SIMULATION AND OPTIMISATION OF DYNAMIC MULTIMODAL TRAFFIC NETWORK WITH SHARED MOBILITY SYSTEMS

Peeranut Jeamchantamook

A thesis submitted for the degree of Doctor of Philosophy of

Imperial College London

Centre for Transport Studies

Department of Civil and Environmental Engineering

Imperial College London, United Kingdom

January 2026

Declaration of Originality

I hereby declare that this thesis and the work originated and conducted by myself. Information and data sourced from other publications are duly acknowledged in the text and references.

Peeranut Jeamchantamook (2026)

"The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law."|

ABSTRACT

Rapid urbanisation continues to intensify traffic congestion, a challenge compounded by the “last-mile” problem and limited understanding of shared mobility systems. Existing research frequently employs isolated models, failing to capture complex interdependencies spanning network effects, matching algorithms, environmental outcomes, HOV lane performance, pricing, and public transport interactions.

This thesis addresses this gap by developing a comprehensive, computationally efficient, macroscopic multimodal traffic simulation framework. It integrates ridesharing, taxi-sharing, private, and public transport within a unified environment, incorporating within-day and day-to-day dynamics. A novel macroscopic matching algorithm, formulated using linear programming and hierarchical clustering, is introduced to efficiently accommodate large-scale demand while maintaining tractability.

Simulations demonstrate that integrating matching algorithms with route choice models is essential, as isolated analyses overestimate uptake by ignoring induced demand. Results show shared mobility systems can reduce congestion by up to 17.5% while improving network speeds by 4.14%. Substantial reductions in emission-related indicators like vehicle-kilometres travelled are also observed.

Policy design is decisive: maximising Vehicle Kilometres Travelled Saved (VKTS) reduces congestion by 17.22%, whereas profit-maximisation achieves 15.9%. Surge pricing raises revenue, but excessive factors may worsen congestion. HOV lanes promote ridesharing but require calibration to avoid displacing congestion. The framework also highlights the competitive relationship between shared mobility and public transport, noting bus ridership declines in low-congestion areas. Finally, the model determines optimal fleet sizes under different objectives.

These findings demonstrate that the proposed framework provides planners and researchers with an advanced tool for investigating shared mobility's long-term impacts. It supports the design of data-driven strategies to reduce congestion, promote sustainability, and integrate shared mobility effectively into urban systems.

ACKNOWLEDGEMENTS

It is with immense gratitude that I acknowledge the support and help of my supervisor, Prof. Washington Yotto Ochieng. His in-depth academic knowledge, nurturing mentorship, and unwavering empathy were instrumental in my journey, guiding me to this point. This thesis would not have been possible without his unyielding guidance.

I would like to pay tribute to Miss Sarah Willis, whose dedication and hard work have been nothing short of extraordinary. Her efforts in assisting me through the various challenges I faced throughout this PhD program have been invaluable.

I owe a profound debt of gratitude to Dr. Ke Han, who provided crucial advice during the early stages of my PhD. His insights on conducting research effectively laid the foundations of my work and helped shape this thesis. I am deeply thankful for his input and guidance.

At a personal level, I would like to express my heartfelt thanks to my parents and my girlfriend, who have always been my support throughout this difficult journey. Their belief in me, even in the most demanding times, has been my source of strength. I am forever indebted to them for their unwavering faith and unrelenting support.

Lastly, I wish to extend my sincere appreciation to the Department of Highways, Ministry of Transport, Thailand, for their generous scholarship. This support made it possible for me to dedicate myself to this research and provide a lifetime experience in London, UK.

I would also like to express my gratitude to Dr. Songrit Chayanan for introducing me to this path and encouraging me to pursue it. His mentorship has been an inspiring beacon on this journey.

สุดท้ายแล้ว เราก็ต้องเดินต่อไป...

TABLE OF CONTENTS

ABSTRACT 3
TABLE OF CONTENTS 7
LIST OF FIGURES 13
LIST OF TABLES 15
LIST OF EQUATIONS 16
CHAPTER 1 INTRODUCTION 19
1.1 Background 19
1.2 Aim and objectives. 23
1.3 Outline of the thesis 24
CHAPTER 2 STATE OF THE ART IN SHARED MOBILITY STUDIES 29
2.1 Background 30
2.2 Development of Ridesharing system 31
2.2.1 Taxi-sharing system 34
2.2.2 Evolution of Ridesharing and Taxi-sharing system 36
2.3 Impact of Shared mobility system 39
2.4 Studies on shared mobility 41
2.4.1 Impact on the road network 42
2.4.2 Better ridesharing and taxi-sharing matching algorithm 44
2.4.3 Environmental impact 46
2.4.4 Pricing scheme 48

2.4.5 Efficiency of High-Occupancy Vehicle lanes 49
2.4.6 Shared mobility and public transport 51
2.4.7 Optimum fleet size 52
2.5 Model integrations 53
CHAPTER 3 ROUTE CHOICE MODEL FOR SHARED MOBILITY SYSTEM 56
3.1 Backgrounds 57
3.2 Optimization model characteristic 57
3.2.1 Objective functions for matching algorithm 58
3.2.2 Constrains in matching algorithms. 60
3.3 Optimization models in shared mobility system 64
3.3.1 Binary or Mixed Integer programming 65
3.3.2 Linear programming 67
3.4 Single Driver – Multiple passengers 68
3.5 Matching algorithm for taxi sharing 73
3.5.1 Vacant taxi movement model 75
3.6 Discussion 79
CHAPTER 4 ROUTE CHOICE MODEL FOR PUBLIC TRANSPORT 82
4.1 Backgrounds 83
4.2 Initial approaches 85
4.3 Strategy approaches 88
4.4 Route section approach 90
4.5 Congestion in transit networks 92

4.6 User Equilibrium and Stochastic assignment in transit network 95
4.7 Discussion.....97
CHAPTER 5 ROUTE CHOICE FOR PRIVATE TRANSPORT 100
5.1 Backgrounds.....101
5.2 All or Nothing assignment 102
5.3 Volume-Delay functions.....104
5.4 Wardrop's equilibrium.....108
5.4.1 User Equilibrium.....109
5.5 Heuristic Equilibration techniques.....112
5.5.1 Incremental assignment 114
5.5.2 Method of Successive Averages 115
5.5.3 Frank-Wolfe algorithm.....118
5.6 Stochastic traffic assignment.....120
5.6.1 Dial's method.....122
5.6.2 Logit-based method 123
5.7 Stochastic User Equilibrium.....125
5.8 Discussion.....127
CHAPTER 6 METHODOLOGY 131
6.1 Overview 131
6.2 Mode choice model 136
6.3 Route choice models 145
6.3.1 Matching algorithms.....146
6.3.2 Vacant taxi model.....152

6.3.3 K-shortest path algorithm 156
6.3.4 Transit assignment model 158
6.3.5 Stochastic traffic assignment 164
6.4 Within day traffic model 165
6.4.1 Actual route travel time 166
6.4.2 Flow propagation model 168
6.5 Day-to-day traffic model. 171
6.6 Discussion 175
CHAPTER 7 NUMERICAL EXAMPLE 178
7.1 Sioux Fall network problem 179
7.2 Numerical Convergence: Ridesharing system 183
7.3 Numerical Convergence: Taxi sharing system. 191
7.4 Sensitivity analysis Framework 196
7.4.1 Rationale for Parameter Selection 196
7.4.2 Specification of Parameter Ranges 197
7.5 Sensitivity analysis: Ridesharing system 198
7.6 Sensitivity analysis: Taxi-sharing system 201
7.7 Stress test 204
7.8 Heuristic approach to reducing computational costs. 208
7.8.1 Similarity measures 210
7.8.2 Clustering Performance 213
7.9 Discussion 216
CHAPTER 8 IMPACT OF SHARED MOBILITY SYSTEMS 219

8.1 Baseline Road Network (No Shared Mobility) 220
8.2 Impact on the road network. 223
8.3 Development of matching algorithm 228
8.4 Environmental Impact 230
8.5 Efficiency of High-Occupancy Vehicle lanes 236
8.6 Impact to the Public transport 246
8.7 Pricing scheme 252
8.8 Optimum fleet size 257
8.9 Discussion 260
CHAPTER 9 CONCLUSION AND FUTURE WORKS 263
9.1 Revisiting Research Objectives 264
9.1.1 First Objective 264
9.1.2 Second objective 266
9.1.3 Third Objective 268
9.2 Contributions in model development 272
9.2.1 Macroscopic matching algorithms 272
9.2.2 Rolling horizon approach 272
9.2.3 Hierarchical clustering 273
9.3 Policy and Practical Implications 274
9.4 Limitations of the Framework 276
9.4.1 Traveller Behaviour and mode choices 276
9.4.2 Public Transport Modelling 277
9.4.3 Shared Mobility Modelling 278

9.4.4 General Modelling and Scope.....279
9.4.5 Validation and Applicability 280
9.5 Future works.....281
9.5.1 Dynamic ridesharing.....281
9.5.2 External validations 286
9.5.3 A more realistic traffic propagation model. 287
9.5.4 Multimodal transportation 288
REFERENCES.....291
APPENDIX A Variable and Equation lookup tables 306
Appendix A1 Variable lookup table.....307
Appendix A2 Equation lookup table.....309
APPENDIX B Simulation Code descriptions.....319
Appendix B1 Input Data.....320
Appendix B2 Definition of Functions 323
Appendix B3 Data Dictionary 326
APPENDIX C Source Code (Ridesharing system) 334
APPENDIX D Source Code (Taxi-sharing system) 365
APPENDIX E Detailed Numerical Examples.....402
Appendix E1 Incremental Assignment 403
Appendix E2 MSA algorithm 404
Appendix E3 Frank-Wolfe algorithm 405
Appendix E4 Modified MSA algorithm 408
APPENDIX F Matching results.....409

