vehicles is a system of multiple agents. AVs can not only communicate each other but also can compete each other for use of roads or roadside facilities, causing congestion, possible collisions, misuse or damage of road facilities and so on. AVs can also cooperate for more efficient use of roads or escaping from traffic hazards. Moreover, self-controlled road facilities can also be part of the multi-agent systems of AVs, playing roles of global/local coordinators, competitors/collaborators with other road facilities and so on.

4. *How to optimise the efficiency of traffic control and management?*

   Autonomous vehicles can make decisions themselves to maximize their benefits or minimize travel costs. However, such an individual optimisation normally cannot achieve global optimisation. Notably, autonomous driving technology makes it possible to achieve optimal social benefits through centralised control. The

choice of centralised or decentralised control of traffic is not only a practical issue but a profound theoretical research topic in game-theory, known as the price of anarchy [Roughgarden and Tardos, 2002a].

This thesis will provide partial answers to the above fundamental research questions on autonomous driving. Question 1 will be addressed in Chapter 2 and part of Charter 3. Question 2 will be addressed in Chapters 3 and 6. Partial answers to Question 3 can be found in Chapter 4 and 5. Chapter 6 is to provide a partial solution to Question 4.

1.2 Related Work

This section discusses the literature related to the following topics: Intelligent Traffic Management Protocols, Road Network Modeling, and Traffic Assignment Problems.

1.2.1 Intelligent traffic management protocols

The management of intersections is crucial to effective road traffic networks. Some studies focus on innovative intersection management techniques to replace signal time allocation. Intersection management has been a research topic in various research institutes for a long time. Still, the rapid development of artificial intelligence and driverless cars has become an even more relevant topic, and many related papers have been published in recent years. To handle AVs at isolated intersections, Dresner and Stone [2004a] developed the reservation-based autonomous intersection management (AIM) system. A CAV sends a request to the control centre when it reaches an intersection and waits for a response. The FCFS principle and space-time resource allocation are the sources of the AIM concept. AIM has since been expanded in several studies. AIM with pure vehicle-to-vehicle communication and no control centre was proposed by VanMiddlesworth et al. [2008] and used at a low-traffic intersection. Au and Stone [2010] looked at the connection between the effectiveness of the AIM controller and

the precision of vehicle movements. Micro-simulation tests were used to assess the performance of AIM under challenging circumstances [Au et al., 2014, Fajardo et al., 2011, Li et al., 2013]. Existing research unequivocally demonstrates that AIM is more effective at an isolated intersection than signal time allocation (STA).

Currently, there are two main mechanisms for autonomous intersection management. The first mechanism is implemented from central control, which is called centralized intersection management. In this approach, all vehicles in the intersection area are under the control of a central control unit. According to this policy, several methods have been proposed recently. Autonomous intersection management based on auctions proposed by Carlino et al. [2013] is a new approach implemented from a stop sign policy. It introduced an auction mechanism to allow autonomous vehicles to be bid through. Another interesting approach is based on a first-come-first-served protocol [Dresner and Stone, 2004b], which allows autonomous vehicles that move in the same direction to form a queue with fixed size and then pass through the intersection as a 'platoon. This is named *platoon-based* intersection management [Bashiri and Fleming, 2017]. The second mechanism is based on the autonomous multi-agent system (MAS). It lets different agents negotiate with each other to decide who will be allowed to pass the intersection [Dresner and Stone, 2008a, Lamouik et al., 2017]. In this scheme, one can regard an Autonomous Vehicle as one kind of agent and the intersection control as another kind of agent [Hausknecht et al., 2011]. Transportation networks can consist of multiple Intersection Agents (IA), and different agreements can be reached between AVs and IAs through negotiation protocols. Vehicle-to-Vehicle(V2V) is an automobile technology designed to allow automobiles to "talk" to each other. It lets vehicles negotiate with each other to solve the traffic problem for themselves, such as Platoon-Based Intersection Management [Bashiri and Fleming, 2017] and Auction-Based Intersection Management [Carlino et al., 2013], Synchronization-Based Intersection Control [Tlig et al., 2014], and Smart Multi-Agent Traffic Coordinator for Autonomous Vehicles at Intersections [Lamouik et al., 2017]. In addition, Vehicle to Infrastructure (V2I) is a communication model that allows vehicles to share information with the components

that support a country's highway system. Many centralized mechanisms have been published to solve the traffic management of autonomous vehicles based on V2I communication [Chan, 2017, Dresner and Stone, 2004b, Kaths et al., 2015, Xu et al., 2017].

1.2.2 Road network modeling and traffic assignment problem

Road networks are made up of numerous interconnected roads. Many methods have been proposed to represent road networks. Graph theory is the most widely used in road network modeling [Jiang and Claramunt, 2004, Mackaness, 1995, Mackaness and Beard, 1993, Thomson and Richardson, 1995, Zhang, 2005]. Several ideas and criteria are taken from graph theory to facilitate the structural study of road networks and route selection. These include connectedness, lowest-cost spanning tree, and shortest-path spanning tree [Zhang, 2004].

Traffic assignment introduced by Wardrop [1952a] in 1952 describes how traffic demand is assigned to different routes when given the topology of the road network. It is one of the most important research topics in transportation. The congestion game, introduced by Rosenthal [1973], is a common-use model to formulate traffic assignment problems. The traffic assignment problem [Dafermos and Sparrow, 1969] is an important research topic in transportation. In a road network, different numbers of vehicles pass between different origins and destinations. However, the different choices of vehicle routes can cause different congestion on different roads. Therefore, rational planning of vehicle choice is another important direction to improve traffic efficiency [Golden et al., 2008].

The traffic assignment problem is usually defined formally using the congestion game model [Gibbons et al., 1992], in which the cost of each player is determined by the resources it selects and the number of other players who also select that resource. Each road in the road network has an independent cost function that outputs the travel time of this road using the number of vehicles travelling on that road as input. In such a transportation system, the travel time of a vehicle depends on the chosen roads and the

number of vehicles selected on the same roads [Wang et al., 2015]. First, [Rosenthal, 1973] proposed the task allocation problem, which later becomes a congestion game like the original traffic assignment problem. [Roughgarden and Tardos, 2002b] uses non-cooperative games to explore the game theoretical properties of traffic assignment problems by considering vehicles as autonomous self-interested agents. Sandholm [2010] further simplifies this idea by focusing on population games and using potential functions to find equilibrium solutions.

Furthermore, some studies use the existing model to solve the traffic assignment problem in road networks. The method of successive averages (MSA) is the algorithm most widely used to find the solution to traffic assignment [Mounce and Carey, 2015]. Liu et al. [2009] proposed a method of successive weighted averages (MSWA) to obtain results faster than the original MSA. There are also some common algorithms, such as origin-based algorithms [Bar-Gera, 2002], path-based algorithms [Jayakrishnan et al., 1994] and the Frank-Wolfe algorithm [Fukushima, 1984]. Autonomous vehicles in our model are assumed to make self-interested decisions, like human drivers, who only consider their own interests, such as how to reach their destination in the shortest possible time. The purpose of the road network model is to allow autonomous vehicles to reason about the network structure and traffic management protocols, which helps them to make decisions. The main motivation for our emphasis on autonomous driving is the basis that vehicles have more powerful means of communication than humans to access more information about the road, helping them make more informed decisions.

1.2.3 Latency/cost functions

In traffic assignment, the cost of individual vehicle or vehicles in a road segment or the whole road network is mainly measured by the volume of delay of their travel. A latency function (or called delay function) is a function maps traffic flow to volume of delay.

Many different latency functions have been proposed and used in practice. All these cost functions follow the basic principles of traffic flow theory, which state that speed decreases as the traffic flow or saturation rate increases. The saturation rate of a road is calculated as the relationship between traffic flow and capacity, where the capacity is unknown and is a problem of estimation in transportation research [Chen et al., 1999, Dheenadayalu et al., 2004, Morlok and Chang, 2004]. Among the well-developed cost functions, the Bureau of Public Roads (BPR) function is the most commonly used one. In addition to the functions above in literature, there are several other functions, such as the Vatzek function [Jastrzebski, 2000], the conical function [Spiess, 1990] and Mosher functions [Mosher Jr, 1963].

1.2.4 Price of anarchy

There is a lot of research on the Price of Anarchy (PoA) from a game-theoretical point of view [Aland et al., 2011, Christodoulou and Koutsoupias, 2005, Feldman et al., 2016]. The PoA in traffic assignment was first investigated by Roughgarden and Tardos [2002b] and shows that, with linear cost functions, the tight upper bound of the PoA is precisely equal to $\frac{4}{3}$. Furthermore, Roughgarden [2003] provides tight upper bounds of the PoA for several common cost functions, shown in Table 1.1. These results reveal only the maximum value the PoA could reach, often significantly higher than the actual value achieved. In previous studies, the main results of the upper bound of the PoA depended on the characteristics of the cost function, such as the highest-power term of all cost functions on the road network, rather than considering the impact of changes in traffic demand on it in a realistic sense [O'Hare et al., 2016a]. But some studies did try to find an expression for the PoA in terms of traffic demand. As an example, an expression to describe the relationship between traffic demand and PoA in the road network is presented in [Cominetti et al., 2021], and O'Hare et al. [2016b] proposed a mechanism to explain the relationship between the PoA and traffic demand. In addition, there is

Description	Representative	Price of Anarchy
Linear	$ax + b$	$\frac{4}{3} \approx 1.333$
Quadratic	$ax^2 + bx + c$	$\frac{3\sqrt{3}}{3\sqrt{3}-2} \approx 1.626$
Cubic	$ax^3 + bx^2 + cx + d$	$\frac{4\sqrt[4]{4}}{4\sqrt[4]{4}-3} \approx 1.896$
Polynomial	$\sum_{i=0}^{p} a_i x^i$	$(1 - p(p+1)^{-\frac{p+1}{p}})^{-1}$
M/M/1 delay functions	$(u - x)^{-1}$	$\frac{1}{2}(1 + \sqrt{\frac{u_{min}}{u_{min}-R_{max}}})$
Exponential	$ae^{bx} + c$	$\frac{2\hat{b}\hat{r}}{log(\hat{b}\hat{r}+1)}$

TABLE 1.1: The Upper Bound of PoA for Common Cost Functions (the last row (see Theorem 5.25) represents our contribution in this thesis).

some data-driven approach to estimate PoA in a given road network [Zhang et al., 2016, 2018].

1.3 Methodologies

This section presents some basic and major methods used in this thesis.

1.3.1 Game theory and multi-agent system

The coordination of systems made up of several agents is the focus of the artificial intelligence study field known as multi-agent systems. An agent is a discrete entity that can act independently to achieve its objectives. These objectives may be expressed as logical statements that must be true, or they may be expressed as a utility function that converts each conceivable condition of any action into a real value. A multi-agent system (MAS) is a collection of agents for each of which the achievement of its goals depends on the actions of the other agents in the MAS.

The fact that the results of one agent's actions may also depend on those of other agents is a crucial component of multi-agent systems. Since there would not be a multi-agent system without this, only a collection of separate individual agents would exist if this were not the case. Agents must therefore take into account the actions other agents have

taken or will take when deciding what to do. Any situation, including a group of self-interested agents and a strategic decision-making process, can be conceived as a game in which individual and collective decisions affect the result.

Such decision-making issues are described by the mathematical theory known as game theory. Although the game theory may appear to be about playing games at first glance, it relates to a considerably wider variety of circumstances. It is relevant to any decision problem in a selfish, multi-agent system and to any circumstances where one person's decision depends on the decisions of others and has various aims. Game theory forecasts rational agent behavior and outlines the necessary course of action. For instance, when you join an auction for a product, the winner is decided by your bidding strategy and your rival's strategy. In game theory, an individual cost function for the agent, often a player, is typically considered, that is, used to give each potential outcome of the game a cost. When cost minimization is the objective, each agent makes a decision. However, if the participants do not work together, a circumstance may eventually develop in which everyone loses.

In this thesis, we assume that the self-driving car is a selfish autonomous agent and that the vehicle's behavior can be expressed as a cost function. The major methodology in this thesis is to treat each autonomous vehicle and smart roadside infrastructure as an intelligent agent or a robot so that the whole transport system becomes a multi-agent (multi-robot) system. Based on this framework, we investigated several game-theoretic properties, such as Nash equilibrium, global optimization of vehicle decisions, and the road network's price of anarchy (PoA).

1.3.2 Simulation environment

Our study assumes that vehicles are fully automated and have independent decision-making capabilities, but current technology does not satisfy this assumption. Therefore, it is necessary to simulate the autonomous driving environment. Second, self-driving vehicles need simulators to help develop them, rather than driving them directly on real

roads, which is a considerable safety risk. Since, in this thesis, traffic assignment and traffic management optimisation are considered for autonomous driving, the simulator is an essential part.

Automated driving simulators are divided into a variety of uses. Some simulators focus on single-vehicle decision-making, using visual analysis or sensor perception of the surrounding environment to provide consideration for vehicle decisions. Some simulators focus on machine learning applications, where vehicles are trained from existing data. Others focus on macro-traffic management, which is used in this thesis. The simulator is used as a simulation environment from a single intersection to a network of multiple intersections to verify the effectiveness of our proposed algorithms and traffic management protocols and to compare the efficiency with existing management protocols.

First, we use Gazebo, an open-source 3D robotics simulator built in the robot operating system (ROS), to create a virtual environment for autonomous driving. In this simulator, we simulated the traffic control protocols of a single intersection. We tested different traffic management protocols for their efficiency with AVs (programmed in Python). However, experiments based on robot simulation can only deal with a minor traffic volume. For a more complicated road network with a large traffic volume, we proposed a formal definition of traffic management protocols for intelligent vehicles based on our spatial model of the road network. We developed a standalone simulation system, AIM4+, based on the autonomous intersection management (AIM) simulation of the University of Texas. The system is written in Java. In addition, we have also implemented it into a generic simulator to test the game-theoretic properties of a transport system. Based on the generic simulation system, we can test various properties of intelligent traffic management systems and autonomous vehicles from macro- to micro-perspectives of transport networks. In addition to the two simulators mentioned above, we used the most commonly used commercial simulator, AIMSUN. We tested traffic assignment algorithms and traffic management optimization problems and obtained good results.

1.3.3 Data analysis

With the development of science and technology, the improvement of intelligent road network technology, and the increasing number of car owners, the traffic pressure on the road network increases daily. Data analysis can effectively deal with a large amount of collected traffic information, which can alleviate traffic pressure on the road network to a greater extent. Traffic data is usually very complicated, and the information channels are diverse, so mining the potential value of traffic data can make traffic data processing efficient and further realize intelligent traffic management of road networks. The traffic data set includes a range of traffic flow conditions and different types of roads (such as ramps, straights, bends, and intersections) (low, moderate, congested, etc.). These data are ideal for studies of traffic flow (speed and density), simulation of microscopic traffic models, and the creation and evaluation of autonomous driving prediction and planning algorithms.

The data used in the data analysis in this thesis relate to a real Australian traffic database, as well as several different automated driving simulators. We are adopting real-world traffic data from selected Australian roads into our implemented simulator. The simulator produces data for autonomous driving traffic management to validate our proposed model, traffic management protocols, and theoretical results. Additionally, we discovered a new latency function by analysing real-time traffic data in Sydney and Melbourne acquired from Intelematics. The function differs from the traditional latency function, which combines an exponential and linear increase in latency with traffic flow.

1.4 Major Contributions

The major contributions of this thesis are the following:

- Introduced a formal model to specify road networks and traffic for autonomous vehicles to understand and reason about road and traffic situations. The model describes roads and traffic from three different perspectives:
  - Micro-level representation of road networks: Describe the connection between intersections and roads, specify connections between the road lanes of an intersection and possible collisions in an intersection. This allows autonomous vehicles to reason about road situations.
  - Macro-level representation of road networks: Describe regional roadworks and their relationships. This allows autonomous vehicles to plan their travel routing to minimise traffic latency.
  - Dynamic of traffic: Describe vehicles moving on the road with variation in time. This allows autonomous vehicles to reason about the dynamic of traffic for collision avoidance and efficiency of travel.
- Designed and tested a number of traffic management protocols for autonomous vehicles based on the spatiotemporal representation of traffic. These management protocols describe how vehicles cross intersections from the micro-level perspective of a road network. We extended the AIM4 system with generic priority-based control protocols and independent control of multiple intersections. We tested a variety of settings for each protocol and related algorithms and compared their efficiency.
- Investigated an exponential cost function in the traffic assignment problem with real-world data support. The results of the data analysis show that the exponential cost function fits the real-world data better than the commonly used BPR cost function for the heavy volume of traffic situations. The exponential cost function can not only be used for the micro-level representation of road networks but also suitable for the macroscopic road network. The research discovery has potential applications to regional transport planning and macroscopic traffic control, as well as the analysis of the price of anarchy.

- Found an expression of the tight upper-bound of the price of anarchy for the class of self-routing games with exponential cost functions. We compared this expression and the corresponding expressions for routing games with the most commonly used cost function, the BPR latency function. This comparison demonstrates that the tight upper bound of the PoA of games with exponential functions is lower than the corresponding value with the BPR function as long as the traffic volume is less than the capacity of the road. The exponential function can obtain a considerably lower upper bound, illustrated by utilizing real-world traffic data and mimicking road latency as closely as a BPR function with even tighter exponential parameters.

- Proposed a multi-agent system based hybrid model for traffic control optimisation with roadside facilities as part of the multi-agent system of AVs. This model combines two levels of optimisation - equilibrium reached via individual optimisation and global optimisation of traffic control. We designed an algorithm to approximate the solution of such a nonlinear optimization problem. This algorithm significantly reduces the total delay in the road network, as demonstrated by the results of our experiments with the AIMSUN simulation software.

1.5 Outline of Chapters

Chapter 2 introduces the formal representation of road networks based on different levels of abstraction, which can be used for autonomous vehicles reasoning about the structure of the road. We first introduce a generic road network model used to abstract roads' connection between positions, called the macro-level model. We then extended the macro-road network model with more detailed information about roads, called the meso-level road network model. It can describe the connection between intersections and internal relations of a single intersection. Finally, we propose a microscopic road network model to describe more specific traffic information. This model is based on discrete time to describe vehicles that travel on the road network.

Chapter 3 investigates various autonomous driving traffic management protocols and will build on the aforementioned paradigm. We provide foundation knowledge on protocols for intelligent traffic management and an autonomous vehicle simulator. Second, we present several priority-based traffic management solutions for autonomous vehicles in a meso-level road network model. The suggested methods are also tested using the general simulator AIM4+. Then, we examine time- and priority-based traffic management methods. Finally, we use a general robot operating system simulator that treats autonomous vehicles as robots to collect data and analyze associated algorithms.

Chapter 4 proposes a cost function in the form of an exponential function in the traffic assignment problem. First, we introduce the formal model of traffic assignment problems with an exponential cost function. We then validate the exponential function using real-world traffic data from three specific local government areas of New South Wales, Australia, with heavy traffic volumes selected for data fitting. We have used this data to compare the exponential function with several other, more commonly used cost functions. Detailed results of the data analysis show that the exponential cost function outperforms the BPR and Akcelik functions to describe the relationship between regional traffic flow and travel cost.

Chapter 5 focuses on traffic assignment in road networks with exponential cost functions by applying the model introduced in Chapter 2 and the cost function proposed in Chapter 4. We model a traffic network as a routing game in which vehicles are selfish agents who choose routes to travel autonomously to minimize the travel delay caused by road congestion. We concentrate on routing games where the latency of road traffic may be characterized by an exponential function, in contrast to previous research where the latency function of traffic congestion was based on BPR or queuing theory. We first calculate a tight upper bound for the price of anarchy for this class of games and then compare this result with the tight upper bound of the PoA for routing games with the BPR latency function. The comparison shows that as long as the traffic volume is less than the road capacity, the tight upper bound of the PoA of the games with exponential functions is less than the corresponding value with the BPR function.

Chapter 6 proposes a multi-agent based method to describe traffic control optimization for autonomous vehicle assignment problems on road networks. We first present a formal model for abstract road networks. We then extend the road network model into a game-theoretical model based on population games to describe the behavior of autonomous vehicles under intelligent traffic control. Based on this model, we investigate a traffic control optimization problem that aims to improve the efficiency of road networks and provides an algorithm to find an approximate solution. Lastly, our algorithm significantly reduces the total delay of the road network, as demonstrated by the results of our experiments with the Aimsun simulation software.

Chapter 7 summarizes this thesis and discusses some directions for future work.

Chapter 2

Road Network Modeling

This chapter addresses the formal representation of road networks based on different levels of abstraction. These road network models allow autonomous vehicles to reason about the network topology, helping them make decisions and enabling self-driving cars to understand how roads are connected according to the formal requirements of different scenarios. First, we briefly overview the background and motivation to introduce road network modelling for autonomous vehicles in section 2.1. Second, we introduce a Macro-level road network model in Section 2.2 that aims to allow vehicles to think about the connection of the road in a large area. We then extend the Macro-level road network framework to more detailed road connections with intersections and road lanes, which we called Meso-level road networks, in Section 2.3. These models can allow vehicles to understand more specific road information and traffic management protocols, which will be discussed in Chapter 3, to make better decisions. Finally, in Section 2.4 we introduce a discrete-time microscopic road network model to describe changes in vehicle location information over time.

2.1 Graph Theory

Graph theory studies a mathematical model consisting of vertices and edges. Many connections between information in the world can be represented using this abstract mathematical way of graphs. There are two critical components for a mathematical model like a graph: (1) Vertex and (2) Edge, where edges connect vertices and vertices, and vertices and edges form a graph. In the field of transportation, many aspects can be expressed using graphs; e.g., each vertex can be a city, each edge can be a road between cities to extend it further, each vertex can be a terminal, each edge can be a corresponding route; each vertex can be a port, each edge can be a corresponding shipping line, or more microscopically, each vertex can be a building in a city. Each edge can be the street between buildings. However, many more abstract data relationships can be represented in a graph.

This thesis uses graph theory as the basic model to express road networks at different levels of abstraction. We have introduced a spatial model of a road network based on graph theory to represent the topological relationship of the road, which contains the connection of road lanes and intersections, the internal connections of an intersection, and traffic management protocols.

Here, we propose the *road graph* in different abstraction levels, a specific way to model roads and traffic that includes fully autonomous vehicles, as the basis of our research. This road graph model describes various elements related to traffic, such as traffic flow, traffic control protocols, vehicle information, and vehicle management processes. It should be mentioned that at this point, our work in this chapter is only theoretical research. Its purpose is to develop an abstract model that allows self-driving vehicles to understand roads and traffic, paving the way for more advanced vehicle negotiation or intelligent traffic management.

2.2 Macro-level Road Network Modeling

In this section, we introduce a general model to describe road networks. Graph theory can provide a formal representation of positions and roads and has found a significant application in the analysis of road networks where there is an intuitive and obvious relationship between links and nodes in a road network [Karimi, 2012, Porta et al., 2006].

Formally, a directed graph is defined as follows:

Definition 2.1. A directed graph G is a tuple (N, E), where

- N is a finite set of nodes.
- $E \subseteq N \times N$ is a set of directed edges.

To specify complicated road networks, we divide a road network into several positions and roads. We use a directed graph to represent a road network. Formally, A **macro-level road network** is a directed graph $G = (N, E)$, where N is a finite and non-empty set of positions and $E \subseteq N \times N$ can be interpreted as a set of roads. A road $(n, n') \in E$ refers to a connection and travel direction from position n to position n'. The position and road of the road network are generalized concepts that depend on the level of abstraction. At the national level, it is possible to express land-road connections between different states. In terms of a state, it is possible to express roads between different cities. And in terms of a city, it can be a road connection between administrative areas. If two positions do not have a road link between them, then no vehicle can travel directly between these two positions. Parallel roads between two positions are allowed in the road network.

Example 2.1. *Fig. 2.1 is an example of a country-size road network that uses an Australian map, which includes eight states: Western Australia (WA), Northern Territory (NT), South Australia (SA), Queensland (QLD), New South Wales (NSW), Victoria*

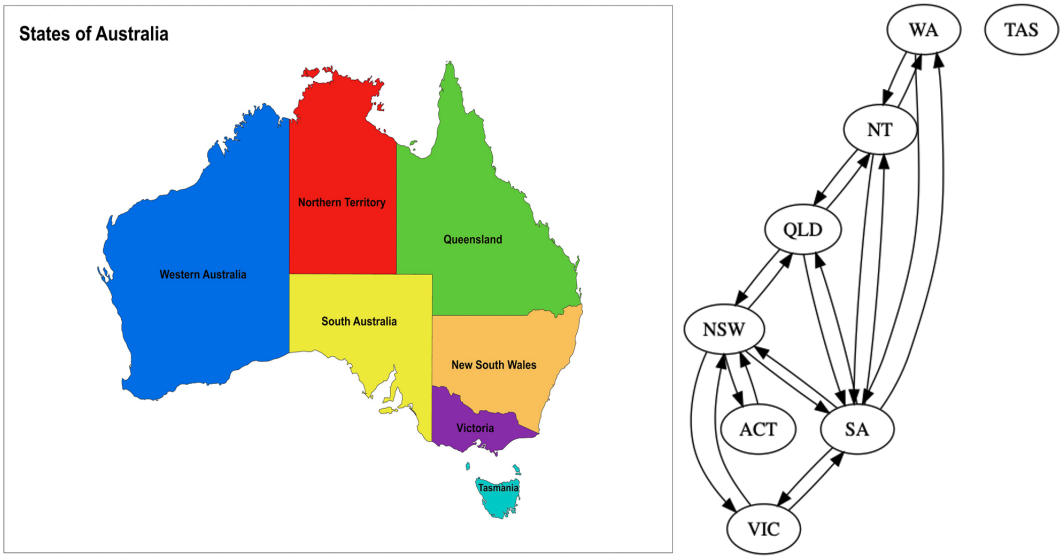


FIGURE 2.1: Example of Macro road network

(VIC), and Tasmania (TAS). The map of Australia is shown on the left side of Fig. 2.1, and the directed graph on the right side represents the road connections between different states in Australia. We formalized each state as a node of the graph, and directed edges represent the roads between the two states. Intuitively, WA is directly bordered by NT and SA so that vehicles can travel between these two states along land-based road connections. However, WA and QLD do not directly border each other, so vehicles must travel via NT to travel between them. TAS is a separate island, so no roads connect it to the mainland. It is important to note that the ACT is the Australian Capital Territory and is included in the NSW region, so ACT vehicles must pass NSW before they can go to VIC, SA or QLD.

The macro-level road network model is the most general model, and the graphical model of the road network can be easily changed depending on the usage requirements. For example, if an edge's length needs to be modelled, a directed graph with labels can be used.

2.3 Meso-level Road Network Modeling

The road network model discussed in the previous section is very macroscopic. In terms of large regions, it can describe the links between different states in a country, while in terms of cities, it can describe the links between different suburbs. However, this model has significant shortcomings when the road network coverage is more fine-grained to describe individual intersections. We summarize these shortcomings as follows.

- Macro-level road networks cannot describe the conflict of vehicles at intersections, as we will see in section 2.3.3.
- Macro-level road network models cannot describe traffic management protocols, as seen in section 3.2.
- When considering traffic management optimization, the macroscopic road network model cannot be used to account for changes in the calculation of waiting time at intersections caused by the change in the traffic management protocol, as we will see in chapter 6.

Since the macro-model has the shortcomings mentioned above, we propose a meso-level road network model in this section to compensate for the shortcomings as mentioned above of the macro-level model. In this section, we present a formal model to describe road networks for autonomous vehicles capable of reasoning about the connection between intersections and roads. At first, we present a graph representation model for abstract road networks that contain intersections and roads. Then, we introduce internal relations for each intersection that specify internal connections and conflicts.

2.3.1 Road networks

As with the macro-level road network model, we still use graph theory as the core theory of the model. Unlike the macro-level road network model, the nodes represent intersections, while the edges represent roads between intersections, called the meso-level road

network. Since roads in the same direction may contain different lanes, we use a directed graph with labels to ensure the uniqueness of the lanes. A labelled directed graph is a generic type of graph and is defined as follows:

Definition 2.2. A labeled directed graph is a tuple (N, L), where

- N is a set of nodes.
- $L \subseteq N \times N \times \mathbb{N}$ is a set of labeled edges.

With the help of Def. 2.2, we can interpret a **meso-level road network** as a tuple $G = (N, L)$, where

- Each node $n \in N$ represents an intersection.
- Each labelled edge $(n, n', x) \in L$ represents a lane of a road between n and n', where x is an identifier to distinguish between the various lanes of that road.
- Each set of labelled edges with the same n and n' represents a road between the intersections n and n'.

It is worth noting that the intersections we are referring to are not intersections in the real sense. Instead, a common area created by the intersection of at least two roads can be called an intersection. For example, a parking lot exit intersecting a city road could be called an intersection, or a private residence intersecting a road could be called an intersection. Intuitively, an intersection links roads, and a road can be divided into several lanes. For example, an arc $(n, n') \in A$ represents a road on which vehicles can travel from n to n'. $(n, n', 1)$ and $(n, n', 2)$ are two lanes of the road[1]. More specifically, the labels guarantee the uniqueness of each lane. Furthermore, for each intersection $n \in N$, let $L_n^{in} = \{(n', n, i) \in L : n' \in N \ \& \ i \in \mathbb{N}\}$ be the set of all incoming lanes, and let $L_n^{out} = \{(n, n', i) \in L : n' \in N \ \& \ i \in \mathbb{N}\}$ be the set of all outgoing lanes.

[1]Instead of denoting a lane as $((n, n'), 1)$, we simply write $(n, n', 1)$.

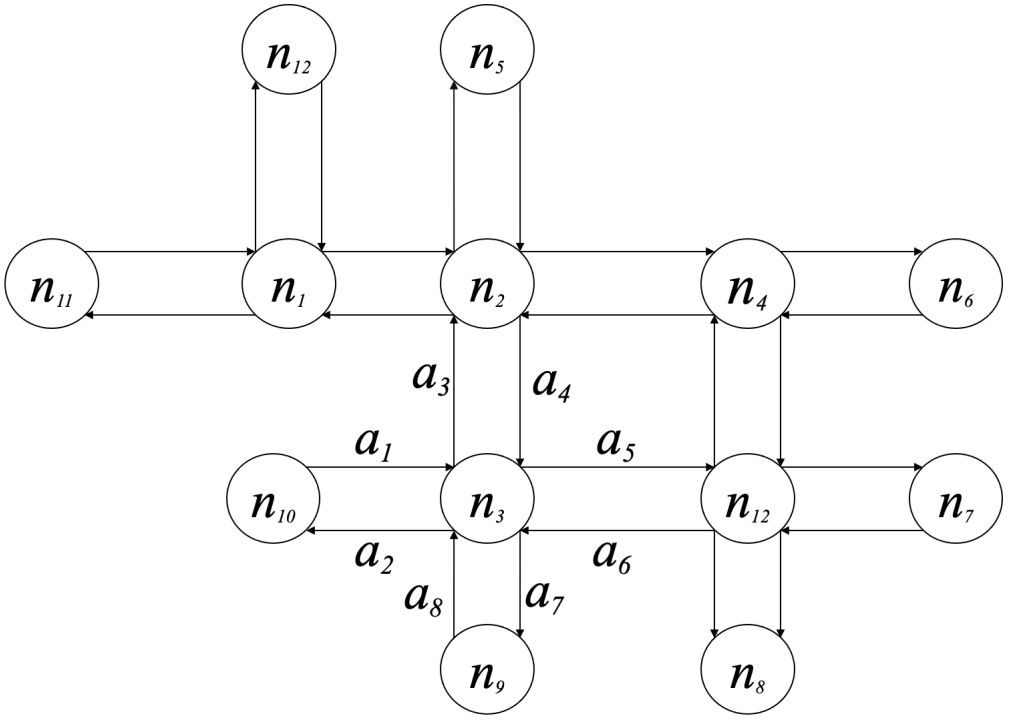


FIGURE 2.2: Example of Meso Road Network

Example 2.2. *The graph in Fig. 2.2 is an example of a road network. There are twelve intersections, and the arcs indicate the travel directions between the intersections. Regarding the graph's structure, we can claim that intersections* n_1 *and* n_4 *are T-junctions, and intersections* n_2, n_3, n_{12} *are traditional four-way intersections in the real world. Fig. 2.3 illustrates the labelled lanes for the intersection* n_3*. We use natural numbers as the labels for lanes. For example, road* a_1 *only has one lane* $l_1 = (n_{10}, n_3, 1)$*, however, road* a_3 *has two lanes* $l_2 = (n_3, n_2, 1)$ *and* $l_3 = (n_3, n_2, 2)$*, respectively.*

It is obvious that the road network model is more detailed than the macroscopic model and can not only describe the relationship between intersections but also be specific to the number of lanes in different directions of the intersection. For example, the intersection in Fig. 2.3 is asymmetric, with the larger north-south road having two adjacent lanes on the road and the narrower east-west road having only one lane on the road.

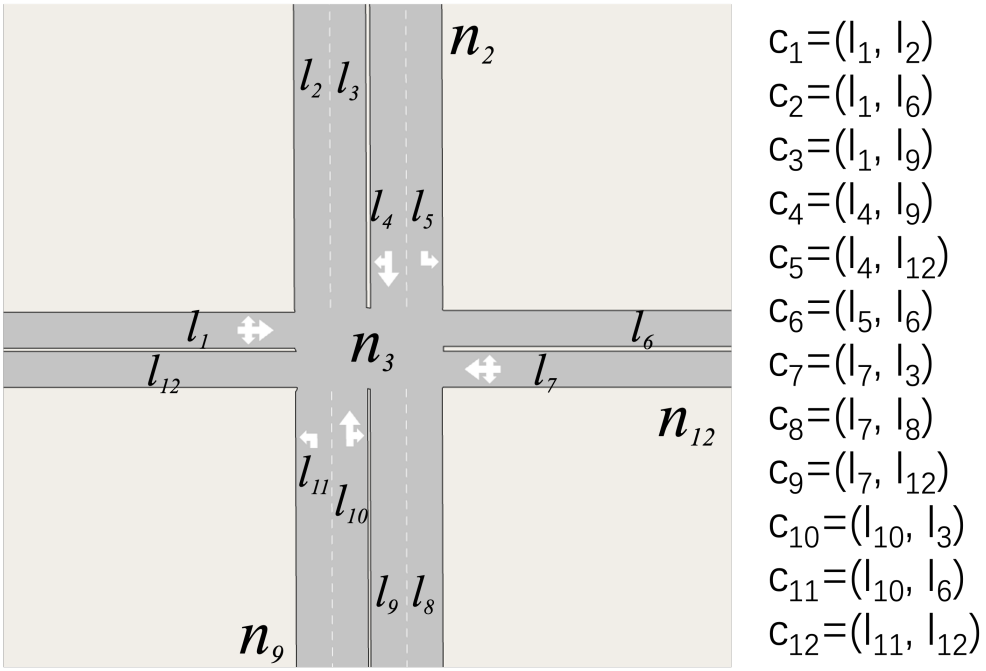


FIGURE 2.3: Example of Labeled Lanes and Connections

2.3.2 Connection relations

The road network model in Def. 2.2 does not represent the connection and conflict relationships between the lanes of the road at an intersection. For example, a vehicle approaching an intersection may wish to proceed straight, turn left, turn right, or make a U-turn, but the vehicle may not be allowed to travel in each of these directions. Therefore, we need to specify the connections between the road lanes at each intersection. Formally, we define the connection relation as follows:

Definition 2.3. Given a meso-level road network $G = (N, L)$, a **connection relation** $C_n \subseteq L_n^{in} \times L_n^{out}$ of the intersection is a binary relation between the sets of incoming and outgoing lanes for each intersection $n \in N$.

The connection relation of an intersection specifies which outgoing lane can be reached from which incoming lane. Each $(l, l') \in C_n$ is called a **connection** between the incoming lane l and the outgoing lane l'. It is worth mentioning that each incoming or outgoing lane of an intersection should be contained in at least one connection.

Example 2.3. *We use Fig. 2.2 and 2.3 as an example to explain the connection relations. The connection relation* $C_{n_3} = \{c_1, \cdots, c_{12}\}$ *in Fig. 2.3 is predefined according to the traffic signals, shown as white arrows, in each direction, but, of course, when the arrows are different, the connection relation must be changed accordingly. Intuitively, the connection* $c_1 = (l_1, l_2) \in C_{n_3}$ *means that the vehicles in* l_1 *are able to turn left at node* n_3 *towards* l_2*. However, we assume that the vehicle in* l_1 *cannot turn left to* l_3*, therefore, we have* $(l_1, l_3) \notin C_{n_3}$*. We also suppose that a U-turn is not allowed at intersection* n_3*, so* (l_1, l_{12}) *is not a connection in this case. Similarly, the connection* $c_{12} = (l_{11}, l_{12}) \in C_{n_3}$ *shows that vehicles in* l_{11} *can turn left to* l_{12}*. However, vehicles in* l_{11} *cannot go straight to* l_2 *by the predefined traffic sign, so that* $(l_{11}, l_2) \notin C_{n_3}$*.*

It is worth mentioning that the above example only explains the connection relation defined in the diagram. Still, the same intersection can have several different connections depending on traffic scenarios. For example, if we change the predefined traffic sign in l_{11} that allowed vehicles able to go straight toward l_2, then $(l_{11}, l_2) \in C_{n_3}$.