LIST OF FIGURES

Figure 1.1 Stages of model development.....24
Figure 3.1 Single driver and single passenger ridesharing problem.....60
Figure 3.2 Single driver - Multiple passengers matching patterns. 69
Figure 3.3 Topology of taxi-sharing system 73
Figure 4.1 Simple transit network with common lines problems.....87
Figure 4.2 Transit network represented with route selection approach. 91
Figure 5.1 Simple traffic network with three routes.....103
Figure 5.2 Traffic network and corresponding link cost functions.....113
Figure 6.1 Overall simulation framework based on the simulation time. 136
Figure 6.2 Factors affecting the mode choice model. 144
Figure 6.3 Detailed framework of the simulation.....146
Figure 6.4 Example of transit network (Original) 159
Figure 6.5 Example of transit network (Modified).....160
Figure 6.6 The within-day traffic model.....166
Figure 7.1 The Sioux network. With two bus lines 182
Figure 7.2 The long-term behaviour of traveller's mode choice with the empirical convergence of model split 187
Figure 7.3 The performance of the matching algorithm under different demand scenarios.....189
Figure 7.4 A convergence result of the taxi-sharing simulation 193
Figure 7.5 The total network cost based on different vacant taxi models. 194
Figure 7.6 A sensitivity analysis test on ridesharing system 199
Figure 7.7 A sensitivity analysis test for taxi-sharing system.....202
Figure 7.8 The sensitivity analysis of modal split when increasing fare. 204
Figure 7.9 Matching possibilities.....205

Figure 7.10 The computational time required by the matching algorithm. 207
Figure 7.11 Sioux Fall network with mixed OD demands (rectangles = origins, triangles = destinations, colours = different OD pairs) 211
Figure 7.12 Similarity measures; OD and Saved Distances 212
Figure 7.13 The performance in different size of cluster 214
Figure 8.1 Network Link flow (the best well knowns) 221
Figure 8.2 Network Link flow (based scenario - no ridesharing) 221
Figure 8.3 Number of matched drivers in the ridesharing system 225
Figure 8.4 Network Link flow (single driver-single passenger) 227
Figure 8.5 Network Link flow (profit maximization) 229
Figure 8.6 Network Average speed (no ridesharing) 233
Figure 8.7 Network Average speed (with ridesharing) 233
Figure 8.8 Network Average speed (with High-Occupancy Vehicle lanes) 241
Figure 8.9 Network with two types of bus operations 248
Figure 8.10 Network Average speed (with taxi-sharing) 254
Figure 8.11 Network Average speed (with taxi-sharing with surge pricing $\rho = 0.25$) 255
Figure 8.12 Impact of surge pricing on the road network 256
Figure 8.13 The initial effects of incrementally adding taxi vehicles to the taxi-sharing system. 259
Figure 9.1 Matching algorithm results (a) without dynamic matching (b) with dynamic matching 284
Figure 9.2 An example of using shared mobility for last-mile problem 289

LIST OF TABLES

Table 6.1 Modes of transport that travellers can be accessed in ridesharing scenario. 141
Table 6.2 Modes of transport that travellers can be accessed in taxi-sharing scenario. 141
Table 6.3 The transit data used as the numerical example..... 162
Table 6.4 Numerical example of flow propagation (vehicle per 15 minutes) .. 170
Table 7.1 An initial mode split on d = 1, for time horizon t = 1:4 184
Table 7.2 The total traveller's origin-destination demands used in the taxi-sharing simulation..... 192
Table 8.1 Example of Sioux Falls link's properties 232
Table 8.2 Links that incorporate High-Occupancy Vehicle lanes..... 238
Table 8.3 Auxiliary links exclusively for High-Occupancy Vehicle lanes. 239
Table 8.4 Implementing High-Occupancy Vehicle lanes without ridesharing system..... 241
Table 8.5 Implementing High-Occupancy Vehicle lanes with ridesharing system (30 percent capacity reduction)..... 243
Table 8.6 Implementing High-Occupancy Vehicle lanes with ridesharing system (10 percent capacity reduction)..... 244
Table 8.7 The performance of buses system (High congestion area buses) 249
Table 8.8 The performance of buses system (Low congestion area buses)..... 249

LIST OF EQUATIONS

Equation 3-1 Standard Optimisation Problem 58
Equation 3-2 Time Constraints 61
Equation 3-3 Cost Constraints 62
Equation 3-4 Binary Integer Programming 66
Equation 3-5 Cost constraints in the single driver – multiple passenger matching problems. 70
Equation 3-6 Cost constraint for Taxi Driver 74
Equation 3-7 Wong's Rate of Return 78
Equation 3-8 Wong's Probabilistic function 78
Equation 4-1 Expected waiting time under random arrivals 86
Equation 4-2 Expected Travel Time with a set of Route 89
Equation 4-3 Expected Travel Time with multiple service frequency 91
Equation 4-4 De Cea's Cost functions 93
Equation 5-1 Smock's Volume-Delay function 105
Equation 5-2 The UK DoT Volume-Delay function 105
Equation 5-3 The UK DoT Time-Flow function $t = t01 + \alpha VC\beta$ Equation 5-4 The US BPR Volume-Delay function 106
Equation 5-5 Akcelik's Volume-Delay function 107
Equation 5-6 Beckmann's UE formulation 111
Equation 5-7 Sheffi's SO formulation 112
Equation 5-8 Accumulate flows 114
Equation 5-9 MSA flow calculation 116
Equation 5-10 Zhang's sequence 117
Equation 5-11 Franke-Wolfe linear function 118
Equation 5-12 Franke-Wolfe algorithm 119

Equation 5-13 Utility function with stochastic attribute.....120
Equation 5-14 Probabilistic function 121
Equation 5-15 Dial's Forward pass 122
Equation 5-16 Dial's Backward pass 123
Equation 5-17 MNL function.....124
Equation 5-18 Commonality function 124
Equation 5-19 C-Logit function.....124
Equation 5-20 SUE optmisation problem.....125
Equation 6-1 MNL function.....137
Equation 6-2 Nested Logit function.....137
Equation 6-3 Revised MNL function.....142
Equation 6-4 Proposed mode cost functions.....143
Equation 6-5 Matching possibility matrices 147
Equation 6-6 Linear Programming 147
Equation 6-7 Proposed optimisation problem for Ridesharing 149
Equation 6-8 Proposed optimisation problem for Taxi-sharing.....150
Equation 6-9 Proposed Rate of Return function.....153
Equation 6-10 ATT Revenue probabilistic function 154
Equation 6-11 Proposed expected travel time 161
Equation 6-12 Stochastic traffic assignment.....164
Equation 6-13 Link flow conservation 167
Equation 6-14 Link Travel Time calculation 167
Equation 6-15 Actual Route Travel Time 168
Equation 6-16 Number of horizons.....169
Equation 6-17 Flow Propagation model 170
Equation 6-18 Crowdedness factor.....173
Equation 6-19 Perceived mode cost.....174

Equation 8-1 Optimisation problem for Single Driver - Single Passenger matching 224
Equation 8-2 Profit maximisation problem 228
Equation 8-3 Surge Pricing 254

CHAPTER 1 INTRODUCTION

This chapter sets the foundation for the thesis, "Simulation and optimisation of dynamic multimodal traffic network with shared mobility systems," by introducing the critical context of intensified urban traffic congestion and the limited understanding of shared mobility systems impacts. It defines the scope, focusing primarily on ridesharing and taxi-sharing, and outlines the research gap: the lack of a comprehensive, computationally efficient, and macroscopic framework capable of assessing the complex network-wide interactions across multiple transport modes and dynamic conditions. The chapter concludes by explicitly stating the aim to develop such a comprehensive framework and defining the three core objectives that guide the subsequent research stages: framework development, matching algorithm innovation, and model evaluation against timely research questions.

1.1 Background

Shared mobility is defined as transportation systems in which rides or trips are shared between travellers. The sharing system could occur simultaneously (Ridesharing system) or over time (Carsharing, Bike sharing and Scooter sharing). It also includes transit services such as Dial-a-rides and paratransit. The shared mobility system emerged in the late 20th century, especially in the U.S. However, the system was not popular until more recent years. With advancements in technology, several taxi e-hailing companies successfully implemented the system. Shared mobility lies between private and public

transport, where the trips are more tailored for each passenger. On the one hand, shared mobility is not private transport as a trip or other individuals sharing a vehicle. On the other hand, the system is also not public transport. Although other users are involved, the trip is more tailored for each individual and demand-responsive.

A Ridesharing system is a shared mobility system where travellers (i.e., passengers) share a trip and split travel costs with the driver, who privately owns the vehicle for the trip. North America is the origin of the ridesharing system. The system began in the U.S. during World War II. when the U.S. government introduced the first organised ridesharing system named the 'Car-Sharing club'. The reason behind this introduction was to help the war effort by rationing gasoline (Frohardt-Lane, 2012). The early matching system is offline and located at community boards or bus stops. This arrangement requires more time for users to be matched, and users require a longer commitment to their trip. Hence such systems are not popular (Oliphant et al., 2013). Today the matching happens in a timely fashion via mobile applications or websites. Some taxi e-haling companies such as Uber, Didi and Lyft implemented ridesharing systems on their fleet. This embedded ridesharing system allows taxis to service more passengers simultaneously, ultimately increasing the company's revenue. Such a system is commonly referred to as a Taxi-sharing system.

The benefit of shared mobility systems is not limited to the operator’s increasing profit but many stakeholders. For example, ridesharing drivers have a new source of income collected from passengers’ fares. In addition, passengers have an alternative mode of transport that costs less than taxis but is more convenient

than public transport. At the network-wide level, the benefits of shared mobility have been assessed in many studies using empirical methods. These studies focused on parameters such as travel delays, fuel consumption and vehicle emission (Chan & Shaheen, 2012; Ferguson, 1997; Kelly, 2007; Minett & Pearce, 2011) Due to these benefits, shared mobility systems such as ridesharing and taxi-sharing are widely accepted by the user community and have become more prevalent during this decade. Therefore, such a system has emerged rapidly in many cities around the globe.

Shared mobility systems have an impact in several aspects. First and foremost is the impact on the road network. Systems such as ridesharing can reduce the need for vehicle ownership by implementing a new mode of transport with accessibility and convenience similar to a private vehicle. This can ultimately reduce traffic congestion in the road network. The second impact is an environmental impact. With a reduction in vehicular demand, emissions generated from vehicles could be reduced. However, sharing vehicles transverse in the network longer than the non-sharing ones due to the pickup and drop-off of other travellers' processes. This could lead to a higher emission generated per trip. The third impact is the impact on public transport. Shared mobility system act similarly to public transport systems such as buses or underground. It is still unclear whether the impact of shared mobility is a substitute or a complement. On the one hand, shared mobility can serve as a last-mile service between travellers' origin/final destination and public transport hubs. On the other hand, the service can compete with the public transport system as a new mode of transport. Due to the rapid emergence of shared mobility systems around the globe, there is a limited number of studies in this area.

We have reviewed several forms of academic documentation, such as journals and conference papers, between 2017-2021. Studies related to shared mobility systems can be categorised into seven topics based on popularity: Impact on the road network, Matching algorithm development, Environmental impact, Pricing scheme, Efficiency of HOV lanes, Effect on public transport and Optimum fleet. Most researchers develop a model that could tackle one or two abovementioned topics. In fact, the seven topics are all related and dependent. For example, developing a matching algorithm can alter the system's performance, which impacts the road network and the environment. Allowing sharing vehicles to enter HOV lanes may impact the public transport system. In addition, the impact of the shared mobility system on public transport is not yet thoroughly explored. Demand balancing tools such as pricing schemes and optimum fleet management are essential not only to the operators but also to the network-wide. To the best of our knowledge, there is no research capable of handling all these topics. Due to a lack of a comprehensive traffic simulation with shared mobility systems that at least tackles all the main research topics mentioned above, the studies on the impact of shared mobility systems are not yet fully explored.

The main challenge of the study is to develop a model to simulate the route choice behaviour of shared mobility users. The model can be divided into two approaches (matching algorithm and traffic assignment). The matching algorithm is a microscopic route choice simulation, while the traffic assignment is a macroscopic simulation. Both approaches have pros and cons depending on the usage and scale. In order to develop a simulation for shared mobility systems that can answer the timely research questions which are reviewed in Chapter 2. These questions are the trending topics in shared mobility research, such as impacts on the road network, development of matching algorithms,

environmental impacts, pricing schemes, the use of High-Occupancy Vehicle (HOV) lanes, optimum fleet size, and the interaction with public transport. To address these questions, it is necessary to design and consider the simulation framework thoroughly, so that the model can provide consistent and realistic outcomes This development includes selecting and modifying multiple route choice models for all related modes of transport that will be encapsulated in the framework and work seamlessly. The process of integration and the complete simulation framework are described in Chapter 6.

1.2 Aim and objectives.

Given the abovementioned issues, the aim and objective of this thesis are:

- To develop a comprehensive traffic simulation with two shared mobility systems: ridesharing and taxi-sharing system with following features:
  1) multiple modes of travel: ridesharing, taxi-sharing, private transport, and public transport.
  2) multiple time scales: within-day and day-to-day dynamics variation.
  3) mode switching, ridesharing drivers and passengers can switch their modes if they are unmatched.
  4) macroscopic model, which suffers less computational costs when simulating the traffic.
  5) multiple route assignments: each mode of transport is represented by its own route assignment model, designed specifically for that mode rather than using a single generic model across all modes, which helps to maintain the realistic behaviour of travellers in the simulation

- To develop a matching algorithm that determines route choice models for shared mobility vehicles in a macroscopic level.
  1) multiple passengers: the matching algorithm can match up to two passengers with different OD simultaneously.
  2) realistic matching algorithm: based on Linear Programming that can exclude unsuitable matchings.

- to evaluate the capability of the proposed model against the timely research questions identified in Chapter 2, by running simulations in Chapter 7 and Chapter 8

1.3 Outline of the thesis

The thesis consists of eight chapters. Each chapter are outlined in the following figure.

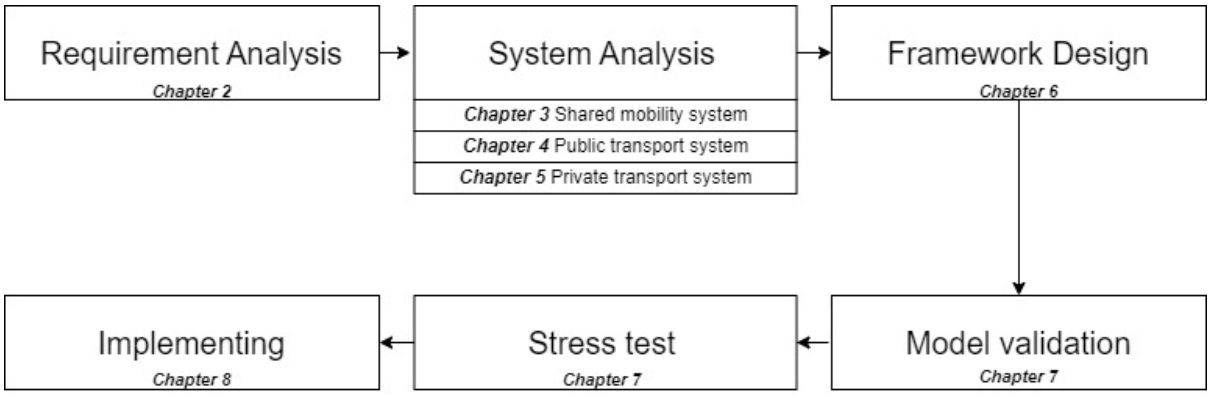

Figure 1.1 Stages of model development

Chapter 1: Introduction

The chapter introduces the background information on shared mobility systems (Ridesharing and Taxi-sharing), defines the problem and states the aim and objectives of the research.

Chapter 2: State-of-the-art in shared mobility studies

This chapter started by paving the basic knowledge of shared mobility systems. It also emphasises the system's benefits, drawbacks, limitations, and impact of such system. The section is followed by a review of the seven most popular research topics in shared mobility systems. The latter part of the chapter discussed the existing studies gap before summarising the requirements and the contributions of the proposed simulation in this thesis. The chapter then concluded with challenges in combining multiple route choice models for different modes of transport into the framework.

Chapter 3: Route choice model for Shared mobility systems

This chapter focused on the route choice model for shared mobility systems. The chapter is started with a basic knowledge of the optimisation procedure and how it applies to shared mobility systems. This section is followed by a review of the development of state-of-the-art ridesharing and taxi-sharing matching algorithms based on optimisation models. Note that algorithms are used at the microscopic level, where demands are treated individually. We then briefly explain how we remodel a microscopic algorithm into a macroscopic model that can be connected with models in other chapters. A more complex route choice model, such as single driver – multiple passengers' arrangement, is explained

excessively in this chapter. In addition, a vacant taxi movement model required for modelling a taxi-sharing system is also introduced within the chapter.

Chapter 4: Route choice for public transport

The chapter is focused on the route choice model for public transport systems called transit assignment models. We first briefly introduce the development of different transit assignment models, separated into two types; frequency and scheduled-based transit assignment. The chapter then focused on the development frequency-based model, which is compatible with route choice models in other modes.

Chapter 5: Route choice for private transport

The chapter is focused on the route choice model for solo-driving vehicles. Studies in the traffic assignment model can be categorised into two parts: static and dynamic traffic assignment. We then introduce our approach to a semi-dynamic traffic assignment developed from the static assignment. The chapter then reviewed different methods to predict road users' route choice behaviour. These methods include the infamous All or Nothing assignment, User equilibrium assignment and stochastic assignment. Studies in this assignment are reviewed in detail with a numerical example. Finally, the chapter reviews different heuristic techniques that reduce the assignment's computational complexity, such as the Frank-Wolfe algorithm.

Chapter 6: Methodology

After we present the route choice models on every mode of transport (Ridesharing, Taxi-sharing, Public Transport and Solo driving), we then propose our general model framework and the model integration. Next, we introduce our day-to-day and within-day traffic assignment model, which will be used to determine the equilibrium.

Chapter 7: Numerical examples

The chapter starts from providing a numerical example to validate our proposed simulation. This includes in finding a converged solution as well as perform some internal validation. These tests are necessitated to ensure that our simulation framework generate mathematically correct solution, generates plausible results and making sense. In addition to this we perform a sensitivity analysis to see how modal splits changes based on various factors. In the latter part of the chapter, we perform a stress test to determine the bottleneck of our framework from a larger scale network and data. Based on findings from the stress test, we proposed a heuristic approach to mitigate the computational cost of the framework while remaining the efficiency

Chapter 8: Impact of shared mobility systems

This chapter aims to demonstrate our proposed simulation and its capabilities of our simulation framework as an advanced comprehensive tool to help researchers studying shared mobility systems more extensively. We perform our simulation with a more complex dataset on a research topic highlighting in Chapter 2 Through multiple simulation and modifying objectives and minor

variables, we successfully performed simulations providing basic insights into these topics. We also claim that our simulation model is necessary to provide a more realistic and comprehensive results compared with those models that develop for a specific topic.

Chapter 9: Conclusion and Future works

In this final chapter, we conclude the main findings of the thesis, summarises main contributions of this work and also provide potential topics for future research in this area.

In concluding this introductory chapter, the context for the subsequent research has been established, defining the problem posed by emerging shared mobility systems within congested urban networks. The core aim—to develop a comprehensive and dynamic multimodal traffic simulation—is linked to three specific objectives, which necessitate a structured approach involving an extensive literature review, detailed methodological development, and robust numerical evaluation. The following four chapters will execute the first required task by systematically reviewing the existing state-of-the-art concerning shared mobility studies (Chapter 2) and the route choice mechanisms for shared mobility, public transport, and private vehicles (Chapters 3, 4, and 5), providing the theoretical foundation required to meet these objectives.

CHAPTER 2 STATE OF THE ART IN SHARED MOBILITY STUDIES

This chapter serves as a comprehensive literature review, fulfilling the necessary foundational step for developing a robust simulation framework. It begins by defining the core concepts of shared mobility, tracing the historical development of ridesharing, from early car-sharing clubs to modern app-based taxi-sharing and microtransit systems. The primary focus is then placed on identifying seven prominent and timely research topics in shared mobility studies—including network impact, algorithm development, and public transport interaction—which collectively highlight the existing academic consensus and, crucially, the gaps where current models operate in isolation, failing to capture complex system-wide interdependencies.