2.3.3 Conflict relations

Furthermore, since these road connections at the same intersection can cross, vehicles crossing the intersection can collide if there is no traffic management protocol to avoid the collision. For example, if a vehicle travels from south to north while another travels from east to west, it may collide at that intersection. To specify such potential collisions between connections, we introduce a 'conflict relation' on top of the connection relation for each intersection. Formally, we define the conflict relation as follows:

Definition 2.4. Given a meso-level road network $G = (N, L)$, a **conflict relation** $Z_n \subseteq C_n \times C_n$ is a symmetric binary relation over the set of connections C_n for any intersection $n \in N$. Here, symmetric means that $(c, c') \in Z_n$ if and only if $(c', c) \in Z_n$ for all $c, c' \in C_n$;

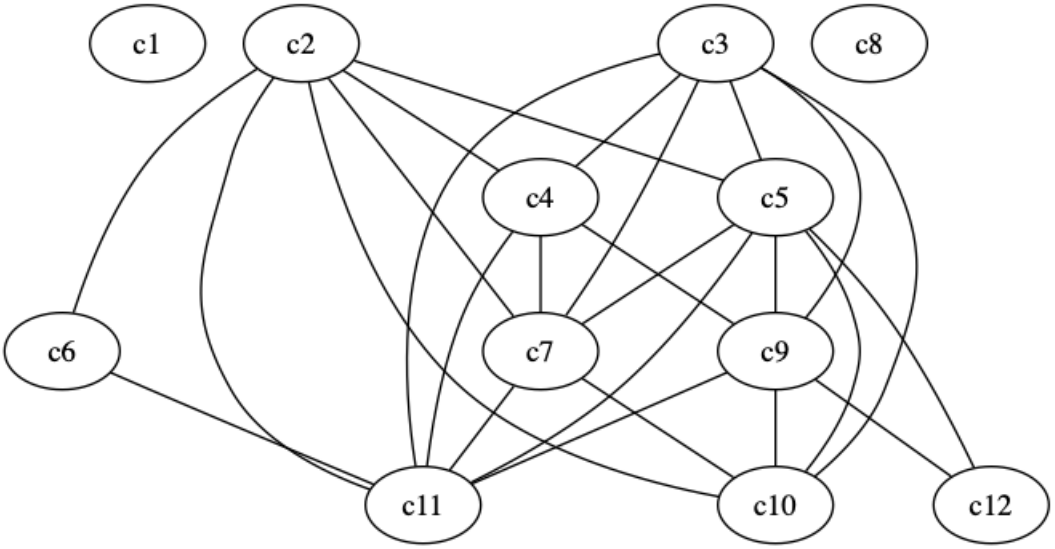


FIGURE 2.4: Example of Conflict Relation

For an intersection n, the conflict relation can be understood as an undirected graph whose nodes are connections in the connection relation C_n. The undirected edges of this graph represent the potential collision of vehicles travelling along two connections at the intersection. Note that we assume that vehicles departing from the same lane will not collide with each other.

Example 2.4. *Fig. 2.4 shows an example of a conflict relation Z_{n_3} of intersection n_3 in Fig.2.2 using the connection relation C_{n_3} shown in Fig. 2.3. The connection c_2 is related to the connection c_4 in the conflict relation Z_{n_3} means that vehicles on l_1 that go straight might collide with vehicles on l_4 that go straight. Similarly, all other connections related to the connection c_2 can be explained in Fig. 2.3. Furthermore, $(c_1, c_8) \notin Z_{n_3}$ can be interrupted since vehicles on l_1 turn left to l_2 have no chance of colliding with vehicles on l_7 turn left to l_8.*

Note that an intersection's collision relation only declares potential collisions between vehicles travelling in different connections, not actual collisions. The meso-level road network and intersection relations allow us to represent complex road networks in the real world.

2.4 Micro-level Road Network Modeling

The above two road network models describe the composition of the road network from a macroscopic point of view. Still, they cannot precisely describe changes in the driving status of vehicles on the road network as a function of time. Therefore, this section proposes a discrete-time-based microscopic road network model to describe the change in vehicle position with the time change. We introduce a formal method to represent roads, traffic flows, and traffic control protocols to allow autonomous vehicles to reason about complicated traffic situations. As a generic assumption of this work and a way of abstraction, time occurs at distinct, separate "points in time" throughout each non-zero region of time ("time period"), represented by natural numbers $T = \{1, 2, 3, \cdots\}$.

2.4.1 Micro-level road networks

To specify any complicated road, we divide a road into several blocks or road segments. Each block of a road allows one car to travel at each time[2], represented as a vertex in a graph. The directed edges in a graph represent connections and travel directions between blocks of road. If two vertices have no edge to link them, no vehicle can travel directly between these two blocks.

Additionally, we assume that each road contains several entrances and several exits. For each entry, the vertex representing it must have at least one outgoing edge, and for each exit, the corresponding vertex must have at least one incoming edge. Formally, we have the following definition.

Definition 2.5. A **micro-level road network** G is a tuple $(B, \mathcal{E}, B_n, B_x)$, where:

- B is a finite and non-empty set of blocks;
- $\mathcal{E} \subseteq B \times B$ is a set of arcs. An arc $(b, b') \in \mathcal{E}$ refers to a connection and travel direction from block b to block b';

[2]We will use the discrete-time to represent traffic flows; thus, a time point represents a period.

- $B_n \subseteq B$ is a set of blocks to represent the road entries;
- $B_x \subseteq B$ is a set of blocks to represent the road exits.

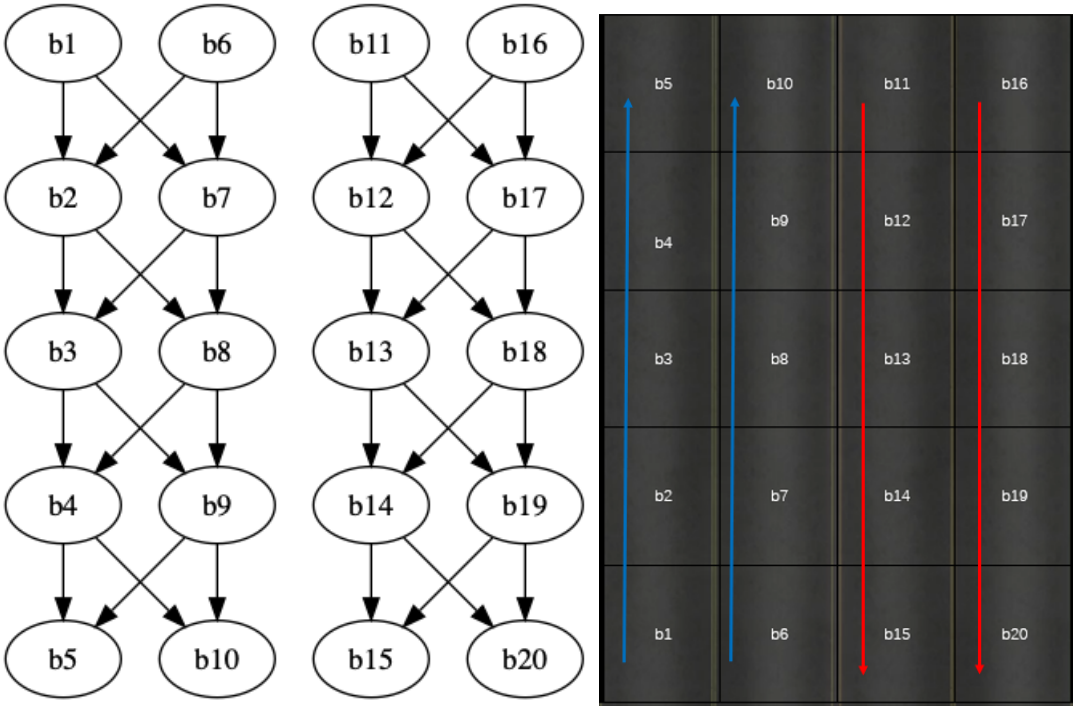


FIGURE 2.5: Example of a two-way road network

Example 2.5. *Fig. 2.5 shows a simple road network that represents a typical two-way road. The vertices* $b_1, \cdots, b_{20}$ *represent the segments of the road, while the arcs indicate the traffic flows that are allowed from segment to segment. For example,* b_1, b_2, b_3, b_4, b_5 *are the blocks of the left-most lane. The arc from* b_2 *to* b_8 *means that a vehicle in block* b_2 *can change lane to block* b_8*, while the absence of an arc from* b_2 *to* b_7 *means that it is not possible to change to block* b_7*. The blocks* b_1*,* b_6*,* b_{11}*, and* b_{16} *are entrances to the road, and the blocks* b_5*,* b_{10}*,* b_{15} *and* b_{20} *are exits from the road.*

As an abstraction of roads, a road graph can represent more complicated road situations and configurations, such as multi-way junctions, roundabouts, no-through roads, and U-turns. The following example shows a representation of a typical four-way intersection.

Example 2.6. *Fig. 2.6 shows an example of a four-way intersection road network that will be used as a running example of the paper. Each direction has one lane. The*

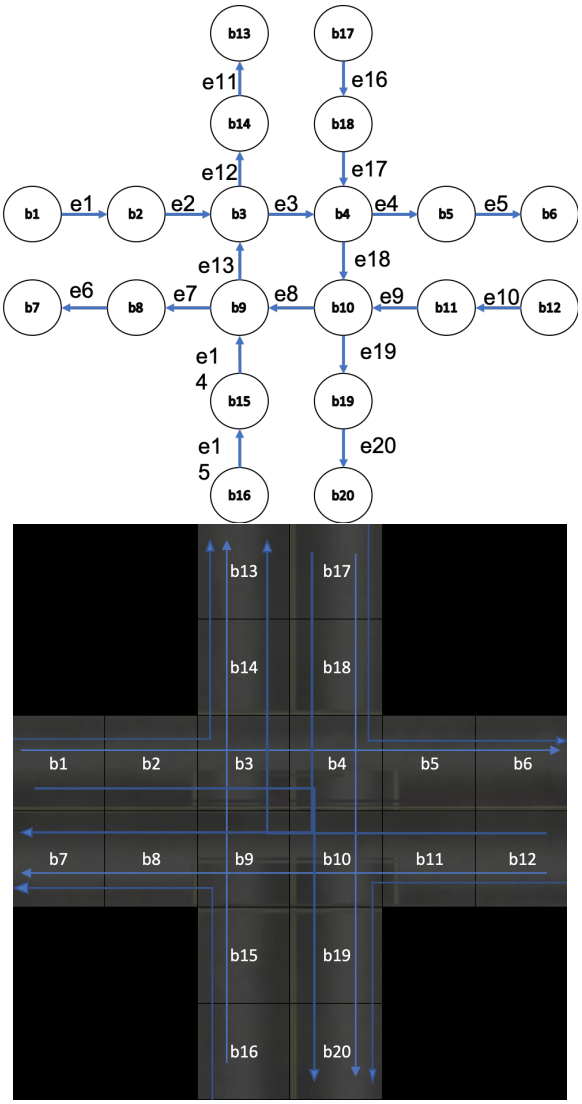


FIGURE 2.6: Example of a four-way intersection

road graph indicates that a vehicle turning right cannot travel diagonally inside the intersection. For example, a vehicle turning b_2 *to* b_{19} *must travel through* b_3*,* b_4 *and* b_{10} *rather than a sharp turn from* b_3 *to* b_{10}*. This is by no means a restriction of road graph representation but reflects an actual road setting.*

Given a road graph, a route along multiple blocks of a road can be easily defined in terms of standard graph theory terminology. Formally, we have the following definition:

Definition 2.6. Given a micro-level road network $(B, \mathcal{E}, B_n, B_x)$, a **path** ρ is a sequence $b_0 \xrightarrow{e_1} b_1 \xrightarrow{e_2} b_2 \cdots \xrightarrow{e_m} b_m$, where

- $b_i \in B$ for all $0 \leq i \leq m$

- $e_i = (b_{i-1}, b_i) \in \mathcal{E}$ for all $0 < i \leq m$
- $b_i \neq b_j$ for any $i \neq j$

ρ is called a *complete path* if $b_0 \in B_n$ and $b_m \in B_x$. We use $\hat{\rho}$ to denote the start block of the path and $\check{\rho}$ the end block of the path, that is, $\hat{\rho} = b_0$ and $\check{\rho} = b_m$.

These conditions express that a complete path is a simple path that links an entry to an exit on the road network. In Fig. 2.6, an example of a complete path is $b_1 \overset{e_1}{\rightarrow} b_2 \overset{e_2}{\rightarrow} b_3 \overset{e_{12}}{\rightarrow} b_{14} \overset{e_{11}}{\rightarrow} b_{13}$.

2.4.2 Vehicles and traffic settings

Vehicles are road users. We assume that all vehicles are fully autonomous, which means that the decision-making of each vehicle is not centralized but is done by the vehicle itself, whether they are driven by humans or computers. We also assume that each vehicle has a designated path, specifying its entry block, exit block, and intended travel path. In the context of automated negotiation between autonomous vehicles, these pieces of information are the initial settings of a vehicle before it enters a road. They are negotiable when they travel on the road. Furthermore, we assume that each vehicle has a designated time point to enter the road. Formally, we specify vehicle information with the following concept:

Definition 2.7. Given a micro-level road network $G = (B, \mathcal{E}, B_n, B_x)$ and a set $\mathcal{V}$ of possible vehicles. The **vehicle information** $\mathcal{I}$ is represented by a tuple $(\mu, \sigma, \eta, \mathcal{P})$ where

- $\mu : \mathcal{V} \rightarrow B_n$ is a function that maps each vehicle to a road entry;
- $\sigma : \mathcal{V} \rightarrow B_x$ is a function that maps each vehicle to a road exit;
- $\eta : \mathcal{V} \rightarrow T$ is a function that maps each vehicle to a time point that indicates the time it is expected to enter the road;

- $\mathcal{P} : \mathcal{V} \to P$ is a function that maps each vehicle to a complete path such that for each vehicle $v \in \mathcal{V}$, $\mathcal{P}\hat{(}v) = \mu(v)$ and $\mathcal{P}\check{(}v) = \sigma(v)$.

In the rest of this section, we call $(G, \mathcal{V}, \mathcal{I})$ a **traffic setting**.

2.4.3 Traffic states and traffic flows

Traffic means that vehicles move on a road. A snapshot of traffic on the road can be viewed as a set of vehicles currently on the road and the positions they occupy. As we mentioned earlier, we assume that each block can contain only one vehicle at each time point. Therefore, the vehicle's location can be represented by an injective function from the set of vehicles to the set of blocks of the road. Formally, we introduce the following concept:

Definition 2.8. Given a traffic setting $(G, \mathcal{V}, \mathcal{I})$, a **traffic state** with respect to this traffic setting is a pair (V, τ), where

- $V \subseteq \mathcal{V}$, indicating the vehicles that are currently on the road;
- $\tau : V \to B$ is an injective function that maps each vehicle to a block of the road. In other words, for any $v, v' \in V$, $\tau(v) = \tau(v')$ implies $v = v'$.

A traffic state represents a snapshot of a traffic flow and thus is a static view of traffic. However, traffic is dynamic. To model a flow of traffic, we define traffic on the road as a set of traffic states in a time sequence:

Definition 2.9. Given a traffic setting $(G, \mathcal{V}, \mathcal{I})$ and a set of time points T, a **traffic flow** $\mathcal{F} = \langle (V_t, \tau_t) \rangle_{t \in T}$ is a temporal sequence of traffic states such that for each time point $t \in T$,

1. $v \in V_{t+1} \setminus V_t$ implies $\tau_{t+1}(v) = \mu(v)$.
2. For each $v \in V_t$, exactly one of the following conditions is met.

(a) $\tau_t(v) = \tau_{t+1}(v)$

(b) $(\tau_t(v), \tau_{t+1}(v)) \in \mathcal{E}$

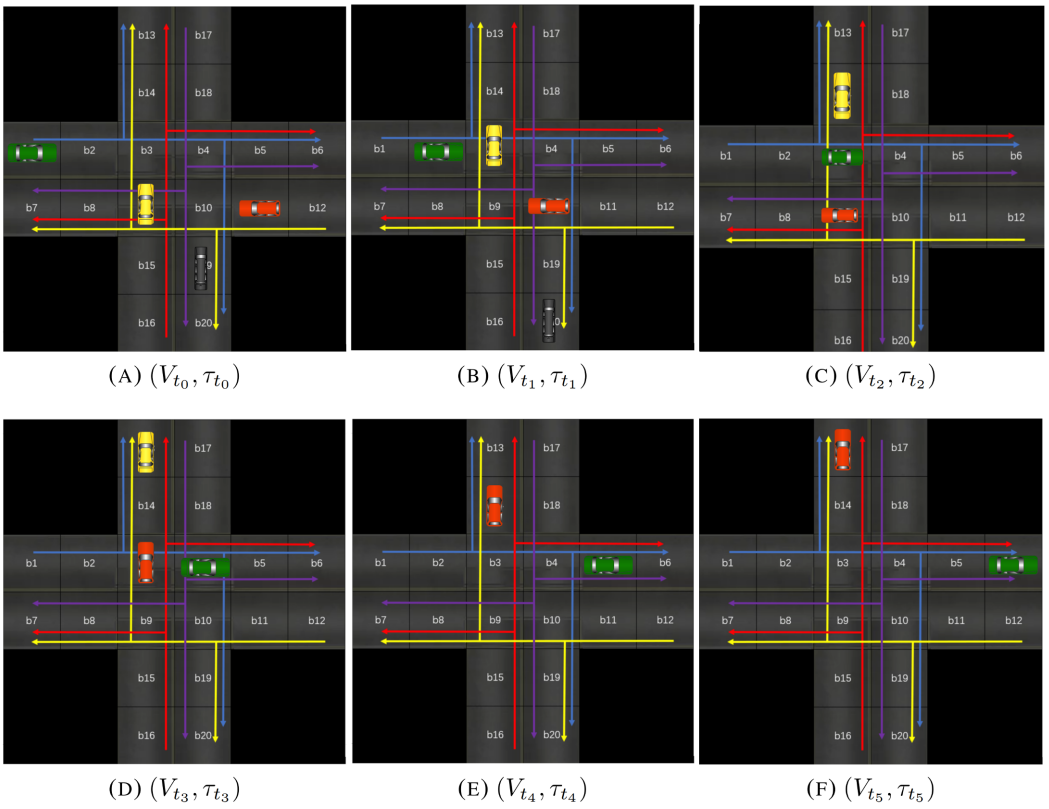
(c) $\tau_t(v) = \sigma(v)$ and $v \notin V_{t+1}$

3. For any $v, v' \in V_t$ such that $v \neq v'$ and $\tau_t(v) = \tau_{t+1}(v')$, $\tau_{t+1}(v) \neq \tau_t(v')$.

Condition 1 says that any new vehicle entering a traffic state must enter the road in its designated entry block. Condition 2 expresses that the vehicle can only have three states in the following time period: stay in the current position, travel to the next block of the path, or leave the road network. And Condition 3 indicates that two vehicles in adjacent traffic states cannot exchange positions with each other (because they would collide).

A traffic flow encodes the complete travel information of each vehicle that travels on the road in general for the entire period of time. For example, if we want to know the trajectory of a vehicle, we can record its position at each time point after it enters the road. Formally, we can represent the trajectory of a vehicle v as a sequence of blocks: $\langle \tau_t(v) \rangle_{t=t_n}^{t_x}$, where t_n and t_x are the entry and exit time of the vehicle, respectively. Note that the trajectory of a vehicle does not have to be precisely the intended path $\mathcal{P}(v)$ specified in the vehicle information $\mathcal{I}$. In negotiations between vehicles, a trajectory is an outcome, and the intended path is just an initial proposal for the negotiation. In certain situations, we need to specify a traffic flow over a time interval, say between time points t_1 and t_2. Then a traffic flow is represented as a temporal sequence of traffic states $\langle (V_t, \tau_t) \rangle_{t=t_1}^{t_2}$.

Example 2.7. *We use Fig. 2.7 as an example to explain the traffic state and traffic flow. The six sub-graphs* $(A) - (F)$ *in the figure are the traffic states of each of the six consecutive periods* $(t_0 - t_5)$ *in order. There are four vehicles, Red* (v_r)*, Green* (v_g)*, Yellow* (v_y) *and Black* (v_b)*, driving on the road. The exit of vehicles are* $\sigma_{v_r} = b_{13}$*,* $\sigma_{v_g} = b_6$*,* $\sigma_{v_y} = b_{13}$ *and* $\sigma_{v_b} = b_{20}$*. The initial state at* t_0 *is* (V_{t_0}, τ_{t_0})*, with* $V_{t_0} = \{v_r, v_g, v_y, v_b\}$*,*$\tau_{t_0}(v_r) = b_{11}, \tau_{t_0}(v_g) = b_1, \tau_{t_0}(v_y) = b_9$ *and* $\tau_{t_0}(v_b) = b_{19}$*. Each period all vehicles will move one grid towards their destination. The black vehicle* v_b *leaves*



(A) (V_{t_0}, τ_{t_0}) (B) (V_{t_1}, τ_{t_1}) (C) (V_{t_2}, τ_{t_2})

(D) (V_{t_3}, τ_{t_3}) (E) (V_{t_4}, τ_{t_4}) (F) (V_{t_5}, τ_{t_5})

FIGURE 2.7: Example of Traffic State and Traffic Flow

at t_2, so that $V_{t_2} = V_{t_1} \setminus \{v_b\} = \{v_r, v_g, v_y\}$. Similarly, the yellow vehicle leaves at t_4, then at $V_{t_4} = V_{t_3} \setminus \{v_y\} = \{v_r, v_g\}$. The period t_6 (one time period after t_5) is the last traffic state because all vehicles will be out during that time period, and then $V_{t_6} = \emptyset$.

2.5 Discussion

Road network models serve as the basis for autonomous vehicles to understand road conditions and can be used in different research directions. In this chapter, we have presented three road network models based on the abstraction level. It formally described the road networks from the macro-level to the micro-level. With running examples, we have demonstrated that the models allow us to abstract any road network depending on the needs of the different scenarios. The road network models in this chapter are the most fundamental part of this thesis; these models will be recalled and expanded in their corresponding chapters. In the next chapter, we will propose a formal representation of

traffic management protocols based on the meso- and micro- level road network models. In chapter 4, we will study the traffic assignment problem with exponential cost functions, described in terms of macro- and meso-level road networks. In chapters 6 and 5, the meso-level road network will investigate a hybrid model of a traffic assignment problem and traffic control.

In addition to the transportation field discussed in this chapter, our road network models can also be applied in the following applications. First, self-driving cars are a kind of mobile robot, and multi-robot coordination is an important road network model application direction. Coordination includes but is not limited to robot scheduling and path planning [Jones et al., 2011, Ulusoy et al., 2013]. There is much research on this, but the application scenarios are different. Pinillos et al. [2016] proposed the application of service robots in a hotel scenario based on a road network model to describe the hotel environment to perform different tasks. Second, many road network models are used in shared roads, such as [Agatz et al., 2012, Ta et al., 2017, Wang et al., 2020]. In addition, there is a large amount of research applied to logistics research [Baker and Ayechew, 2003, Bellman, 1958, Golden et al., 2008, Laporte, 1992, Toth and Vigo, 2002].

In general, road network models based on graph theory are widely used in different research directions of multi-robot systems and transportation, with the primary purpose of allowing vehicles or robots to understand the road structure to accomplish a particular task.

Chapter 3

Traffic Management Protocols for Autonomous Vehicles

Traffic must be controlled to ensure safety and efficiency on the road. Traffic control consists in instructing vehicles to take the appropriate measures or actions to avoid collisions or delays. Complex operational procedures, rules and laws, and physical equipment (such as signs, markings, and lights) have been used in real-world traffic control systems. The most common traffic control devices and methods are traffic lights, stop signs, roundabouts, and other facilities.

This chapter will use the above model to propose various traffic management protocols models and simulators for testing and data collection in simulation scenarios for data analysis. The structure of this chapter is as follows. Section 3.1 introduces some background about intelligent traffic management protocols and existing simulators for autonomous vehicles. In Section 3.2, multiple priority-based traffic management protocols are proposed for autonomous vehicles in the meso-level road network model. Additionally, we have implemented a new simulator called AIM4+, to simulate traffic and test the proposed protocols, based on the AIM4 [Au and Stone, 2010, Dresner and Stone, 2004a]. In Section 3.3, we investigate time-based and priority-based traffic management protocols based on the formulation of the micro-level road network model. In

Section 3.4, we use a robot operating system simulator to view the robots as self-driving cars and perform data collection and analysis of the relevant algorithms.

3.1 Background

Self-driving cars are becoming tangible and will change our lives more than we can imagine [Banerjee et al., 2017, Bimbraw, 2015, Pettersson and Karlsson, 2015, Thorpe et al., 1991]. In the future, when all cars are self-driving, traffic situations will be dramatically different and require different methods and infrastructure for control and management Chan [2017]. With the emergence of AVs and connected vehicles (CVs), traffic facilities designed for human driving, such as traffic lights, stop signs, and roundabouts would be replaced by less visible but more efficient algorithmic policies [Gruel and Stanford, 2016]. According to the American National Highway Traffic Safety Administration (NHTSA), AVs are expected to be greener, more efficient and safer [Choi, 2010]. With approximately 2.5 million accidents related to intersections, of which fifty per cent of these incidents cause serious injuries, and twenty per cent are fatal, intersections are a safety concern and the main cause of traffic congestion. As such, using new technologies with connected vehicles and intelligent traffic control for driverless cars has become one of the most important research topics for autonomous driving [Bashiri and Fleming, 2017, Belkhouche, 2017, Carlino et al., 2013, Fok et al., 2012, Hausknecht et al., 2011, Lin et al., 2017]. This chapter aims to investigate the behaviour of AVs under different intersection control protocols.

It is believed that autonomous vehicles will significantly improve driving safety by reducing road accidents, human error injuries, and traffic jams [Wei et al., 2017]. In the future, when all cars are autonomous, the situation of road traffic can be dramatically different from what we have now and, therefore, will require different methods and infrastructures for road management and traffic control [Chan, 2017]. With new technologies for vehicle-based communication and intelligent traffic control, traditional

vision-based traffic control facilities, such as traffic lights, roundabouts, and stop signs, are likely to be replaced by less visible but more efficient and more effective algorithmic controlled road facilities [Gruel and Stanford, 2016].

Intersections are the weakness of the intelligent traffic system and a major cause of traffic accidents. Traffic at intersections is the main delay in driving time and the main reason why accidents occur in vehicles; most intersections are not intelligent, so they still have a lot of room for improvement. Traffic light control or stop sign has been used at the intersection for more than 150 years. The newest traffic light intersection control, called dynamic traffic light control, can change the signal time in real-time by performing a big data analysis of the car between each traffic light. The most common intersection traffic management strategies are Cybercars-2 [de La Fortelle, 2010], Intersafe-2 [Roessler, 2010], Autonomous Intersection Management [Fok et al., 2012, Hausknecht et al., 2011, Wuthishuwong and Traechtler, 2013], Intelligent and Cooperative Intersection Collision Avoidance System [Basma et al., 2011, Rawashdeh and Mahmud, 2008]. There are three typical control mechanisms used almost anywhere in the world: *traffic signals*, *stop signs*, and *roundabouts*. Traffic signals are considered the most efficient mechanism for heavy traffic intersections, stop signs for light and unbalanced traffic intersections, and roundabouts built to accommodate moderate traffic with balanced flow from all directions. These traffic control facilities were designed for human drivers. Despite the development of new technologies such as smart intersections to optimize traffic control [Geng and Cassandras, 2015, Younis and Moayeri, 2017], in environments where all vehicles are fully connected and autonomous, these facilities are no longer necessary and efficient.

The most common traffic control protocols on major roads are based on traffic lights. The earliest versions of these traffic signal systems assigned a fixed amount of time for each traffic light to turn green, regardless of the number of vehicles or traffic density in the corresponding lane. However, as technology advanced, these traffic signal systems started to consider different parameters, such as a distinction between day and night or between peak and off-peak periods, to determine the ratio between time in green and

time in red. Some vision-based traffic management systems [Esteve et al., 2007, Javaid et al., 2018, Reza et al., 2021] use vision sensors to capture the flow of cars coming from different directions. To our knowledge, these advanced management approaches target individual intersections and do not synergise with other intersections. Some roads may be busier than others at different times, which requires additional time to clear congestion on the road. An in-depth study of autonomous vehicle traffic management is in high demand. Almost all current road infrastructures and traffic control technologies depend on human driving [Wagner, 2016]. Even self-driving cars are being trained to recognize human-oriented traffic signs and mimic human driving behaviours, which is no longer necessary for an efficient or reliable traffic management system.

Furthermore, since fully autonomous vehicles are not yet available on a large scale in the real world, it is not likely that traffic management for autonomous driving will be studied through real-world tests. Therefore, the management protocols we mention in this chapter are simulated using virtual simulators to obtain data and analyze them against existing traffic management protocols. Many simulators are designed to model human behaviour rather than testing custom agent algorithms. One of the most widely used simulation platforms in robotics and related research areas is Gazebo [Koenig and Howard, 2004]. Due to their modular nature, different sensor types and physics engines can be added to the simulator. However, Gazebo makes it challenging to develop huge, complicated settings. AirSim [Shah et al., 2017] and CARLA [Dosovitskiy et al., 2017] are some more notable open-source simulators for autonomous driving. However, when this research began, none of them gave us the ability to easily replace the mechanism by which intersections are governed. One of the essential related works is the Autonomous Intersection Management (AIM) designed by Dresner and Stone [2008b], which could be an efficient way to handle road junctions for autonomous vehicles. However, we cannot use this simulator in our research since the vehicles and traffic management facilities in the simulator cannot make independent judgments and instead rely on pre-programmed tactics. As a result, more sophisticated simulators are required to mimic in a multi-agent context.

3.2 Priority-based Traffic Management Protocols for Meso-level Road Networks

In this section, we introduce priority-based traffic management protocols based on the meso-level road network model $G = (N, L)$ (see Def 2.2) and its intersection relations $(C_n, Z_n)_{n \in N}$ (see Def 2.3 and Def 2.4). The meso-level road network shown in Fig. 3.1 is used as a running example in this section.

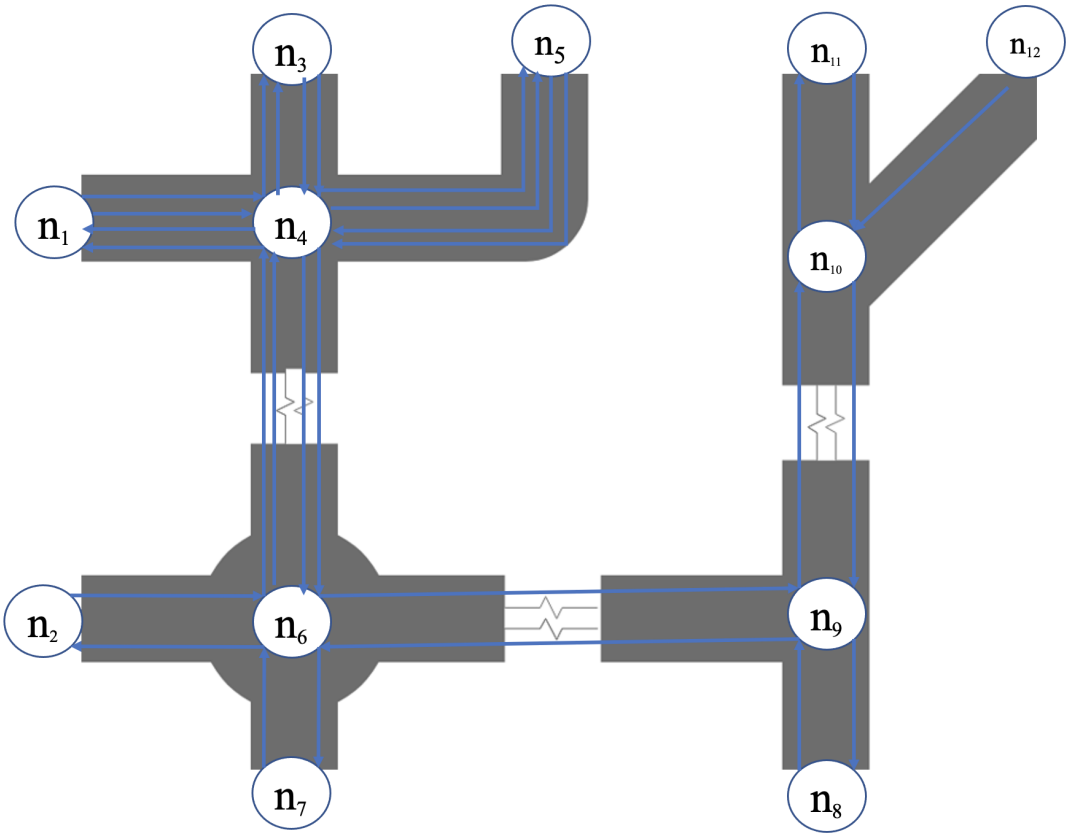


FIGURE 3.1: An example of a road network.

Example 3.1. *Fig. 3.1 shows an example of a road network. It includes an intersection* (n_4) *with multiple lanes, a roundabout* (n_6)*, a T-junction* (n_9)*, and a merging intersection* (n_{10})*. Fig. 3.2 shows the connection relation and conflict relation for the intersection* n_9 *in Fig. 3.1. The left graph shows the detailed road of intersection* n_9*. The middle graph represents the connection relation* $C_{n_9} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$*, which means that traffic from* n_6 *is allowed to turn left and turn right; traffic from* n_{10} *are*

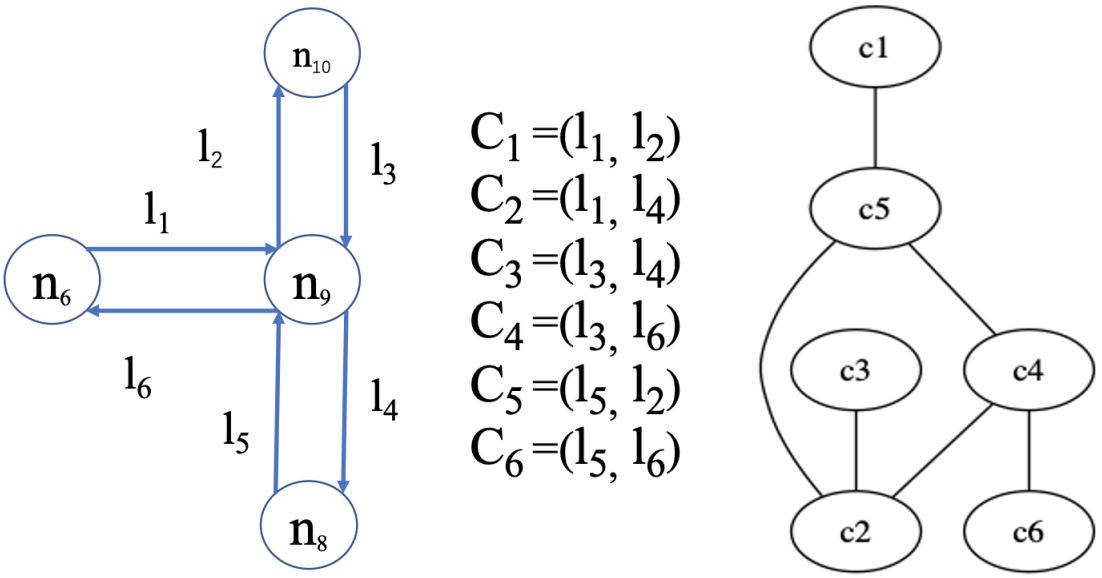


FIGURE 3.2: An example of connection relation and conflict relation

allowed to go straight and turn right; traffic from n_8 *are allowed to turn left and go straight. However, U-turns are not allowed at* n_9 *from any direction. If it is allowed for the traffic from, say* n_6*, you may add* (l_1, l_6) *to* C_{n_9}*. The right graph shows an example of conflict relation* Z_{n_9} *at the intersection* n_9*. In this relation,* c_1 *conflicts with* c_5*, which means that vehicles on lane* l_1 *turning left to lane* l_2 *have a potential collision with vehicles on lane* l_5 *go straight to lane* l_2*.*

Vehicles often collide with other vehicles because their routes intersect and thus obstruct the paths of others. The general rule determining who has the right of precedence is called *right of way*. It defines who has the right to use the conflicting portion of the route and who must wait for the other to do so. For example, in a country where cars drive on the left, vehicles will give way to traffic on the right whenever they approach an intersection, which means that the road on the right has higher priority than the road on which the vehicle is travelling. Despite the significant differences in traffic management methods and systems, almost all traffic management methods can be classified into two types: road-based priority protocols and vehicle-based priority protocols. Road-based priority management can be divided into static priority and dynamic priority. A

more detailed explanation of traffic management protocols is given in the following subsections.

3.2.1 Static priority management protocols

Static priority management protocols are based on predefined priorities of the connections at each intersection. We introduce a graph representation to describe the priority of connections for each intersection based on the formal representation of the road networks and the intersection relations in Chapter 2. Static priority management protocol means that for any two conflicting connections, there is always one that has priority over the other. In short, a vehicle travelling on a lower-priority connection must yield to a vehicle travelling on a higher-priority connection if it must cross the intersection simultaneously. For each intersection $n \in N$, we use a directed graph to specify the priority of connections in the connection relation C_n. If there is a directed edge from c to c', vehicles driving along connection c have priority over c' to pass the intersection n. Formally, we define a static priority management protocol ξ_n as follows:

Definition 3.1. Given a meso-level road network $G = (N, L)$ and intersection relations $(C_n, Z_n)_{n \in N}$. For each intersection $n \in N$, a static-priority management protocol $\xi_n = (C_n, \Psi_n)$ is a directed graph, where $\Psi_n \subseteq \mathcal{C}_n \times C_n$. It must satisfy the following conditions:

- Antisymmetric: If $(c, c') \in \Psi_n$, then $(c', c) \notin \Psi_n$ for any connection $c, c' \in C_n$, ;
- Transitive: If $(c, c'), (c', c'') \in \Psi_n$, then $(c, c'') \in \Psi_n$ for any connections $c, c', c'' \in C_n$;
- Complete relative to Z_n: If $(c, c') \in Z_n$, then $(c, c') \in \Psi_n$ or $(c', c) \in \Psi_n$ for any connections $c, c' \in C_n$,.

We say that a 'deadlock' occurs when two conflicting connections have the same priority or multiple conflicting connections form a closed priority loop. The condition of

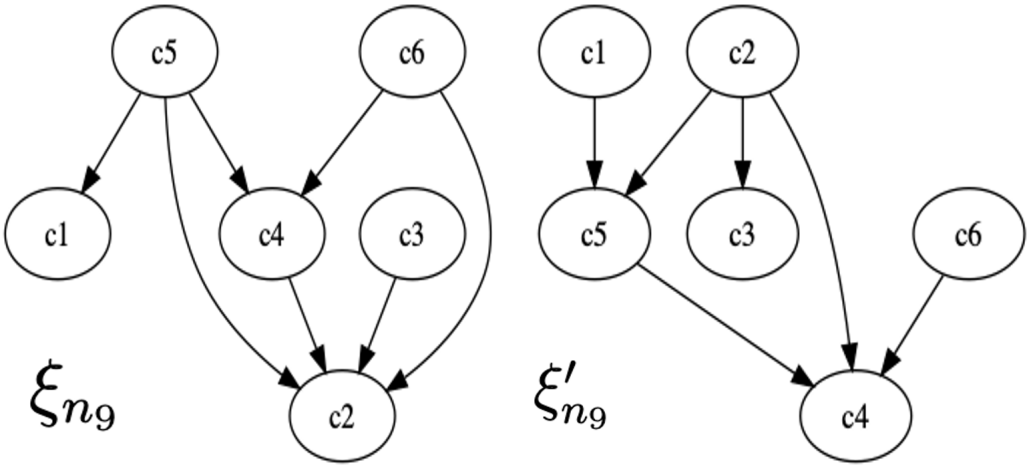


FIGURE 3.3: Example of a static-priority management protocol

antisymmetry and transitivity ensures that no deadlock can occur. For example, suppose the two connections in the directed graph meet symmetry. In that case, the vehicles on the two roads will give way to each other due to the unclear priority for multiple connections with closed loops. The fact that the graph is relative to Z_n means that for every pair of conflicting connections, one must have priority over the other. A collision or deadlock will occur if two conflicting connections do not satisfy completeness. Let Ξ_n represent all possible priorities that satisfy all the conditions above for intersection $n \in N$.

Example 3.2. *In Fig. 3.3, the graph on the left-hand side shows a simple example of a priority graph for the T-junction* (n_9)*. The example can be explained as a stop sign on* l_1 *in the real world. Vehicles on edge* l_5 *that turn left or go straight and vehicles on* l_4 *that go straight have the highest priority. Then, vehicles on* l_1 *that turn left and vehicles on* l_3 *that turn right have the second-highest priority. Last, vehicles on* l_1 *that turn right have the lowest priority. For the graph on the right-hand side, we can use a similar explanation.*

3.2.2 Dynamic priority management protocols

Although the static priority protocol has certain advantages in managing unevenly distributed traffic flow, it has specific fairness issues if it is relatively balanced. For example, in the case of balanced traffic flow, vehicles on lower-priority connections have higher delays than vehicles on higher-priority connections. Formally, we define a dynamic priority management protocol as follows.

Definition 3.2. Let $G = (N, L)$ be a road network, and, for some intersection $n \in N$, let C_n and Z_n be the corresponding connection- and conflict relation, respectively. A dynamic-priority management protocol $\eta_n : T \to 2^{\Xi_n}$ is a function that maps each point in time to a priority graph, where T represents time.

Intuitively, a dynamic priority management protocol specifies which priority graph is used at each point in time. An example of a dynamic priority traffic management protocol is shown below.

Example 3.3. *Consider an intersection n_9 as shown in Fig. 3.2 on the left. A dynamic-priority traffic management protocol at intersection n_9 can be defined as follows:*

$$\eta_{n_9}(t) = \begin{cases} \xi_{n_9}, & \text{if } 0 \leq t \bmod \lambda < \lambda_1 \\ \xi'_{n_9}, & \text{if } \lambda_1 \leq t \bmod \lambda < \lambda \end{cases}$$

where ξ_{n_9} (Fig. 3.3 left) and ξ'_{n_9} (Fig. 3.3 right) are two priority graphs of intersection n_9 and $0 < \lambda_1 < \lambda$. The protocol specifies two-time intervals in each period of length $[0, \lambda_1)$ and $[\lambda_1, \lambda)$. The priority graph ξ_{n_9} is used in the first time interval, and the priority graph ξ'_{n_9} is used in the second time interval.

3.2.3 Vehicle-based priority management protocols

The static and dynamic priority management protocols are based on elaborating the right of way. These two protocols are not only for autonomous vehicles but can also

Algorithm 3.2.1: Vehicle-based priority management protocol

1 **Input:** $\mathcal{V}_n$ is a set of vehicles that cross an intersection n.
2 **Output:** $CS[|\mathcal{V}_n|]$ is sequence of vehicles to pass the intersection n.

1: $round = 0$;
2: $Participants(round)$: A set of first vehicles on every incoming lane of intersection i;
3: **while** $round < |\mathcal{V}_n|$ **do**
4: $\quad Proposal(round) = \{t(v) | \forall v \in Participants(round)\}$;
5: $\quad Winner = \arg\min_{v \in Proposal(round)} t(v)$;
6: $\quad CS[round] = Winner$;
7: $\quad round = round + 1$;
8: $\quad Participants(round) = Participants(round) \cup \{new\_vehicle\} \setminus winner$, where $new\_vehicle$ is the vehicle behind of the winner vehicle.
9: **end while**

be used for existing traffic control. When autonomous vehicles equipped with communication facilities are implemented, a vehicle-based priority management protocol can be used as one of many protocols to manage the passage of autonomous vehicles through intersections. The vehicle-based priority management protocol can be interpreted as communication between vehicles to determine the order of passing through the intersection.

First-come-first-serve (FCFS) is a typical communication criterion; the vehicle arriving can pass through the intersection first. Algorithm 3.2.1 shows the vehicle-based management protocol based on FCFS, which means that the sequence in which the vehicles are allowed to cross is based on the arrival time of those vehicles. Suppose that $\mathcal{V}_n$ is a set of vehicles that cross intersection n, and $t(v)$ denotes the arrival time of vehicle $v \in \mathcal{V}_n$. In each iteration, the algorithm compares the arrival time of all the lead vehicles entering the lane, and the winner is the vehicle that reaches the intersection first. Then, the set of vehicles allowed to pass the intersection in the next round is the collection of the remaining vehicles from the previous round and the vehicle after the winning vehicle. It is worth mentioning that the vehicle-based priority management protocol can not only use the arrival time as a criterion for vehicles but also use other criteria based on, for example, auctions. In this case of an auction, vehicles may propose a

price they are willing to pay. If the arrival time in Algorithm 3.2.1 is changed to the bid of each vehicle, then the winner is determined as the vehicle with the highest bid so that Algorithm 3.2.1 can represent the agreement of the passing right of the auction. However, in the rest of this section, we only use the arrival time as the vehicle-based priority management protocol criterion.

3.2.4 Experimental setting and results

AIM4 provides traffic simulation at multiple intersections, offering the possibility of traffic control at the road network level for autonomous driving. It is based on the open-source project "Autonomous Intersection Management (AIM)" conducted by the AI Laboratory Learning Agents Research Group in the Department of Computer Sciences at the University of Texas at Austin [Dresner and Stone, 2008b].[1] [2] However, it has some unrealistic drawbacks that make its simulation results different from real-world traffic management. First, in AIM4, although a simulated environment with multiple intersections is provided, each intersection uses the same traffic light management strategy rather than each intersection operating independently. Second, vehicle departures follow a uniform distribution, meaning that a vehicle is generated at fixed time intervals from the origin. Finally, in AIM4, vehicles cannot choose their routes but always follow the shortest route from the origin to the destination.

This subsection describes a new simulation platform, Aim4+, extended and implemented from AIM4 that can simulate traffic on any road network with a graph representation of roads and a configuration of priorities among roads and vehicles. Based on the meso-level road network model, we developed a system that can simulate complex road networks with autonomous vehicle traffic under the management of different traffic control protocols at different intersections. With this simulation system, we can test various properties of traffic management protocols from macro- and micro-perspectives

[1] https://www.cs.utexas.edu/ aim/

[2] Although the system can take any input of a road graph and a configuration of priorities, the capacity of roads and vehicles are limited by computer hardware and GUI setting.

of a traffic network with autonomous vehicles. We can set different speed limits for each road between the intersections. Furthermore, vehicles can autonomously choose their routes and travel speeds. The intersections are independently managed under different traffic control protocols based on preset priorities of roads and vehicles. Furthermore, the simulator provides a variety of data collection APIs, which allow automated data collection for different traffic scenarios.

In experiments in the following section, we assume that all vehicles are homogeneous autonomous vehicles, so the vehicle parameters are the same (acceleration/deceleration, angular velocity, length, and width). In this section, we first test the average delay of vehicles driving on the road network, separated by different priorities. Second, we tested whether there is a so-called 'bullwhip effect' (explained below) at a single intersection. Then, we simulated and compared the average delay experienced by vehicles when waiting at the intersection due to the various traffic management protocols. Finally, based on our test results, we determine, for each protocol, a 'delay function' that describes the delay of the vehicles."

3.2.4.1 Average delay for the static-priority management protocol

For our simulation environment, we use an intersection with one lane for each direction of entry. In Fig. 3.4, the image on the left shows the intersection we used in our simulation, including nine connections inside the intersection. The image on the right provides a simple static priority protocol for that intersection. To avoid overly complex expressions of static priority protocols, we assume that no vehicles are entering the intersection from lane l_7, so any connections from l_7 are not shown in the figure. The connections in Fig. 3.4 left are $C_{n_2} = \{c_1 = (l_1, l_2), c_2 = (l_1, l_4), c_3 = (l_1, l_6)\}, c_4 = (l_3, l_4), c_5 = (l_3, l_6), c_6 = (l_3, l_8)\}, c_7 = (l_5, l_6), c_8 = (l_5, l_8), c_9 = (l_5, l_2)\}$. To simplify the presentation, we divide the connections into several different layers, such that in each of these layers, the connections in that layer do not conflict with each other. Specifically, the static-priority protocol in Fig. 3.4 is divided into three layers: $H_0 = \{c_1, c_2, c_3\}$,

$H_1 = \{c_4, c_5, c_6\}$, and $H_2 = \{c_7, c_8, c_9\}$. The connections in H_0 have the highest priority over H_1 and H_2; this means that vehicles on the connections located in H_1 and H_2 should give way to vehicles on the connections located in H_0. Similarly, connections in H_1 have a higher priority than H_2, and connections in H_2 have the lowest priority in this example.

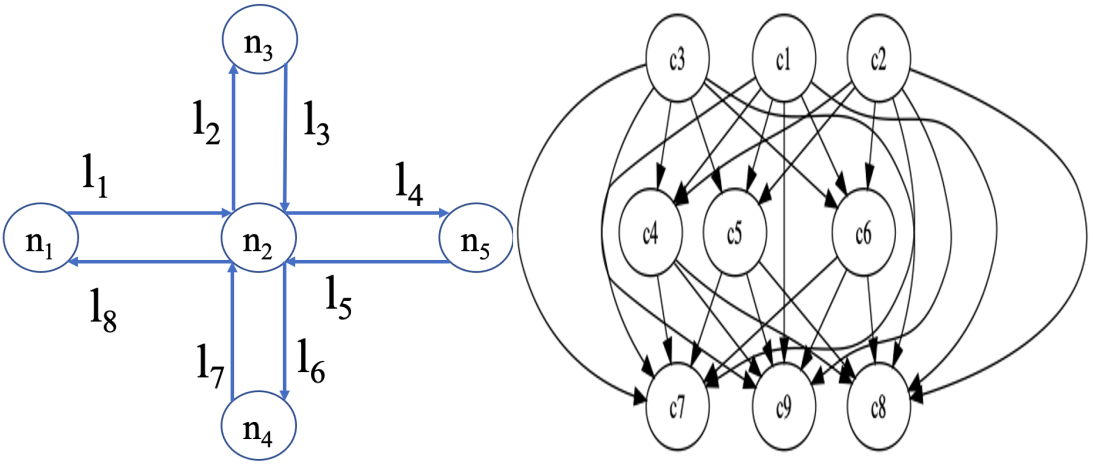


FIGURE 3.4: Conflict relation of n_2

The average delay here means how many times all vehicles wait before passing the intersection. Fig. 3.5 shows a vehicle's average delay at an intersection to give way to other vehicles with higher priority in different priority hierarchies. The X axis is the traffic flow per hour, and the Y axis is the average delay for all vehicles in a layer. It can be seen from the experimental data that the delay of high-priority vehicles does not increase with increasing traffic flow. This means that the delay of vehicles travelling on high-priority connections is negligible. In contrast, as traffic flow increases, the delay time of the second-highest-priority and lowest-priority vehicles increases. As traffic flow gradually increases, the average delay increases non-linearly.

3.2.4.2 Bullwhip effect for the static-priority management protocol

This subsection shows that our extended AIM4+ simulator can perform several functions to test traffic and simulate traffic management on the road network. We use the

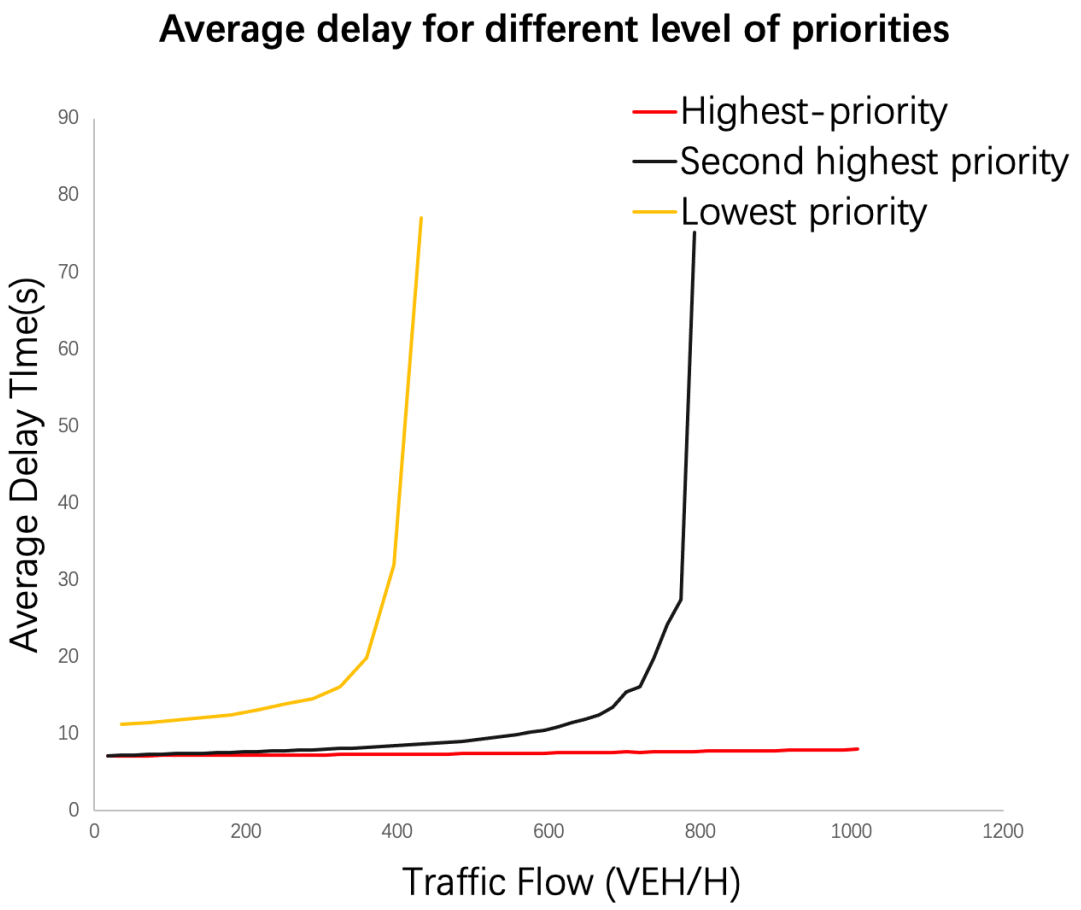


FIGURE 3.5: Average delay for different levels of priorities

so-called 'bullwhip effect' in supply chain management to test the impact of intersections with different priorities on traffic delays. The bullwhip effect (also known as the Forrester effect) is a demand distortion that flows upstream in the supply chain from the retailer to the wholesaler and the manufacturer due to order volatility greater than the variance of sales [Lee et al., 1997]. Information cannot be exchanged efficiently along the supply chain when transmitted from the ultimate customer to the original provider, resulting in increased fluctuations in demand information. This section uses the bullwhip effect of the delay impact on vehicles of different priority levels at an intersection, which means that when the traffic flow on the high-priority road fluctuates, how it affects the delay of other priorities.

In our experiments, we use a single intersection, like the one depicted in Fig 3.4, and

use two priority layers H_0 and H_1 mentioned in section 3.2.4.1 as simulation environment. Vehicles are generated at l_1 and l_3 in Fig 3.4 with pre-defined traffic volume. At first, we assume that new vehicles will enter the intersection from l_1 or l_3 at fixed time intervals, for example, every 6 seconds. Then, we compare this with the case that they appear at random moments determined by some given probability distribution. Our experiment then consists of varying the standard deviation of this probability distribution and measuring the effect this has on the delay of the other vehicles that are driving in the H_1 lanes. Note that if the standard deviation is 0, it means we are looking at the original situation in which new cars appear at fixed time intervals. On the other hand, for larger standard deviations, the cars will appear at the intersection in a more chaotic fashion.

To change the standard deviation of the probability distribution, we created a new algorithm to generate vehicles from the spawn point, as shown by Algorithm 3.2.2. The core of this algorithm is divided into several steps. The first step is to fix the traffic flow rate and the number of time intervals in an hour. Suppose that traffic flow is expressed as time intervals of $720\,Veh/hr$ is 40, which means that $90s$ for each time interval. First, we randomly generate 40 random numbers from $0 - 20$, then obtain the sum of the generated random numbers. The second step is calculating the number of vehicles that need to be generated for each time interval. The calculation method is to take the random number and the total of each interval in the first step as the ratio and then multiply it by the total number of vehicles per hour. The third step is to generate the vehicle spawn time from the respective origins. For example, if we assume that 30 vehicles need to appear during the first time interval, then we randomly draw 30 numbers between 0 and 90, where each number represents the time that each vehicle should appear. As the number of time intervals increases, the randomness of vehicle generation decreases and leads to a decrease in standard deviation.

Fig. 3.6 shows our experimental results to measure the bullwhip effect according to different traffic flows. From the two graphs in the first row, it can be concluded that the standard deviation of the second-highest-priority connections increases as the standard

Algorithm 3.2.2: Vehicle generation for the Bullwhip effect

Input: Traffic flow per hour f and number of time interval N.
Output: Vehicle spawn time in an hour.

$Number\_of\_vehicles[N]$;
$sum = 0$;
$Time\_of\_interval = \frac{3600}{N}$;
for $i < N$ **do**
    $Number\_of\_vehicles[i] = Random(0, 20)$;
    $sum = sum + Number\_of\_vehicles[i]$;
end for
$sum\_real\_number = 0$;
for $i < N$ **do**
    $Number\_of\_vehicles[i] = \frac{Number\_of\_vehicles[i]}{sum} * f$;
end for
while $current\_time < MAX\_TEST\_TIME$ **do**
    $Current\_interval = Current\_time\%Time\_of\_interval$;
    if $Current\_time\%Time\_of\_interval = 0$ **then**
        for $i < Number\_of\_vehicles[i]$ **do**
            $Spawn\_time[i] = Random(0, Time\_of\_interval)$;
        end for
        Sort $Spawn\_time[]$ from smallest to largest;
        $Spawned\_vehicle = 0$;
    end if
    if $Current\_time = Spawn\_time[Spawned\_vehicle]$ **then**
        Spawn a vehicle;
        $Spawned\_vehicle = Spawned\_vehicle + 1$;
    end if
end while

deviation of the highest-priority connections increases. We can therefore conclude that the bullwhip effect exists in our experiment. The second row of the graph represents the increase in the standard deviation of the second-highest-priority connections, causing the increase in the average delay of vehicles on those connections. Generally, when we follow the experimental setup for the test when the standard deviation for the high-priority road increases, it causes the delay time of vehicles on the low-priority lanes to follow a monotonic increase.

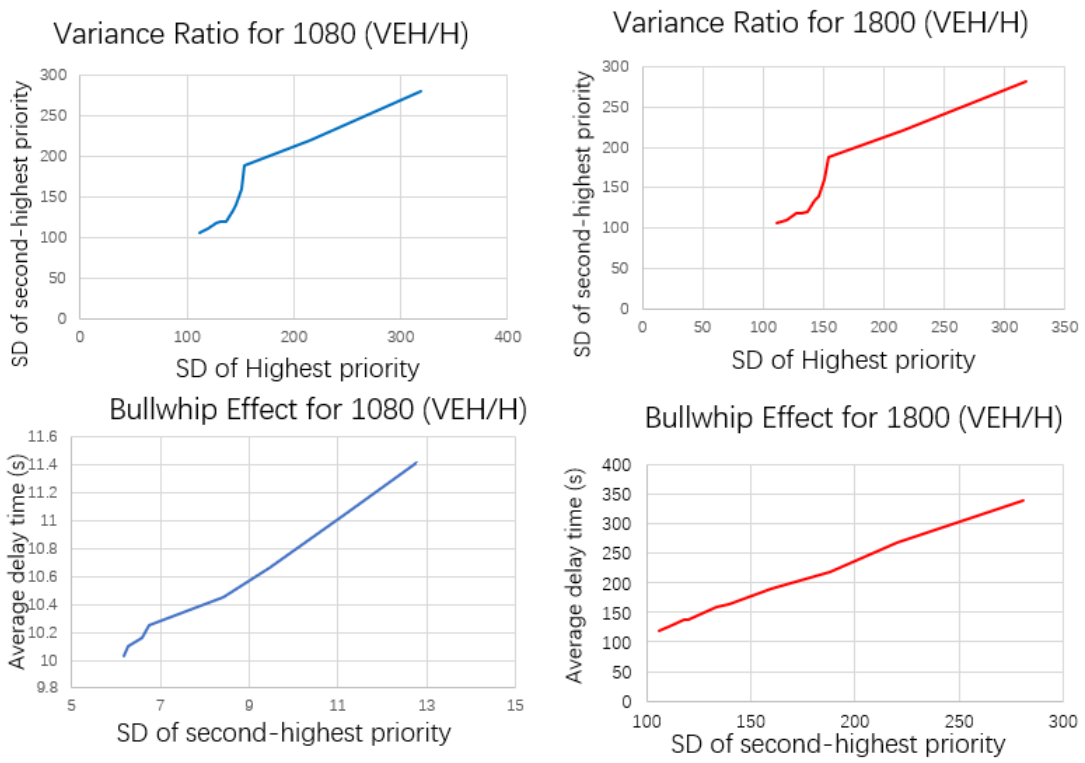


FIGURE 3.6: Bullwhip effect with different traffic flows

3.2.4.3 Average delay for different management protocols

Next, we compare the average delay time under three traffic management protocols: static-priority, dynamic-priority, and vehicle-based. In this test, we use a single intersection, and vehicles are generated from two incoming lanes. This means that, in any given time interval, the dynamic priority traffic management protocol allows vehicles to pass through one incoming edge while the other direction is blocked. In the static-priority traffic management protocol, there are two priority levels, such as H_0 and H_1, in Section 3.2.4.1. And the third protocol is the vehicle-based priority management protocol, which uses FCFS as its basic rule.

Fig. 3.7 shows the test results under three different traffic management protocols proposed above. The x-axis represents the ratio between the respective traffic flows on the two incoming lanes l_1 and l_2, and the Y-axis is the average delay of all the vehicles tested. For example, suppose there are 1000 vehicles per hour heading to an intersection from two incoming lanes l_1 and l_2. We use the notation $90 : 10$ to indicate that there

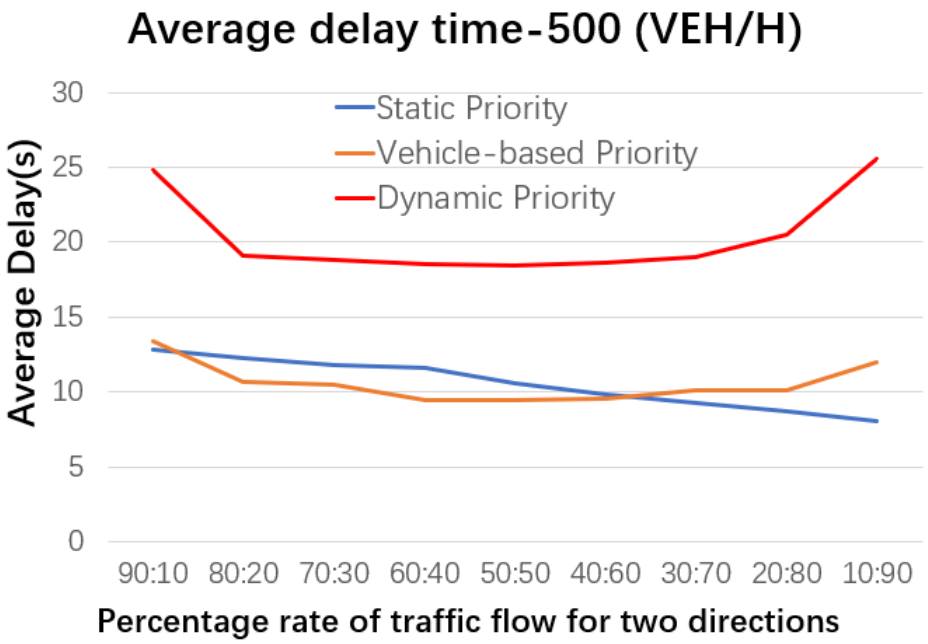


FIGURE 3.7: Average delay of different management protocols

are 900 vehicles heading from lane l_1 and 100 vehicles heading from lane l_2. This is an extremely unbalanced traffic flow. Similarly, $50:50$ represents balanced traffic flow from the two incoming lanes (500 vehicles per hour for each lane).

We see from Fig. 3.7 that the delay increases significantly for the dynamic priority protocol when the traffic flow is unbalanced. For the static priority management protocol, we see that as the amount of traffic with high priority increases relative to the amount of traffic with low priority, the delay time decreases. The vehicle-based priority protocol has the same trend as the dynamic priority in Fig. 3.7 but has a lower delay time. Furthermore, we see that vehicle-based and static priority protocols yield shorter delays than dynamic priority protocols.

3.2.4.4 Delay function

Each intersection has an independent delay function in which the input of the delay function is the traffic flow of each incoming lane, and the output is the delay time of that lane. Among the literature on the research of delay functions with traffic management protocols, there is a famous delay function so-called the ”Webster function” [Webster,

1958], which expresses, for an intersection with traffic lights, the relationship between the average delay d and the traffic flow f in one particular direction.

$$d = \frac{c(1-\lambda)^2}{2(1-\lambda x)} + \frac{x^2}{2f \cdot (1-x)} \tag{3.1}$$

where c is the time it takes from the traffic light to go through the cycle from green, to orange, to red and back to green again, called *cycle time*; $\lambda = \frac{g}{c}$ is the proportion of the cycle in which the light is green (where g is the total amount of time that the traffic light is green during a single cycle); f is the traffic flow in the direction; $x = \frac{f}{s}$ is known as the *degree of saturation*, which is the ratio of the actual flow to the so-called *saturation flow* s. The saturation flow is the maximum amount of traffic that can pass through the intersection in an hour.

In this experiment, we have tried to find a general expression between delay time and traffic volume in different traffic control protocols. For the experiments in this chapter, we use a single intersection with 4 different entry directions, similar to Fig. 3.4. We tested the intersection's static priority, dynamic priority and vehicle priority management protocols. For static priority protocol, we define one direction as one hierarchy, like the one shown in Fig. 3.4, but with the difference of having four hierarchies. For the dynamic priority protocol, we use 120s as the cycle time and change the priority graph every 30 seconds so that each direction will be the highest priority in a certain time interval. Vehicle priority uses the original FSFC rules, meaning that the sooner a vehicle reaches the intersection, the sooner it passes. We collected data on the average waiting time of vehicles in different directions at simulated intersections. We did this by running a simulation and measuring the delay time as a function of the traffic flow. We then tried to find the function that best fits the data. Specifically, We found that the function that fits the data best has the following exponential form:

$$d = ae^{bf} + c \tag{3.2}$$

where a, b, and c are non-negative numbers, and f is the traffic flow. It is worth noting that the dynamic priority protocol is similar to a traffic light but not identical. In the dynamic protocol, we allow connections that do not collide with the highest priority connections to pass directly through the intersection.

Fig. 3.8 shows the curve-fitting results for different priority traffic management protocols. Blue crosses, orange squares, and green dots are the data collected from our simulator. The yellow line Eq. (3.3) is the delay function obtained by the static-priority based management protocol using the composite function. The blue line Eq. (3.4) represents the delay function of the vehicle-based priority management protocol. The red line Eq. (3.5) represents the delay function of the dynamic-priority based management protocol. And the black line represents the Webster function that uses the same parameters as the dynamic-priority management protocol, where $c = 66, g = 30, s = 600$.

$$D_s = (4.233 \times 10^{-8}) \cdot e^{0.02834f} + 7.0244 \tag{3.3}$$

$$D_v = (1.002 \times 10^{-9}) \cdot e^{0.03f} + 7.24 \tag{3.4}$$

$$D_t = (6.34 \times 10^{-4}) \cdot e^{0.01253f} + 16.17 \tag{3.5}$$

From Fig. 3.8, we can conclude that, in our scenario, using exponential functions can better simulate the delay compared with the Webster function. The reason for this difference is that the Webster function only describes the delay time for one specific direction, while we have used the average delay time in each direction as the analysis data. Interestingly, our delay function can describe different traffic management protocols by changing variables, which means that our results are suitable for more generic traffic scenarios.

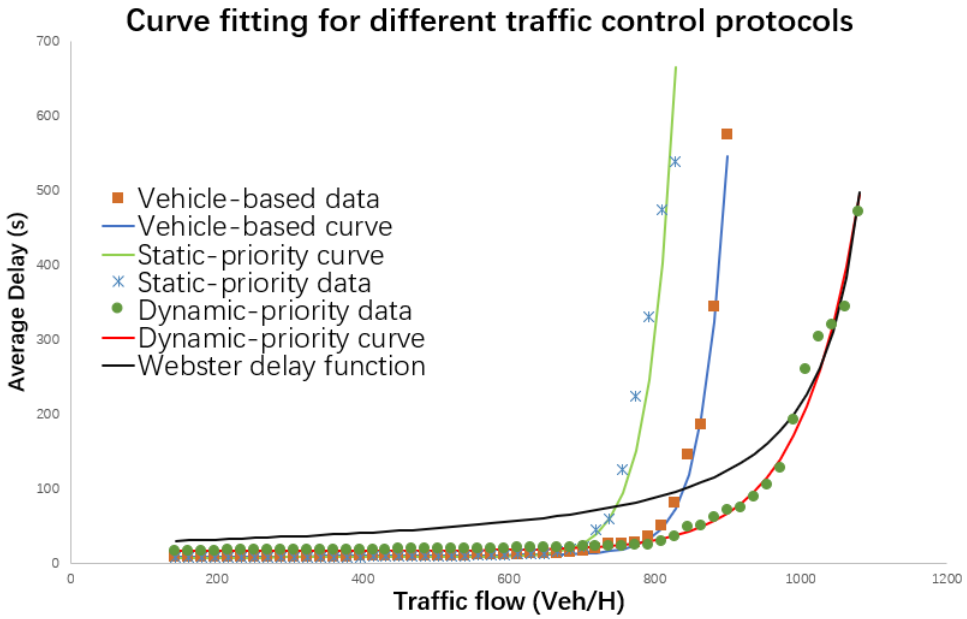


FIGURE 3.8: Experimental results and regression for different management protocols

3.3 Traffic Management Protocols for Micro-level Road Networks

In this section, we investigate traffic management protocols for the micro-level road network model $(B, \mathcal{E}, B_n, B_x)$ (see Def 2.5). Several traffic management protocols based on the meso-level road network model were proposed in the previous section. While it can be used to describe a road network of multiple intersections or a single intersection, it is not suitable for describing how the traffic state changes over time when the time factor is taken into account in the model. Therefore, the main motivation in this chapter is to use a microscopic road network model and introduce a time concept to describe the change of traffic state as time changes. This makes it possible to adapt control protocols over time depending on the traffic volume from different directions at an intersection.

Despite the significant differences between the different methods and traffic control systems, the mechanisms of all traffic control methods can be categorized into two fundamental traffic control protocols: *time-based traffic control* and *priority-based traffic control*.

3.3.1 Time-based traffic control protocols

Time-based traffic control uses protocols that control traffic by restricting the accessibility or impassibility of roads in different periods. With the road graph representation, we can define a time-based traffic protocol as a temporal sequence of arc groups, indicating which road segments are accessible at each time. A typical application of time-based traffic control protocols is traffic light systems. Formally, we define a time-based traffic control protocol as follows:

Definition 3.3. Given a road graph $G = (B, \mathcal{E}, B_n, B_x)$. A *time-based traffic control protocol* $\alpha : T \to 2^{\mathcal{E}}$ is a function from each time point to a subset of arcs. Furthermore, a traffic flow $\mathcal{F} = \langle (V_t, \tau_t) \rangle_{t \in T}$ is said to *comply with* a time-based protocol α if for any time point t and any vehicle v, if $v \in V_t$, $\tau_t(v) \neq \sigma(v)$ and $\tau_t(v) \neq \tau_{t+1}(v)$, then $(\tau_t(v), \tau_{t+1}(v)) \in \alpha(t)$.

Intuitively, a time-based traffic control protocol specifies which road segment can be passed at each time point. In other words, for each time point t, all arcs in $\alpha(t)$ are passable (green light), while all arcs in $\mathcal{E} \setminus \alpha(t)$ are impassable (red light). When a traffic control protocol is enforced on the road, the traffic is shaped to form specific traffic flow patterns. If a traffic flow is consistent with a time-based protocol, it means that all vehicles only travel through passable arcs. The following example shows a representation of traffic in a four-way intersection when a time-based protocol is enforced.

Example 3.4. *Consider a road graph for a four-way intersection in Example1, and a time-based traffic control protocol α on the road graph as follows:*

$$\alpha(t) = \begin{cases} E_g \cup \{e_2, e_3, e_9, e_8\}, & \text{if } 0 \leq t \bmod \lambda < \lambda_1 \\ E_g \cup \{e_2, e_3, e_{18}, e_{14}\}, & \text{if } \lambda_1 \leq t \bmod \lambda < \lambda_2 \\ E_g \cup \{e_{14}, e_{13}, e_{17}, e_{18}\}, & \text{if } \lambda_2 \leq t \bmod \lambda < \lambda_3 \\ E_g \cup \{e_9, e_8, e_{13}, e_{17}\}, & \text{if } \lambda_3 \leq t \bmod \lambda < \lambda \end{cases}$$

where $0 < \lambda_1 < \lambda_2 < \lambda_3 < \lambda$ *and* $E_g = \{e_1, e_4, e_5, e_6, e_7, e_{10}, e_{11}, e_{12}, e_{15}, e_{16}, e_{19}, e_{20}\}$. $t \bmod \lambda$ *means "t modulo λ".*

The protocol specifies four time intervals in each period of length λ: $[0, \lambda_1)$, $[\lambda_1, \lambda_2)$, $[\lambda_2, \lambda_3)$ and $[\lambda_3, \lambda)$. In the first time interval, vehicles from the east or the west can travel straight or take a left turn. In the second interval, traffic from the west is allowed to take a right turn and traffic from the south is allowed to take a left turn. The other two intervals are similar for traffic from other directions.

3.3.2 Priority-based traffic control protocol

A priority-based protocol controls traffic based on preset priorities of roads at each road junction. For instance, in a left-driving country, vehicles give way to the traffic on the right whenever they are approaching an intersection, which means that the road on the right has higher priority than the road a vehicle travels. With the help of a graph representation of roads, we can formalise a priority-based traffic control protocol as follows:

Definition 3.4. Given a road graph $G = (B, \mathcal{E}, B_n, B_x)$. A priority-based protocol $\beta : \mathcal{E} \to 2^{\mathcal{E}}$ is a function from each arc of the road to a subset of the arcs $\mathcal{E}$ such that

1. for any $(b_1, b_2) \in \mathcal{E}$, if $(b_1', b_2') \in \beta(b_1, b_2)$, then $b_2 = b_2'$;
2. if $e' \in \beta(e)$, $e \notin \beta(e')$.

To understand the conditions of the definition, the first condition means that an arc gives priority to another arc only if they meet at the same block. The second condition means that two vehicles on different roads do not give way to each other. Similar to time-based traffic control protocols, we can also define whether a traffic flow complies with a priority-based traffic control protocol.

Definition 3.5. Let $(G, \mathcal{V}, \mathcal{I})$ be a traffic setting. A traffic flow $\mathcal{F} = \langle (V_t, \tau_t) \rangle_{t \in T}$ is said to *comply with a priority-based protocol* β if for any $t \in T$ and any $v, v' \in V_t$, such that $v \neq v'$, $\tau_t(v) \neq \tau_{t+1}(v)$ implies $(\tau_t(v'), \tau_{t+1}(v)) \notin \beta(\tau_t(v), \tau_{t+1}(v))$ unless $(\tau_t(v'), \tau_{t+1}(v)) \notin \mathcal{P}(v)$.

It means that a vehicle does not have to give way to another vehicle only if the road that the other vehicle travels on does not have a higher priority or the other vehicle does not travel into the same block.