The review strategy focused on peer-reviewed journals and conference papers published between 2017 and 2021. Search platforms included Web of Science and Google Scholar, utilizing keywords such as "shared mobility," "ridesharing," "carpooling," "taxi-sharing," "network impact," and "matching algorithm". Only studies focused on quantitative modelling or system-level analysis were included; purely qualitative studies or those focused on traditional, static carpooling were excluded to maintain focus on modern, app-based shared systems.

2.1 Background

The term shared mobility is used in transport systems in which rides or trips are shared between travellers with some common means. The sharing system could occur simultaneously (Ridesharing system) or over time (Carsharing, Bike sharing and Scooter sharing). Shared mobility system such as ridesharing, involves multiple individuals sharing a single vehicle to reach a common or nearby destination. This type of shared mobility is typically facilitated by ride-hailing companies like Uber and Lyft, which use mobile applications to connect passengers with nearby drivers who are traveling in the same direction. Ridesharing services can help reduce traffic congestion, lower emissions, and provide cost savings for users, as they only pay for the portion of the ride they use. Over time shared mobility, on the other hand, involves multiple individuals sharing a vehicle over a longer period, such as days, weeks, or months. This type of shared mobility is often facilitated by carsharing services like Zipcar and Car2Go, which allow users to rent vehicles by the hour or day. Over time shared mobility services can help reduce the number of cars on the road, provide increased flexibility for users, and reduce the financial burden of car ownership. It also includes transit services such as Dial-a-rides and paratransit.

On the one hand, shared mobility is not a private transport as a trip, or a vehicle is shared by other individuals. On the other hand, the system is also not public transport. Although other users are involved, the trip is more tailored for each individual and demand responsive. Several studies concluded that shared mobility has a wide range of impacts, such as reducing traffic congestion, pollution, and vehicle ownership. In this thesis, we will focus on a shared

mobility system in which the sharing occurs among travellers simultaneously (Ridesharing and Taxi sharing). Although there have been several studies regarding the system, the wording is still ambiguous. Terminology such as carpooling, taxi-sharing, and taxi-ridesharing are also popular and broadly used. Throughout this thesis, we denote the words' Ridesharing' as a shared mobility system in which traveller share their trips in a privately owned vehicle and 'Taxi-sharing' in a taxi vehicle. These forms of shared mobility have evolved through various models, ranging from structured corporate carpooling programs designed for commuter benefits to demand-responsive taxi-sharing services operating at key transit hubs such as train stations and airports, ultimately leading to today's highly dynamic, app-based systems

2.2 Development of Ridesharing system

A Ridesharing system is a shared mobility system that involves multiple individuals sharing a single vehicle to reach a common destination. In this system, passengers share a trip and split travel costs with the driver who privately owns the vehicle for the trip. Ridesharing has become increasingly popular in recent years due to its potential benefits, including reduced traffic congestion, lower emissions, and cost savings for users. North America is considered to be the origin of the ridesharing system. The concept of ridesharing can be traced back to World War II when the United States government introduced the first organized ridesharing system named the "Car-Sharing Club." The government introduced this system as part of the war effort to help ration gasoline and conserve energy resources Frohardt-Lane, S. (2012). Although the name was 'Car-Sharing club', the government encouraged their citizens to share

their rides with others. Hence, the system was ridesharing. Then after the end of World War II, the system became less attractive and finally disappeared from public notice (Ferguson, 1991).

The concept of ridesharing remerged in the 1970s in response to the oil crisis, which led to fuel shortages and skyrocketing gas prices. In the United States, many major cities responded to the crisis by creating High Occupancy Vehicle (HOV) lanes, which were reserved for vehicles with two or more occupants. The HOV lanes were intended to encourage carpooling and reduce congestion on the roads. During this time, drivers and passengers had no formal matching process for ridesharing. Instead, drivers privately offered rides to passengers at various bus stops or other public locations. Carpooling became a popular and effective way for commuters to save money on gas and reduce their carbon footprint (Concas & Winters, 2007; Javid, Nejat & Salari, 2016) .

In the early 1990s, many cities in the United States initiated telephone-based ridesharing as a new approach to encourage more people to carpool. The system aimed to connect passengers and drivers who were interested in ridesharing by registering their travel needs via a telephone line. This approach was designed to be more accessible to a wider range of users compared to the traditional approach, where drivers and passengers had to physically meet and coordinate their rides. The telephone-based ridesharing system used human operators and automated systems to match the travel needs of passengers and drivers. Passengers would call a central phone number to register their desired pick-up and drop-off locations, and the system would match them with drivers who were traveling in the same direction. The service fees for this system were generally

higher than for traditional carpooling, which may have contributed to its lack of success (Kowshik, Kitamura & Jovanis, 1996).

There was an attempt to enhance telephone-based ridesharing at the University of Washington. The university introduced the system along with Internet and E-mail capabilities. However, the system was not widely used and limited to the university staff and students but was more successful than the prior system. There was a record with over 500 requests and 150 potential matchings. One of the main reasons for this success was its closed environment. This relieved some personal security concerns, which is a major drawback of the system.

In recent years, ridesharing systems have undergone significant developments due to the emergence of new communication capabilities such as the Global Positioning System (GPS) and mobile internet technology. This modernized version of ridesharing is often referred to as 'Dynamic Ridesharing' or 'Real-time Ridesharing', depending on the nature of the system. Dynamic Ridesharing offers a booking process that can be made via computers and mobile applications. Sophisticated matching algorithms undertake bookings and can match drivers and passengers within a short period. This rapid matching enables drivers and passengers to post their requests any time before the trip, providing ridesharing users with more flexibility and less commitment compared to traditional approaches.

Dynamic Ridesharing systems offer improved security measures, where all users must prove their identity before registering on the platform. These measures help to ensure that both drivers and passengers feel safe when using the service. The user base of Dynamic Ridesharing systems is far superior to that of earlier systems, covering national levels compared to previous versions that only served a regional level. Dynamic Ridesharing systems have gained popularity rapidly in recent years. BlablaCar, a French-based Dynamic Ridesharing company, reported in 2019 that the company had over 87 million ridesharing users and covered 400 cities in Europe, with a 71% growth in revenue. The success of BlablaCar and other ridesharing companies is a testament to the effectiveness and convenience of Dynamic Ridesharing systems in meeting the transportation needs of people. However, the industry continues to face challenges and scrutiny, such as concerns around safety, labour practices, and regulation, which must be addressed for the industry's sustainable growth and continued success.

2.2.1 Taxi-sharing system

The Taxi-sharing system is a relatively new ridesharing system that specifically caters to taxi services. This system has emerged in recent years and can be found in Taxi E-hailing companies such as Uber, Lyft, and Didi Chuxing. Unlike a regular taxi service where customers wait on the pavement for a vacant taxi, these companies provide a hailing platform on websites and mobile applications where customers can request a taxi from the company's fleet. The hailing platform calculates the waiting time based on the customers' and vacant taxi's GPS coordinates, ensuring a quick and efficient service. E-hailing taxis are prevalent worldwide and are gradually taking market share from street-hailing taxis. To increase their customer base and profits, some companies adopt the

ridesharing concept in their system. Customers can request both regular and taxi-sharing services at the same time, with a reduced fare rate applied if the passenger opts for the latter system. Similar to the ridesharing system, passengers who participate in the taxi-sharing occasionally have a longer journey than the regular taxi service due to the possibility of picking up other passengers along the way. Nonetheless, the reduced fare rate for taxi-sharing makes it a popular choice for many customers, particularly for those looking for an affordable transportation option or when there is a limited number of vehicles in that area and time.

The Taxi-sharing system is a relatively new form of ridesharing that caters specifically to taxi services. However, it is considered more problematic than traditional ridesharing due to the complexities involved in matching two passengers with different origins and destinations. In Taxi-sharing, the matching algorithm aims to find the best two customers for each vacant taxi, determining which one to pick up and drop off orders. However, this process can be challenging, as there are a total of four cases of pick-up and drop-off orders for matching up to two customers simultaneously. These cases can have a significant impact on the total trip length, passengers' in-vehicle time, and waiting time, which must be considered to optimize operator revenue and customer satisfaction.

A more sophisticated version of the matching algorithm has been developed, which can even capture more customers. This algorithm matches the vehicle multiple times and is referred to as a dynamic matching algorithm. After the first customer has reached their destination but before the second, more space

becomes available for other ridesharing users to be matched. This approach improves the efficiency of the Taxi-sharing system, reducing waiting times and maximizing vehicle occupancy. Despite its potential benefits, the Taxi-sharing system still faces challenges related to safety, regulation, and labour practices. Nonetheless, the continued evolution of the matching algorithm in Taxi-sharing and other ridesharing systems demonstrates the potential for innovative solutions to address the challenges of modern transportation.

2.2.2 Evolution of Ridesharing and Taxi-sharing system

The development of shared mobility systems can be understood as a gradual evolution from static, institutionally supported initiatives to increasingly dynamic and technology-enabled services. The earliest forms of organised ridesharing emerged in the 1970s in response to energy crises and concerns over urban congestion. Corporate-sponsored carpooling programs, often supported by federal or local government initiatives, were promoted as a transportation demand management (TDM) tool to reduce single-occupancy vehicle use. Employers played a central role in these schemes, providing ride-matching services, financial incentives, and reserved parking spaces for carpool vehicles (Chan & Shaheen, 2012). For many companies, carpooling not only reduced the pressure on limited parking facilities but also supported broader goals of employee retention and sustainability. However, adoption remained uneven. Numerous studies highlighted the behavioural barriers that limited participation, including the perceived inflexibility of carpool arrangements, difficulties in coordinating schedules, and concerns about reliability and privacy (Ferguson, 1997; Song et al., 2014). Despite these shortcomings, corporate carpooling demonstrated that institutional sponsorship could lower entry barriers for shared

mobility, while also revealing the limits of static matching in accommodating diverse traveller needs.

Building on these early experiences, a second generation of shared mobility services emerged around transit nodes, particularly airports and railway stations. Airport shuttle services became a notable example of demand-responsive transport, where algorithms and scheduling tools were applied to group travellers with similar origins or destinations into shared vehicles. By batching passengers and optimising pick-up and drop-off sequences, these services reduced curb congestion, lowered costs for travellers compared to exclusive taxis, and offered convenient first- and last-mile access (Coogan, 2008). These systems represented an important step towards dynamic allocation, but they also faced competitive pressures. With the rapid growth of share mobility providers such as Uber and Lyft, traditional airport shuttles experienced declining market share. Research shows that ride-hailing services often increased overall vehicle activity around airports, in some cases exacerbating congestion and offsetting the efficiency gains of shared shuttles (Schaller, 2017). Thus, while hub-based shared services highlighted the potential of flexible routing and dynamic assignment, their sustainability was undermined by competition from app-based services that privileged individual convenience.

The most recent stage in this progression is represented by on-demand microtransit, enabled by advances in mobile platforms, GPS tracking, and real-time optimisation algorithms. Unlike employer-based carpools or hub-based shuttles, microtransit services offer a flexible, app-enabled service to the general public, often filling gaps left by conventional public transport. Providers such as

Bridj in Boston, Chariot in San Francisco, and Via in multiple cities sought to combine the efficiency of public transit with the flexibility of demand-responsive transport. Municipal partnerships in Arlington, Texas, and Belleville, Ontario, demonstrate how microtransit can be deployed as a complement to scheduled bus services, particularly for low-density areas or first/last-mile connections (Villagran, 2019). The literature highlights, however, that the long-term viability of microtransit remains uncertain. Studies report high operating costs per passenger, limited scale, and a reliance on public subsidie. While dynamic ride-matching algorithms increase efficiency relative to static carpooling, the complexity of real-time routing, coupled with modest ridership levels, often constrains financial sustainability

The evolution of shared mobility follows a clear trajectory. Workplace carpooling first introduced the principle of sharing private vehicle trips within structured, employer-supported arrangements. This concept was then extended to transport hubs, where taxi-sharing and airport shuttle services operationalised shared travel at larger scales by grouping passengers with similar destinations. More recently, advances in digital platforms and optimisation algorithms have enabled microtransit, which generalises ride-sharing to the wider public and provides flexible, real-time services. Despite this progression, most studies have examined each of these models in isolation and within narrow contexts. What remains absent is a comprehensive framework that can evaluate how shared mobility systems, in their varied forms, interact with private cars and public transport across the network as a whole.

This thesis addresses that gap by developing an integrated simulation framework that unifies ridesharing, taxi-sharing, and public transport within a single environment, enabling the assessment of network-wide impacts such as congestion, emissions, pricing, HOV lane policies, and fleet optimisation. The framework is flexible enough to be adapted to earlier models such as corporate carpooling, hub-based shuttle services, and more recent innovations like on-demand microtransit, through adjustments in cost functions, fleet structures, and matching mechanisms. For the purposes of this thesis, however, the focus remains on two representative cases: ridesharing, in which private drivers use their own vehicles to carry passengers, and taxi-sharing, in which vehicles belong to operators and remain in circulation after dropping off passengers. These two systems form the foundation for the subsequent analysis, and the next section turns to the literature on the impacts of shared mobility to situate their potential contributions within the broader transport context.

2.3 Impact of Shared mobility system

The emergence of ridesharing and taxi-sharing systems has had a significant impact on multiple levels, including personal, cooperative, and system levels. At the personal and cooperative level, ridesharing and taxi-sharing systems provide users with a new mode of transport that offers lower travel costs than self-driving or standard taxi services. By using these systems, passengers can save money on transportation, and taxi-sharing passengers can pay lower fees compared to traditional taxi services. However, the trade-off is that ridesharing, and taxi-sharing services can result in longer travel times and extra waiting times.

Ridesharing and taxi-sharing systems can also reduce the need for vehicle ownership, thereby increasing transport accessibility for individuals. This is particularly useful in urban areas where owning a car may not be practical or necessary. For ridesharing and taxi-sharing operators, there is a new source of income from fare collection. In the taxi-sharing system, increased revenue is expected from reduced empty vehicle mileage and the overlap of multiple passengers' journeys for a single taxi.

On a system level, both systems offer the benefit of reduced vehicle number and mileage, which can lead to a reduction in traffic congestion and environmental impact. By sharing rides, the number of vehicles on the road can be reduced, leading to lower emissions and less traffic on the road. This is important in addressing the environmental challenges of urban transportation. At the network-wide level, the impact of shared mobility has been studied extensively using empirical methods (AlKheder, 2021; Cai et al., 2019; Zhang et al., 2020).

Shared mobility systems, such as ridesharing and taxi-sharing, offer several benefits that are similar to those provided by traditional public transport systems like buses, underground, and trams. These benefits include reduced traffic congestion, decreased carbon emissions, and increased transport accessibility. Moreover, ridesharing and taxi-sharing can supplement public transport by providing a last-mile connection between public transport hubs and individual origins/destinations. However, the rise of shared mobility systems has also raised concerns about competition with traditional public transport systems. While shared mobility can complement public transport, it also offers a

financially attractive option to users, which may result in a competitive relationship with public transport (Babar & Burtch, 2020).

In summary, while there are several studies on the impact of shared mobility systems (Chen, Liu & Wei, 2019; Hang et al., 2019), many of these studies rely on analysis from existing traffic data and are unable to demonstrate the relationship of various network parameters through a meticulous modelling approach. As a result, there is a need for more sophisticated modelling approaches that can better capture the complex interactions between different transport modes and their impact on the wider transport network. These modelling approaches can help policymakers and transport planners to make more informed decisions about the development and implementation of shared mobility systems and their integration with other modes of transport. By adopting a more comprehensive and integrated approach to transport planning, it is possible to evaluating the benefits of shared mobility systems on the wider transport network.

2.4 Studies on shared mobility

Ridesharing and Taxi-sharing have gained significant popularity as modes of transport, with governments worldwide promoting their adoption to alleviate traffic congestion and reduce vehicle emissions. Consequently, research in this domain has witnessed an upsurge, covering a wide range of topics. To provide an overview of the existing academic literature, we conducted a comprehensive

review of scholarly documents, including peer-reviewed journals and conference papers, published between 2017 and 2021. By analysing this body of work, we identified seven prominent topics that have garnered substantial attention within the research community. These topics serve as indicators of the current trends and interests in the field of ridesharing and taxi-sharing research.

2.4.1 Impact on the road network

The impact of ridesharing and taxi-sharing on the road network has emerged as a highly popular and researched topic within the field of shared mobility studies. Researchers have developed models to investigate the implications of integrating shared mobility systems into the road network, examining aspects such as traffic congestion and changes in vehicle movement. Additionally, studies have focused on understanding mode and route choice behaviour following the implementation of shared mobility systems.

(Rayle et al., 2016) conducted a study in San Francisco and discovered that ridesharing services could decrease traffic congestion by reducing the number of vehicles on the road. This is particularly relevant in urban areas where personal car ownership is high. By offering a more convenient and affordable alternative, ridesharing can potentially reduce the need for personal car usage, thereby reducing the number of cars on the road. Conversely, (Henao, 2017) argued that these services could increase congestion due to increased vehicle miles travelled and deadheading (time spent by ridesharing vehicles traveling without passengers). Research on the impact of taxi-sharing on road networks is still nascent. However, early studies, such as those by (Alonso-Mora et al., 2017),

have suggested that smart taxi-sharing systems could improve road network efficiency by reducing the total distance travelled and improving vehicle utilization rates.

The need to understand and predict the impact of ridesharing on road networks has led to the development of various models, notably the Ridesharing User Equilibrium (RUE) model. This model, as discussed by (Wang et al., 2020), aims to predict the behaviour of users within a ridesharing system, assuming that each user seeks to minimize their own travel cost. The RUE model takes into account various factors, including user preferences, travel costs, and system constraints (Chen, X. & Di, 2021; Li, M. et al., 2020; Wang, Yang & Zhu, 2018). The primary assumption of the RUE model is to find an equilibrium point where, within the same origin and destination points, all modes and route choices have the same generalised travel cost.

The RUE model's strength lies in its ability to analyse and predict the impacts of ridesharing on a granular level, enabling transportation planners and policymakers to better plan and manage urban road networks. However, (Chen, X. & Di, 2021) argued that the model's reliance on the assumption of rational user behaviour might limit its accuracy, as it doesn't account for unpredictable or irrational user behaviours. The Ridesharing User Equilibrium (RUE) model, while powerful in analysing the behaviour of users within a ridesharing system, does have some limitations. A significant drawback is its lack of consideration for the matching mechanism between passengers and drivers.

In the real world, the efficiency and effectiveness of a ridesharing system is heavily dependent on how well the algorithm can match passengers to drivers (Ma et al., 2013). However, the RUE model, in its current form, does not account for these complex dynamics. This is mainly due to, the RUE model is a macroscopic model that focuses on system-level behaviour, assuming that each user seeks to minimize their own travel cost. It is not designed to consider individual-level interactions or behaviours, such as the matching mechanism between passengers and drivers. The incompatibility between the macroscopic nature of the RUE model and the microscopic nature of matching algorithms presents a significant challenge to incorporating the latter model into the former model.

To address this limitation, future research could explore the development of multi-scale models that incorporate both macroscopic and microscopic perspectives. These models could potentially offer a more comprehensive and accurate tool for predicting and managing the impacts of ridesharing services on urban road networks. Therefore, while the RUE model provides valuable insights into user behaviour within a ridesharing system, it might benefit from considering the matching mechanism between drivers and passengers to enhance its predictive accuracy and practical application (Agatz et al., 2011).

2.4.2 Better ridesharing and taxi-sharing matching algorithm

The rapid proliferation of ridesharing services has led to an increased focus on the development of efficient matching algorithms. These algorithms play a central role in ridesharing systems, influencing their effectiveness, and shaping

the user experience. The matching process involves pairing drivers with passengers in a way that optimizes various factors such as distance, time, cost, and user preferences. Similarly, in the context of taxi-sharing, the matching process is critical to the efficient allocation of taxis to passengers. To address the complexity of this process, researchers have employed various optimization models to develop effective matching algorithms. For instance, (Li, Y. & Chung, 2020) utilized an integer programming model to optimize the matching process in ridesharing services. Their approach accounted for multiple factors, including travel time, cost, and passenger preferences, to create the most efficient matches.