Example 3.5. *Fig. 3.9 shows a road graph representing a T-junction. Imagine a stop sign in block* b_{15}*. Then all vehicles entering* b_9 *via* e_{14} *must stop at the stop sign and observe the coming vehicles towards* b_9 *from other roads. Assume that we enforce the following priority-based traffic control protocol* β *at this T-junction:*

- $\beta(e) = \emptyset$*, where* $e \in \{e_1, e_3, e_4, e_5, e_6, e_7, e_8, e_{10}, e_{11}, e_{13}, e_{14}, e_{15}, e_{16}\}$.
- $\beta(e_2) = \{e_{13}\}$; $\beta(e_{12}) = \{e_8\}$; $\beta(e_9) = \{e_{14}\}$;

Let $\mathcal{F}(t) = (V_t, \tau_t)$ *be a traffic state at time* t *where* $V_t = \{v_1, v_2\}$, $\tau_t(v_1) = b_{10}$ *and* $\tau_t(v_2) = b_{15}$*. Assume that both vehicles* v_1 *and* v_2 *travel towards block* b_9*. Since* e_8 *has a higher priority than* e_{14}*, only vehicle* v_1 *can go through but* v_2 *must stay in block* b_{15}*. If* $\mathcal{F}(t+1)) = (V_{t+1}, \tau_{t+1})$ *represents the next state, we then have* $\tau_{t+1}(v_1) = b_9$ *and* $\tau_{t+1}(v_2) = b_{15}$.

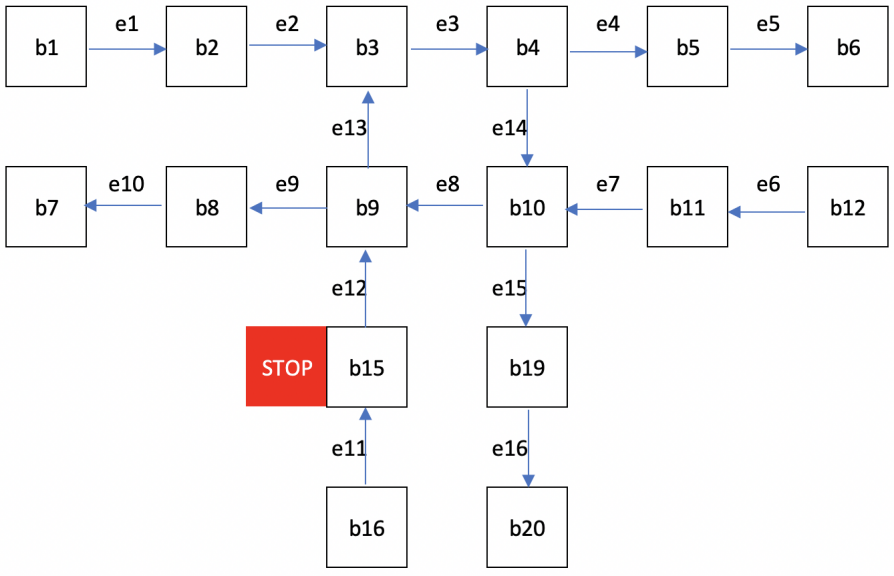


FIGURE 3.9: An example of a priority-based protocol on a T-junction

3.4 Traffic Management Protocol Simulation with Robot Operating System

In the previous two sections, we introduced traffic management protocols for the meso- and micro- level road network models and used AIM4+ to simulate traffic. However, this simulator simulates traffic from a macroscopic point of view, where the vehicles do not have a real physical sense. The main motivation in this chapter is to use a more realistic simulator to simulate the autopilot at a single intersection. We treat the autonomous vehicle as a mobile robot and use a built-in 3D simulation software in the ROS platform called Gazebo, in which vehicles have realistic physical properties, such as acceleration, deceleration, angular velocity, and volume, and in which the vehicle has sensors, such as LIDAR, to provide a more realistic simulation of road traffic compared to the previous two sections. However, one disadvantage of using a real simulator is that the number of simulated vehicles depends on the computing power of the simulator. Hence, we only test up to 20 vehicles in this section to avoid system downtime.

In this section, we investigate intersection management protocols in an environment where all vehicles are autonomous and capable of communication. It is worth mentioning that this chapter does not involve any road network models from chapter 2, but is purely a simulator test for a realistic traffic point of view. We first investigate a traffic control algorithm for AVs, called the virtual roundabout protocol, to manage traffic at an intersection. After that, we present two algorithms based on the traffic light mechanism and the first-come-first-serve policy. These three algorithms for intersection control are considered and implemented for Gazebo. Although these protocols mimic conventional control mechanisms for human driving, their behaviour in autonomous driving environments differs. We found that the virtual roundabout, a protocol in which all vehicles follow the rules of a roundabout without a physical roundabout, is the most effective. Based on our results, we conclude that the virtual roundabout outperformed the other two protocols concerning average delays regardless of safety distance and traffic load and handled unbalanced traffic reasonably well.

3.4.1 The model of intersection

We consider an intersection as a four-way junction where two roads cross. It has a 100m radius intersection area (see Fig. 3.10) which is controlled by a software agent that we called the *intersection manager*. The intersection manager collects information on the location of vehicles within the intersection area and transmits traffic control signals. Within the intersection area, there is a *stop line* on each road to the intersection; A vehicle must stop before the stop line if the intersection is occupied by another vehicle or if it encounters a red light. The area between the stop lines is called *central area*.

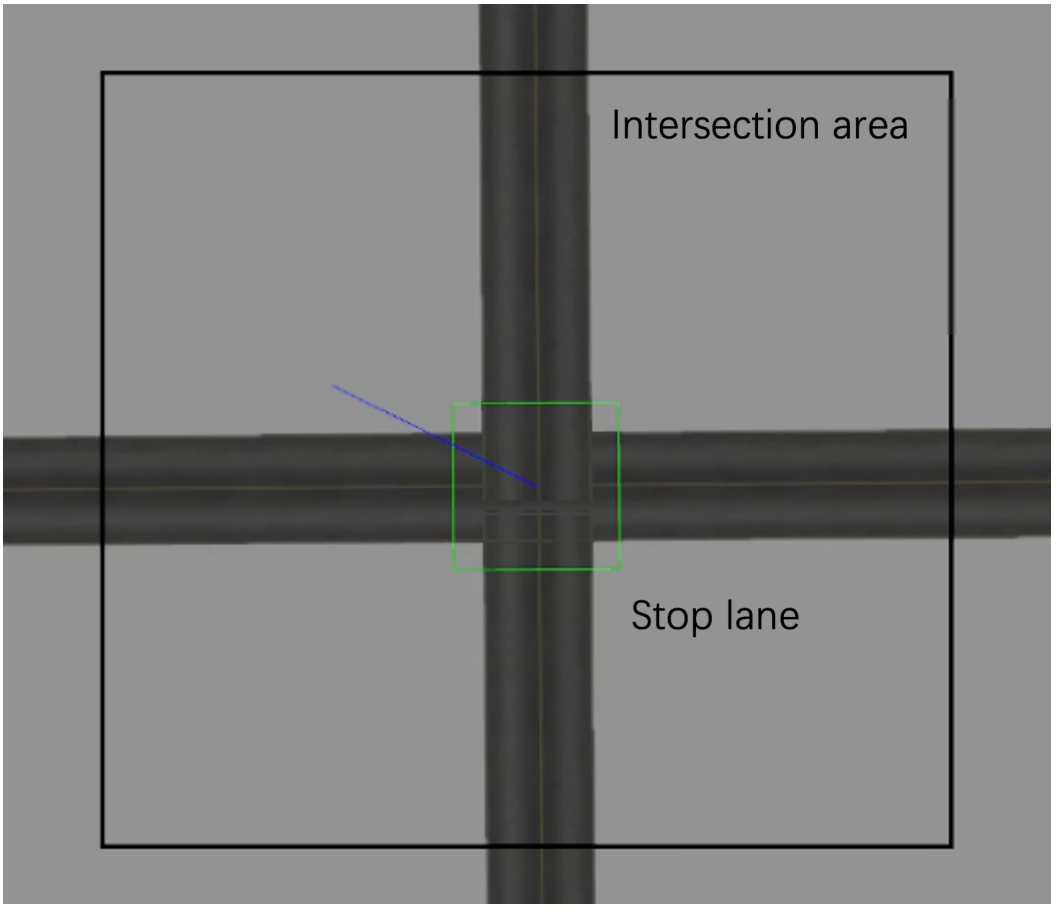


FIGURE 3.10: The model of intersection

Vehicles on the road are assumed to have software and hardware to follow the appropriate protocols and be able to communicate with each other. Communication amongst the vehicles can be peer-to-peer or broadcast via the facilities of the intersection, allowing vehicles to negotiate with each other. Within the intersection area, all vehicles will be

aware of the status of other vehicles in the intersection area. For simulations, the status includes the following information:

- Vehicle ID: an identification number to identify vehicles inside the intersection area.
- Travel direction: the direction towards the intersection.
- Position: current coordinates in the intersection area.
- State: used to determine whether the vehicle has or has not passed the intersection. It can be any of the following four values: "*before central area*", "*waiting at stopping line*", "*inside central area*", "*after central area*".
- Speed: current speed of the vehicle.

The state of a vehicle can be derived from its position and direction; however, explicitly including the state allows a more straightforward implementation of the algorithm. In this study, an intersection consists of four entrances: *east*, *west*, *south*, and *north*, with all cars entering the intersection proceeding straight without turning. Road1 consists of north-south lanes by a solid line; Road2 is the east-west lane.

Consider an observation window period T. We let n denote the vehicles that pass through the intersection area (excluding vehicles that entered before or left after the period). Let W denote the width of the intersection area[3]. We assume that each vehicle $i \in n$ travels in and out of the *intersection area* with velocity V_i. Let s_i be the time it enters the *intersection area* and f_i be the time it leaves the *intersection area*. Then, the travel time $d_i = f_i - s_i$.

[3]For simplicity, we assume that the intersection is a square.

3.4.2 Traffic signal protocol

In Algorithm 3.4.1, we divide the system time into two phases to replicate traffic signals. $Phase\ 1$ allows the vehicles on $Road\ 1$ to pass through while restricting vehicles on $Road\ 2$. In other words, $Phase\ 1$ means a green light for $Road\ 1$, while $Road\ 2$ has a red light, and vice versa for $Phase\ 2$. Instead of simulating yellow light, vehicles on one road do not enter the intersection until vehicles on the other road have left the intersection.

Algorithm 3.4.1: Traffic signal protocol

Input: The set of vehicles N and the array of states $\{state[i] : i \in N\}$
Output: Time record for each vehicle: the finish time f_i and time duration d_i
Function: $traffic\_signals()$:

```
while System is not shutdown do
    Get Systemtime
    for i ∈ N do
        if state[i] is "after central area" then
            Continue
            if Systemtime in phase1, vehicle i in Road1 then
                Vehicle i crosses the intersection and state[i] change to
                "inside central area"
            end if
            if Systemtime in phase2, vehicle i in Road2 then
                Vehicle i crosses the intersection and state[i] change to
                "inside central area"
            end if
        end if
    end for
end while
```

3.4.3 FIFO protocol

The second protocol shown in Algorithm 3.4.2 follows the idea of first-in-first-out queues (FIFO), the earlier a vehicle enters the *intersection area*, the earlier it can pass through. Vehicles arriving later must wait at the stop line until all previous vehicles have passed the intersection. Although this protocol is covered in previous sections in

this chapter, the algorithm in this chapter is based on the intercommunication between vehicles and intersection managers as a basis and is specific to the Gazebo simulator.

Algorithm 3.4.2: FIFO Protocol

Input: The vector of vehicles N sorted by their arrived time and the array of states $\{state[i] : i \in N\}$
Output: Time record for each vehicle: the finish time f_i and time duration d_i
Function: $fifo()$:
  while System is not shutdown **do**
    for $i = 1$ *to* $|N|$ **do**
      occupied=false
      if $state[i]$ is "*inside central area*" **then**
        occupied=true
      end if
    end for
    for $i = 1$ *to* $|N|$ **do**
      if state[i] is "*after central area*" **then**
        Continue
      end if
      if Vehicles i and $i + 1$ are on $Road1$ or $Road2$, $state[i]$ and $state[i + 1]$ are "waiting at stop line", and !occupied **then**
        Vehicle i and vehicle $i + 1$ crosses the intersection and set state[i] and state[i+1] to "*inside central area*"
      else if state[i] is "waiting at stop line" and !occupied **then**
        Vehicle i crosses the intersection and set state[i] to "*inside central area*"
      end if
    end for
  end while

If we strictly follow FIFO, only a single vehicle can pass the intersection at any time. In our implementation, we allow two consecutively arriving vehicles to pass the intersection simultaneously if they are on the same road (in different directions).

3.4.4 Virtual roundabout protocol

In Algorithm 3.4.3, we have implemented a new protocol called the *Virtual Roundabout*. It is a protocol designed to mimic roundabouts without needing a physical roundabout. If the central area is empty, the vehicle that arrived earlier is allowed to enter the intersection. However, if the central area is not empty, vehicles on the same road as the vehicle currently in the central area are allowed to enter the intersection. The difference between FIFO and the virtual roundabout protocol is the ability to allow multiple vehicles to cross the intersection regardless of the arrival time.

Algorithm 3.4.3: Virtual Roundabout Protocol

Input: The vector of vehicles N sorted by their arrived time and the array of states $\{state[i] : i \in N\}$
Output: Time record for each vehicle: the finish time f_i and time duration d_i
Function:
  while System is not shutdown **do**
    occupied=false
    for $i = 1$ *to* $|N|$ **do**
      if $state[i]$ is "*inside central area*" **then**
        occupied=true
      end if
    end for
    for $i = 1$ to $|N|$ **do**
      if state[i] in "*after central area*" **then**
        Continue
      end if
      if Vehicle i in the waiting state, and occupied **then**
        Vehicle i crosses the intersection and set $state[j]$ to "*inside central area*"
      end if
      if Vehicle i on $Road1$ or $Road2$, $state[i]$ is "*inside central area*", and !occupied **then**
        for $j = 1$ to $|N|$ **do**
          if $state[j]$ in "*after central area*" **then**
            Continue
          else if Vehicle j on same road with Vehicle i, and $state[j]$ is "*waiting at stop line*" **then**
            Vehicle j crosses the intersection and set $state[j]$ to "*inside central area*"
          end if
        end for
      end if
    end for
  end while

3.4.5 Simulation

In our simulations, we compare the performance of conventional traffic signals, FIFO, and the virtual roundabout protocol by measuring the average travel delay in each case. To simulate different traffic conditions, we repeat experiments with varying numbers of vehicles and safety distances, which is the distance between two vehicles in the same lane. Furthermore, we varied the 'road balance' to examine the efficiency of the virtual roundabout under asymmetric traffic load.

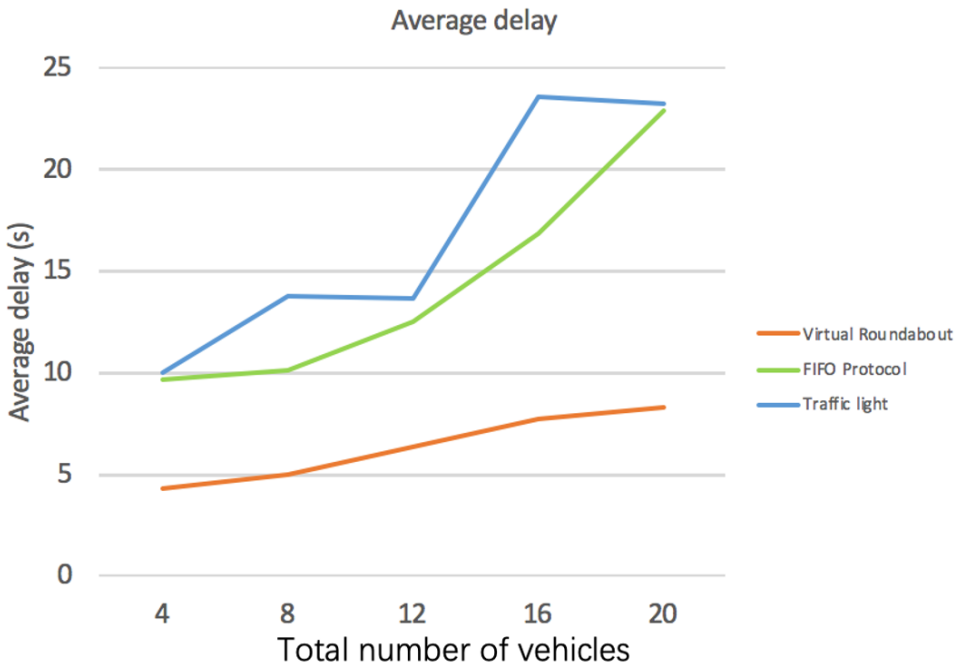


FIGURE 3.11: Average delay

3.4.5.1 Testing environment

We used Gazebo, a simulator designed to simulate real-world robots for our simulation environment. The communication frequency was set to $10Hz$, which means that vehicles broadcast their information 10 times per second.

We define a unit of distance to be exactly the width of the central area. The central and intersection areas were set to have 1×1 $unit^2$ and 10×10 $unit^2$, respectively, with speed in the intersection area of 0.5 $unit/s$. With a scale of $1 : 30$, the central area is 30×30 m^2, the intersection area is 300×300 m^2, and the speed V_i inside the intersection area is $54km/h$. Start times s_i for each car were generated at random. A screenshot of the simulation is shown in Fig. 3.10.

In the rest of this subsection, we analyze our simulations' results and discuss the virtual roundabout's performance against the traffic light and the FIFO protocol.

3.4.5.2 Different traffic flow

For each of the protocols, we measured the average delay time D^T for increasing numbers of vehicles. Regardless of the protocol implemented, the average delay also increased as the number of cars increased (see Fig. 3.11). However, the growth rate of the delay time was different for each protocol.

As the traffic signal protocol was implemented with 30-second phases, traffic that exceeded the maximum number that can be processed in a single cycle must stop and wait, resulting in staggering growth. For the FIFO protocol, the growth rate increases exponentially as each traffic lane is rerouted to the stop line and must wait for cars arriving at the central area first. For the virtual roundabout, the growth rate appears to increase at a linear rate with an average delay of 62% compared to the traffic lights and 56% of the FIFO protocol.

3.4.5.3 Change safety distance

This subsection analyses the influence of different safety distances on vehicle average delay times. The safety distance is the spacing between two vehicles in one road lane, and we measure the safety distance with different vehicle body lengths. We denote the length of the vehicle body by b and set the safety distance at $0.5b$, $1b$, $1.5b$, and $2b$. For example, assuming a body length of 5 meters, $0.5b$ safety distance represents a front-to-rear distance of 2.5 meters, while $1.5b$ represents a front-to-rear distance of 7.5 meters.

In Figs. 3.12, 3.13 and 3.14, we show the relationship between average delays, safety distance, and the number of vehicles for the three protocols. It is evident for the FIFO and the virtual roundabout that as the safety distance increases, so does the average delay. Interestingly, for the traffic light protocol, it was found that the safety distance has no measurable impact on the average delay time.

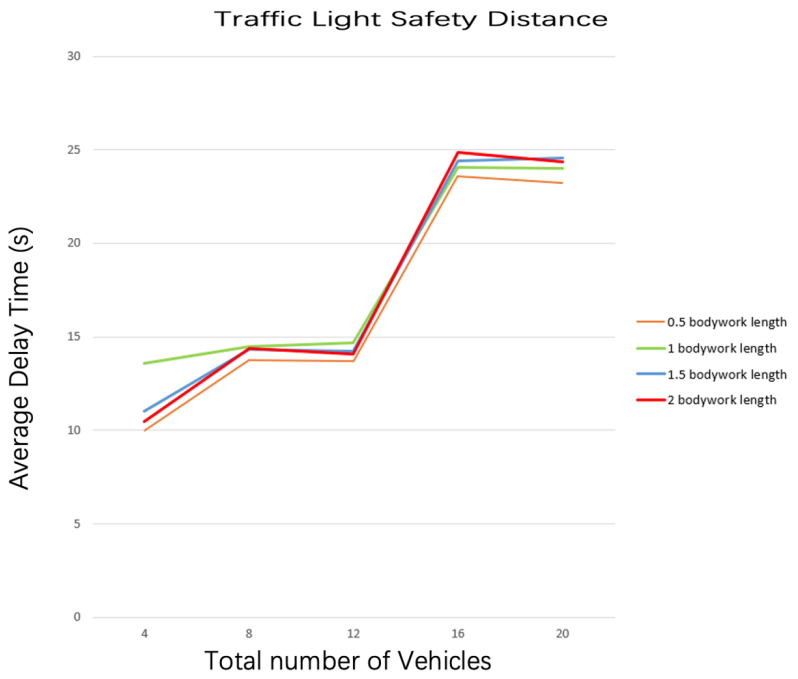


FIGURE 3.12: Varying safety distance with traffic lights

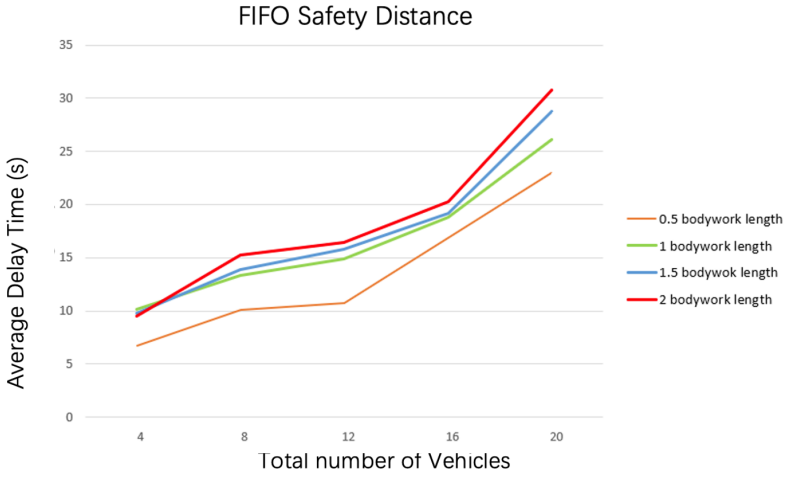


FIGURE 3.13: Varying safety distance with FIFO

3.4.5.4 Unbalanced traffic flow for the virtual roundabout protocol

In our previous simulations, an equal number of vehicles passed through the central area from each direction. In this section, we present our experiments with the virtual roundabout protocol on unbalanced roads: more vehicles coming from one road than from the other. Under realistic conditions, the ratio and direction of vehicles entering an intersection are not uniform.

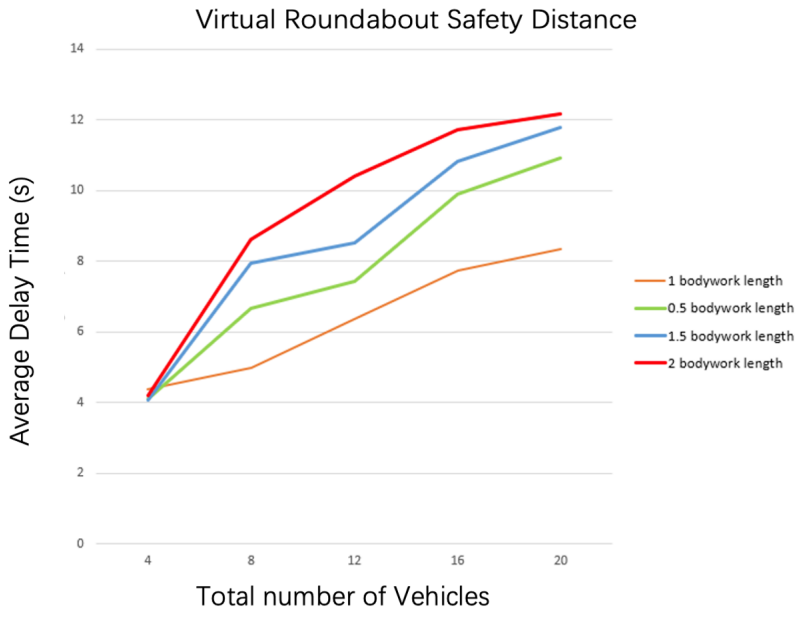


FIGURE 3.14: Varying safety distance with virtual roundabout

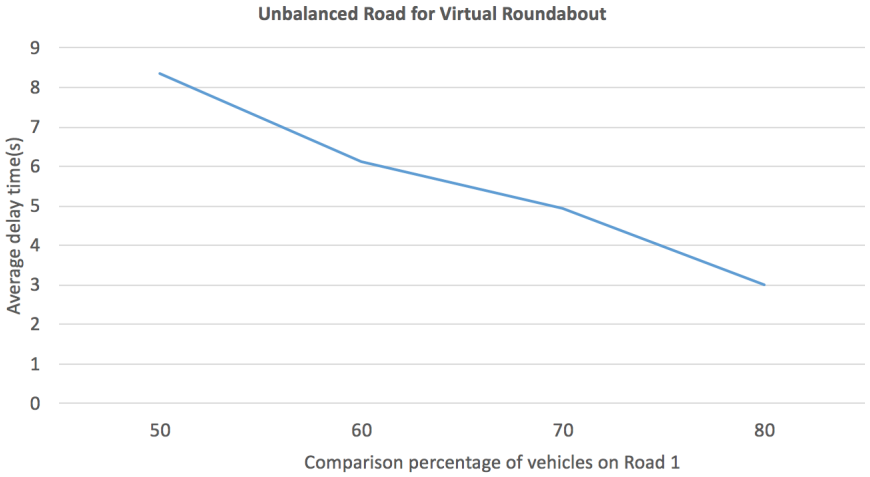


FIGURE 3.15: Increased system efficiency as traffic load becomes imbalanced.

In Fig. 3.15, as the intersection becomes increasingly imbalanced due to the composition of the traffic load increasing on Road1, the average delay decreases.

3.5 Summary

This chapter proposed various autonomous driving traffic management protocols. Furthermore, we used traffic simulation software to collect data and experiment with those

protocols. First, we have provided some background information on intelligent traffic management protocols and an autonomous vehicle simulator. We have provided several priority-based traffic management strategies for autonomous cars based on the meso-level road network model. We extended and implemented a general simulator AIM4+ from the simulator AIM4 and used it to test the suggested protocols. Following that, we have looked at the time- and priority-based traffic management algorithms that are built on the formulation of the micro-level road network model. Finally, we have proposed a virtual roundabout traffic management protocol to manage the passage of self-driving vehicles at a single intersection. We collected data and analysed the relevant algorithms using a ROS simulator called Gazebo that treats robots as self-driving vehicles.

Chapter 4

Exponential Cost Functions for Road Networks

In transportation, a cost function is a function that describes the relationship between traffic flow and travel time, which is essential for the analysis of traffic state, traffic assignment, and road planning. More specifically, a region-based macroscopic cost function is a function to model the travel cost of a specific region, which has potential applications for regional transport planning and macroscopic control of traffic management. Chapter 3 showed that exponential delay functions might be more realistic than previously studied delay functions. This chapter shows that something similar holds for cost functions. We show that an exponential cost function may be more realistic than the commonly used BPR function.

This chapter investigates an exponential latency function for a road network. We focus on the real-world implications of exponential cost-flow functions and use real data to compare them with existing functions to show the advantages of exponential functions. The structure of this chapter is as follows. Section 4.1 briefly introduces this chapter's motivation and related work. Section 4.2 discusses the database used in this chapter and the data processing approach. Section 4.3 then details the data analysis and the

goodness-of-fit results. Section 4.4 summarizes the results of this article and the direction of future work. It is worth mentioning that both Macro-level (Chapter 2.2) and Meso-level (Chapter 2.3) road network models can be used in this chapter. For simplifying the representation, we use the macro road network model $G = (N, P)$, where N is a set of positions and P is a set of roads, in this chapter. Note that we use the symbol P to represent a set of edges instead of E in this section due to the reuse of the symbol for exponential symbol e.

4.1 Introduction

The classic transportation planning model consists of four phases, namely 'trip generation', 'trip distribution', 'modal division', and 'traffic assignment' [Liu et al., 2010]. As explained in previous chapters, traffic assignment refers to the selection of routes (also known as pathways) for vehicles between the origins and destinations of a traffic network [Patriksson, 2015]. The origin-destination matrix Bell [1983] is an essential component of the four-stage model, which describes traffic demand (that is, the number of vehicles between each origin-destination pair (OD)) [Graham and Glaister, 2004]. Travel time is a widely used metric in the field of transportation. Estimated travel time is used to study the effect of traffic load on the road network, which also helps assess traffic management systems' effectiveness in urban areas.

In most road network models, the effect of traffic flow on travel time is specified using cost functions. Each road has a separate cost function that calculates the travel time of that road when the number of vehicles on that road is known. The cost function accounts for the effects of congestion on the road network. Using these functions, travel time and average speed can be calculated under congested and uncongested conditions given parameters such as free flow speed (or free flow travel time) and traffic demand.

As an initial work, the cost function proposed by Smock [1962] was described as an exponential curve in the Detroit Area Transportation Study, and the value of the cost

function shown in Equation (4.1) was estimated by averaging the intersection capacities at the ends.

$$t(x) = t_0 \cdot e^x \tag{4.1}$$

In 1967, Overgaard [1967] proposed another function in the form below.

$$t(x) = t_0 \cdot ae^{x^b} \tag{4.2}$$

Among the well-developed cost functions, the Bureau of Public Roads (BPR) function is the most commonly used one. Since its first publication in 1964, this model has been widely used among researchers in various traffic models due to its good balance between simplicity and effectiveness. In the BPR function shown in Equation (4.3), the ratio of travel time (or average travel speed) per unit of distance to free flow is defined by the parameter a. In contrast, the change of the average travel speed from free-flow to crowded conditions is determined by the parameter b [Mtoi and Moses, 2014].

$$t(x) = t_0 \cdot (1 + ax^b) \tag{4.3}$$

The default values of the parameters a and b are 0.15 and 4, respectively. However, these numbers do not reflect traffic conditions on all types of roads or in all traffic control methods [Márquez et al., 2014]. In actual applications, the parameters need to be adjusted accordingly. Therefore, a calibration process with extensive and accurate field data is needed. Despite its simplicity, this paradigm also has inherent drawbacks, mainly when parameter b is high. It slows down the convergence by giving undue weight to overloaded links with a high value of b. Secondly, the outcome of the BPR function with a high value of b and low traffic flow will be too close to the free-flow travel time [Spiess, 1990].

Besides the BPR function, Akcelik [1978] cost function, which is a variant of the function initially proposed by Taylor [1997], is widely used. Formally, the Akcelik function

is defined as follows.

$$t(x) = t_0 + \frac{3600}{4} a[(x-1) + \sqrt{(x-1)^2 + \frac{8bx}{da}}] \tag{4.4}$$

in which the default values are $a = 1$, $b = 1$, $c = 1$ and $d = 1800$. Akçelik's function can cover more traffic situations, including undersaturated and oversaturated circumstances [Singh and Dowling, 2002].

de Grange et al. [2019] provides a traffic assignment model based on link density, which is the collective result of the responses of road users to the existing transport network and the interactions between road users at different levels of traffic demands. Region-based cost functions, also known as macroscopic cost functions (MCF) [Wong and Wong, 2016], describe the relationship between travel time and traffic demand in the entire network area. The MCF helps planners and engineers understand how road users and the road network interact at different levels of traffic demand. Using MCFs to analyze and plan complex urban networks has several advantages over link-based cost functions. For example, they can significantly reduce computation time. The MCFs, which model travel costs at different network usage levels, have received increasing attention for their potential applications in regional traffic management, control, and land use planning. Kurth et al. [1996] proposed an approach for the regional traffic assignment problem. Kucharski and Drabicki [2017] provided an approach to estimate regionalized cost functions using traffic and speed data transformed into the traffic density of each road.

Most existing studies are based on the regional cost function extended from the BPR function without mentioning other types of functions. Second, the available studies use traffic data in fixed-size regions rather than political region data for function fitting. Finally, we divide the data according to different periods to get more realistic results of the fitted function. The objective of this chapter is to present an exponential cost function and compare it with other relevant functions using a real-world database. The same data is analyzed at different levels of areas in a highly urbanized area of Sydney, comparing

the relevant parameters from the LGA to the suburb and analyzing the relationship between the individual parameters in the function. To analyze the data and calibrate the functions, we divide it into three groups: peak hours, working day off-peak hours, and holidays. The results of the analysis show that the exponential cost function matches the extracted data better than previously studied cost functions. This macroscopic exponential function has deep research implications for regional traffic control, traffic diversion and intelligent cities, mainly for traffic assignment problems.

4.2 Data Description

In this section, we first introduce the exponential cost function for road networks. Then, we describe the database that we used for our experiments. Finally, the literature's BPR function and Akcelik function are used to fit the same database and compared to our proposed exponential cost function.

4.2.1 Exponential cost function

To our knowledge, there is no standard or best practice for cost functions in travel demand models. Field data are usually used to calibrate or validate relevant parameters or construct unique cost functions [Singh and Dowling, 2002]. Since traffic management controls traffic from a relatively macroscopic perspective, we focus on the average travel time through each road instead of focusing on the travel time of each specific vehicle.

Each arc $p \in P$ has an independent exponential cost function as follows:

$$l_p(f_p) = a_p e^{b_p f_p} + c_p \tag{4.5}$$

where $l_p(f)$ represents the time it takes for a vehicle to travel along road p (measured in sec/km), a_p is the congestion sensitivity parameter, b_p is the congestion index, and

$a_p + c_p$ is the free-flow travel time, which depends on the speed limit. It is worth mentioning that the definition of free-flow travel time for the exponential function is similar to that of the BPR function, interpreted as the time required to complete a unit length. Remark that, although Eq. (4.5) has the same expression as Eq. (3.2), it has a different interpretation because this is a cost function, while the other one was a delay function.

4.2.2 Data source

In this study, we have employed Insight, a traffic database provided by Intelematic[1]. The database contains data that covers more than $40,000$km of roads in New South Wales (NSW) and Victoria (VIC) in Australia, with traffic flow data and speed data from 2019 to date. In terms of data frequency, it records data every 15 minutes. Regarding the coverage scope, it ranges from individual link-based data (microscopic level) to Local Government Area (LGA) (macroscopic level). The data used in this chapter is within NSW, covering 39 LGAs, 1058 suburbs and nearly 5000 covered roads in NSW, which have a total length of nearly $19,500$ km. Located on the southeast coast of Australia, the Greater Sydney region is the capital of the Australian state of New South Wales. It is the largest and most populous city in Australia. The Sydney metropolitan area has an area of approximately 1687 km^2 and a population of approximately 5.73 million in 2019. The data between January 2019 and March 2022 are used in time intervals of years, months, weeks, days and even 15 minutes.

The data in the Insight database was collected from numerous sources and validated to provide road and traffic statistics. Real-time detectors (GPS hardware) from commercial and private fleets of road vehicles and millions of nodal sensors at highways and intersections are used to collect real-time traffic flow and speed. Real-time data on road

[1] https://www.intelematics.com/

conditions from emergency services and road management provide the context for further calibration of the data sources. Cross-validation of $24/7$ field control groups and machine learning algorithms ensures the validity of the data.

We use vehicle data from the Insight database in multiple NSW LGAs, and suburbs to compare the proposed exponential function to other existing cost functions. The LGA and suburban data are macroscopic data from January 1, 2019, to February 28, 2022, consisting of speed and traffic flow data. The speed data contains the average speed, average delay time, average travel time, road ID, area ID, total length of the entire road, and average speed limit for the peak and non-peak periods within the area during the time interval. The flow data contains the total traffic flow in an area, road ID, area ID, and total length of roads throughout the area during peak and off-peak periods within the time interval.

The macroscopic level data is obtained by integrating the data of all road sections in the jurisdiction every 15 minutes. The traffic data downloaded from the database is automatically divided into peak and off-peak periods. The peak period is 8 hours in total, including $7\ am$ to $10\ am$ and $3\ pm$ to $6\ pm$ from Monday to Friday, while the off-peak period is 16 hours (excluding the peak period) plus 24 hours per day on weekends (Saturday and Sunday).

Since public holidays are not grouped in the database, it is obvious that the data for these days have more off-peak characteristics, so we manually grouped the public holidays into the off-peak data. The peak period data is averaged over all 8 hours, the weekday off-peak data is divided by 16 to obtain the hourly average, and the non-working day and holiday data are divided by 24 hours to obtain the average. We did not perform any additional processing on the database, except that we manually excluded data for public holidays from the weekday data and moved them to the weekend data.

We filter the data following the criteria below to derive more accurate traffic and travel time correlation functions.

- To best fit the macroscopic function, the sampling region must include as many types of roads as possible, including viaducts, freeways, and urban roads.
- The absence of overlapping areas in the sampling area ensures the regional road network's independence and distinctive topological features.
- The sampling region must feature a variety of land uses, including commercialized areas, high-density office sectors, residential areas with a mix of high and low densities, and a well-connected road system.
- Fitting function data must be sampled over a wide time interval to prevent the impact of crises on traffic data, such as automobile accidents, special holidays, and lengthy traffic control.
- The selected area needs to have prevailing traffic conditions.

After filtering, three LGAs (i.e., Sydney, North Sydney and Parramatta) belonging to the Greater Sydney Area with high traffic flow and population density are selected for analysis, considering that these areas can cover a wide range of traffic conditions. Note that Sydney, North Sydney and Parramatta are three independent political area without any overlap area. In addition to the three LGAs, we select a suburb from each LGA with similar characteristics as described above.

4.2.2.1 Sydney

The City of Sydney is a local government district in the Sydney Metropolitan Area of NSW, comprising the Sydney Central Business District (CBD) and 31 adjacent suburbs. The City of Sydney has a population of approximately $170,000$ and a land area of 6.19 square kilometres. The area is a major financial, commercial and tourist centre with a large transient population and an extensive road network. As a result, the traffic situation is complex, consisting of many urban roads, underground tunnels and highways connecting neighbouring areas. The data we use covers 461 roads in Sydney's local government area, which are 433 km long.