In a different approach, (Hou et al., 2018) used a non-linear programming model to optimize the matching process. The objective of their model was to minimize the total travel time of all passengers, thereby enhancing system-wide efficiency. Another study by (Aydin, Gokasar & Kalan, 2020) used a binary programming model to optimize the matching process. Their model was designed to balance supply and demand while also reducing the total travel distance for drivers. This approach aimed to increase system efficiency and reduce environmental impact.

More recently, heuristic approaches have garnered attention in the development of matching algorithms for ridesharing and taxi-sharing services. Heuristics, which are strategies derived from previous experiences to aid in problem-solving, can offer practical and efficient solutions, particularly when dealing with large-scale or complex problems. (Papoutsis et al., 2021) proposed a heuristic approach using a clustering method in machine learning to optimize the matching process. The authors applied the k-means clustering algorithm to segment the geographical area into clusters and then matched drivers and

passengers within those clusters. This approach effectively reduced the problem's complexity and improved the efficiency of the matching process.

Another commonly used heuristic in the literature is the greedy algorithm, which makes the locally optimal choice at each stage with the hope of finding a global optimum. (Jia, Xu & Liu, 2017; Zhang, C. et al., 2020) applied a greedy algorithm in their study, showing that it can effectively improve the performance of matching algorithms in large-scale, real-time ridesharing services. In another study, (Zhang, C. et al., 2020) also used a greedy algorithm for their matching process. They developed a multi-objective greedy algorithm that considered both the travel time and the number of passengers in a vehicle to optimize the matching process.

While existing matching algorithms have significantly contributed to the optimization of ridesharing services, their focus is often limited to immediate interactions between drivers and passengers. These algorithms might fall short when it comes to incorporating equilibrium principles, such as those in the Ridesharing User Equilibrium (RUE) model, which could result in an unable to analyse the impact of the system on road networks and the environment.

2.4.3 Environmental impact

The environmental impact of transportation systems has become a prominent and widely discussed topic in recent years, driven by increasing concerns about global warming and the need to address climate change. Among the various aspects of transportation that have attracted attention, self-driving vehicles have emerged as one of the primary subjects of interest due to their potential

implications for emissions reduction. Recognizing the potential benefits of shared mobility in terms of reducing vehicle emissions, authorities have been supportive of initiatives promoting the adoption of shared transportation services, which require relatively less investment compared to traditional modes of transportation.

The rapid expansion of shared mobility systems in numerous cities has prompted scientific inquiry into the environmental consequences of this emerging transportation paradigm. Researchers have employed emission models to assess the environmental impact of various factors associated with shared mobility, such as reductions in individual vehicle trips, changes in vehicle purchase behaviour, and increased trip lengths. In a study conducted by(Yan et al., 2020) used a microscopic emission model called The COPERT III (Ntziachristos et al., 2000), to estimate the emissions of vehicles complying with European emission standards. The findings of this study revealed that taxi-sharing alone could result in a significant reduction of fuel consumption by 22.88% in Shanghai, China. For Beijing, another group of researchers (Yu et al., 2017) focused on the environmental benefits of taxi-sharing in Beijing. Using a similar approach, the researchers estimated that taxi-sharing initiatives in the city annually contributed to the mitigation of approximately 46.2 kilotons of CO2 emissions and 257.3 tons of NO2 emissions. These findings demonstrate the potential of shared mobility systems to make substantial contributions to environmental sustainability in densely populated urban areas. Furthermore, (Tikoudis et al., 2021) conducted a comprehensive analysis across a sample of 247 cities to forecast the potential reduction in CO2 emissions achievable through the implementation of taxi-sharing systems.

Understanding the environmental impacts of ridesharing and taxi-sharing systems necessitates a comprehensive approach that includes evaluating the efficiency of matching algorithms and applying the concept of Ridesharing User Equilibrium (RUE).

Matching Algorithms: These determine how drivers and passengers are paired in ridesharing and taxi-sharing systems, influencing overall system efficiency, including fuel consumption and emissions. A robust matching algorithm optimizes routes and reduces idle time, directly contributing to lower emissions.

Ridesharing User Equilibrium (RUE): This transportation planning concept describes an optimized state where no user can improve their travel conditions by choosing a different route or transport mode. Considering RUE provides insights into ridesharing's long-term impact on traffic, congestion, and emissions. In conclusion, integrating matching algorithms and RUE into the analysis of ridesharing and taxi-sharing's environmental impacts allows for a more complete understanding and the identification of potential avenues for enhancing sustainability.

2.4.4 Pricing scheme

The fare structure in ridesharing services represents a pivotal component of the system dynamics, critically influencing the behaviour of both passengers and drivers. Passengers' decision to utilize ridesharing services hinges largely on the fare; excessively high fares might render ridesharing services unattractive compared to other transportation alternatives. Conversely, if fares are set too low, it could lead to diminished driver participation, as the financial incentives

to offer rides decrease, potentially undermining the efficiency and reliability of the system.

As a result, achieving a delicate balance in the pricing strategy is essential to ensure the sustainability and effectiveness of ridesharing and taxi-sharing systems. One specific pricing mechanism that has been widely adopted, particularly in the taxi-sharing domain, is surge pricing. Surge pricing is a dynamic pricing model that adjusts fares in real-time according to variations in passenger demand. During periods of high demand, fares increase in a bid to incentivize more drivers to provide services.

While this pricing model can theoretically optimize resource allocation and ensure maximized profit for the company, it has attracted substantial criticism from passengers. The primary concern stems from the perception that surge pricing exploits demand-supply imbalances to the company's advantage while disproportionately burdening the passengers with higher fares. Empirical studies corroborate this view, indicating that while surge pricing might contribute to higher revenues for drivers and the platform, passengers tend to fare worse off in comparison to a static pricing strategy (Dong & Leng, 2021; Zha, Yin & Xu, 2018).

2.4.5 Efficiency of High-Occupancy Vehicle lanes

High-Occupancy Vehicle (HOV) lanes, dedicated traffic lanes reserved for vehicles with one driver and one or more passengers, including ridesharing

vehicles, have garnered significant attention due to their potential impact on transportation systems. (Zhong et al., 2020) conducted a study that concluded HOV lanes have a positive impact on the ridesharing system and social welfare. They found that the wider implementation of HOV lanes throughout the road network enhances the effectiveness of ridesharing, resulting in increased ridesharing usage and improved overall social welfare. This indicates that HOV lanes play a crucial role in promoting sustainable transportation alternatives. Another study conducted by (Li, Huang & Shang, 2020) concluded HOV lanes have a positive impact on the ridesharing system and social welfare. They found that the wider implementation of HOV lanes throughout the road network enhances the effectiveness of ridesharing, resulting in increased ridesharing usage and improved overall social welfare. This indicates that HOV lanes play a crucial role in promoting sustainable transportation alternatives.

In a network wide-level, a study concluded that HOV lanes have a positive impact on the ridesharing system and social welfare. They found that the wider implementation of HOV lanes throughout the road network enhances the effectiveness of ridesharing, resulting in increased ridesharing usage and improved overall social welfare. This indicates that HOV lanes play a crucial role in promoting sustainable transportation alternatives (Ashraf Javid & Al-Khayyat, 2021).

Incorporating the traffic assignment model such ash Ridesharing User Equilibrium (RUE) model into studies on High-Occupancy Vehicle (HOV) lanes and ridesharing is crucial. The RUE model provides a framework for understanding the equilibrium state of a ridesharing system, considering the

interplay between user behaviour, route choices, and travel times. By integrating a traffic assignment model, researchers can more accurately assess the impact of HOV lanes on ridesharing, accounting for factors such as travel demand, congestion levels, and overall system efficiency.

2.4.6 Shared mobility and public transport

The impact of ridesharing on the public transport system is characterized by a combination of complementarity and competition. Ridesharing services can improve first-mile and last-mile connectivity, addressing the "first and last mile problem" where passengers face challenges in accessing public transport stations or reaching their final destinations from transit stops. (Thao, Imhof & von Arx, 2021) used a mixed-method approach with observed data in Switzerland and concluded that the ridesharing system could complement the public transport system, especially rail network. The researchers also suggested that integrating ridesharing information boards at bus stops improved traveller choices and reduced waiting times for connections to the railway system. More specifically, (Kumar & Khani, 2021) developed an algorithm for a ridesharing system as the first mile to the public transport system. They summarised that their algorithm could significantly reduce overall in-vehicle time. However, there are still a limited number of studies on the effect of these two modes, and it cannot be concluded that the effect is either complementary or competitive.

2.4.7 Optimum fleet size

The study of an optimum fleet size originated from the study of conventional and e-hailing taxis. The underlying assumption is that taxi-sharing systems have the potential to accommodate more passengers per vehicle, thereby reducing the overall number of taxis required in the transportation network. This topic is the least commonly found in the literature. Nevertheless, (Farhan & Chen, 2018) developed an optimisation framework for an autonomous taxi-sharing system. Through their analysis, they found that the fleet size in their autonomous taxi-sharing system could be significantly reduced by more than half compared to regular e-hailing services. This reduction in fleet size not only signifies potential cost savings for service providers but also contributes to the optimization of vehicle utilization and environmental sustainability.

The study of optimum fleet size in taxi-sharing systems is closely depended on matching algorithm models and route choice models. These models are essential for accurately determining the optimal fleet size, as they consider factors such as passenger preferences, driver availability, and spatial-temporal constraints of the road network. Without these external models, the optimum fleet size solution may not reflect the actual scenario. Since, some limitations of such system are unforeseen.

Based on this wide range of topics and the number of studies, one can conclude that shared mobility research has different aspects. Researchers used different models and approaches to answer their topic of interest; only a fracture of these models can be used for more than one topic. For example, traffic assignment

models such as RUE models can be used to assess an environmental impact based on their outputs, such as reduced vehicle flow and travel time. In addition, RUE models can adjust road characteristics to support HOV lanes or incorporate them into the public transport system. In another case, researchers can customise their optimisation models in matching algorithms to deploy pricing schemes and calculate the optimum fleet size based on their matching rate.

2.5 Model integrations

The boundary between these two main approaches (matching algorithms and traffic assignment) is distinctively clarified. Studies in matching algorithms consider only matching processes between passengers and drivers, whose demands, locations, and requests are exogenously defined. In reality, the demand for ridesharing services is influenced by various factors, such as waiting time and travel costs, which fluctuate dynamically due to changes in network conditions. Moreover, existing algorithms in the literature all assume exogenously given link/path travel times; this limitation is acceptable in case the number of ridesharing users is low and does not significantly impact link/path costs. Furthermore, all matching algorithms found in the literature are on the microscopic level, which is computationally expensive for modelling large-scale transport systems. For such reasons, these studies are insufficient to assess the network-wide impact of a shared mobility system when travellers' mode and route choices are influenced by dynamically changing network states, user preferences, and pricing strategies.

On the other hand, we have not found any study considering matching for ridesharing services as an inherent model component of traffic assignment. All the previously mentioned work that involves matching within traffic assignment is concerned with conventional taxis or e-hailing. The main challenge lies in the potentially sophisticated optimisation procedures in the matching process and potential unmatched trips, which should be considered in a realistic setting.

In addition, all studies involving traffic assignment and ridesharing systems are conceived in the nature of a static modelling framework. No studies have yet attempted endogenizing these variables in a time-varying traffic assignment modelling framework and capturing the dynamic balance between travel demand and network supply – including those offered by ridesharing services and travellers' adaptive mode and route choices.

In order to develop a simulation framework capable of addressing the key aspects of a shared mobility system, a multimodal traffic simulation model is needed. These require an integration of several models in the mode and route choice models. Given the existence of numerous models developed for each mode of transport (Ridesharing, Taxi-sharing, Solo-driving and Public-transport), a careful selection and modification of these models is required to ensure seamless integration and compatibility among them. In the following chapters, we will conduct a comprehensive review and development of route choice models for each mode of transport within the research scope. The strengths and limitations of these models will be analysed, and their adaptation and integration will be demonstrated in the latter chapter (Research Methodology).

This chapter has successfully reviewed the state-of-the-art in shared mobility, establishing a critical foundation for the proposed simulation framework and fulfilling the requirement to understand the prevailing research landscape. The review confirmed that a significant research gap exists: current models often address only one or two specific aspects (e.g., matching algorithms or environmental impact) and lack the integrated approach necessary for comprehensive policy analysis. By identifying this gap and defining the timely research questions (such as HOV lane efficiency, pricing schemes, and public transport interaction), this chapter directly supports the third objective of this thesis—to evaluate the capability of the proposed model against these complex topics in later chapters (Chapter 8).

CHAPTER 3 ROUTE CHOICE MODEL FOR SHARED MOBILITY SYSTEM

This chapter functions as a specialized literature review, focusing exclusively on the mathematical modelling underpinning route choice for shared mobility systems, commonly called matching algorithms. It reviews the fundamental characteristics of optimization models, detailing objective functions (such as maximizing Vehicle Kilometres Travelled Saved (VKTS) or minimizing travel time) and constraints (time and cost limits) essential for determining feasible matching pairs. Crucially, the review examines state-of-the-art microscopic solutions, including Binary/Mixed Integer Programming for single and multiple passenger matching, highlighting the inherent computational complexity of these methods, a challenge the thesis framework must overcome to achieve large-scale application.

The review strategy centered on searching academic databases for keywords related to transport optimization, including "matching algorithms," "Optimisation," "integer programming," and "linear programming". Search platforms included Imperial College Library, Science Direct and Google Scholar. Literature was included if it provided mathematical formulations of objective functions and constraints relevant to ridesharing optimization (such as multi-passenger matching). This review specifically focused on existing models operating at the microscopic level, which treats users individually.

3.1 Backgrounds

The ridesharing system's route choice model is commonly called the matching algorithm. The traditional process of the matching algorithm was performed via community boards or the internet. After both matching requests and offers had been posted, the matching process was performed by users or human operators. During that period, there was no automated system to handle this task. The matching process, therefore, took a lot of time, and the success rate was low. This delay raises inflexibility as passengers and drivers must show long commitment. Owing to this drawback, the number of users is limited. Nowadays, with more technological advancements and mobile communication, the matching algorithm is automated based on sophisticated mathematical models with different objective functions and a range of constraints. This makes the system capable of providing a successful matching between passengers and drivers at concise notice (Wang, Xing, 2013). Successful matchings are directly associated with ridesharing operator intensive and an essential part of the system.

3.2 Optimization model characteristic

Matching between drivers and passengers is an optimisation problem. The problem involves finding the best solution among all feasible solutions. Optimisation problems consist of three parts: objective function constraints and decision variables. Depending on the mathematical model used in the optimisation and the assumption, decision variables can be binary {0,1}, integer

or real numbers. The following Equation 3-1 illustrates a standard form of an optimisation problem.

$$\begin{aligned} \textbf{given} \quad & \vec{x} = (x_1, x_2, x_3, \dots, x_n) \\ \textbf{minimize} \quad & f(\vec{x}) \\ \textbf{subject to} \quad & g_1(\vec{x}) \leq b_1 \\ & g_2(\vec{x}) \leq b_2 \\ & g_3(\vec{x}) \leq b_3 \end{aligned}$$

Equation 3-1 Standard Optimisation Problem

Objective functions define the objective of the optimisation model. This optimisation model is used to maximize or minimise certain value from functions. In the figure, the objective function is minimizing the function $f(\vec{x})$. Constrains in optimisation models are used to remove (hard constrains) or reduce (soft constrains) infeasible variables. Functions $g_m(\vec{x}) \leq b_m$, where $m = 3$, represent three constrain functions in the example. The last part is the decision variable. Decision variables are unknown quantities that optimisation models seek, in this case, the vector (or set) $\vec{x}$.

3.2.1 Objective functions for matching algorithm

The objective functions play a crucial role in the operation of service providers. The provider can be either a private firm or a public organisation. If a private firm operates the system, the system's goal is to maximise the profit from commissions. For example, the firm may charge the commission per successful matching trip. This commission can be either a fixed rate or distance based. In the public case, the system may have an objective in social benefits, such as

reducing congestion and pollution. Hence, shared mobility systems may have a variety of objective functions in their matching algorithm. Agatz, Niels et al. (2012) stated that most studies considered one or a combination of the following objectives.

- Minimising the system vehicle-kilometres (or miles) travelled (Agatz, Niels et al., 2011; Amey, 2011)**.** The measure represents the total vehicle distance driven by travellers from their origins to destinations with a ridesharing system. The objective is crucial from a transport authorities' point of view because the measure is directly related to reducing traffic congestion and vehicle emissions. The objective is also compatible with minimising total travel costs, which is an essential factor for attracting potential users and the service provider.

- Minimising the system travel time (Winter & Nittel, 2006)**.** The system travel time means the total travel time that all ridesharing vehicles spend between their origins and final destinations. This measure is also essential for society and transport authorities. This is because the travel time value relates to vehicle emission and traffic network performance. In addition, travel time is also an essential factor for ridesharing users.

- Maximising the number of users (Ghoseiri, Haghani, & Hamedi, 2011; Xing et al., 2009)**.** The objective is to maximise the number of successful matchings between drivers and passengers in the system. The objective is beneficial for the system provider whose revenue is relied on the success rate. The measure can also reflect the ridesharing system's performance

from the users' perspective. A high success rate may induce users to maintain the service.

3.2.2 Constrains in matching algorithms.

To help us clarify constrains in matching problem, we demonstrated a single driver and with a single passenger ridesharing problem (see Figure 3.1). In this example, the driver d_1 and the passenger p_1 have their origins and destinations indicated by circle and rectangle symbols respectively. In order to match the driver d_1 with the passenger p_1, there are constrains that need to be consider if the matching between this pair is feasible and plausible. The travel time and cost that a vehicle between node i and j and indicated by $T_{(i,j)}$ and $C_{(i,j)}$ respectively.

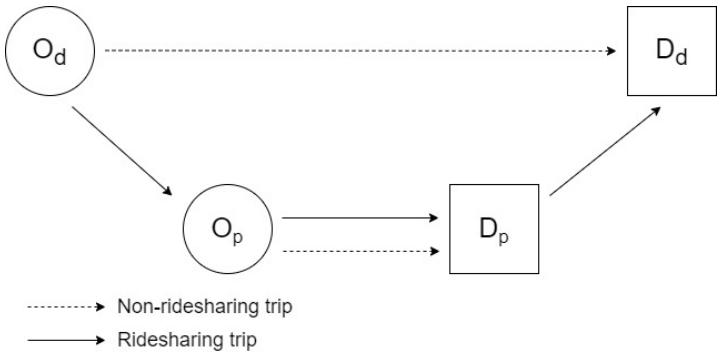

Figure 3.1 Single driver and single passenger ridesharing problem

- Time constraints

Constraints in the ridesharing problem can be commonly categorized into two types: time and cost. For time constraints, it is assumed that ridesharing drivers and passengers provide not only their origin and destination information but also their time preferences. Time preference involves two values: earliest departure time and latest arrival time. The optimization model then finds the best matching

and its route (based on the objective function), given that each traveller's trip begins and ends between their time preference (Agatz, Niels A. H. et al., 2011). Note that each traveller will have a difference in start and end time if their origin and destination differ. In the example, the driver d has their earliest departure time and latest arrival time denoted by T_d^0 and T_d^1 respectively, and for the passenger p are T_p^0 and T_p^1. If these two users are matched, the ridesharing route is $O_d \rightarrow O_p \rightarrow D_p \rightarrow D_d$ which started at time T. The trip is feasible is it passes following time constrains.

Driver earliest departure	$T_d^0 - T \leq 0$
Passenger earliest departure	$T_p^0 - T_{(O_d,O_p)} - T \leq 0$
Driver latest arrival	$T + T_{(O_d,O_p)} + T_{(O_p,D_p)} + T_{(D_p,D_d)} \leq T_d^1$
Passenger latest arrival	$T + T_{(O_d,O_p)} + T_{(O_p,D_p)} \leq T_p^1$

Equation 3-2 Time Constraints

Another approach that considers time constraints was undertaken by (Baldacci, Maniezzo, & Mingozzi, 2004), which allows users to determine their maximum excess travel time prior and ensure that all users in successful matchings do not suffer their in-vehicle time above the limit. However, our proposed model in this thesis does not directly consider time constraints. In our model, time is one of the factors we consider as a cost which can be derived by the Value of Time (VOT) parameters. The parameter then transfers to our cost constraints in the model.