4.2.2.2 North Sydney

North Sydney is a local government area on the Lower North Shore of Sydney. It has a population of $67,658$ and covers an area of approximately 10.9 square kilometres. The area has not only a large number of high-density commercial areas and a large number of complex urban roads but also toll roads and free expressways that connect the rest of Sydney. The North Sydney LGA data covers 156 roads with 166 km total length and 14 suburbs under its jurisdiction.

4.2.2.3 Parramatta

The City of Parramatta spans 84 square kilometres. According to the 2016 census, Parramatta has a population of $226,149$ and contains 38 suburbs within its jurisdiction. This busy area covers several highways that run through the Sydney area and complex urban roads. Parramatta LGA's database includes 222 roads with a total length of 470 km.

4.2.2.4 Suburb selection

For the selection of suburban data, we use the traffic data from the central city of the three LGAs mentioned above as base data for comparison. The study area includes a central business district with multiple high-rise buildings, a mixed commercial and residential high-density region, a relatively low-density residential neighbourhood, and a substantial urban highway network. Even under identical traffic conditions, the transportation networks in the selected areas have different traffic capacities and road conditions due to differences in topographic characteristics.

4.2.3 Evaluation criteria

For comparison, we fit the function using travel times from the traffic database. The exponential function, BPR function and Akcelik function are then compared based on

the fitted results. The aim is to identify and analyze the advantages and disadvantages of each of the functions. To compare with a unified standard, we simplify the three functions in the following way.

$$t(x) = ax^b + c \quad (BPR) \tag{4.6}$$

$$t(x) = c + \frac{3600}{4} a[(x-1) + \sqrt{(x-1)^2 + \frac{8b \cdot x}{\cdot a}}] \quad (Akcelik) \tag{4.7}$$

The coefficient of determination, abbreviated R^2, is the proportion of variation in the dependent variable that the independent variable can predict in statistics. It is a statistical criterion used to predict future outcomes or to evaluate hypotheses based on other data. Based on the fraction of the total variation of effects explained by the model, it measures how well the observed results are represented by the model [Carpenter, 1960, Glantz and Slinker, 2001]. The function fit analysis uses traditional R^2 and root-mean-square error (RMSE). We use the R-squares of different equations in the same region to illustrate how well the function fits.

$$R^2 = 1 - \frac{\sum_{i=0}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{N}(y_i - \overline{y}_i)^2} \tag{4.8}$$

where N is the number of samples, y_i is a dependent variable, $\hat{y}_i$ is the output of the regression model, both indexed by data set i and $\overline{y}_i$ is the mean of the dependent variable. R^2 is always less than or equal to 1; the larger it is, the more the variance of the dependent variable is explained by the regression model.

The RMSE is a commonly used metric for comparing predicted and observed values (sample or population values) by a model or estimator [Li et al., 2014, Xie et al., 2007], and is also one of the parameters commonly used in the transportation field for the degree of fit between functions and data. The square root of the second sample moment of the discrepancies between anticipated and observed values, or the quadratic mean of

these differences, is represented by the RMSE. The RMSE combines the magnitudes of prediction mistakes for different data points into a single measure of predictive capacity. To compare the forecasting mistakes of other models, the RMSE measures accuracy.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{4.9}$$

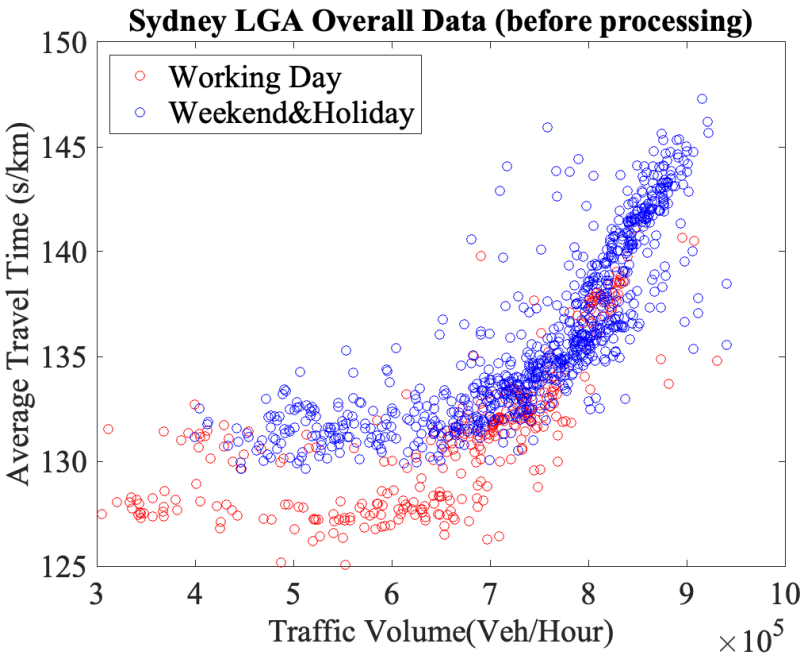
where N is the number of samples, y_i is a dependent variable, $\hat{y}_i$ is the output of the regression model. We use R^2 and RMSE as reference quantities for the degree of fit of the data to the function. The closer the R^2 is to 1, the better the function fits the data, and conversely, the higher the RMSE, the worse the function fits the data.

4.3 Numerical Results

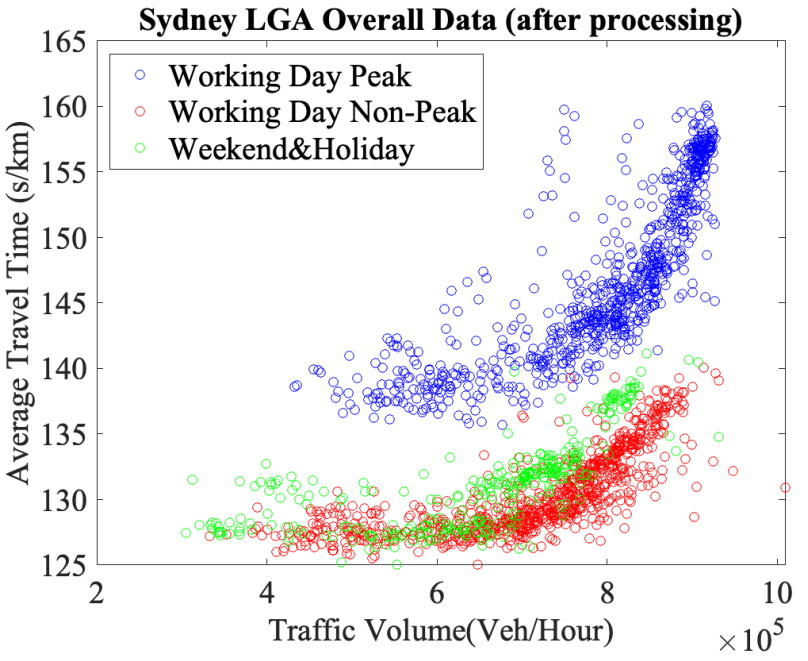
This section analyses the differences between the different cost functions in different regions. Firstly, we perform statistical analysis and function fitting on peak-hour and nonpeak-hour data at the LGA level. Then we narrow down the regions from LGAs to suburbs. The results of our analysis indicate a clear data stratification based on the data set. The phenomenon can be described as a clear regionalization of the data point set, where we cut the Y-axis data within the same X-axis interval into multiple mutually independent fetches. We used data from Sydney as an example to explain data stratification.

We used the average speed of vehicles, measured in different time intervals, as raw data for processing. The average travel time required to complete one kilometre in different traffic flows is obtained using the equation that relates speed to the length of the road.

Fig. 4.1a shows the daily data in Sydney LGA from January 1, 2019, to February 28, 2022. The X-axis represents the traffic flow per hour, while the Y-axis represents the average time (in seconds) that it takes a car to drive one kilometre. It is worth mentioning that each data point represents the traffic volume and average travel time for all



(A) Before processing



(B) After processing

FIGURE 4.1: Daily data in Sydney

traffic in one day. The upper stratification (blue points) consists of weekdays (Monday-Friday), while the lower one (red points) consists of weekends (Saturday-Sunday) and public holidays. Since the data are divided into multiple data point sets, the data can be divided into multiple separate subsets.

Since we can see from Fig. 4.1a that the data representing the working days are still stratified, it is unsuitable for data fitting by a single cost function. Therefore, we split the data into peak and off-peak periods. Fig. 4.1b shows the data points after splitting the data. As can be seen in the figure, there are three clusters. Further analysis of the data reveals that the top group (blue points) is the set of data points during peak hours on weekdays; the bottom group (red points) is the set of data points during off-peak hours on weekdays, and the middle group (green points) is the set of data points on weekends or holidays.

The data in Fig. 4.1 indicates that even for a given value of traffic volume, there can be many different values for the travel time. This is because each data point represents data sampled over an extended period of time, such as 8 hours, over which traffic volume was not constant. Therefore, even if we compare two different time periods with the same average traffic volume, the average travel time may be much higher during one of these periods than during the other because, at some point, during the first period, the traffic density was much higher than in the rest of that period, or than during the other period.

For example, the morning peak period is 7:00-10:00 a.m. Suppose there are 500 vehicles per hour during the 7:00-9:00 a.m. period and 5,000 vehicles per hour during the 9:00-10:00 a.m. period, so on average, there are 2,000 vehicles per hour during the morning peak period. So, since the travel time depends non-linearly on the traffic volume, the average travel time during the entire morning peak period is mainly determined by the traffic between 9:00 and 10:00. Therefore, if we compare this with an off-peak period with the same amount of traffic but where the traffic is more uniformly distributed, we may find a much lower travel time during the off-peak period.

Additionally, traffic control schemes are inconsistent over time, resulting in travel time variations, and some temporary traffic controls may be implemented during off-peak hours. Therefore, we fit the Sydney LGA traffic data into three cases, as mentioned above.

In the rest of this section, we first fit the data for the three LGA peak hours to three different cost-flow functions. Second, we fit the data for the Sydney LGA as an example to distinguish between working off-peak hours and holidays&weekend by comparing the exponential function, the BPR function, and the Akcelik function. Next, we reduce the area size from LGAs to suburbs and use the same approach to fit the data and compare the related functions. Finally, we summarize the results of the data analysis.

4.3.1 Working day peak hours

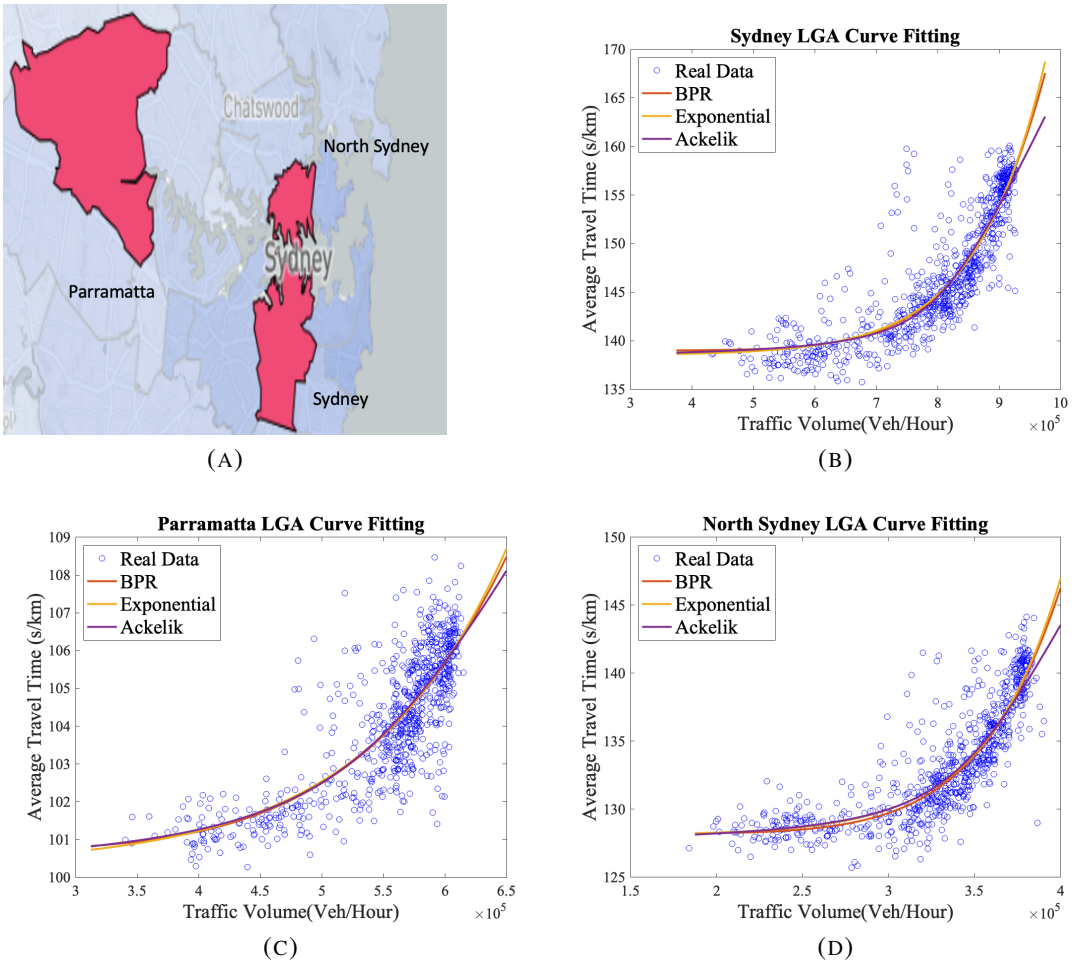


FIGURE 4.2: LGA peak hour results

We first fit the functions using the working-day peak hour data for the three regions (Sydney, Parramatta, and North Sydney) mentioned in Section 4.2. Fig. 4.2 shows how the three cost functions fit the peak hours in these three LGAs. Fig. 4.2(a) shows the

geographic information of the three LGAs. Figs. 4.2(b)-(d) offer the road data fitted by different functions in the Sydney, Parramatta, and North Sydney LGAs, respectively. It can be observed that the fit of the exponential function is very similar to that of the BPR function, but the exponential function has a more considerable gradient change than the BPR function.

TABLE 4.1: Results in LGA Peak Hours

Region	Model	a	b	c	R^2	RMSE
Sydney	BPR	6.494	8.12	139	0.7847	2.971
	Akcelik	55737	60.88	138.4	0.7813	2.994
	Exponential	0.004602	7.325	138.5	0.7856	2.965
Parramatta	BPR	3.463	5.598	100.7	0.6434	1.062
	Akcelik	11501.4	535.5	100.3	0.6438	1.059
	Exponential	0.02734	4.956	100.3	0.6443	1.058
North Sydney	BPR	5.735	8.592	128.2	0.7562	2.088
	Akcelik	14280	371	127.6	0.756	2.089
	Exponential	0.001908	8.049	128.1	0.7568	2.08

Table 4.1 shows the parameters of different functions for different LGAs. The value of c in the table reflects a surprising consistency between different equations in the same region. It can be concluded that the exponential function has the best R^2, and the value of RMSE for the exponential function is the lowest. Since the different distribution of data concentrations in each region can lead to an irregular variation of R^2, we only compare the fit between different functions in the same region rather than performing a uniform analysis of all regions.

Next, we analyze how well the BPR and exponential functions fit the data across various sections of the X-axis. Specifically, we divide the X-axis into three separate regions: low flow $(0 \leq x < 0.5)$, high flow $(0.5 \leq x < 1)$, and overflow $(x > 1)$. Table 4.2 shows the results obtained by partitioning and refining the data and fitting this data to the BPR- and exponential- function, with the same parameters as in Table 4.1. The results show that the exponential function always fits better than the BPR function indicating that the exponential function is better matched to the real data under different flow conditions.

TABLE 4.2: Curve fitting R^2 results for different LGAs

Region	Functions	Low flow $(0 \leq x < 0.5)$	High flow $(0.5 \leq x < 1)$	Overflow $(x > 1)$	Overall
Sydney	BPR	0.2926	0.241	0.5151	0.7847
	Exponential	0.3053	0.2438	0.5161	0.7856
Parramatta	BPR	0.6121	0.4822	0.5846	0.6434
	Exponential	0.6141	0.4838	0.5850	0.6443
North Sydney	BPR	0.2884	0.1755	0.6398	0.7562
	Exponential	0.2979	0.1768	0.6418	0.7568

4.3.2 Working day off-peak hours, and holidays & weekends

Fig. 4.3(a) shows the results of the road data and curve fitting for off-peak working day hours. We see that, compared to the peak period, the variation in the travel time is relatively small during the weekday off-peak period. Fig. 4.3(b) shows the results of the road data and curve fitting for holidays and weekends. Here, we see that for weekends and holidays, the variation in travel time is similar to the peak period.

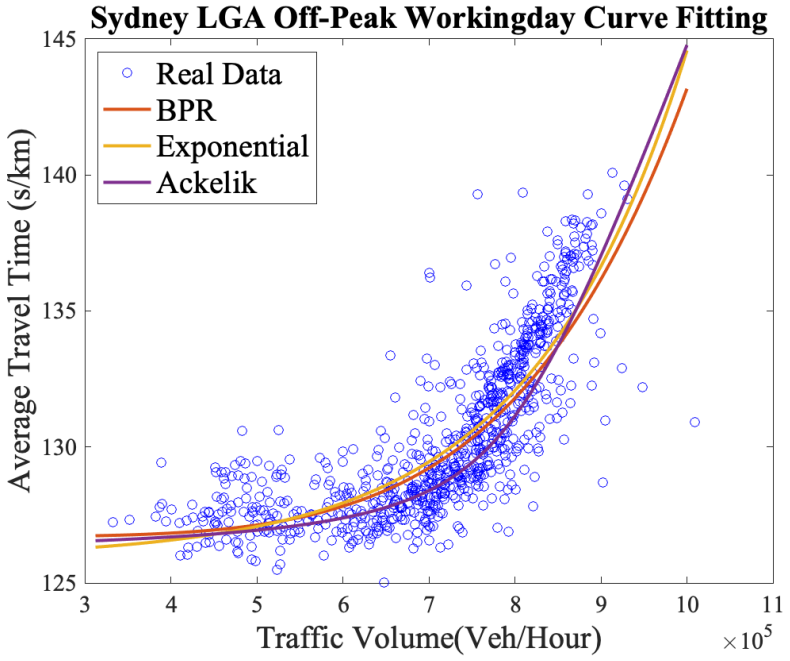
Table 4.3 shows the coefficients and parameters fitted for each function in different cases. In terms of parameters and fits, they are essentially the same as for the data of the LGAs level, except that the Akcelik function fits slightly better than the other two functions on weekends, which is caused by the small sample size of the data.

TABLE 4.3: Results in the Sydney LGA off-peak hour

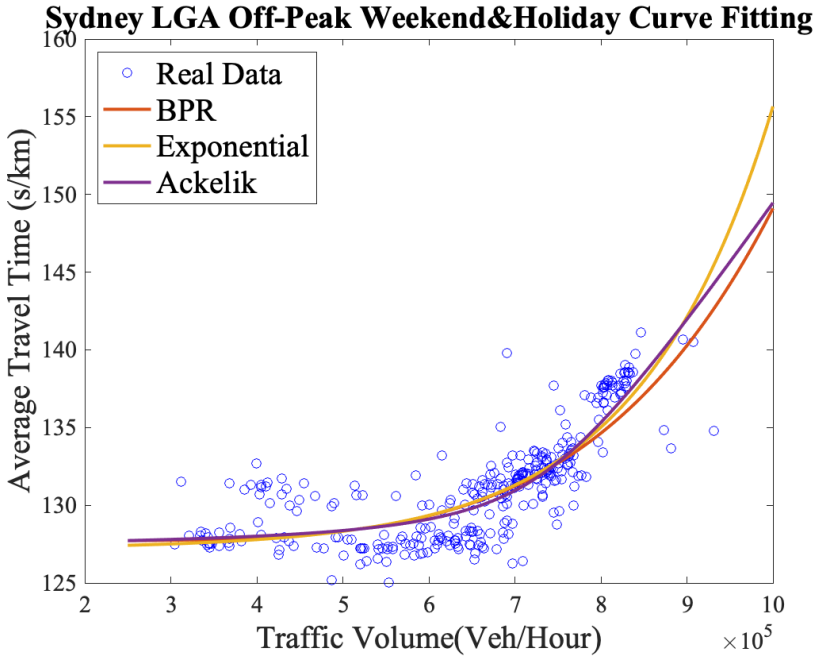
Off-hour	Model	a	b	c	R^2	RMSE
Working day	BPR	5.526	5.26	126.7	0.6666	1.82
	Akcelik	41670	1492	126.7	0.6668	1.809
	Exponential	0.07374	4.497	125.9	0.6672	1.793
Weekend&Holiday	BPR	5.01	5.048	127.7	0.6784	2.002
	Akcelik	30429	2273	127.5	0.6795	1.990
	Exponential	0.04448	4.846	127.2	0.6793	1.992

4.3.3 Peak hours in suburbs

Fig. 4.4 summarizes the fit results in the suburban area. Fig. 4.4(A) shows the geographic information of the three suburban areas. Fig. 4.4(B) displays curve fitting



(A) Working day



(B) Weekend & Holiday

FIGURE 4.3: Sydney LGA off-peak hour fitting results

functions for Sydney suburban road data. Fig. 4.4(C) displays a separate function for the Parramatta suburban road data, and Fig. 4.4(D) displays a different function fit for the North Sydney suburban road data. Each point represents the value of off-peak travel time for weekdays. The distribution of these data is similar to that of the LGAs. The

results are also identical to those obtained in the three LGAs.

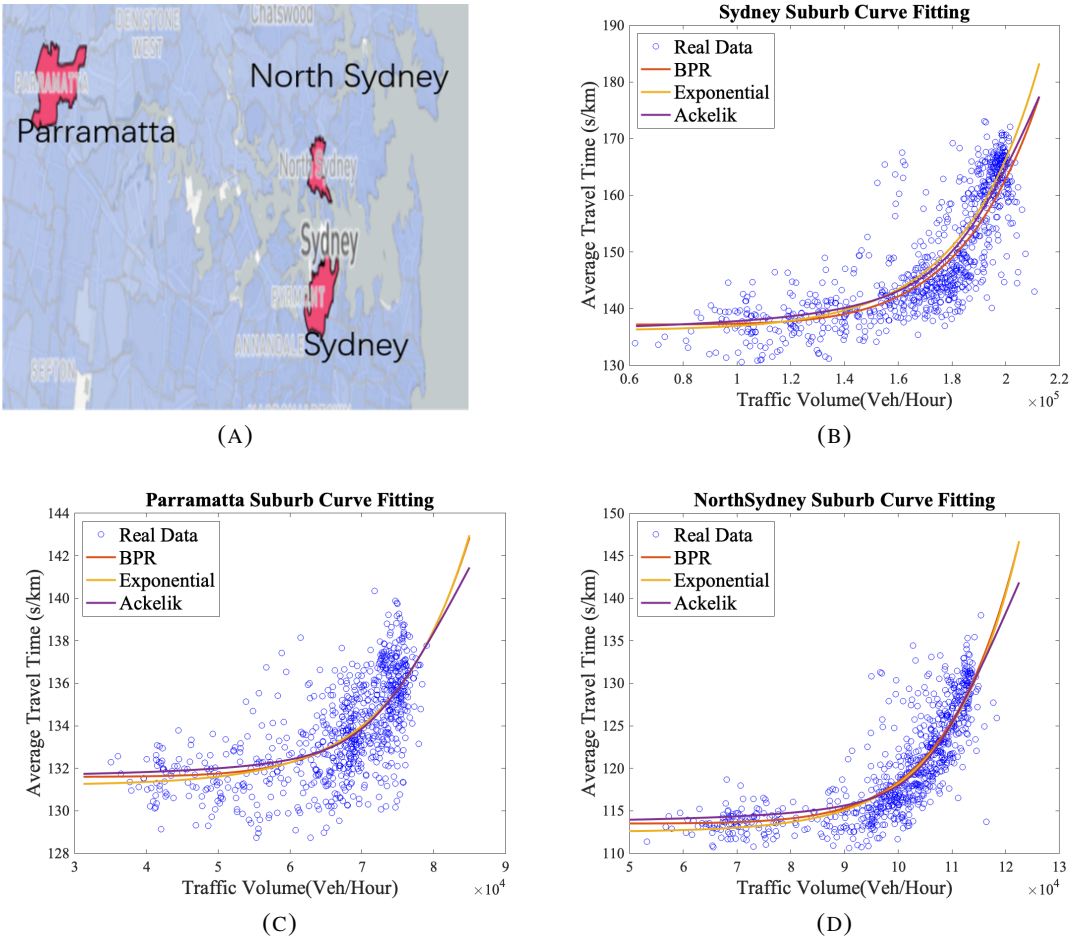


FIGURE 4.4: Peak hour results at the suburb level

Table 4.4 provides detailed information on the function parameters. In the case of parameter c, the value continues to express the travel time under ideal conditions. For the matching results of R^2 and $RMSE$, it can be seen from the data of the three suburbs that the BPR function and the exponential function both fit the data very well. On the contrary, the Akcelik function fits the data relatively less than the other two functions.

Using the information from Tables 4.1, 4.3, and 4.4, we can now formally describe the cost function for a whole region as a piecewise function. Here, we use the cost function for Sydney LGA as an example of a piecewise function shown below.

TABLE 4.4: Results of the suburb peak hour

Suburb	Model	a	b	c	R^2	RMSE
Sydney	BPR	15.94	7.383	137.2	0.6857	5.982
	Akcelik	23922	592.5	136.2	0.6862	5.979
	Exponential	0.0249	6.559	136.1	0.6864	5.973
Parramatta	BPR	3.106	8.1	131.6	0.4501	1.714
	Akcelik	2111.25	29.78	131.6	0.4452	1.772
	Exponential	0.003403	6.95	131.2	0.4505	1.715
North Sydney	BPR	8.682	9.426	113.5	0.6967	3.308
	Akcelik	10221.2	93.04	113.6	0.6973	3.31
	Exponential	0.002008	8.451	112.5	0.6974	3.305

$$t_{sydney}(x) = \begin{cases} 0.004602 \cdot e^{7.325x} + 138.5(s/km) & \text{Working day peak hour} \\ 0.07374 \cdot e^{4.497x} + 125.9(s/km) & \text{Working day non-peak hour} \\ 0.009562 \cdot e^{4.846x} + 127.2(s/km) & \text{Holiday\&weekend} \end{cases} \tag{4.10}$$

4.4 Conclusions

In this chapter, we have proposed using an exponential function as a cost function and compared it to the more commonly used BPR and Akcelik functions. We have shown that, in general, for different regional categories and temporal statistical analysis methods, the exponential function fits slightly better than the most widely used BPR and Akcelik functions. This suggests that the exponential function is a more realistic representation of real-life cost functions. We have compared these cost functions with the support of the real Australian traffic database.

In future work, we plan to use more general regional data to show that the exponential function is more generally applicable than just to Australian traffic conditions. Further theoretical studies on exponential functions in road networks need to be carried out, such as Wardrop's principles, game-theoretical-related properties, and traffic planning problems.

Chapter 5

Price of Anarchy for Traffic Assignment with Exponential Cost Function

The advancement of technologies for connected autonomous vehicles (CAVs) provides great potential for intelligent traffic control and management in the future. The deployment of Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I) and Vehicle-to-Everything (V2X) communications enables traffic control on road segments, intersections or regional road networks with more options, either centralized or decentralized. However, these options are not purely technical considerations but a trade-off between autonomous decision-making and system optimization. An applicable quantitative criterion for such a trade-off is the price of anarchy (PoA) of autonomous decision-making. This chapter analyses the price of anarchy for road networks with autonomous vehicles. Unlike existing research in which the latency function of road congestion was based on BPR or queuing theory, we focus on routing games where the latency of road traffic is expressed as an exponential function.

The rest of this chapter is organized as follows. Section 5.1 introduces some background knowledge and motivation in this chapter. In Section 5.2, we formally present the traffic

assignment model and the related notation. In Section 5.3, we give a brief explanation and theoretical calculations to obtain the expression of the tight upper bound of the PoA for the road network with exponential cost functions. In Section 5.4, we compare the expressions of the tight upper bound of the PoA for the BPR function and the exponential function. Finally, we summarize this chapter and discuss future work in Section 5.5.

5.1 Introduction

The main motivation for our emphasis on autonomous driving is based on the fact that vehicles have more powerful means of communication than humans to access more information about the road, which helps them make more rational decisions. However, we assume that vehicles make decisions regarding which route to take when there are multiple routes that lead from the same origin to the same destination. This is a game-theoretical decision because if one route is shorter and all cars take that route, they only end up going slower due to congestion. Human-driven vehicles cannot have centralized control because it is impossible to force the human driver to make decisions against his or her will. However, with the advent of self-driving vehicles and the gradual improvement of vehicle-to-everything (V2X) technology, it will be possible to control and route vehicles centrally. At that time, the vehicle can be controlled by a central computer to complete the trip. This allows us to study the impact of the social behavior of distributed control and central control.

If all vehicles make their decisions selfishly, traffic will reach a so-called user equilibrium (UE). This is known as Wardrop's first principle [Wardrop, 1952b]. At the user equilibrium state, no vehicle can achieve a shorter travel time by changing its route, implying that all vehicles with the same origin and destination will have exactly the same travel time. The selection of global optimization routes for a road network is called the

system optimum (SO), known as Wardrop's second principle [Wardrop, 1952b], which reflects that the total travel time of all vehicles on the road network is minimized.

This chapter discusses the social aspects of autonomous driving, in particular, the behavior of vehicles either under the assumption of selfish behavior, or under the assumption of centralized control, from a game-theoretical point of view. It is well known that self-decision making of intelligent agents can cause a degradation in the performance of the whole network [Akella et al., 2002, Johari and Tsitsiklis, 2004]. The price of anarchy PoA is a value that quantifies the cost of self-decision making and was first introduced by Koutsoupias and Papadimitriou [1999]. The PoA is defined as the ratio between the UE flow's social cost and the SO flow's social cost in the worst case. In other words, the PoA in traffic assignment is the cost of letting vehicles make individual decisions rather than letting a centralized control mechanism dictate them to achieve global optimization. We can use this concept as a benchmark to change the behavior of vehicles under different traffic conditions. The PoA can be used as an index to determine which traffic situations allow vehicles to make decisions or which traffic situations require centralized control. When the value of PoA is low, the difference between the social cost of distributed control and the social cost of centralized control is small, and the vehicle can be allowed to self-determine at that time. On the contrary, when the value of PoA is high, the cost of self-determination is too high, instead allowing central control to obtain more efficient traffic.

The main objective of this research is to calculate the upper bound of the PoA in a large family of road networks with the same type of cost function [O'Hare et al., 2016a]. This chapter aims to highlight an expression of the tight upper bound of the PoA for traffic assignment with exponential cost functions, and we compare it with the expression of the PoA when using the BPR function. We show that the tight upper bound of the PoA for exponential latency functions depends on the traffic demand. We conclude that vehicles can make decisions by them-self when the traffic volume is low, while when the traffic volume is high, global optimization should be preferred. We present a traffic assignment model with related notations to derive our results. Then, we provide an

expression for the tight upper bound of the PoA, which depends on the traffic demand, for road networks where all cost functions are exponential. Furthermore, we present a simpler expression for this upper bound, which is not tight. Next, we compare the difference between using the BPR function to calculate the PoA and the exponential function to calculate the PoA.

In this chapter, we assume that all vehicles satisfy the following properties.

- Vehicles are fully autonomous, which means that the decision-making of each vehicle is not centralized, but by the vehicle itself, no matter whether humans or computers drive it.
- Vehicles are homogeneous users, which means that vehicle attributes are the same, such as acceleration, body length, and behavior. For example, vehicles in the road network cannot be distinguished as SUVs, sedans or trucks.
- Each vehicle has its origin and destination and is free to choose any available routes to reach its destination.

5.2 Problem Formulation

This section's traffic assignment problem is formalized based on the road network, as mentioned in Chapter 2. It is worth mentioning that both macro-level (Chapter 2.2) and Meso-level(Chapter 2.3) road network models can be used in this chapter. However, to simplify the representation, we only use the macro road network model $G = (N, P)$, where N is a set of positions and P is a set of roads, to formalize the traffic assignment problem in this chapter. Note that we use the symbol P to represent a set of edges instead of E in this section to avoid confusion with Euler's number e.

In any given road network $G = (N, P)$, each road $p \in P$ has a **traffic capacity** $\Phi_p \in \mathbb{R}$, which represents the maximum number of vehicles that can pass the road in an

hour, assuming "*traffic flow is not so great as to cause unreasonable delay, hazard or restriction to the driver*" [Olcott, 1955, Schleicher et al., 2011]. This definition of traffic capacity is also known in the literature as the *practical capacity*.

The traffic assignment problem is formalized as a congestion game, a type of game proposed by Rosenthal [1973]. In a congestion game, each player's payoff depends on the resources it chooses and the number of players choosing the same resource. However, the payoff function is not player specific. Formally, a congestion game is defined as follows:

Definition 5.1. [Rosenthal, 1973]A congestion game is a tuple $(V, R, (S_i)_{i \in V})$, where

- $V = \{1, \ldots, v\}$ is a set of players.
- $R = \{1, \ldots, m\}$ is a finite set of resources.
- Each S_i is the set of pure strategies of player $i \in V$, where each pure strategy $s_i \in S_i$ is a subset of R (i.e. $s_i \subseteq R$).

The cost of each player $i \in V$ is $\pi_i(s) = \sum_{r \in s_i} c_r(s)$, where $s = (s_1, \cdots, s_i, \ldots, s_v)$ is the strategy profile, and $c_r(s) = |\{r \mid \exists i \in V : r \in s_i\}|$ is the number of players that have chosen to use resource r in the strategy profile s.

It is worth mentioning that every congestion game has at least one pure-strategy Nash equilibrium [Rosenthal, 1973].

5.2.1 Origin-destinations and routes

Next, we define $O \subseteq N$ as a set of origin positions and $D \subseteq N$ as a set of destination positions. An origin-destination is a pair of positions that describe the road network's starting and end positions. Formally, we have

Definition 5.2. Given a road network $G = (N, P)$, an **origin-destination** $(o, d) \in O \times D$ is a pair of positions, where $d \neq o$, and $OD = \{(o_1, d_1), \ldots, (o_i, d_i), \ldots, (o_k, d_k)\}$ is a set of all possible origin-destinations on the road network G.

An origin-destination represents the start point and the end point of a vehicle. For example, using a map of Australia, an origin-destination could be Sydney-Melbourne or Sydney-Canberra. Furthermore, we simply refer to an origin-destination (o_i, d_i) as a subscript $i \in [1, k]$.

A route γ, for a given origin-destination (o_i, d_i), is a simple path in the graph (no cycles) that links its origin o_i to its destination d_i. Formally, we have the following definition of a route in a road network.

Definition 5.3. Given a road network $G = (N, P)$. A route γ, for a given origin-destination (o_i, d_i) is a sequence of roads $p_1 \to \cdots \to p_m$ that links origin o_i to destination d_i, where

- $p_x \in P$ for all $1 \leq x \leq m$;
- $p_1 = (o_i, n')$ and $p_m = (n'', d_i)$;
- $p_x \neq p_y$ for any $x \neq y$.
- If $p_{x-1} = (n, n')$, then $p_x = (n', n'')$ for any $1 < x \leq m$, where $n, n', n'' \in N$;

The first condition shows that any edges on a route must exist in the graph, while the second condition defines a route from origin to destination. The following condition is discussed in which a route is a simple path in graph theory. The last condition states that two adjacent edges on a route must be connected. Furthermore, let Γ_i denote all possible routes for (o_i, d_i) and $\Gamma = \bigcup_{i \in [1,k]} \Gamma_i$ define all possible routes according to the given topology of the road network G. We assume that for any $i \in [1, k]$, $\Gamma_i \neq \emptyset$. It is worth mentioning that the road network model in this chapter is a generalized model that can express the road network on a large scale and describe the road network within a specific region.

5.2.2 Vehicles, traffic demand and traffic flow

Vehicles are road users. Each vehicle has its origin and destination to travel by selected route. Formally, we specify the information of vehicles with the following concepts.

Definition 5.4. Given a road network $G = (N, P)$, the information of vehicle v is represented by a tuple (o_v, d_v, γ_v), where

- $o_v \in O$ is the origin position of vehicle v;
- $d_v \in D$ is the destination of vehicle v;
- $\gamma_v \in \Gamma_{(o_v, d_v)}$ is the route selected by vehicle v.

Furthermore, we assume that vehicles do not have special routing preferences except travel time to reach their destination. For simplicity, we regroup all vehicles traveling between the ith origin-destination as a set of vehicles $V_i = \{v : o_v = o_i, d_v = d_i\}$, and define $V = \bigcup_{i \in [0,k]} V_i$ as all vehicles on the road network G. For each origin-destination (o_i, d_i), we define the **traffic demand** $r_i = |V_i|$ to be the total number of vehicles per hour that travel between o_i and d_i.

Definition 5.5. A **traffic flow** $f : \Gamma \to \mathbb{R}^+$ is a function that maps each route γ to a positive number representing the traffic volume of that route, measured in the number of vehicles per hour.

To simplify the notation, we use f_γ as a shorthand for $f(\gamma)$. We say that a flow f is **feasible** if and only if it satisfies $\sum_{\gamma \in \Gamma_i} f(\gamma) = r_i$ for all $i \in [1, k]$, and let F denote all feasible flows. Furthermore, we define $f_p = \sum_{\gamma \in \Gamma : p \in \gamma} f_\gamma$ as the traffic flow of the road p for a feasible flow f.

5.2.3 Cost functions and social cost

Each road has a cost function that takes the traffic rate of that road as its input and outputs the travel time (in seconds) for a vehicle on that road.

Definition 5.6. Given a road network $G = (N, P)$. For each road, $p \in P$, a **cost function** $l_p : \mathbb{R} \to \mathbb{R}$ is a function that takes **traffic flow** as input and outputs the travel time on that road.

For obvious reasons, the cost function should be non-negative, differentiable, and non-decreasing; more traffic leads to more congestion on roads and, thus, higher travel times. We use $\mathcal{L}$ to denote the set of all possible cost functions, and, for some given road network G, we use $l : P \to \mathcal{L}$ to indicate the function that maps each road p to its corresponding cost function l_p. The travel time of a route $\gamma \in \Gamma$ by a feasible traffic flow $f \in F$ can be calculated by the sum of the travel time of the edges contained in the route, indicated by $l_\gamma(f) = \sum_{p \in \Gamma} l_p(f_p)$. The cost of the vehicle is the travel time of the route it has selected. It should be noted that the travel time for each route is the average travel time of all cars that use the route. Furthermore, we define $C(f) = \sum_{\gamma \in \Gamma} l_\gamma(f_\gamma) f_\gamma$ as the **social cost** incurred by feasible flow f, which is the total travel time of the traffic. By adding roads in a route γ and reversing the order of summation, we can rewrite

$$C(f) = \sum_{\gamma \in \Gamma} l_\gamma(f_\gamma) f_\gamma = \sum_{p \in P} l_p(f_p) f_p. \tag{5.1}$$

Definition 5.7. An instance of the traffic assignment problem is now defined as a tuple $(G, \vec{r}, L)$, where

- $G = (N, P)$ is road network.
- $\vec{r} = (r_1, \ldots, r_k)$ is a tuple containing the traffic demand r_i of each origin-destination (o_i, d_i)
- $L : P \to \mathcal{L}$ is a set of cost functions, one for each road p.

5.2.4 User equilibrium

In the field of transportation, *Wardrop's first principle*, also known as *user equilibrium*, has been accepted as a simple and sound conduct principle to explain the distribution

of traffic among alternative routes due to congestion. Traffic flows that adhere to this principle are referred to as user equilibrium flows since each user chooses the best route. Intuitively, each vehicle travels along the route with the minimum travel time. Otherwise, this vehicle would re-select another route with a shorter travel time. The user equilibrium flow is the result of purely self-interested decision-making. Formally, the user equilibrium is defined as follows:

Definition 5.8. [Roughgarden and Tardos, 2002a] A feasible flow for instance $(G, \vec{r}, L)$ is a **user equilibrium** (UE) flow if for all $i \in [1, k]$, all $\gamma_1, \gamma_2 \in \Gamma_i$ with $f_{\gamma_1} > 0$ and all $\eta \in (0, f_{\gamma_1})$, we have $l_{\gamma_1}(f) \leq l_{\gamma_2}(f')$, where

$$f'_\gamma = \begin{cases} f_\gamma - \eta & \text{if} \quad \gamma = \gamma_1 \\ f_\gamma + \eta & \text{if} \quad \gamma = \gamma_2 \\ f_\gamma & \text{if} \quad \gamma \notin \{\gamma_1, \gamma_2\} \end{cases} \tag{5.2}$$

If all traffic is divided over the roads according to a UE flow, each vehicle cannot unilaterally change to a different route to obtain a shorter travel time. Letting η tend to 0, the continuity and strictly monotonically of the cost latency functions give the following practical proposition of a flow at user equilibrium.

Proposition 5.9. *[Roughgarden and Tardos, 2002a] Given an instance $(G, \vec{r}, L)$, a feasible flow $f \in F$ is a user equilibrium (UE) **flow** if and only if for any origin-destination $i \in [1, k]$ and any $\gamma \in \Gamma_i$ with $f_\gamma > 0$, we have $l_\gamma(f_\gamma) \leq l_{\gamma'}(f_{\gamma'})$ for any $\gamma' \in \Gamma_i$.*

This means that when the traffic flow is at UE, for each origin-destination (o_i, d_i), the travel time along each route between o_i and d_i that has a positive traffic flow is the same. Formally, $l_\gamma(f) = l_{\gamma'}(f)$ for all $\gamma, \gamma' \in \Gamma_i$ and $i \in [1, k]$. From existing work, it is known that any traffic assignment problem in the form of a congestion game (See Def.5.1) is a potential game [Monderer and Shapley, 1996], which is a game for which the incentive of all players to change their strategy can be expressed using a single global function called the potential function. It is well-known that for such games, there exists at least one pure strategy user equilibrium [Sandholm, 2010].

Specifically, the user equilibrium flow of an instance is a traffic flow that minimizes the potential function, which can be calculated from a non-linear program [Sandholm, 2001]. Although in this chapter, we will not discuss how to calculate the user equilibrium flow in general. Furthermore, we note that if there exist two or more user equilibrium flows f and f', then $C(f) = C(f')$ for any instance $(G, \vec{r}, L)$.

5.2.5 System optimum

A traffic flow is said to satisfy *Wardrop's second principle* (also known as *System Optimum*) if the average travel time of a feasible flow is minimum, achieving the global optimum of an instance $(G, \vec{r}, L)$. In other words, the SO flow is a feasible flow with minimal social cost $C(SO)$ among all feasible flows. Note that a system optimum can only be reached if all vehicles choose their paths cooperatively to ensure the most efficient utilization of the system as a whole.

Definition 5.10. Given an instance $(G, \vec{r}, L)$, a feasible flow $f^* \in F$ is a **system optimum** (SO) **flow** if and only if $C(f^*) = \min_{f \in F} C(f)$.

To explain how the minimal social cost can be calculated, we need the following definition.

Definition 5.11. For any cost function l its corresponding **marginal cost function** l^* is defined by $l^*(x) := \frac{d}{dx}(x \cdot l(x))$.

It is known from existing research [Beckmann et al., 1956, Dafermos and Sparrow, 1969, Roughgarden and Tardos, 2002a], that for any instance $(G, \vec{r}, L)$ a flow $f^* \in F$ is an SO flow if and only if f^* is the UE flow for the corresponding instance $(G, \vec{r}, L^*)$, where $L^* : P \to \mathcal{L}$ is a function that maps each road p to a cost function l_p^*, which is the marginal cost function corresponding to l_p. Therefore, we can get the SO flow of instance $(G, \vec{r}, L)$ by finding the traffic flow with the minimum value of the potential function, for instance $(G, \vec{r}, L^*)$.

5.2.6 Price of anarchy

It is commonly known that when intelligent agents decide for themselves, the entire network's performance may suffer. The PoA is the ratio between the social cost of the UE flow and the social cost of the SO flow.

Definition 5.12. Given an instance $(G, \vec{r}, L)$, the **price of anarchy** (PoA) is defined as

$$PoA(G, \vec{r}, L) := \frac{C(UE)}{C(SO)}$$

It is worth mentioning that the PoA is a value greater than or equal to 1 because the social cost of user equilibrium is greater than the cost of system optimization. A higher value of the PoA represents a higher cost of self-determination. On the contrary, the more value of the PoA converges to 1, the smaller the difference between the social cost of global optimization and the social cost of the solution that appears from purely self-interested decision-making.

5.3 Price of Anarchy with Exponential Cost Functions

In this section, we derive an expression for an upper bound of the PoA in the case where all cost functions in the road network are exponential and show that this upper bound is tight. Furthermore, we present another upper bound with a simpler expression that is not tight.

5.3.1 Exponential Cost Function

In this chapter, we are interested in instances of the traffic assignment problem in which each road has a cost function with exponential format (see Eq.(4.5)). Specifically, we assume that for each road $p \in P$, the cost function can be expressed as:

$$l_p(f_p) = ae^{bf_p} + c \tag{5.3}$$

We use $\mathcal{L}_{exp}$ to denote the set of all possible cost functions of the form (5.3). Furthermore, we use $(G, \vec{r}, L_{exp})$ to represent an instance with only exponential cost functions, so $L_{exp} : P \to \mathcal{L}_{exp}$ is a function that maps each road to an exponential cost function, and we write l_p instead of $L_{exp}(p)$ in the rest of this chapter.

Furthermore, we note that an instance $(G, \vec{r}, L_{exp})$ with only exponential cost functions has the special property of having a unique user equilibrium solution, as shown in the following theorem.

Lemma 5.13. *For any instance $(G, \vec{r}, L_{exp})$, there is a unique user equilibrium flow.*

Proof. From existing work [Aashtiani and Magnanti, 1981, Konishi, 2004, Milchtaich, 2005], it is known that a unique user equilibrium flow exists for any instance with monotonically increasing cost functions. □

5.3.2 Anarchy value and price of anarchy

This subsection explores finding an upper bound of the PoA for instances with exponential functions. We need to define the concept of the 'anarchy value' for each road. The idea of anarchy value was proposed in [Roughgarden, 2003]. The motivation to define the anarchy value is to find the worst-case ratio between the cost of UE flow and SO flow for a given set of cost functions.

We should stress here, however, that our definition of 'anarchy value' is different from the original one. The first main difference is that our definition depends on the traffic demand $\vec{r}$ of the instance, while Roughgarden's original definition took the supremum over all possible values of the traffic demand. The motivation for this difference is that

otherwise, the upper bound of the PoA would go to infinity in the case of exponential cost functions. Secondly, our definition assumes that traffic is at user equilibrium. Formally, our definition of anarchy value is as follows.

Definition 5.14. Let $(G, \vec{r}, L_{exp})$ be an instance with exponential cost functions, and let f denote its UE flow. Then the **anarchy value** $\phi_p(\vec{r})$ of a road p is defined as follows:

$$\phi_p(\vec{r}) := [\lambda_p \mu_p + (1 - \lambda_p)]^{-1} \tag{5.4}$$

where $\lambda_p \in [0, 1]$ is the solution of the equation $l_p^*(\lambda_p f_p) = l_p(f_p)$, and μ_p is defined as $\mu_p := \frac{l_p(\lambda_p f_p)}{l_p(f_p)} \in [0, 1]$.

Note that the user equilibrium flow f depends on the traffic demand $\vec{r}$, so λ_p and μ_p depend on $\vec{r}$, and therefore ϕ_p also depends on $\vec{r}$. We need the following lemma to show that the anarchy value is well-defined.

Lemma 5.15. *For any function l of the form of Eq.(5.3) and for any positive value $x \in \mathbb{R}^+$, there is a unique value $\lambda \in [0, 1]$ that solves the equation $l^*(\lambda x) = l(x)$ (where $l^*(\cdot)$ is the marginal cost function of $l(\cdot)$, see Def. 5.11)*

Proof. Secondly, we note that l and l^* are monotonically increasing on the domain $x > 0$. The idea is then to show that $l^*(0) = l(0) \leq l(x) \leq l^*(x)$ is true for any positive x. It is easy to see that, indeed, we have $l^*(0) = a + c = l(0)$. Furthermore, since l is monotonically increasing, we have $l(0) \leq l(x)$. And finally, since a, b and x are all non-negative, we have:

$$l(x) \quad = \quad ae^{bx} + c \quad \leq \quad ae^{bx} + abxe^{bx} + c \quad = \quad l^*(x)$$

Since l and l^* are monotonically increasing and $l^*(0) = l(0)$, it is now easy to see that there exists a unique value $x' \leq x$ such that $l^*(x') = l(x)$. Then, we can simply define $\lambda := \frac{x'}{x} \in [0, 1]$, so that indeed we have $l^*(\lambda x) = l(x)$. □

This lemma shows that λ_p and μ_p of Def. 5.14 are well-defined. We now define $\phi(L_{exp}) := \max_{p \in P} \phi_p(\vec{r})$ as the anarchy value of the instance $(G, \vec{r}, L_{exp})$. Next, we need the following lemmas from [Roughgarden, 2003] to derive the main results of this chapter.

Lemma 5.16. *[Roughgarden, 2003]Let f and f^* be UE flow and SO flow, respectively, for instance $(G, \vec{r}, L_{exp})$. For a road p, let $\lambda_p \in [0, 1]$ solve $l_p^*(\lambda_p f_p) = l_p(f_p)$. Then,*

$$C(f) \geq \sum_{p \in P} [l_p(\lambda_p f_p) \cdot \lambda_p f_p + (f_p^* - \lambda_p f_p) l_p(f_p)]$$

Proof. See page 11 in [Roughgarden, 2003] □

Lemma 5.17. *[Roughgarden, 2003]Let f be UE flow and f^* be a feasible flow for instance $(G, \vec{r}, L_{exp})$. Then,*

$$\sum_{p \in P} l_p(f_p) f_p \leq \sum_{p \in P} l_p(f_p) f_p^*$$

Proof. See page 12 in [Roughgarden, 2003] □

With all of the preliminaries now in place, we can show the relationship between the anarchy value and the PoA of an instance $(G, \vec{r}, L_{exp})$ in the following Lemma. It is worth mentioning that this Lemma is largely copied from [Roughgarden, 2003] with only a small adaption for my alternative definition of 'anarchy value'.

Lemma 5.18. *For any instance $(G, \vec{r}, L_{exp})$, we have:*

$$PoA(G, \vec{r}, L_{exp}) \leq \phi(L_{exp})$$

Proof. Let f and f^* be user equilibrium flow and system optimum flow, respectively, for the given instance $(G, \vec{r}, L_{exp})$. By combing the lemma 5.16 and lemma 5.17, it is easy to rewrite the social cost of SO in a form that is easier to relate to the social cost of UE as follows:

$$\begin{aligned} C(f^*) &\geq \sum_{p\in P}[l_p(\lambda_p f_p)\lambda_p f_p + (f_p - \lambda_p f_p)l_p(f_p)] \\ &\geq \sum_{p\in P}[\mu_p\lambda_p + (1-\lambda_p)]l_p(f_p)f_p \end{aligned} \tag{5.5}$$

with λ_p, μ_p and f_p as in Def. 5.14. Thus, we can rewrite this inequality as follows.

$$\begin{aligned} C(f^*) &\geq \sum_{p\in P}\frac{l_p(f_p)f_p}{[\mu_p\lambda_p + (1-\lambda_p)]^{-1}} = \sum_{p\in P}\frac{l_p(f_p)f_p}{\phi_p(\vec{r})} \\ &\geq \frac{\sum_{p\in P} l_p(f_p)f_p}{\max_{p\in P}\phi_p(\vec{r})} = \frac{C(f)}{\phi(L_{exp})} \end{aligned} \tag{5.6}$$

Eq. (5.6) leads to

$$PoA(G,\vec{r},L_{exp}) = \frac{C(f)}{C(f^*)} \leq \phi(L_{exp}) \tag{5.7}$$

□

From Lemma 5.18, we see that the PoA of any instance is always less than or equal to the maximum anarchy value among all roads in that instance.

5.3.3 The Lambert W function

Before continuing with the rest of this chapter, we here need to briefly discuss the Lambert W function [Corless et al., 1996]. For any positive real number x, the value $W(x) \in \mathbb{R}$ is defined as the unique real number that satisfies:[1]

$$W(x)e^{W(x)} = x \tag{5.8}$$

The Lambert W function is monotonically increasing and satisfies the following well-known properties [Hoorfar and Hassani, 2007, Weisstein, 2002], which will be useful

[1]For real numbers $x \in [-\frac{1}{e}, 0]$ this equation has two solutions, which are denoted as $W_{-1}(x)$ and $W_0(x)$ respectively. Still, because we are not interested in a negative value of x (not only in this chapter), we are not interested in such values.

to us later on:

$$W(e) = 1 \tag{5.9}$$

$$\frac{d}{dx}W(x) = \frac{W(x)}{x \cdot (1 + W(x))} \tag{5.10}$$

For any $x \geq e$:

$$log(x) - loglog(x) \;\leq\; W(x) \;\leq\; log(x) - \frac{1}{2}loglog(x) \tag{5.11}$$

5.3.4 Upper bound of the PoA

At this point, we have defined all the concepts required to calculate our upper bound of the PoA on the set of all instances $(G, \vec{r}, L_{exp})$ with exponential cost functions. We present this upper bound below, in Theorem 5.22, which depends on two more lemmas.

Lemma 5.19. *For any instance $(G, \vec{r}, L_{exp})$, let f denote its UE flow and $x = f_p$ is the UE flow of the road p. The anarchy value $\phi_p(\vec{r})$ of any road $p \in P$ with cost functions of the form $l(x) = ae^{bx} + c$, where a, b, and c are non-negative coefficients, satisfies the following:*

$$\phi_p(\vec{r}) \leq \frac{bx}{bx + 2 - W(e^{bx+1}) - \frac{1}{W(e^{bx+1})}} \tag{5.12}$$

where $W(\cdot)$ is the Lambert W function.

Proof. Recall that the anarchy value is defined as $\phi(\vec{r}) = [\lambda\mu + (1 - \lambda)]^{-1}$, and to calculate λ we have to solve $l^*(\lambda x) = l(x)$. That is, we have to solve:

$$ae^{\lambda bx} + a\lambda bx \cdot e^{\lambda bx} + c = ae^{bx} + c \tag{5.13}$$

Subtracting c from both sides and then dividing by a into both sides, we get the following.

$$\begin{aligned} e^{\lambda bx} + \lambda bx \cdot e^{\lambda bx} &= e^{bx} \\ (1 + \lambda bx) \cdot e^{\lambda bx} &= e^{bx} \\ (1 + \lambda bx) \cdot e^{\lambda bx+1} &= e^{bx+1} \end{aligned}$$

Next, if we replace $(\lambda bx + 1)$ by δ, then we get $\delta \cdot e^{\delta} = e^{bx+1}$, which can be solved using the Lambert W function. This gives us: $\delta = W(e^{bx+1})$. Now, if we substitute back $\lambda bx + 1$ for δ, and solve for λ, then we get:

$$\lambda = \frac{W(e^{bx+1}) - 1}{bx} \tag{5.14}$$

Next, recall that μ was defined as $\frac{l(\lambda x)}{l(x)}$, so we have:

$$\mu = \frac{ae^{\lambda bx} + c}{ae^{bx} + c}$$

We note that μ therefore satisfies the following:

$$\mu \geq \frac{ae^{\lambda bx}}{ae^{bx}} = e^{(\lambda-1)bx} \tag{5.15}$$

Plugging these expressions (5.14) and (5.15) for λ and μ back into $[\lambda\mu + (1-\lambda)]^{-1}$, we obtain:

$$\begin{aligned} &\phi(\vec{r}) = [\lambda\mu + (1-\lambda)]^{-1} \\ &\leq [\frac{W(e^{bx+1}) - 1}{bx} \cdot e^{(\frac{W(e^{bx+1})-1}{bx} - 1) \cdot bx} + 1 - \frac{W(e^{bx+1}) - 1}{bx}]^{-1} \\ &= [\frac{W(e^{bx+1}) - 1}{bx} \cdot e^{(W(e^{bx+1})-1-bx)} + \frac{bx}{bx} - \frac{W(e^{bx+1}) - 1}{bx}]^{-1} \\ &= \frac{bx}{(W(e^{bx+1}) - 1) \cdot e^{(W(e^{bx+1})-1-bx)} + bx - (W(e^{bx+1}) - 1)} \end{aligned}$$

We see from Eq. (5.8) that for any $x > 0$ we have $e^{W(x)} = \frac{x}{W(x)}$, so we have

$$e^{W(e^{bx+1})-1-bx} = \frac{e^{bx+1}}{W(e^{bx+1})} \cdot e^{-1-bx} = \frac{1}{W(e^{bx+1})}$$

From this, we get the following:

$$\begin{aligned}\phi(\vec{r}) \leq & \frac{bx}{(W(e^{bx+1})-1)\cdot\frac{1}{W(e^{bx+1})}+bx-(W(e^{bx+1})-1)} \\ = & \frac{bx}{bx+2-W(e^{bx+1})-\frac{1}{W(e^{bx+1})}}\end{aligned} \tag{5.16}$$

□

This lemma shows an upper bound of the anarchy value for a road p. The following two lemmas give us a better idea of how the expression at the end of Eq.(5.16) behaves.

Lemma 5.20. *The following equation holds:* $\lim_{x\to 0}\frac{x}{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}} = 1$

Proof. Instead of proving this directly, we will prove the 'reversed' equation, which is equivalent:

$$\lim_{x\to 0}\frac{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}}{x} = 1 \tag{5.17}$$

This means we need to prove the following:

$$\lim_{x\to 0}\frac{2-W(e^{x+1})-\frac{1}{W(e^{x+1})}}{x} = 0 \tag{5.18}$$

Now, note that since $\lim_{x\to 0} W(e^{x+1}) = 1$, we can multiply this by $W(e^{x+1})$. So, equivalently we can prove the following:

$$\lim_{x\to 0}\frac{2\cdot W(e^{x+1})-W(e^{x+1})^2-1}{x} = 0 \tag{5.19}$$

Note that, in this equation, the limit of both the numerator and the denominator is 0, so we can calculate this limit using L'Hopital's rule. Indeed, using equation 5.10 we get:

$$\begin{aligned}
\lim_{x\to 0}\frac{2W(e^{x+1})-W(e^{x+1})^2-1}{x} &= \lim_{x\to 0}\frac{\frac{d}{dx}(2W(e^{x+1})-W(e^{x+1})^2-1)}{\frac{d}{dx}x}\\
&= \lim_{x\to 0}\frac{\frac{d}{dx}(2W(e^{x+1})-W(e^{x+1})^2-1)}{1}\\
&= \lim_{x\to 0}\frac{d}{dx}(2W(e^{x+1})-W(e^{x+1})^2-1)\\
&= \lim_{x\to 0}\frac{2W(e^{x+1})}{W(e^{x+1})+1}-\frac{2W(e^{x+1})^2}{W(e^{x+1})+1}\\
&= 0
\end{aligned} \tag{5.20}$$

□

Lemma 5.21. *The expression* $\frac{x}{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}}$ *is monotonically increasing for* $x > 0$.

Proof. To prove this, we calculate its derivative and show that it is non-negative for any positive x.

$$\begin{aligned}
&\frac{d}{dx}\frac{x}{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}}\\
&=\frac{(x+1-W(e^{x+1}))\cdot(W(e^{x+1})-1)\cdot W(e^{x+1})}{(W(e^{x+1})^2-(x+2)\cdot W(e^{x+1})+1)^2}
\end{aligned} \tag{5.21}$$

It is easy to see that the denominator is non-negative. For the numerator, we divide into three parts, $x+1-W(e^{x+1})$, $W(e^{x+1})-1$, and $W(e^{x+1})$. Since the Lambert W function is monotonically increasing and $W(e)=1$, we can easily see that $W(e^{x+1})-1$ and $W(e^{x+1})$ are both non-negative. For the remaining expression, we note that it is 0 when $x=0$, and that (by Eq.(5.10)) we have

$$\frac{d}{dx}(x+1-W(e^{x+1}))=\frac{1}{W(e^{x+1})+1}\geq 0$$

From this, it follows that $x + 1 - W(e^{x+1})$ is also non-negative for any positive value of x. □

The following Theorem gives us an upper bound for the PoA of any instance $(G, \vec{r}, L_{exp})$ with exponential cost functions. We first need to introduce the following notation: $\hat{r} := \sum_{i\in[1,k]} r_i$ and $\hat{b} := \max_{p\in P} b_p$.

Theorem 5.22. *For any instance $(G, \vec{r}, L_{exp})$ with exponential cost functions, the price of anarchy satisfies the following.*

$$PoA(G, \vec{r}, L_{exp}) \leq \frac{\hat{b}\hat{r}}{\hat{b}\hat{r} + 2 - W(e^{\hat{b}\hat{r}+1}) - \frac{1}{W(e^{\hat{b}\hat{r}+1})}} \tag{5.22}$$

Proof. We know from Lemma 5.18 that:

$$PoA(G, \vec{r}, L_{exp}) \leq \phi(L_{exp}) = \max_{p\in P} \phi_p(\vec{r})$$

Then we know from Lemma 5.3.4 that $\phi_p(\vec{r})$ can be replaced by the right-hand side of Eq. (5.12). Furthermore, thanks to Lemma 5.21 and the fact that $f_p \leq \hat{r}$ we can replace f_p by $\hat{r}$. Similarly, again due to Lemma 5.21 and the fact that $b_p \leq \hat{b}$, we can remove the maximization over $p \in P$ and instead replace b_p by $\hat{b}$. Hence, we obtain Eq. (5.22). □

5.3.5 Tightness of the upper bound

The following Theorem shows that the expression we presented in Theorem 5.22 is, in fact, a *tight* upper bound for the set of all instances with exponential functions.

Theorem 5.23. *For any positive numbers $\hat{b}$ and $\hat{r}$, there exists an instance $(G, \vec{r}, L_{exp})$ with exponential cost functions, for which $PoA(G, \vec{r}, L_{exp})$ is exactly equal to*

$$\frac{\hat{b}\hat{r}}{\hat{b}\hat{r} + 2 - W(e^{\hat{b}\hat{r}+1}) - \frac{1}{W(e^{\hat{b}\hat{r}+1})}} \tag{5.23}$$

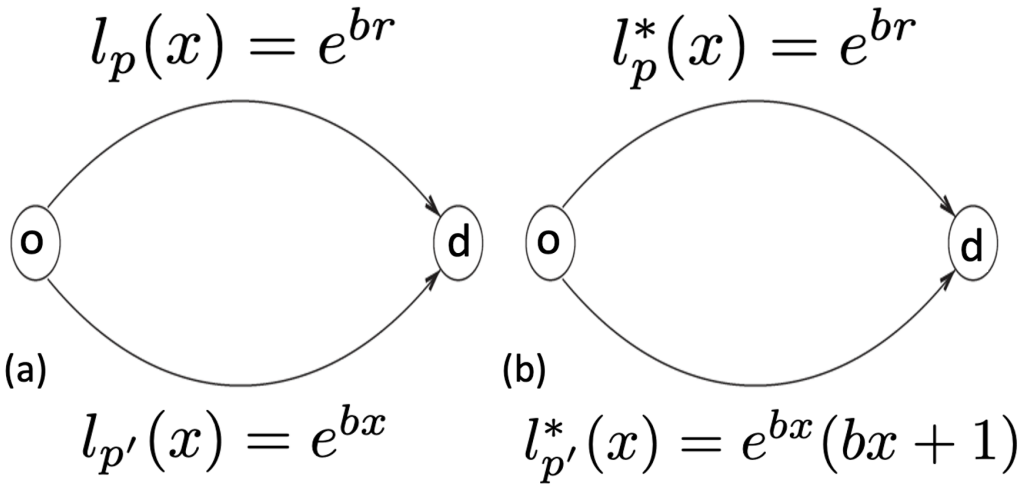


FIGURE 5.1: A variation of Pigou's example

Proof. We prove this theorem using a simple road network which is an adaptation of a road network known as *Pigou's example* [Pigou and Aslanbeigui, 2017], displayed in Fig 5.1. The network (a) in this figure is the network for which we will calculate the PoA. We can choose any arbitrary number r as the traffic demand for this network (that is, $\vec{r} = (r)$). The network (b) is the same, but with the cost, functions replaced by their corresponding marginal cost functions. To calculate the PoA of the network (a), we first need to calculate its UE flow. As explained above, to do this, we need to set the latency $l_{p'}$ of the lower road p' equal to the latency l_p of the upper road p. That is, we need to solve the equation $e^{br} = e^{bx}$, which leads to $x = f_{p'} = r$. This means that all traffic is choosing the lower road p'. The total cost $C(UE)$ is then given by:

$$C(UE) = f_{p'} \cdot l_{p'}(f_{p'}) = r \cdot l_{p'}(r) = r \cdot e^{br}$$

Next, we need to calculate the SO flow for network (a), which equals the UE flow of network (b). Therefore, we need to set the latency of the two roads in the network (b) equal. That is, we need to solve: $e^{br} = e^{bx}(bx+1)$, for which the solution is given by: $x = \frac{W(e^{br+1})-1}{b}$.This means that in SO, the total number of vehicles choosing the lower road is given by $f_{p'} = \frac{W(e^{br+1})-1}{b}$ and the total number of vehicles choosing the upper

road is $f_p = r - \frac{W(e^{br+1})-1}{b}$. The total cost of $C(SO)$ for all vehicles is then given by:

$$\begin{aligned} C(SO) &= f_p \cdot l_p(f_p) + f_{p'} \cdot l_{p'}(f_{p'}) \\ &= \frac{W(e^{br+1})-1}{b} \cdot e^{W(e^{br+1})-1} + (r - \frac{W(e^{br+1})-1}{b}) \cdot e^{br} \end{aligned}$$

where p is the upper road of the network and p' is the lower road. Combining these two results, we get:

$$\begin{aligned} PoA &= \frac{r \cdot e^{br}}{\frac{W(e^{br+1})-1}{b} \cdot e^{W(e^{br+1})-1} + (r - \frac{W(e^{br+1})-1}{b}) \cdot e^{br}} \\ &= \frac{br \cdot e^{br}}{(W(e^{br+1})-1) \cdot e^{W(e^{br+1})-1} + (br - W(e^{br+1}) + 1) \cdot e^{br}} \end{aligned}$$

From Eq. (5.8) we see that for any $x > 0$ we have $e^{W(x)} = \frac{x}{W(x)}$, so we have

$$e^{W(e^{br+1})-1} = \frac{e^{br+1}}{e \cdot W(e^{br+1})} = \frac{e^{br}}{W(e^{br+1})} \tag{5.24}$$

.

Using this, we get the following:

$$\begin{aligned} PoA &= \frac{br \cdot e^{br}}{(W(e^{br+1})-1) \cdot \frac{e^{br}}{W(e^{br+1})} + (br - W(e^{br+1}) + 1) \cdot e^{br}} \\ &= \frac{br}{(W(e^{br+1})-1) \cdot \frac{1}{W(e^{br+1})} + (br - W(e^{br+1}) + 1)} \\ &= \frac{br}{br + 2 - W(e^{br+1}) - \frac{1}{W(e^{br+1})}} \end{aligned}$$

Finally, since this example only had one value b_p and one value r_i, we have $\hat{b} = b$ and $\hat{r} = r$, so we indeed have obtained the expression mentioned in the theorem. □

5.3.6 Alternative upper bound

The tight upper bound we presented in Theorem 5.22 has a rather complex expression. Therefore, we will now derive two simpler expressions that form a lower- and upper-bound for this expression. It shows that the original function grows less than linearly as a function of $\hat{b}$ and $\hat{r}$.

Lemma 5.24. *For any non-negative x, we have*

$$\frac{x}{log(x+1)} \leq \frac{x}{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}} \leq \frac{2x}{log(x+1)}$$

Proof. Since x is non-negative we have $e^{x+1} \geq e$, so we can substitute e^{x+1} for x in Eq. (5.11) and obtain:

$$x+1-log(x+1) \quad \leq \quad W(e^{x+1}) \quad \leq \quad x+1-\frac{1}{2}log(x+1)$$

From this, we get the following:

$$\begin{aligned}\frac{x}{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}} &\leq \frac{x}{\frac{1}{2}log(x+1)+1-\frac{1}{x+1-log(x+1)}} \\ &\leq \frac{2x}{log(x+1)}\end{aligned} \tag{5.25}$$

where we used the fact that $\frac{1}{x+1-log(x+1)} < 1$, and we get:

$$\begin{aligned}\frac{x}{x+2-W(e^{x+1})-\frac{1}{W(e^{x+1})}} &\geq \frac{x}{log(x+1)+1-\frac{1}{x+1-\frac{1}{2}log(x+1)}} \\ &\geq \frac{x}{log(x+1)}\end{aligned} \tag{5.26}$$

where we used the fact that $\frac{1}{x+1-\frac{1}{2}log(x+1)} < 1$. □

It is worth mentioning that the functions in Eq. (5.25) and Eq. (5.26) are monotonically increasing for non-negative x. From Lemma 5.24, we can easily get the following theorem.

Theorem 5.25. *For any instance $(G, \vec{r}, L_{exp})$ with exponential cost functions, its price of anarchy satisfies:*

$$PoA(G, \vec{r}, L_{exp}) \leq \frac{2\hat{b}\hat{r}}{log(\hat{b}\hat{r}+1)}$$

Proof. This is simply a combination of Theorem 5.22 and Lemma 5.24 □

5.4 Comparing with the Upper Bound for BPR functions

In this section, we claim that the PoA is lower under realistic circumstances when roads have exponential cost functions than BPR cost functions. We assume this value is the same for every road p, denoted by Φ. Note that by the definition of traffic capacity, in the real world, the actual traffic flow cannot exceed the traffic capacity by much.

As explained above, the most common cost function used in the literature is the Bureau of Public Roads (BPR) function [Bureau of Public Roads, 1964], which has the form

$$l(f) = t_0(1 + m \cdot (\frac{f}{\Phi})^n). \tag{5.27}$$

where t_0 is the free-flow travel time. It is known from the literature [Roughgarden, 2003] that a tight upper bound $\hat{PoA}_{BPR}$ for the PoA over the set of all instances where the cost functions are BPR functions with the degree at most $\hat{n}$ is given by:

$$\hat{PoA}_{BPR} = (1 - \hat{n}(\hat{n}+1)^{-\frac{\hat{n}+1}{\hat{n}}})^{-1} \tag{5.28}$$

In the next section, we will show that if we try to model the true cost functions of a real-world road network using BPR functions or by using exponential cost functions, then the respective values of Φ, $\hat{b}$, and $\hat{n}$ that we get will typically satisfy $\Phi \cdot \hat{b} \leq \hat{n}$, which is an essential assumption for the rest of this section.

We are now ready to state our main claim in this section. A formal proof of this conjecture is left as future work.

Conjecture 5.26. Given a road network G and any traffic demand $\vec{r}$, we have $(G, \vec{r}, L_{BPR})$ and $(G, \vec{r}, L_{exp})$. If $\Phi \cdot \hat{b} \leq \hat{n}$ and $f_p^* \leq \Phi$ for all $p \in P$ (where f is the equilibrium flow of the instance with exponential functions), then $PoA(G, \vec{r}, L_{exp}) \leq \hat{PoA}_{BPR}$.

Proof Idea: We know from Lemma 5.18 that:

$$PoA(G, \vec{r}, L_{exp}) \leq \phi(L_{exp}) = \max_{p \in P} \phi_p(\vec{r})$$

Combining this with Eq.(5.12) and Lemma 5.21, we get

$$PoA(G, \vec{r}, L_{exp}) \leq \frac{\beta}{\beta + 2 - W(e^{\beta+1}) - \frac{1}{W(e^{\beta+1})}}$$

where $\beta = \max_{p \in P} b_p f_p^*$, and from Eq.(5.28 we know that

$$\hat{PoA}_{BPR} \leq (1 - \hat{n}(\hat{n}+1)^{-\frac{\hat{n}+1}{\hat{n}}})^{-1}$$

Due to the assumption that $f_p \leq \Phi$, we have $b_p \cdot f_p \leq b_p \Phi$ and therefore $\beta \leq \hat{b} \cdot \Phi$. Then, since we also assumed that $\Phi \cdot \hat{b} \leq \hat{n}$ we have $\beta \leq \hat{n}$. So, by Lemma 5.21 we have

$$\frac{\beta}{\beta + 2 - W(e^{\beta+1}) - \frac{1}{W(e^{\beta+1})}} \leq \frac{\hat{n}}{\hat{n} + 2 - W(e^{\hat{n}+1}) - \frac{1}{W(e^{\hat{n}+1})}} \tag{5.29}$$

We then only need to show that, for any positive $\hat{n}$, we have

$$(1-\hat{n}(\hat{n}+1)^{-\frac{\hat{n}+1}{\hat{n}}})^{-1} - \frac{\hat{n}}{\hat{n}+2-W(e^{\hat{n}+1})-\frac{1}{W(e^{\hat{n}+1})}} \geq 0 \tag{5.30}$$

Rather than formally proving this inequality, we argue that it is true by showing a plot of the left-hand side of Eq. (5.30). This plot is displayed in Fig 5.2. The graph on the left shows a range of $\hat{n}$-values of $0-10$, and the graph on the right shows a range of $\hat{n}$-values of $0-100$.

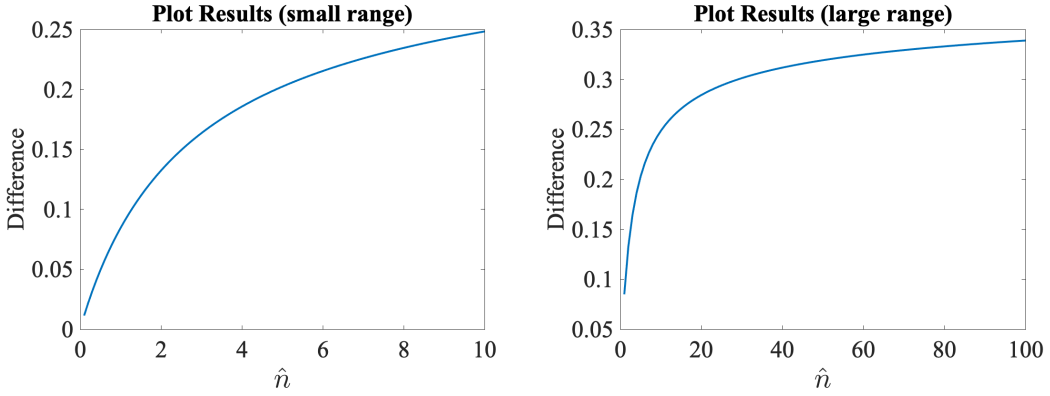


FIGURE 5.2: Plot Results for Conjecture 5.26

5.5 Summary

Traffic assignment with selfish routing is inefficient because it generally does not achieve the optimal solution that could be achieved if all vehicles cooperated. The price of anarchy describes this inefficiency. The traditional approach to calculating the upper bound of the PoA upper bound uses different cost functions, such as linear function and polynomial function, for simplification. The resulting expression for the upper bound of the PoA depends on the exponent of the cost functions.

This chapter focused on the tight upper bound of the PoA in road networks using exponential cost functions and discussed the changes in the tight upper bound due to changes in traffic demand. For realistic cases, traffic demand cannot be much greater

than capacity, so we compare the trend of the PoA as a function of traffic demand when the road network topology is fixed, with the cost functions being exponential or BPR functions. And the results show that when the traffic rate is lower than capacity, using the exponential cost function yields a tight upper bound for the PoA that is lower than the BPR cost function. The conjecture mentioned in this chapter is not mathematically proven, and the more formal theoretical proof is an aspect of future work. Second, extending the exponential function to a wider range of application scenarios is the focus of future work. In addition, more research on PoA can be considered instead of simply considering traffic scenarios.

Chapter 6

A Hybrid Model of Traffic Assignment and Control for Autonomous Vehicles

In the previous chapters, we introduced road networks, the traffic assignment problem, and traffic management protocols. Autonomous vehicles in our model are assumed to make self-interested decisions, like human drivers, who only consider their interests, such as how to reach their destination in the shortest possible time. This chapter proposes a multi-agent based method to describe traffic control optimization for autonomous vehicle assignment problems on road networks. We extend the road network model into a game-theoretical model based on population games to describe the behaviour of autonomous vehicles under intelligent traffic control. Based on this model, we investigate a traffic control optimization problem that aims to improve the efficiency of road networks and provides an algorithm to find an approximate solution.

6.1 Introduction

Traditional traffic control systems continue to suffer from several well-known flaws [Djahel et al., 2014]. Current traffic control systems show ineffective time management at

intersections, requiring vehicles to wait unnecessarily, resulting in congestion, pollution, and additional delays, among other things. Americans travel nearly 3 trillion miles a year, meaning the number of hours spent in traffic worldwide increases dramatically. With more than 3,000 people dying daily, driving can be dangerous, and most accidents result from human error [Administration et al., 2016].

The most common traffic control protocols on major roads are based on traffic lights. The first versions of these traffic signal systems assigned a fixed time for each traffic light to turn green, regardless of the number of vehicles or traffic density in the corresponding lane. However, as technology advanced, these traffic signal systems started to consider different parameters, such as a distinction between day and night or between peak and off-peak periods, to determine the ratio between time in green and time in red. Some vision-based traffic management systems [Esteve et al., 2007, Javaid et al., 2018, Reza et al., 2021] use vision sensors to capture the flow of cars from different directions. To the best of our knowledge, these advanced management approaches target individual intersections and do not synergize with other intersections. Some roads may be busier than others at different times, which requires additional time to clear congestion on the road. An in-depth study of autonomous vehicle traffic management is in high demand. Almost all current road infrastructures and traffic control technologies depend on human driving [Wagner, 2016]. Even self-driving cars are being trained to recognize human-oriented traffic signs and mimic human driving behaviors, which is by no means necessary for an efficient or reliable traffic management system.

Optimizing traffic control for autonomous vehicles has been studied for many years. These studies range from optimizing vehicle routing to matching traffic control [Liard et al., 2020, Xu et al., 2012, You et al., 2019]. Others have proposed new traffic control methods to make autonomous vehicles more efficient [Chen et al., 2021, Dresner and Stone, 2008b, Fernandes and Nunes, 2010]. Furthermore, some articles optimize existing traffic management protocols to adapt to changes in traffic flow [Sun et al., 2020, Wu et al., 2022]. Traditional vision-based traffic control facilities, such as traffic lights, roundabouts, and stop signs, will likely be replaced by less visible but more

efficient and effective algorithm-controlled road facilities as new technologies emerge for vehicle-based communication and intelligent traffic control [Gruel and Stanford, 2016]. With the arrival of autonomous vehicles, better solutions have become necessary for intelligent traffic management systems that use the most advanced technologies [Campisi et al., 2021]. Optimizing traffic control based on vehicle decisions has been an ongoing challenging problem.

This chapter presents a formal method to optimize traffic control with fully autonomous vehicles by combining a nonlinear optimization problem and a game-theoretical approach. Since we assume that the vehicles make autonomous self-interested decisions, we use a game-theoretical model based on population games and congestion games to describe the behavior of the vehicles. We then assume that an independent agent, which we call the *intersection manager*, has control over all intersections by dynamically changing their respective traffic management protocols. The goal of this intersection is to minimize the total delay of all vehicles. Specifically, we formalize this scenario as a traffic control optimization problem.

We have created a simple simulation environment in *Aimsun*, which is commercial software that provides simulation and services for transportation planning and traffic management [Ims and Pedersen, 2021]. We provide an algorithm using Aimsun to find approximate optimization solutions for traffic control protocols. Our proposed algorithm can successfully reduce the delay in the road network caused by static traffic assignment and traffic setting based on the traffic flow distribution generated by static traffic assignment, as demonstrated by the test results.

6.2 Traffic Network Game

This section presents the formal definition of a traffic network game. In this model, we consider the cost of the vehicle as the time the vehicle is driving along a road, as well as the time it takes for the vehicle to wait at the intersections it goes through. Unlike

the traffic assignment model (see Section 5.2) in Chapter 5, we consider the change of the intersection management strategy in this chapter and optimize the management approach for the route choice of vehicles.

A traffic network game is a model that describes the behavior of autonomous vehicles on a road network with intelligent traffic control. Each vehicle has a fixed origin and destination but can choose which road it will take to reach its destination. Some of the concepts from chapters 2, 3, and 4 are recalled and expanded here to make this chapter easy to follow.

6.2.1 Road network and traffic management protocols

As we mentioned in Chapter 2, the macro-level road network model (see Section 2.2) has a limit to describe traffic protocols, and the meso-level road network model we mentioned in Section 2.3 compensates for this drawback. The road network models and related concepts mentioned in this chapter are extended from the meso-level road network model. In this subsection, we briefly outline the road network model and the traffic assignment problem.

Given a **meso-level road network** $G = (N, L)$, an **origin-destination** (o_i, d_i) is a pair of intersections. $I = \{(o_1, d_1), \ldots, (o_i, d_i), \ldots, (o_k, d_k)\}$ is the set of all possible origin-destinations on the road network G (see Def 5.2). We simply write an origin-destination (o_i, d_i) as a subscript $i \in I$. For each origin-destination i, let $\Gamma_i = \{\gamma_1, \ldots, \gamma_{m^i}\}$ denote a set of all paths from o_i to d_i in G, where m^i is the total number of such paths. From now on, we will refer to such paths as *routes*. Formally, we define a route as follows:

Definition 6.1. Given a meso-level road network $G = (N, L)$ and its intersection relations $(C_n, Z_n)_{n \in N}$. For each origin-destination $(n, n') \in P$ and $n'', n''' \in N$, a route γ is a sequence $(l, l'') \to c_1 \to \cdots c_{m-1} \to (l''', l')$, where $l, l', l'', l''' \in L$, $l = (n, n'', i)$ and $l' = (n''', n', j)$, and satisfies for any u with $0 \leq u \leq m - 1$, such

that $tail(c_u) = head(c_{u+1})$, where $head(c) = l$ and $tail(c) = l'$ for any connection $c = (l, l') \in C_n$.

$\Gamma = \bigcup_{i \in I} \Gamma_i$ defines all possible routes on the road network, and the total number of routes for all origin-destination pairs is denoted by $m = \sum_{i \in I} m^i$. We assume that $\Gamma_i \neq \emptyset$ for all $i \in I$.

For each origin-destination pair, a population of vehicles must travel from that origin to that destination. To simplify matters, we will not regard vehicles as discrete, countable objects but rather as a continuous quantity (like a liquid). For any origin-destination pair, $i \in I$, the value $v_i > 0$ represents the vehicle 'density'. That is the number of vehicles that travel from o_i to d_i per unit of time.

Intuitively, the traffic management protocols of an intersection on a road network can be interpreted as the proportion of allocation of all connections per time unit from a macro-perspective in our road network model. For example, at an intersection under the management of traffic lights, the difference in the allocation proportion leads to changes in the green light time of that connection per unit time for each intersection connection. Obviously, the longer the traffic light is green, the more traffic can pass through that connection. Therefore, we can define an intersection's traffic management protocol as the allocation proportion to each intersection's connection. Let $\overline{Z}_n := Z_n \bigcup \{(c, c) : c \in C_n\} \bigcup \{(c, c'') : (c, c'), (c', c'') \in Z_n\}$ denote the reflexive and transitive closure of Z_n, so it forms an equivalence relation on C_n. For each $c \in C_n$, let $[c] = \{c' \in C_n : (c, c') \in \overline{Z}_n\}$ denote the equivalence class of c under C_n [Schechter, 1996]. Formally, the traffic management protocol for an intersection is defined as follows.

Definition 6.2. Given a meso-level road network $G = (N, L)$ and its intersection relations $(C_n, Z_n)_{n \in N}$. For each intersection $n \in N$, a traffic management protocol x_n for the intersection is a function $x_n : C_n \to [0, 1]$ such that $\sum_{c' \in [c]} x_n(c') \leq 1$, which allocates a percentage of resources, such as time, flow, or capacity, to each connection at the intersection.

The condition shows that the sum of allocation proportion for all connections in the same equivalence class must be less than or equal to the unit of time. If we consider the traffic light as an example of the traffic management protocol at the intersection n, then the value $x_n(c)$ represents the percentage of time that the traffic light for connection c is green. Furthermore, let X_n be the set of all possible traffic management protocols for intersection n. We assume that all intersections on a road network are controlled by a single intersection manager device. This device focuses on reducing total delay by changing the traffic management protocols of the road network. We define $X = \prod_{n \in N} X_n = \{x = (x_{n_1}, \ldots, x_{n_{|N|}}) \in [0,1]^\sigma : n \in N, x_n \in X_n\}$ as the set of *traffic settings*, where $\sigma = \sum_{n \in N} |C_n|$ is the total number of connections in the road network. An element of X represents a joint state used to describe intersection traffic management protocols, one for each intersection.

6.2.2 Traffic network model

In this subsection, we use the defined road network model and traffic management protocol to describe the traffic network game model.

6.2.2.1 Players and strategy Space

In the traditional traffic assignment game model, the vehicle acts as a player while its strategy is to choose the road to travel from its origin to its destination. But the weakness is that the game model becomes very complex when the number of vehicles is large enough. This thesis focuses on the macroscopic impact on the road network when vehicles make their decisions. Therefore, to simplify the model, we combine all vehicles with the same origin-destination into a single independent player, which aims to assign individual vehicles to different routes. Therefore, the set of strategies for an origin-destination pair $i \in I$ is $S^i = \{s^i \in \mathbb{R}^{m^i} : \sum_{\gamma \in \Gamma_i} s^i_\gamma = v^i\}$. The element $s^i_\gamma \in \mathbb{R}^+$ represents the mass of vehicles in the origin-destination pair p choosing strategy $\gamma \in \Gamma_i$.

Specifically, a strategy for a given player assigns a certain percentage of all vehicles to each possible route. For example, it can assign 60% of the cars to route γ_1 and 40% of the cars to route γ_2.

6.2.2.2 Strategy profiles

Let $S = \prod_{i \in I} S^i = \{s = (s^{p_1}, \ldots, s^{i_{|I|}}) \in \mathbb{R}^m : s^i \in S^i\}$ denote the set of strategy profiles that describe the behavior of all players at once. Each strategy profile s is a vector of vehicle mass distributions, one for each origin-destination. In a congestion game (see Def.5.1, the cost of each player is influenced by the resources they select and the number of other players who also select those resources. Next, we consider connections as facilities for a congestion game. For each node $n \in N$, each connection $c \in C_n$, $f_c(s)$ is the total mass of vehicles that use the connection c. That is, $f_c(s)$ is determined from the strategy profile in the following way:

$$f_c(s) = \sum_{i \in I} \sum_{\gamma \in \Gamma_i} s^i_\gamma \delta_{c,\gamma} \tag{6.1}$$

where $\delta_{c,\gamma}$ is an indicator function that indicates whether a connection c is contained in a route γ. That is, $\delta_{c,\gamma} = 1$ if $c \in \gamma$; otherwise, $\delta_{c,\gamma} = 0$.

6.2.2.3 Cost functions

Each connection (See Def 2.3) has a cost function $d^x_c : \mathbb{R}^+ \to \mathbb{R}$ with the traffic flow of connection $f_c(s)$, which is the total number of vehicles using the connection under strategy profile s, by given traffic setting x. The cost of the vehicle is the travel time on the route it takes, which is the sum of the delays on its constituent connections (See Def 2.3) and the free-flow travel time of the route. The delay on a connection depends on the number of vehicles that use that connection and the traffic management protocol applied to the connection. Subsequently, the free-flow travel time of a route

is a constant number that depends on that route's length and speed limit. It is worth mentioning that the route's travel time does not represent the actual travel time for each vehicle; however, it shows the average travel time for all vehicles that choose that route. Formally, the cost for each route is calculated as follows:

$$F_\gamma^x(s) = \sum_{c \in \gamma} d_c^x(f_c(s)) + T_c \tag{6.2}$$

where x is a traffic setting and T_c is the free-flow travel time of connection c, which is a constant number depending on the speed limit and length of the road. $F^x : S \to \mathbb{R}^m$ is a continuous map that assigns each strategy profile $s \in S$ to a vector of costs for a given traffic setting x, one for each route in each origin-destination.

In summary, the traffic network game is defined as follows:

Definition 6.3. Given a road network $G = (N, L)$ and its intersection relations $(C_n, Z_n)_{n \in N}$. A tuple is called a traffic network game $(I, \Gamma, X, S, (F^x)_{x \in X})$, where

- I is the set of origin-destination pairs.
- Γ is the set of all routes.
- X is the set of traffic settings.
- S is the set of strategy profiles.
- $F^x : S \to \mathbb{R}^m$ is the cost function given the traffic setting $x \in X$.

6.2.3 User equilibrium

Next, we need to define the equilibrium strategy profile of the traffic network games. Here, we recall Def 5.8 and Proposition 5.9 from Chapter 5 to define the user equilibrium. Specifically, a user equilibrium strategy profile is obtained if no vehicle can

reduce travel time with one-sided measures. Furthermore, we use the following lemmas to show the properties of traffic network games based on the concepts of population games [Sandholm, 2010], congestion games [Rosenthal, 1973], and potential games [Monderer and Shapley, 1996].

Lemma 6.4. *Given a traffic setting x, any traffic network game $(I, \Gamma, \{x\}, S, F^x)$ is a congestion game.*

Lemma 6.4 is trivial. Based on the definition of congestion game [Rosenthal, 1973], the cost of each player depends on the facility it chooses and the number of players who choose the same facility. In the traffic network game, we consider each origin-destination $p \in P$ to be a player and each connection c to be a facility. From the definition of the cost function (2), the cost of a player depends only on the traffic flow of the connections when the traffic setting x is given, which satisfies the definition of the congestion game. We assume that for each intersection $n \in N$ and each connection $c \in C_n$, suppose $g(s) = d_c^x(f_c(s))$, then $g(s)$ is a continuously differentiable function, and it satisfies the following properties:

- For any $n \in N$, $c \in C_n$ and $x \in X$, if $f_c(s) \leq f_c(s')$, then $d_c^x(f_c(s)) \leq d_c^x(f_c(s'))$ for any $s, s' \in S$;
- For any $n \in N$, $c \in C_n$ and $x^1, x^2 \in X$, if $x_n^1(c) \leq x_n^2(c)$, then $d_c^{x^1}(f_c(s)) \geq d_c^{x^2}(f_c(s))$ for any $s \in S$;

The first property says that for a fixed traffic setting x, if the number of vehicles $f_c(s)$ that drive through connection c increases (or stays the same), then the delay at $d_c^x(\cdot)$ will also increase (or stay the same. And the second property shows that for a connection c at intersection n, if the traffic setting $x_n^1(c)$ is less than the traffic setting $x_n^2(c)$, then all vehicles travel through connection c for the traffic setting x^1 have more delay than the traffic setting x^2. That is, $d_c^x(f_c(s))$ does not decrease with increasing $f_c(s)$ when x is fixed, while it does not increase with increasing $x_n(c)$ when s is fixed.

Lemma 6.5. *Given a traffic setting x, the traffic network game $(I, \Gamma, \{x\}, S, F^x)$ is a potential game.*

Proof. It is easy to see that a game is a full potential game if and only if it satisfies full externality symmetry(see observation 3.1.1 in [Sandholm, 2010]). In a congestion game, a vehicle taking route $\gamma \in \Gamma_q$ affects the cost of other vehicles choosing route $\gamma \in \Gamma_p$ through the marginal increases in congestion in the connections $c \in \gamma \bigcap \gamma'$ that the two routes have in common. Formally, we have

$$\frac{\partial F^x_\gamma(s)}{\partial s^q_{\gamma'}} = \sum_{c \in \gamma \bigcap \gamma'} d^x_c(f_c(s))' = \frac{\partial F^x_{\gamma'}(s)}{\partial s^p_\gamma} \tag{6.3}$$

for all $p, q \in I, \gamma \in \Gamma_p, \gamma' \in \Gamma_q$ and $s \in S$. Eq. (6.3) satisfies full external symmetry, which means that the traffic network game is a potential game in a given traffic setting x.

□

The Eq. 6.3 means that We can use the properties of the potential game to prove the existence of a user equilibrium.

Theorem 6.6. *Given a traffic setting $x \in X$, the traffic network game $(I, \Gamma, \{x\}, S, F^x)$ has at least one user equilibrium strategy profile.*

Proof. Based on Lemma 6.4 and Lemma 6.5, it is possible to find a potential function [Sandholm, 2010] for the congestion game $(I, \Gamma, \{x\}, S, F^x)$ as follows:

$$\mathcal{F}^x(s) = \sum_{\gamma \in \Gamma} \int_0^{f_c(s)} F^x_\gamma(z) dz \tag{6.4}$$

So the task of finding the user equilibrium can be considered as the following non-linear optimization problem:

$$\min_{s \in S} \mathcal{F}^x(s) \tag{6.5}$$

subject to:

$$\sum_{\gamma \in \Gamma_i} s_\gamma^i = v^i, \quad \forall i \in I \tag{6.6}$$

$$s_\gamma^i \in [0, v^i], \quad \forall \gamma \in \Gamma_i \quad \& \quad \forall i \in I \tag{6.7}$$

According to Theorem 3.1.3 in [Sandholm, 2010], the solution to this non-linear problem is a user equilibrium. □

For some given traffic network game, let $UE(x) = \{s \in S : i \in I, \gamma, \gamma' \in \Gamma_i, s_\gamma^i > 0 \rightarrow F_\gamma^x(s) \leq F_{\gamma'}^x(s)\}$ denote all user equilibrium strategy profiles for a given traffic setting x. We know from Theorem 6.7 that $UE(x) \neq \emptyset$ for all traffic settings $x \in X$. Next, we investigate an optimization problem to optimize traffic control for fully autonomous vehicles based on the model proposed in this section.

6.3 Traffic Control Optimization

In this section, we first introduce an optimization problem that aims to reduce the total delay of vehicles based on the traffic network game model. Then, we present algorithms to find an approximate solution to this optimization problem. Lastly, we create a simple test environment to verify our algorithms and show some results.

6.3.1 Optimization problem

The intersection manager aims to minimize the total delay caused by changing traffic settings. To find the minimum total delay for the intersection manager, we can formalize the problem as the following traffic control optimization problem:

$$\min_{x \in X, s \in UE(x)} \sum_{n \in N} \sum_{c \in C_n} f_c(s) d_c^x(f_c(s)) \tag{6.8}$$

subject to:

$$0 \leq x_n(c) \leq 1 \quad \forall c \in C_n \quad \& \quad \forall n \in N \tag{6.9}$$

$$\sum_{c' \in C_n, s.t. c' \in [c]} x_n(c') \leq 1 \quad \forall c, c' \in C_n \quad \& \quad \forall n \in N \tag{6.10}$$

$$f_c(s) \geq 0 \quad \forall c \in C_n \quad \& \quad \forall n \in N \tag{6.11}$$

$$\sum_{\gamma \in \Gamma_i} s_\gamma^i = v^i \quad \forall i \in I \tag{6.12}$$

$$s_\gamma^i \in [0, v^i] \quad \forall \gamma \in \Gamma_i \quad \& \quad \forall i \in I \tag{6.13}$$

The objective function Eq. (6.8) calculates the total delay of the road network. The $s \in UE(x)$ corresponds to the user equilibrium, which guarantees that no vehicle can experience a shorter travel time by unilateral deviation in an optimized traffic setting. The constraints Eq. (6.9)-(6.10) ensure the feasibility of traffic setting x. The constraint Eq. (6.11) specifies the positive traffic flow for all connections. Furthermore, the constraints Eq. (6.12)-(6.13) identify the feasibility of the strategy profile. The objective function cannot be solved by linear optimization when user equilibrium is used as a constraint. Therefore, we have created a simple road network in *Aimsun* and implemented an algorithm to provide an approximate solution to the problem, which we present in the next section.

6.3.2 Simulation-based solution

The method of successive averages (MSA) is the algorithm most widely used to find the solution to traffic assignment [Mounce and Carey, 2015]. Liu et al. [2009] proposed a method of successive weighted averages (MSWA) to obtain results faster than the original MSA. There are also some common algorithms, such as origin-based algorithms [Bar-Gera, 2002], path-based algorithms [Jayakrishnan et al., 1994], and the Frank-Wolfe algorithm [Fukushima, 1984]. The simulation software Aimsun provides

Algorithm 6.3.1: MSA Algorithm

Input: Traffic network game $(I, \Gamma, X, S, (F^x)_{x\in X})$, traffic setting x, gap tolerances θ, and the maximum iterations Λ.

Output: Approximate user equilibrium strategy profile s^*.

For each $i \in I$, $\gamma_i^* = \min_{\gamma\in\Gamma_i} \sum_{c\in\gamma} T_c$;

Initial strategy profile $s^0 = (s^{i,0}, \ldots, s^{i_{|I|},0})$, where $s^{i,0}_{\gamma_i^*} = v^i$ and $s^{i,0}_{\gamma} = 0$ for all $i \in I$;

$y^{i,0}_{\gamma} = v^i$, if $\gamma = \gamma_i^*$ for all $\gamma \in \Gamma_i \& i \in I$;

$y^{i,0}_{\gamma} = 0$, if $\gamma \neq \gamma_i^*$ for all $\gamma \in \Gamma_i \& i \in I$;

Run simulation in $Aimsun(s^0)$;

$j = 0$;

$R_{gap}(0) = \frac{\sum_{\gamma\in\Gamma} s^{i,0}_{\gamma}(tt^{i,0}_{\gamma} - \pi^0_i)}{\sum_{\gamma\in\Gamma} s^{i,0}_{\gamma}\pi^0_i}$;

while $R_{gap}(j) > \theta$ and $j < \Lambda$ **do**

  $y^{i,j}_{\gamma} = v^i$, if $\gamma = \pi_i^{j-1}$ for all $\gamma \in \Gamma_i \ \& \ i \in I$;

  $y^{i,j}_{\gamma} = 0$, if $\gamma \neq \pi_i^{j-1}$ for all $\gamma \in \Gamma_i \ \& \ i \in I$;

  $s^{i,j+1}_{\gamma} = \frac{1}{j}(y^{i,1}_{\gamma} + \cdots + y^{i,j}_{\gamma})$, for all $\gamma \in \Gamma_i \ \& \ i \in I$;

  $s^j = (s^{i,j}, \ldots, s^{i_{|I|},j})$;

  Run simulation in $Aimsun(s^*)$;

  $R_{gap}(j) = \frac{\sum_{\gamma\in\Gamma} s^{i,j}_{\gamma}(tt^{i,j}_{\gamma} - \pi^j_i)}{\sum_{\gamma\in\Gamma} s^{i,j}_{\gamma}\pi^j_i}$;

  $j = j + 1$;

  $s^* = s^j$

end while

an approximate user equilibrium strategy profile based on the method of successive averages (MSA) [Daskin, 1985], which is a general approach to compute User equilibrium with well-behaved link cost functions [Powell and Sheffi, 1982]. Due to its simplicity and efficiency in computation, static traffic assignment has been widely used not only to estimate traffic demands on specific networks but also for transportation planning and demand management policies related to infrastructure investment [Du and Wang, 2014, Wang and Du, 2016]; it is the preferred tool for strategic transport planning [Du and Wang, 2016].

The algorithm 6.3.1 describes MSA, which aims to find the user equilibrium of a given road network and traffic demand based on our definition of notation. We use $s = MSA(x)$ as user equilibrium strategies for a given traffic setting x in the remainder of the paper. The basic idea of MSA is explained as follows.

Algorithm 6.3.2: Traffic Control Optimization

Input: Traffic network game $(I, \Gamma, X, S, (F^x)_{x\in X})$, objective function Ψ, gap tolerances ϵ, step size $\tau \in [0, 1]$, and the maximum iterations Λ.

Output: Optimized traffic setting x^*.

Initial state $x^0 \in X$ and $s^0 = MSA(x^0)$;
$\Psi_c^0 = f_c(s^0)d_c^{x^0}(f_c(s^0))$, for all $c \in C_n$ and $n \in N$;
$\Psi^0 = \sum_{n\in N}\sum_{c\in C_n} \Psi_c^0$;
$j = 0$;
while $j < \Lambda$ **do**
  $k = 0$;
  $x_c^j = 1$, for all $n \in N$ such that $c, c' \in C_n$ and $(c, c') \notin Z_n$;
  $\eta_{(c,c')}^j = |\frac{\Psi_c^{j-1} - \Psi_{c'}^{j-1}}{\Psi_c^{j-1} + \Psi_{c'}^{j-1}}|$, for all $n \in N$, $c, c' \in C_n$ and $(c, c') \in Z_n$;
  while $k \in [0, 1]$ **do**
    $x_c^j = x_c^{j-1} + k\eta_{(c,c')}^j$, for all $n \in N$, $c, c' \in C_n$, $(c, c') \in Z_n$ such that $\Psi_c - \Psi_{c'} > 0$;
    $x_{c'}^j = x_c^{j-1} - k\eta_{(c,c')}^j$, for all $n \in N$, $c, c' \in C_n$, $(c, c') \in Z_n$ such that $\Psi_c - \Psi_{c'} < 0$;
    $s^j = MSA(x^j)$;
    $\Psi_c^j = f_c(s^j)d_c^{x^j}(f_c(s^j))$, for all $c \in C_n$ and $n \in N$;
    $\Psi^j = \sum_{n\in N}\sum_{c\in C_n} \Psi_c^j$;
    if $\frac{\Psi^{j-1} - \Psi^j}{\Psi^{j-1}} > \epsilon$ **then**
      $Break$;
    else
      $k = k + \tau$;
    end if
  end while
  $j = j + 1$;
  $x^* = x^j$
end while

- Step 1: Find the shortest path for all origin-destination by a given road network.
- Step 2: Assign all traffic demand to the shortest path for each origin-destination. This step is called all-or-nothing-assignment [Dial, 1971].
- Step 3: Checking equilibrium state:
  - If the equilibrium state is reached, the algorithm stops.
  - If the equilibrium is not reached, a portion (called step size) of the traffic is shifted to the new shortest path based on the existing traffic calculation. Then the current step is repeated until the equilibrium state is reached.

Let the predetermined sequence of step size in the MSA algorithm be $\theta_j = \frac{1}{j}$, where j is the iteration number. Conventional MSA is calculated as

$$s_\gamma^{i,j+1} = s_\gamma^{i,j} + \theta_j(y_\gamma^{i,j} - s_\gamma^{i,j}) \tag{6.14}$$

The Eq. (6.14) is used to calculate the traffic volume of route γ at iteration $j+1$, where $s_\gamma^{i,j}$ is the traffic volume of route γ from origin-destination i at iteration j, $y_\gamma^{i,j}$ is equal to the traffic demand of origin-destination v^i of route γ at iteration j. Also we can simplify the equation (6.14) as

$$s_\gamma^{i,j+1} = \frac{1}{j}(y_\gamma^{i,1} + \cdots + y_\gamma^{i,j}) \tag{6.15}$$

The proposed algorithm is considered to be converged if the average value of the user equilibrium relative gap becomes stable, where the relative gap can be calculated as:

$$R_{gap}(j) = \frac{\sum_{\gamma\in\Gamma} s_\gamma^{i,j}(tt_\gamma^j - \pi_i^j)}{\sum_{\gamma\in\Gamma} s_\gamma^{i,j}\pi_i^j} \tag{6.16}$$

where j is the iteration number, $s_\gamma^{i,j}$ is the traffic flow on the route γ, $tt_\gamma^{i,j}$ is the travel experience time of the route γ collected by *Aimsun*, and π_i^j is the minimum travel time route from the origin to destination i.

Algorithm 6.3.2 is used to find an approximate solution to the optimization problem of traffic control. The basic idea is to compare the total travel time of vehicles in each connection at each iteration. Under a traffic management protocol, if the overall vehicle travel time of a connection is higher than that of another connection, a higher percentage is allocated to the connection with a higher total travel time in the next iteration, and the percentage of the connection with lower travel time is reduced accordingly. The proportion of allocations for each iteration increases or decreases proportionally to the difference between the total delays of two connections. If the new traffic setting reduces the total delay by more than a given threshold, then this traffic setting will be used as

input for the next iteration by the MSA algorithm to find a user equilibrium. When the maximum number of iterations is reached, the algorithm is terminated. It is worth mentioning that the algorithm 6.3.2 can only guarantee convergence to a locally optimal solution but cannot guarantee that the algorithm output is the globally optimal solution. Therefore, an accurate algorithm to find the global optimal solution is a critical task for the future.

6.3.3 Aimsun setting and results

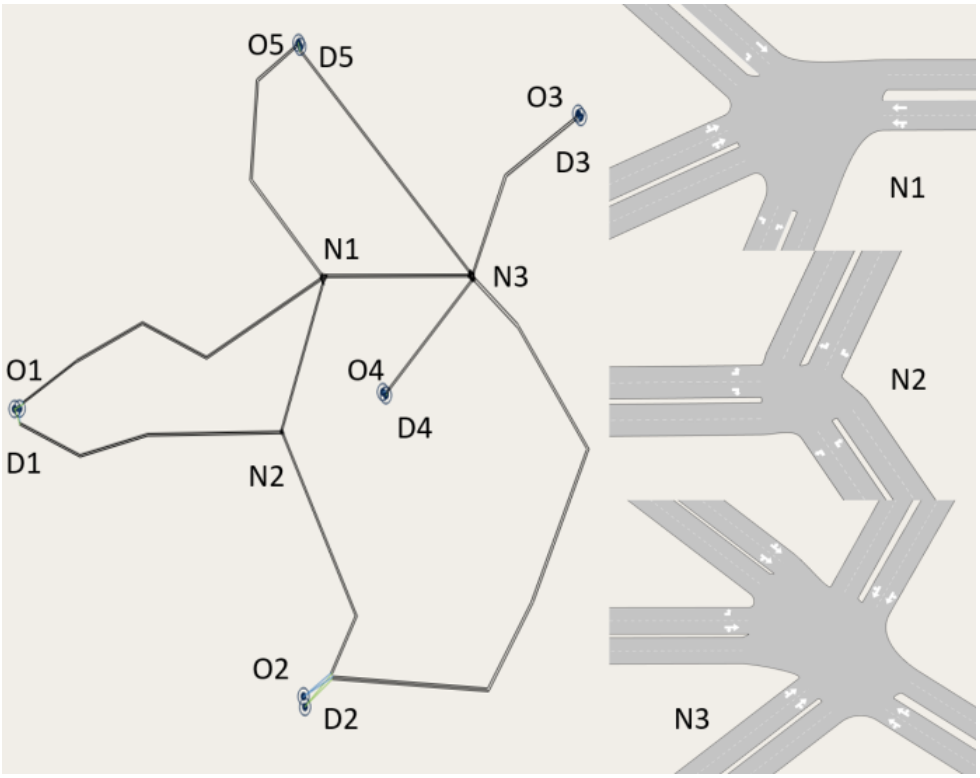


FIGURE 6.1: Simulation Setting

We built a simple testing environment to test the algorithms in *Aimsun*, as shown in Fig. 6.1. It includes three intersections, five origins, five destinations, and 20 origin-destination pairs. Each origin-destination pair has independent traffic demand. The total number of vehicles in the simulation is 5400. The simulation was performed using an Intel(R) Core(TM) $I9-10900K$ CPU @ 3.70GHz and $64GB$ RAM. Since the algorithm only seeks to optimize the solution to the problem, it is not considered from the

perspective of fast solutions. Therefore, iteration efficiency is not a research direction of the algorithm in this paper, and this issue is part of future work.

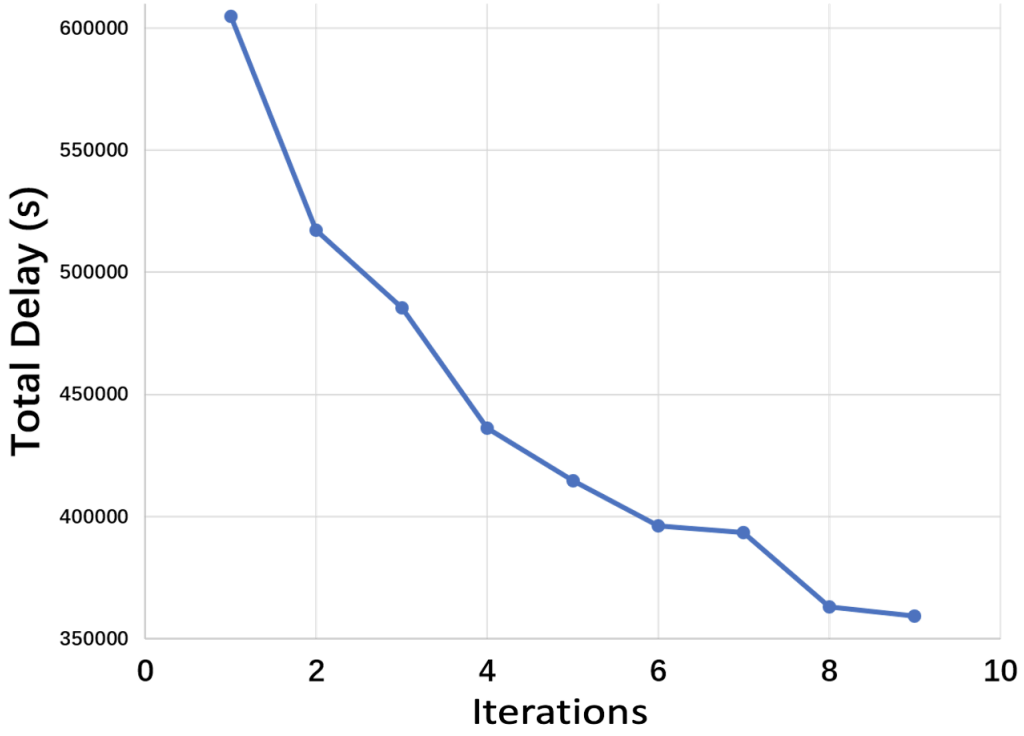


FIGURE 6.2: Total Delay on Road Network

In the experimental setup, each OD has a different traffic flow generated as a constant flow. In the initial traffic management setup, the ratio assigned to each connection is the same as the ratio between the total traffic passing through the intersection and the traffic of that connection in a static user equilibrium, and each segment uses the BPR function, which is expressed as

$$t_0 \cdot (1 + \alpha \cdot (\frac{volume}{capacity})^{\beta}) \tag{6.17}$$

where t_0 is the free flow travel time of the road lane, $\alpha = 0.15$, $\beta = 4$, $volume$ is the assigned traffic in the road lane and $capacity = 1800$ for all the road lanes. The initial traffic x_0 is set to have the same allocation ratio of connections at each intersection. For

example, if an intersection has four connections that conflict, the initial traffic management protocol is 0.25 for each connection. It is worth mentioning that road networks and traffic demand are the same during different iterations.

Fig. 6.2 shows the result of *Aimsun* using Algorithm 6.3.2, and it is evident that each iteration reduces the total delay of the simulated road network. Although we used a simple road network model, the algorithm can still reduce the total delay in static traffic assignment by 40% without changing vehicle behavior based on the assumption that vehicles make individual decisions. Further analysis shows that when changes in traffic settings, traffic assignment still has much room for optimization to reduce overall delay in a road network.

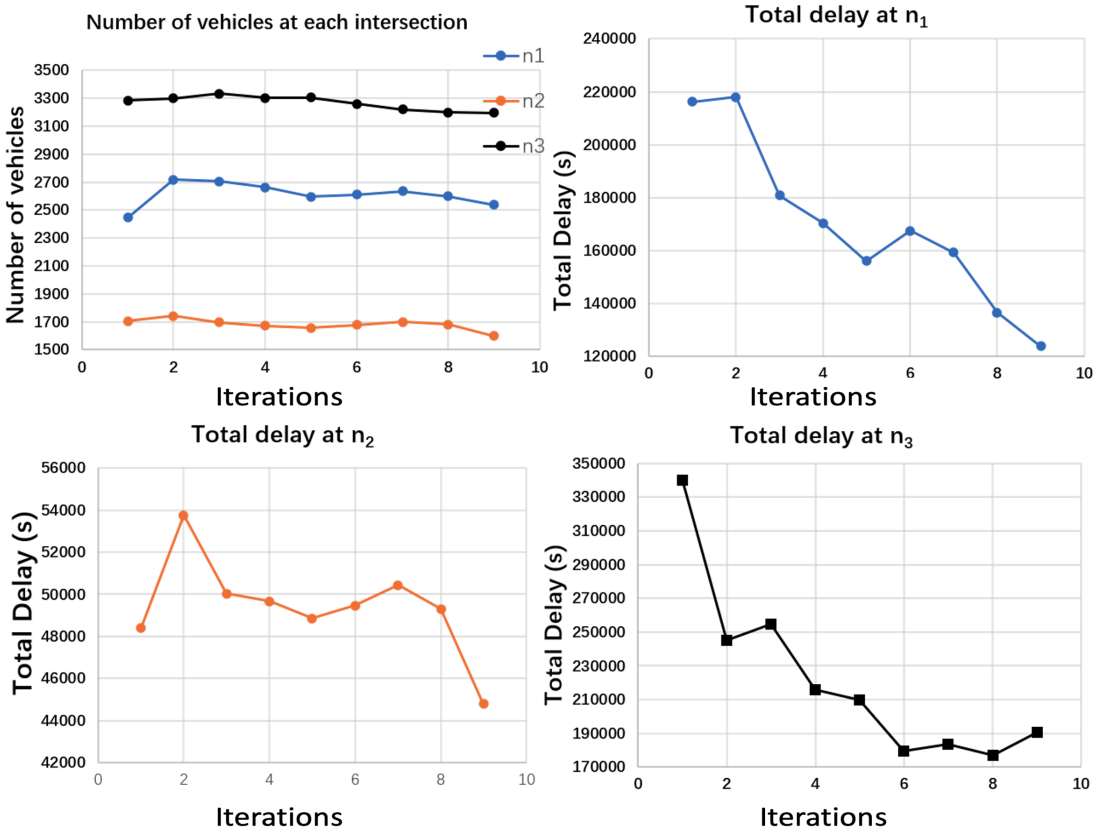


FIGURE 6.3: Total Number of Vehicles and Total Delay at Each Intersection

The upper left of Fig. 6.3 shows the total number of vehicles that pass at each intersection during each iteration. The coloured lines represent the number of cars at different intersections, one for each intersection in *Aimsun*. It can be seen from the graph that

the change in traffic flow is almost constant at each intersection throughout the iteration. The rest of the sub-figures in Fig. 6.3 show the total delay at each intersection for each iteration, one figure for one intersection. The overall delay in the road network decreases monotonically with increasing iterations. However, the total delay of vehicles at each intersection does not decrease monotonically. To put it simply, the autonomous decision of a car causes congestion at some intersections to make others more accessible, thus reducing the total delay on the road network.

6.4 Summary

In this chapter, we considered problems in the future transport system that optimise traffic control when fully autonomous vehicles can make individual decisions. We proposed a novel model to combine optimization and user equilibrium methods to increase the efficiency of road networks from a macroscopic point of view. Due to the complexity of the theoretical solution for traffic control optimization, we used a simulation-based approach to reach an approximate solution by implementing an iteration algorithm based on *Aimsun*. It can be seen from the experimental results that our proposed algorithm can significantly reduce the delay in the road network.

Our work leaves some interesting unsolved problems in this chapter. More efficient algorithms have been implemented that are used to find global optimization. From a game theory point of view, future research should consider intersections as intelligent agents, giving them utility, such as toll policies, making the traffic network a different model. Another suggestion for the future is to extend our model to handle the traffic with both autonomous and human drive vehicles.

Chapter 7

Conclusion and Future Work

This chapter summarises and discusses the work done so far and future research directions.

7.1 Summary of the Major Contributions

This thesis proposes a collection of formal models to define and analyze traffic management utilizing AI and multi-agent systems from a macroscopic perspective with fully autonomous vehicles, which can make independent judgments on the road. More specifically, the following work has been done:

1. Road network modelling: To make it possible for mobile robots, particularly those used in autonomous cars, to comprehend information about the composition and management of roads, formalized road network models based on many levels of scenarios have been introduced. These models offer three different perspectives on the road link.

    - Macro-level Road Network: Describes the relationships between locations and routes from a broad perspective, such as an urban road network.

- Meso-level road network: To enable autonomous vehicles to make better-informed decisions, the model can specify the internal link and potential collision for a single intersection and the relationship between roads and intersections. We typically use it to examine how traffic control procedures affect certain high-traffic areas or the central part of a city's road network.
- Micro-level road network: A discrete-time-based road network model represents how cars move on the road over time. The model can be applied to single-intersection traffic situations and vehicle trajectory monitoring in limited areas.

2. Intelligent Traffic Control Protocols: Multiple traffic management protocols based on road network representation should be available for autonomous cars. From the point of view of the road network, these management protocols define how cars cross intersections. We have employed several different simulators to develop and verify traffic management methods. Therefore, in addition to formally defining traffic management protocols, we have simulated and compared existing ones. The data comparison findings show that the traffic management techniques we recommend are more effective in regulating vehicle movements.

3. Traffic Assignment with Exponential Cost Functions:

   - We have investigated, with the help of empirical data, the impact of exponential cost functions on the traffic assignment problem. The data analysis findings demonstrate that the exponential cost function better fits the experimental and real-world data than the cost functions used in previous studies. The region-based macroscopic cost flow function that is used to simulate the travel costs of a particular region can also be utilized with the exponential cost function, which has potential uses for regional transport planning and macroscopic control of traffic management.
   - Using an exponential cost function, we have analyzed the tight upper bound of the price of anarchy in a road network and how the tight upper bound

changes when traffic demand changes. Since traffic demand cannot exceed capacity, we examine the trend of PoA with traffic demand for the same road network architecture, whether the cost function is exponential or the BPR function.

4. Hybrid of Traffic Assignment and Control: We have suggested a hybrid model based on a multi-agent system that tries to optimize traffic control at the network level under the assumption that vehicles make independent decisions. The model integrates the macroscopic traffic assignment problem and the microscopic traffic control problem by combining congestion game and non-linear optimization. Based on the suggested approach, we have examined a traffic control optimization problem as a non-linear optimisation problem aiming to increase road network effectiveness. We have also implemented an algorithm to resolve the non-linear optimization approximation of a solution. The results of our tests using the Aimsun simulation software show that this approach greatly minimizes the overall delay of the road network.

7.2 Future Work

In this thesis, we have defined and examined the route planning and traffic management of autonomous driving in road networks from different perspectives. This thesis has raised many more questions, which have been introduced throughout this thesis, some of which have been addressed in the relevant chapters. We here present more potential directions for future work.

1. This thesis used the traffic scenario as one of the application directions of the multi-agent system. The game theory model used in this paper is based on the non-cooperative game with selfish behaviour. At the same time, the autonomous driving scenario can also be modelled as a cooperative game. Existing studies include cooperative games [Song et al., 2014, Wang et al., 2021, Wurman et al.,

2008], automated negotiation Chater et al. [2018], Keferböck and Riener [2015], Kneissl et al. [2018] or mechanism design [Arslan et al., 2007, Liu et al., 2017, Lovellette and Hexmoor, 2021]. It makes more sense to use cooperative mechanisms for some traffic scenarios. For example, how do we solve the problem of two vehicles driving in different directions while the road is too narrow for the cars to pass each other [Imbsweiler et al., 2018, Mavromatis et al., 2020, Zhao et al., 2015]? We intend to extend the existing non-cooperative game model to a cooperative game model to describe the social behaviour arising from autonomous driving decisions and develop related algorithms.

2. In Chapter 6, we have proposed a hybrid model for traffic assignment and control based on the game theory where the players are purely autonomous vehicles. Some studies of game theory have been conducted in which individual devices or several devices act as game players to assist in traffic management in a region, rather than together using games and vehicles to describe traffic, such as [Bui and Jung, 2018, Elhenawy et al., 2015, Li et al., 2018, Wei et al., 2018]. However, with the development of V2X technology, smart roadside devices can become independent agents shortly, and they make decisions individually based on current traffic situations [Jameel et al., 2020, Wang et al., 2019]. We intend further to extend the model in Chapter 6 to describe the group decision-making of infrastructure and vehicle agents. The aim is to provide models and related algorithms for more intelligent transportation.

3. The research in this thesis can be extended to ride-sharing. Ride-sharing is a scenario in which a private vehicle driver who wants to offer a ride and a passenger who wants to be picked up by a private vehicle utilizes a network (such as one accessed through an app or website) to coordinate the sharing of personal car rides, with the passenger paying the fare. Some of the approaches of the road network [Agatz et al., 2012, Ta et al., 2017, Wang et al., 2020] and the game theory approaches [Chau et al., 2020, Wang et al., 2018, Yan et al., 2021] on

ride-sharing are proposed from a micro-perspective. However, we anticipate adjustments to the macro-road network model to regulate or enhance ride-sharing for autonomous vehicles in a given area.

4. Additionally, this research can also be applied to fields related to vehicle routing problems (VRP). Vehicle routing problem [Economides et al., 1991, Golden et al., 2008, Toth and Vigo, 2002] is an optimization and integer optimization problem to rationally allocate the fleet of vehicles to deliver items to the user. Package delivery, warehouse-supermarket shipments, and express delivery can be vehicle routing problems. Game theory is widely applied to existing research on VPR, such as non-cooperative game [Gansterer and Hartl, 2018, Hollander and Prashker, 2006] and automated negotiation [de Jonge et al., 2021, 2022], current traffic scenario with a human driver. We want to use the research mentioned in this thesis to solve the problem of vehicle routing in fully autonomous driving situations. But how to reasonably get goods from the merchant to the user in an autonomous driving scenario?

5. Finally, we expect to be able to switch the research scenario from transportation to multi-robot systems for more practical applications, such as warehouse [Bolu and Korçak, 2021, Claes et al., 2017, Li et al., 2020] and service robots [Cavallo et al., 2014, Di Nuovo et al., 2018]. We envision applications in these areas using road network models and management protocols similar to those in this thesis.

Appendix A

Published Work

Some of the results presented in this thesis are included in the following papers which have been published:

- Jianglin Qiao, Dongmo Zhang, and Dave de Jonge. Virtual Roundabout Protocol for Autonomous Vehicles. AI 2018: Advances in Artificial Intelligence, pages 773–782. *The results in this paper are investigated in Chapter 2 and 3.*
- Jianglin Qiao, Dongmo Zhang, and Dave de Jonge. Graph Representation of Road and Traffic for Autonomous Driving. PRICAI 2019: Trends in Artificial Intelligence, pages 377–384. *The results in this paper are included in Chapter 2 and 3.*
- Jianglin Qiao, Dongmo Zhang, and Dave de Jonge. Priority-based Traffic Management Protocols for Autonomous Vehicles on Road Networks. AI 2021: Advances in Artificial Intelligence, pages 240-253. *The results in this paper are shown in Chapter 2 and 3.*
- Jianglin Qiao, Dave de Jonge, Dongmo Zhang, Carles Sierra and Simeon Simoff. A Hybrid Model of Traffic Assignment and Control for Autonomous Vehicles. PRIMA 2022: Principles and Practice of Multi-Agent Systems. *The results in this paper are presented in Chapter 6.*

Some of the results presented in this thesis are included in the following papers, which have been submitted but are currently still under review:

- Jianglin Qiao, Dave de Jonge, Bo Du, Carles Sierra, Dongmo Zhang, and Simeon Simoff. Estimating Region-based Macroscopic Cost Flow Function: An Exponential Approach. Submitted to Journal of Advanced Transportation. *The results in this paper are proposed in Chapter 4.*

- Jianglin Qiao, Dave de Jonge, Dongmo Zhang, Simeon Simoff and Carles Sierra. Price of Anarchy of Traffic Assignment with Exponential Cost Functions. Submitted to the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023) *The results in this paper are included in Chapter 5.*

Bibliography

Aashtiani, H. Z. and Magnanti, T. L. (1981). Equilibria on a congested transportation network. *SIAM Journal on Algebraic Discrete Methods*, 213–226.

Administration, N. H. T. S. et al. (2016). 2015 motor vehicle crashes: overview. *Traffic safety facts: research note*, 1–9.

Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 295–303.

Akcelik, R. (1978). A new look at davidson's travel time function. *Traffic Engineering & Control*, 19.

Akella, A., Seshan, S., Karp, R., Shenker, S., and Papadimitriou, C. (2002). Selfish behavior and stability of the internet: A game-theoretic analysis of tcp. *ACM SIGCOMM Computer Communication Review*, 117–130.

Aland, S., Dumrauf, D., Gairing, M., Monien, B., and Schoppmann, F. (2011). Exact price of anarchy for polynomial congestion games. *SIAM Journal on Computing*, 1211–1233.

Arslan, G., Marden, J. R., and Shamma, J. S. (2007). Autonomous vehicle-target assignment: A game-theoretical formulation.

Association, A.-E. A. M. et al. (2015). The automobile industry pocket guide 2015-2016.

Au, T.-C. and Stone, P. (2010). Motion planning algorithms for autonomous intersection management. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Au, T.-C., Zhang, S., and Stone, P. (2014). Semi-autonomous intersection management. In *AAMAS*, 1451–1452.

Baker, B. M. and Ayechew, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 787–800.

Banerjee, T., Bose, S., Chakraborty, A., Samadder, T., Kumar, B., and Rana, T. K. (2017). Self driving cars: A peep into the future. In *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference*, 33–38.

Bar-Gera, H. (2002). Origin-based algorithm for the traffic assignment problem. *Transportation Science*, 398–417.

Bashiri, M. and Fleming, C. H. (2017). A platoon-based intersection management system for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium*, 667–672.

Basma, F., Tachwali, Y., and Refai, H. H. (2011). Intersection collision avoidance system using infrastructure communication. In *2011 14th International IEEE Conference on Intelligent Transportation Systems*, 422–427.

Beckmann, M., McGuire, C. B., and Winsten, C. B. (1956). Studies in the economics of transportation. Technical report.

Belkhouche, F. (2017). Control of autonomous vehicles at an unsignalized intersection. *2017 American Control Conference*, 1340–1345.

Bell, M. G. (1983). The estimation of an origin-destination matrix from traffic counts. *Transportation Science*, 198–217.

Bellman, R. (1958). On a routing problem. *Quarterly of applied mathematics*, 87–90.

Bimbraw, K. (2015). Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th International Conference on Informatics in Control, Automation and Robotics*, 191–198.

Bolu, A. and Korçak, Ö. (2021). Adaptive task planning for multi-robot smart warehouse. *IEEE Access*, 27346–27358.

Bui, K.-H. N. and Jung, J. J. (2018). Cooperative game-theoretic approach to traffic flow optimization for multiple intersections. *Computers & Electrical Engineering*, 1012–1024.

Campisi, T., Severino, A., Al-Rashid, M. A., and Pau, G. (2021). The development of the smart cities in the connected and autonomous vehicles (cavs) era: From mobility patterns to scaling in cities. *Infrastructures*, 100.

Carlino, D., Boyles, S. D., and Stone, P. (2013). Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems*, 529–534.

Carpenter, R. (1960). Principles and procedures of statistics, with special reference to the biological sciences. *The Eugenics Review*, 172.

Cavallo, F., Limosani, R., Manzi, A., Bonaccorsi, M., Esposito, R., Di Rocco, M., Pecora, F., Teti, G., Saffiotti, A., and Dario, P. (2014). Development of a socially believable multi-robot solution from town to home. *Cognitive Computation*, 954–967.

Chan, C.-Y. (2017). Advancements, prospects, and impacts of automated driving systems. *International Journal of Transportation Science and Technology*, 208 – 216.

Chater, N., Misyak, J., Watson, D., Griffiths, N., and Mouzakitis, A. (2018). Negotiating the traffic: Can cognitive science help make autonomous vehicles a reality? *Trends in cognitive sciences*, 93–95.

Chau, S. C.-K., Shen, S., and Zhou, Y. (2020). Decentralized ride-sharing and vehicle-pooling based on fair cost-sharing mechanisms. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, A., Yang, H., Lo, H. K., and Tang, W. H. (1999). A capacity related reliability for transportation networks. *Journal of advanced transportation*, 183–200.

Chen, S., Wang, H., and Meng, Q. (2021). An optimal dynamic lane reversal and traffic control strategy for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*.

Choi, E.-H. (2010). Crash factors in intersection-related crashes: An on- scene perspective, 366.

Christodoulou, G. and Koutsoupias, E. (2005). The price of anarchy of finite congestion games. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 67–73.

Claes, D., Oliehoek, F., Baier, H., Tuyls, K., et al. (2017). Decentralised online planning for multi-robot warehouse commissioning. In *AAMAS'17*, 492–500.

Cominetti, R., Dose, V., and Scarsini, M. (2021). The price of anarchy in routing games as a function of the demand. *Mathematical Programming*, 1–28.

Corless, R. M., Gonnet, G. H., Hare, D. E., Jeffrey, D. J., and Knuth, D. E. (1996). On the lambertw function. *Advances in Computational mathematics*, 329–359.

Cui, S., Seibold, B., Stern, R., and Work, D. B. (2017). Stabilizing traffic flow via a single autonomous vehicle: Possibilities and limitations. In *2017 IEEE Intelligent Vehicles Symposium*, 1336–1341.

Dafermos, S. C. and Sparrow, F. T. (1969). The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards B*, 91–118.

Daskin, M. S. (1985). Urban transportation networks: Equilibrium analysis with mathematical programming methods.

de Grange, L., Marechal, M., and González, F. (2019). A traffic assignment model based on link densities. *Journal of Advanced Transportation*.

de Jonge, D., Bistaffa, F., and Levy, J. (2021). A heuristic algorithm for multi-agent vehicle routing with automated negotiation. In Dignum, F., Lomuscio, A., Endriss, U., and Nowé, A., editors, *AAMAS '21*, 404–412.

de Jonge, D., Bistaffa, F., and Levy, J. (2022). Multi-objective vehicle routing with automated negotiation. *Applied Intelligence*, 16916–16939.

de La Fortelle, A. (2010). Analysis of reservation algorithms for cooperative planning at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, 445–449.

Dheenadayalu, Y., Wolshon, B., and Wilmot, C. (2004). Analysis of link capacity estimation methods for urban planning models. *Journal of transportation engineering*, 568–575.

Di Nuovo, A., Broz, F., Wang, N., Belpaeme, T., Cangelosi, A., Jones, R., Esposito, R., Cavallo, F., and Dario, P. (2018). The multi-modal interface of robot-era multi-robot services tailored for the elderly. *Intelligent Service Robotics*, 109–126.

Dial, R. B. (1971). A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation research*, 83–111.

Djahel, S., Doolan, R., Muntean, G.-M., and Murphy, J. (2014). A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *IEEE Communications Surveys & Tutorials*, 125–151.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, 1–16.

Dresner, K. and Stone, P. (2004a). Multiagent traffic management: A reservation-based intersection control mechanism. In *Autonomous Agents and Multiagent Systems, International Joint Conference on*, 530–537.

Dresner, K. and Stone, P. (2004b). Multiagent traffic management: A reservation-based intersection control mechanism. In *AAMAS '04*, 530–537.

Dresner, K. and Stone, P. (2008a). A multiagent approach to autonomous intersection management. *J. Artif. Int. Res.*, 591–656.

Dresner, K. and Stone, P. (2008b). A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 591–656.

Du, B. and Wang, D. Z. (2014). Continuum modeling of park-and-ride services considering travel time reliability and heterogeneous commuters – a linear complementarity system approach. *Transportation Research Part E: Logistics and Transportation Review*, 58–81.

Du, B. and Wang, D. Z. W. (2016). Solving continuous network design problem with generalized geometric programming approach. *Transportation Research Record*, 38–46.

Economides, A. A., Silvester, J. A., et al. (1991). Multi-objective routing in integrated services networks: A game theory approach. In *Infocom*, 1220–1227.

Elhenawy, M., Elbery, A. A., Hassan, A. A., and Rakha, H. A. (2015). An intersection game-theory-based traffic control algorithm in a connected vehicle environment. In *2015 IEEE 18th international conference on intelligent transportation systems*, 343–347.

Esteve, M., Palau, C. E., Martínez-Nohales, J., and Molina, B. (2007). A video streaming application for urban traffic management. *Journal of network and computer applications*, 479–498.

Fajardo, D., Au, T.-C., Waller, S. T., Stone, P., and Yang, D. (2011). Automated intersection control: Performance of future innovation versus current traffic signal control. *Transportation Research Record*, 223–232.

Feldman, M., Immorlica, N., Lucier, B., Roughgarden, T., and Syrgkanis, V. (2016). The price of anarchy in large games. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 963–976.

Fernandes, P. and Nunes, U. (2010). Platooning of autonomous vehicles with intervehicle communications in sumo traffic simulator. In *13th International IEEE Conference on Intelligent Transportation Systems*, 1313–1318.

Fok, C.-L., Hanna, M., Gee, S., Au, T.-C., Stone, P., Julien, C., and Vishwanath, S. (2012). A platform for evaluating autonomous intersection management policies. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, 87–96.

Fukushima, M. (1984). A modified frank-wolfe algorithm for solving the traffic assignment problem. *Transportation Research Part B: Methodological*, 169–177.

Gansterer, M. and Hartl, R. F. (2018). Collaborative vehicle routing: a survey. *European Journal of Operational Research*, 1–12.

Geng, Y. and Cassandras, C. (2015). Multi-intersection traffic light control with blocking. *Discrete Event Dynamic Systems*, 7–30.

Gibbons, R. et al. (1992). A primer in game theory.

Glantz, S. A. and Slinker, B. K. (2001). *Primer of applied regression & analysis of variance, ed.*

Golden, B. L., Raghavan, S., Wasil, E. A., et al. (2008). *The vehicle routing problem: latest advances and new challenges*.

Graham, D. J. and Glaister, S. (2004). Road traffic demand elasticity estimates: a review. *Transport reviews*, 261–274.

Gruel, W. and Stanford, J. M. (2016). Assessing the long-term effects of autonomous vehicles: A speculative approach. *Transportation Research Procedia*, 18 – 29.

Hausknecht, M., Au, T.-C., and Stone, P. (2011). Autonomous intersection management: Multi-intersection optimization.

Hollander, Y. and Prashker, J. N. (2006). The applicability of non-cooperative game theory in transport analysis. *Transportation*, 481–496.

Hoorfar, A. and Hassani, M. (2007). Approximation of the lambert w function and hyperpower function. *Research report collection*, 10.

Imbsweiler, J., Ruesch, M., Weinreuter, H., León, F. P., and Deml, B. (2018). Cooperation behaviour of road users in t-intersections during deadlock situations. *Transportation research part F: traffic psychology and behaviour*, 665–677.

Ims, A. B. and Pedersen, H. B. (2021). Simulation of automated vehicles in aimsun. Master's thesis, NTNU.

Jameel, F., Javed, M. A., Zeadally, S., and Jäntti, R. (2020). Efficient mining cluster selection for blockchain-based cellular v2x communications. *IEEE Transactions on Intelligent Transportation Systems*, 4064–4072.

Jastrzebski, W. (2000). Volume delay functions. In *15th International EMME/2 Users Group Conference*.

Javaid, S., Sufian, A., Pervaiz, S., and Tanveer, M. (2018). Smart traffic management system using internet of things. In *2018 20th international conference on advanced communication technology (ICACT)*, 393–398.

Jayakrishnan, R., Tsai, W. T., Prashker, J. N., and Rajadhyaksha, S. (1994). A faster path-based algorithm for traffic assignment.

Jiang, B. and Claramunt, C. (2004). A structural approach to the model generalization of an urban street network. *GeoInformatica*, 157–171.

Johari, R. and Tsitsiklis, J. N. (2004). Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 407–435.

Jones, E. G., Dias, M. B., and Stentz, A. (2011). Time-extended multi-robot coordination for domains with intra-path constraints. *Autonomous robots*, 41–56.

Karimi, K. (2012). A configurational approach to analytical urban design:'space syntax'methodology. *Urban Design International*, 297–318.

Kaths, J., Papapanagiotou, E., and Busch, F. (2015). Traffic signals in connected vehicle environments: Chances, challenges and examples for future traffic signal control. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 125–130.

Keferböck, F. and Riener, A. (2015). Strategies for negotiation between autonomous vehicles and pedestrians. *Mensch und Computer 2015–Workshopband*.

Kneissl, M., Molin, A., Esen, H., and Hirche, S. (2018). A feasible mpc-based negotiation algorithm for automated intersection crossing. In *2018 European Control Conference (ECC)*, 1282–1288.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2149–2154.

Konishi, H. (2004). Uniqueness of user equilibrium in transportation networks with heterogeneous commuters. *Transportation science*, 315–330.

Koutsoupias, E. and Papadimitriou, C. (1999). Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, 404–413.

Kucharski, R. and Drabicki, A. (2017). Estimating macroscopic volume delay functions with the traffic density derived from measured speeds and flows. *Journal of Advanced Transportation*.

Kurth, D. L., Van den Hout, A., and Ives, B. (1996). Implementation of highway capacity manual–based volume-delay functions in regional traffic assignment process. *Transportation research record*, 27–36.

Lamouik, I., Yahyaouy, A., and Sabri, M. A. (2017). Smart multi-agent traffic coordinator for autonomous vehicles at intersections. In *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 1–6.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 345–358.

Lee, H. L., Padmanabhan, V., and Whang, S. (1997). The bullwhip effect in supply chains. *Sloan management review*, 93–102.

Li, N., Kolmanovsky, I., Girard, A., and Yildiz, Y. (2018). Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control. In *2018 Annual American Control Conference (ACC)*, 3215–3220.

Li, R., Zhao, Z., Zhou, X., Palicot, J., and Zhang, H. (2014). The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice. *IEEE Communications Magazine*, 234–240.

Li, Z., Barenji, A. V., Jiang, J., Zhong, R. Y., and Xu, G. (2020). A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. *Journal of Intelligent Manufacturing*, 469–480.

Li, Z., Chitturi, M. V., Zheng, D., Bill, A. R., and Noyce, D. A. (2013). Modeling reservation-based autonomous intersection control in vissim. *Transportation research record*, 81–90.

Liard, T., Stern, R., and Delle Monache, M. L. (2020). Optimal driving strategies for traffic control with autonomous vehicles. *IFAC-PapersOnLine*, 5322–5329.

Lin, P., Liu, J., Jin, P. J., and Ran, B. (2017). Autonomous vehicle-intersection coordination method in a connected vehicle environment. *IEEE Intelligent Transportation Systems Magazine*, 37–47.

Liu, C., Lin, C.-W., Shiraishi, S., and Tomizuka, M. (2017). Distributed conflict resolution for connected autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 18–29.

Liu, H. X., He, X., and He, B. (2009). Method of successive weighted averages (mswa) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and Spatial Economics*, 485–503.

Liu, Y., Bunker, J., and Ferreira, L. (2010). Transit users' route-choice modelling in transit assignment: A review. *Transport Reviews*, 753–769.

Lovellette, E. and Hexmoor, H. (2021). Lane and speed allocation mechanism for autonomous vehicle agents on a multi-lane highway. *Internet of Things*, 100356.

Mackaness, W. (1995). Analysis of urban road networks to support cartographic generalization. *Cartography and Geographic Information Systems*, 306–316.

Mackaness, W. A. and Beard, K. M. (1993). Use of graph theory to support map generalization. *Cartography and Geographic Information Systems*, 210–221.

Márquez, L., García, D. E., and Guarín, L. C. (2014). Conical and the bpr volume-delay functions for multilane roads in bogota. *Revista de Ingeniería*, 30–39.

Mavromatis, I., Tassi, A., Piechocki, R. J., and Sooriyabandara, M. (2020). On urban traffic flow benefits of connected and automated vehicles. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 1–7.

Milchtaich, I. (2005). Topological conditions for uniqueness of equilibrium in networks. *Mathematics of Operations Research*, 225–244.

Monderer, D. and Shapley, L. S. (1996). Potential games. *Games and economic behavior*, 124–143.

Morlok, E. K. and Chang, D. J. (2004). Measuring capacity flexibility of a transportation system. *Transportation Research Part A: Policy and Practice*, 405–420.

Mosher Jr, W. W. (1963). A capacity-restraint algorithm for assigning flow to a transport network. *Highway Research Record*, (6).

Mounce, R. and Carey, M. (2015). On the convergence of the method of successive averages for calculating equilibrium in traffic networks. *Transportation science*, 535–542.

Mtoi, E. T. and Moses, R. (2014). Calibration and evaluation of link congestion functions: applying intrinsic sensitivity of link speed as a practical consideration to heterogeneous facility types within urban network. *Journal of Transportation Technologies*.

Nieuwenhuijsen, J. (2015). Diffusion of automated vehicles: a quantitative method to model the diffusion of automated vehicles with system dynamics.

of Public Roads, B. (1964). *Traffic assignment manual for application with a large, high speed computer*.

O'Hare, S. J., Connors, R. D., and Watling, D. P. (2016a). Mechanisms that govern how the price of anarchy varies with travel demand. *Transportation Research Part B: Methodological*, 55–80.

O'Hare, S. J., Connors, R. D., and Watling, D. P. (2016b). Mechanisms that govern how the price of anarchy varies with travel demand. *Transportation Research Part B: Methodological*, 55–80.

Olcott, E. S. (1955). The influence of vehicular speed and spacing on tunnel capacity. *Journal of the Operations Research Society of America*, 147–167.

Overgaard, K. R. (1967). Urban transportation planning'traffic estimation. *Traffic Quarterly*, 21.

Patriksson, M. (2015). *The traffic assignment problem: models and methods*.

Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., Rus, D., and Ang Jr, M. H. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 6.

Pettersson, I. and Karlsson, I. C. M. (2015). Setting the stage for autonomous cars: a pilot study of future autonomous driving experiences. *IET Intelligent Transport Systems*, 694–701.

Pigou, A. C. and Aslanbeigui, N. (2017). *The economics of welfare*.

Pinillos, R., Marcos, S., Feliz, R., Zalama, E., and Gómez-García-Bermejo, J. (2016). Long-term assessment of a service robot in a hotel environment. *Robotics and Autonomous Systems*, 40–57.

Porta, S., Crucitti, P., and Latora, V. (2006). The network analysis of urban streets: A dual approach. *Physica A: Statistical Mechanics and its Applications*, 853–866.

Powell, W. B. and Sheffi, Y. (1982). The convergence of equilibrium algorithms with predetermined step sizes. *Transportation Science*, 45–55.

Rawashdeh, Z. Y. and Mahmud, S. M. (2008). Intersection collision avoidance system architecture. In *2008 5th IEEE Consumer Communications and Networking Conference*, 493–494.

Reza, S., Oliveira, H. S., Machado, J. J., and Tavares, J. M. R. (2021). Urban safety: an image-processing and deep-learning-based intelligent traffic management and control system. *Sensors*, 7705.

Roessler, B. (2010). Status of european project intersafe-2 on cooperative intersection safety. In *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*, 381–386.

Rosenthal, R. W. (1973). A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 65–67.

Roughgarden, T. (2003). The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 341–364.

Roughgarden, T. and Tardos, É. (2002a). How bad is selfish routing? *Journal of the ACM*, 236–259.

Roughgarden, T. and Tardos, E. (2002b). How bad is selfish routing? *Journal of the ACM*, 236–259.

Sandholm, W. H. (2001). Potential games with continuous player sets. *Journal of Economic theory*, 81–108.

Sandholm, W. H. (2010). *Population games and evolutionary dynamics*.

Schechter, E. (1996). *Handbook of Analysis and its Foundations*.

Schleicher, S., Gelau, C., et al. (2011). The influence of cruise control and adaptive cruise control on driving behaviour–a driving simulator study. *Accident Analysis & Prevention*, 1134–1139.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2017). Airsim: High-fidelity visual and physical simulation for autonomous vehicles.

Shi, L. and Prevedouros, P. (2016). Autonomous and connected cars: Hcm estimates for freeways with various market penetration rates. *Transportation Research Procedia*, 389–402.

Shladover, S. E. (2018). Connected and automated vehicle systems: Introduction and overview. *Journal of Intelligent Transportation Systems*, 190–200.

Singh, R. and Dowling, R. (2002). Improved speed-flow relationships: application to transportation planning models. In *Seventh TRB Conference on the Application of Transportation Planning MethodsTransportation Research Board*

Smock, R. (1962). An iterative assignment approach to capacity restraint on arterial networks. *Highway Research Board Bulletin*, 347.

Song, J., Gupta, S., and Hare, J. (2014). Game-theoretic cooperative coverage using autonomous vehicles. In *2014 Oceans-St. John's*, 1–6.

Spiess, H. (1990). Conical volume-delay functions. *Transportation Science*, 153–158.

Sun, C., Guanetti, J., Borrelli, F., and Moura, S. J. (2020). Optimal eco-driving control of connected and autonomous vehicles through signalized intersections. *IEEE Internet of Things Journal*, 3759–3773.

Ta, N., Li, G., Zhao, T., Feng, J., Ma, H., and Gong, Z. (2017). An efficient ride-sharing framework for maximizing shared route. *IEEE Transactions on Knowledge and Data Engineering*, 219–233.

Taylor, M. (1997). The effects of lower urban speed limits on mobility, accessibility, energy and the environment: Trade-offs withincreased safety.

Thomson, R. C. and Richardson, D. E. (1995). A graph theory approach to road network generalisation. In *Proceeding of the 17th international cartographic conference*, 1871–1880.

Thorpe, C., Herbert, M., Kanade, T., and Shafer, S. (1991). Toward autonomous driving: the CMU Navlab. I. Perception. *IEEE Expert*, 31–42.

Tlig, M., Buffet, O., and Simonin, O. (2014). Decentralized traffic management: A synchronization-based intersection control. In *2014 International Conference on Advanced Logistics and Transport*, 109–114.

Toth, P. and Vigo, D. (2002). *The vehicle routing problem*.

Ulusoy, A., Smith, S. L., Ding, X. C., Belta, C., and Rus, D. (2013). Optimality and robustness in multi-robot path planning with temporal logic constraints. *The International Journal of Robotics Research*, 889–911.

VanMiddlesworth, M., Dresner, K., and Stone, P. (2008). Replacing the stop sign: Unmanaged intersection control for autonomous vehicles. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, 1413–1416.

Wagner, P. (2016). Traffic control and traffic management in a transportation system with autonomous vehicles. In *Autonomous Driving*, 301–316.

Wang, D. Z. and Du, B. (2016). Continuum modelling of spatial and dynamic equilibrium in a travel corridor with heterogeneous commuters—a partial differential complementarity system approach. *Transportation Research Part B: Methodological*, 1–18.

Wang, H., Meng, Q., Chen, S., and Zhang, X. (2021). Competitive and cooperative behaviour analysis of connected and autonomous vehicles across unsignalised intersections: a game-theoretic approach. *Transportation research part B: methodological*, 322–346.

Wang, J., Shao, Y., Ge, Y., and Yu, R. (2019). A survey of vehicle to everything (v2x) testing. *Sensors*, 334.

Wang, M., Chen, Z., Mu, L., and Zhang, X. (2020). Road network structure and ride-sharing accessibility: A network science perspective. *Computers, environment and urban systems*, 101430.

Wang, X., Agatz, N., and Erera, A. (2018). Stable matching for dynamic ride-sharing systems. *Transportation Science*, 850–867.

Wang, X., Xiao, N., Xie, L., Frazzoli, E., and Rus, D. (2015). Analysis of price of anarchy in traffic networks with heterogeneous price-sensitivity populations. *IEEE Transactions on Control Systems Technology*, 2227–2237.

Wardrop, J. G. (1952a). Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 325–362.

Wardrop, J. G. (1952b). Road paper. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 325–362.

Webster, F. V. (1958). Traffic signal settings. Technical report.

Wei, H., Mashayekhy, L., and Papineau, J. (2018). Intersection management for connected autonomous vehicles: A game theoretic framework. In *2018 21st International Conference on Intelligent Transportation Systems*, 583–588.

Wei, Y., Avcı, C., Liu, J., Belezamo, B., Aydın, N., Li, P., and Zhou, X. (2017). Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. *Transportation Research Part B: Methodological*, 102–129.

Weisstein, E. W. (2002). Lambert w-function. *https://mathworld. wolfram. com/*.

Wong, W. and Wong, S. (2016). Network topological effects on the macroscopic bureau of public roads function. *Transportmetrica A Transport Science*, 272–296.

Wu, Q., Wu, J., Shen, J., Du, B., Telikani, A., Fahmideh, M., and Liang, C. (2022). Distributed agent-based deep reinforcement learning for large scale traffic signal control. *Knowledge-Based Systems*, 108304.

Wurman, P. R., D'Andrea, R., and Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 9–9.

Wuthishuwong, C. and Traechtler, A. (2013). Vehicle to infrastructure based safe trajectory planning for autonomous intersection management. In *2013 13th International Conference on ITS Telecommunications* , 75–180.

Xie, Y., Zhang, Y., and Ye, Z. (2007). Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, 326–334.

Xu, B., Ban, X. J., Bian, Y., Wang, J., and Li, K. (2017). V2i based cooperation between traffic signal and approaching automated vehicles. In *2017 IEEE Intelligent Vehicles Symposium*, 1658–1664.

Xu, W., Wei, J., Dolan, J. M., Zhao, H., and Zha, H. (2012). A real-time motion planner with trajectory optimization for autonomous vehicles. In *2012 IEEE International Conference on Robotics and Automation*, 2061–2067.

Yan, P., Lee, C.-Y., Chu, C., Chen, C., and Luo, Z. (2021). Matching and pricing in ride-sharing: Optimality, stability, and financial sustainability. *Omega*, 102351.

You, C., Lu, J., Filev, D., and Tsiotras, P. (2019). Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 1–18.

Younis, O. and Moayeri, N. (2017). Employing cyber-physical systems: Dynamic traffic light control at road intersections. *Ieee Internet Of Things Journal*, 2286–2296.

Zhang, J., Pourazarm, S., Cassandras, C. G., and Paschalidis, I. C. (2016). The price of anarchy in transportation networks by estimating user cost functions from actual traffic data. In *2016 IEEE 55th Conference on Decision and Control*, 789–794.

Zhang, J., Pourazarm, S., Cassandras, C. G., and Paschalidis, I. C. (2018). The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies. *Proceedings of the IEEE*, 538–553.

Zhang, Q. (2004). Modeling structure and patterns in road network generalization. In *ICA Workshop on Generalisation and Multiple Representation*.

Zhang, Q. (2005). Road network generalization based on connection analysis. In *Developments in spatial data handling*, 343–353.

Zhao, L., Ichise, R., Yoshikawa, T., Naito, T., Kakinami, T., and Sasaki, Y. (2015). Ontology-based decision making on uncontrolled intersections and narrow roads. In *2015 IEEE intelligent vehicles symposium*, 83–88.