Thus, time preference values such as departure and arrival time are neglected within our framework.

- Cost constraints

The primary reason that travellers adopt shared mobility systems, particularly ridesharing, is the potential to save on travel costs when compared with self-driving. Agatz et al. (2012) highlighted that cost-saving is a key incentive, especially for those without access to a private vehicle. In some cases, such individuals may be willing to accept a slightly higher fare than public transport in exchange for the added flexibility of ridesharing. In the literature, travel costs that have been considered are proportional to distance and time. Therefore, one must reject their ridesharing trip if the total cost is higher than self-driving. To ensure that this scenario will not happen, cost constraints should be embedded in the matching algorithm and ensure that all users have a lower cost. In most cases, ridesharing drivers have a more considerable travel cost compared with self-driving. To compensate for this, passengers pay some amount of travel fare to the drivers. However, from the passenger's point of view, this fare must be less than the total cost of their non-ridesharing trip. Next, we demonstrate cost constraints shown in Equation 3-3 similarly to the previous example.

Driver cost $$\left[C_{(O_d,O_p)} + C_{(O_p,D_p)} + C_{(D_p,D_d)} - F^d_{d,p}\right] - C_{(O_d,D_d)} \leq 0$$

Passenger cost $$\left[F^p_{d,p} + C'_{(O_p,D_p)}\right] - C_{(O_p,D_p)} \leq 0$$

System cost $$\left[C_{(O_d,O_p)} + C_{(O_p,D_p)} + C_{(D_p,D_d)}\right] - \left[C_{(O_d,D_d)} + C_{(O_p,D_p)}\right] \leq 0$$

Equation 3-3 Cost Constraints

Cost variable denoted as $C_{(i,j)}$ included both vehicle cost and value of time while $C'_{(i,j)}$ included only value of time. Drivers have costs in two folds. (1) Costs that generated from vehicle fuel consumption and other maintenance costs that based on travelled distance. (2) Costs is generated from travel time that the driver spends. However, the driver has an incentive of $F^d_{d,p}$ which is a travel fare that driver d get if he/she is matched with passenger p. Passenger bare cost from travel fare $F^p_{d,p}$ plus $C'_{(o_p,D_p)}$ which is the travel time cost they spent in the ridesharing trip. Note that the passenger's fare may not equal to driver's incentive, in other word $F^d_{d,p} \neq F^p_{d,p}$, if the service provider decides to collect some fee. The algorithm can also have an extra constraint for the system benefit to ensure that the system does not generate negative impact to the road network and the environment. Based on these constraints, travel fare is the key factor determining the success of the matching. Geisberger et al. (2009) suggested that the ridesharing fare should be increased until both driver and passenger split the total cost in ridesharing evenly. Agatz et al. (2011) proposed a fare allocation system based on their non-ridesharing actual distances. Some studies adopted the ridesharing fare based on passengers' travelled distances in their models (Sun, S. & Szeto, 2021; Xu et al., 2015).

3.3 Optimization models in shared mobility system

Optimisation models are commonly found in transport research, especially in the logistic system. For example, optimisation models are used to calculate an order picking process in a warehouse (Bučková, Krajčovič & Edl, 2017) and schedule crude oils in multiple modes of transport (Zhang, X. et al., 2018). For shared mobility systems, optimisation models are used in the matching algorithm, originating from dial-a-ride problems (Masoud & Jayakrishnan, 2017b). Dial a ride system is a paratransit system that offers door-to-door transportation for elderly and disabled travellers. The system does not provide a regular route and schedule services. Instead, it operates from a bespoke service based on passengers' demand. In order to register their demands, passengers make contact with the service provider. Using an optimisation model, the ridesharing provider then assigns vehicles from their depot and determines routes based on multiple requests. Savelsbergh & Sol (1995) were very early researchers who proposed an optimisation model for solving a fundamental dial-a-ride problem. However, the study was limited to some assumptions, such as all vehicles must begin and end at the same depot, and all vehicles and passengers are assumed to be homogeneous. Over years of studies, many models have been introduced with more usable forms, such as multiple depots (Cordeau & Laporte, 2007), considering time windows (Braekers, Caris & Janssens, 2014) and multi hop journey (Masson, Lehuédé & Péton, 2014).

For ridesharing and taxi-sharing systems, early development were presented by (Herbawi & Weber, 2012; Lalos et al., 2009; Tao, 2007). Their studies introduced heuristic frameworks based on methods such as greedy algorithms, genetic

algorithms and insertion heuristics. These approaches demonstrated the potential of heuristic optimisation, but they did not emphasise the core challenges of matching users to routes, and they also ignored fare structures. As a result, their applications were limited to relatively small-scale matching problems.

Subsequent research proposed more advanced heuristics. Najmi, Rashidi & Liu. (2020) proposed a clustering - based method, where ridesharing requests and offers were grouped by trips origins and destinations before matched independently Similarly, Kleiner, Nebel & Ziparo. (2011) followed by Nourinejad & Roorda (2016) introduced an agent-based algorithm based on auction mechanisms. Although heuristic approaches cannot guarantee optimal solutions, they can still provide acceptable results within reasonable time. In contrast, exact optimisation methods such as binary or linear programming ensure optimality but are often computationally demanding. These trade-offs highlight the need for models that balance solution quality with scalability. This issue is examined further in the next section, which reviews optimisation-based approaches in greater detail.

3.3.1 Binary or Mixed Integer programming

A binary integer programming problem is a mathematical optimization or feasibility program in which decision variables are restricted to be {0,1} integer. Equation 3-4 shows the mathematical formulation of the ridesharing problem in the single driver – single passenger case. The objective function is to maximize the weight of the successful match. Two constraints in the figure are used to ensure that each user is assigned only once. The decision variable x_{dp} have two

possible values; $x_{dp} = 1$ if driver d is matched with passenger p, otherwise $x_{dp} = 0$. .

$$\textbf{given} \qquad x_{dp} \in \{0,1\} \quad \forall\, p \in P, d \in D$$

$$\textbf{Maximize} \qquad \sum_{d \in D} \sum_{p \in P} w_{dp} x_{dp}$$

$$\textbf{subject to} \qquad \sum_{p \in P} x_{dp} \leq 1 \quad \forall\, p \in P,$$

$$\sum_{d \in D} x_{dp} \leq 1 \quad \forall\, d \in D,$$

Equation 3-4 Binary Integer Programming

Many studies have used binary integer programming embedded in their matching algorithm (Hao, 2018; Kumar & Khani, 2021; Masoud & Jayakrishnan, 2017a; Nourinejad & Roorda, 2016; Wang, Dessouky & Ordonez, 2016; Xia et al., 2015). Most of these studies considered only a single driver – single passenger problem. Researchers commonly used mixed integer programming to match a single driver with multiple passengers or a more complex matching algorithm. This mathematical programming allows more decision variables, which leads to more assumptions, to be encapsulated in their optimization models. Decision variables in mixed integer programming are relaxed to any integer number between the predetermined lower and upper bound. For example, Armant & Brown (2014) presented a mixed integer programming model which have two decision variables; 1.) x_{dp} represents a matching between driver and passenger which explained previously 2.) t_{d,O_d} represents the departure time that driver d departs its origin location O_d. Tamannaei & Irandoost (2019) used

the same programming model that optimizes drivers' matching pairs and departure times. They introduced a beam search algorithm that yielded near-optimal solutions while consuming less computational cost in large-scale networks. Di Febbraro, Gattorna & Sacco. (2013) introduced another single driver – multiple passengers matching algorithm with a mixed integer programming. The feature increases the system's capacity. However, the algorithm will become more complex and consume a higher computational cost. In the next section, we will explain the characteristic of the feature and how we implemented them in our framework while minimizing the drawback.

3.3.2 Linear programming

Similar to binary and integer, linear programming has been used in many transport studies. For example, to minimize transportation costs by determining optimal distribution of goods Prifti et al. (2020), calculated optimal routes for intermodal freight transport problem in Rotterdam (Li, Negenborn & Schutter, 2013) and assigned optimum traffic flows while balancing system and user objectives (Angelelli et al., 2021). In shared mobility systems, we have shown that many studies used a mixed-integer linear programming approach, in which the objective function and constraints are in the realm of linear functions. However, decision variables are limited to integer numbers. In standard linear programming, decision variables must be a real number.

To the best of our knowledge, we have not found a single study that used linear programming as a matching algorithm in shared mobility systems. All existing matching algorithms typically model the ridesharing system at a microscopic

level, treating individual users separately, which results in computationally demanding problems that struggle to scale to larger networks. To ensure computational efficiency and tractability for large-scale urban analysis, this thesis necessitates a shift toward a macroscopic matching methodology. The formal adaptation of Linear Programming principles to model aggregated demand, along with the detailed discussion of the resulting computational benefits (P class complexity), is fully provided in the Methodology (Chapter 6).

3.4 Single Driver – Multiple passengers

Earlier examples of ridesharing systems can only match a single driver with a single passenger, i.e., one-to-one matching. In this section, we will describe a more complex matching arrangement. The single driver – multiple passengers matching, i.e., one-to-many matching, can increase the benefit of the ridesharing system. This matching arrangement is most suitable when drivers have more time flexibility and may be willing to accept more than one passenger in their trip to further reduce travel costs (Hosni, Naoum-Sawaya & Artail, 2014). From the passenger's point of view, the new arrangement can reduce the ridesharing fare but may increase their in-vehicle travel time. This arrangement is economically logical only when all users have a lower cost compared with the single driver – single passenger arrangement and the non-ridesharing trip. As mentioned in the early section, the total travel cost is based on several factors such as travel time, waiting time and ridesharing fare. Fundamentally, each user's travel time and waiting time are based on distances of their origin and destination points, as well as the similarity of their optimum route. In the single driver – single passenger matching the ridesharing route is straightforward

$O_d \rightarrow O_p \rightarrow D_p \rightarrow D_d$. However, in the single driver – multiple passengers matching, determining the ridesharing route and its pick-up and drop-off sequence are more complex. For the ridesharing system that allows up to two passengers to be matched simultaneously, there is a total of four matching patterns which are illustrated below:

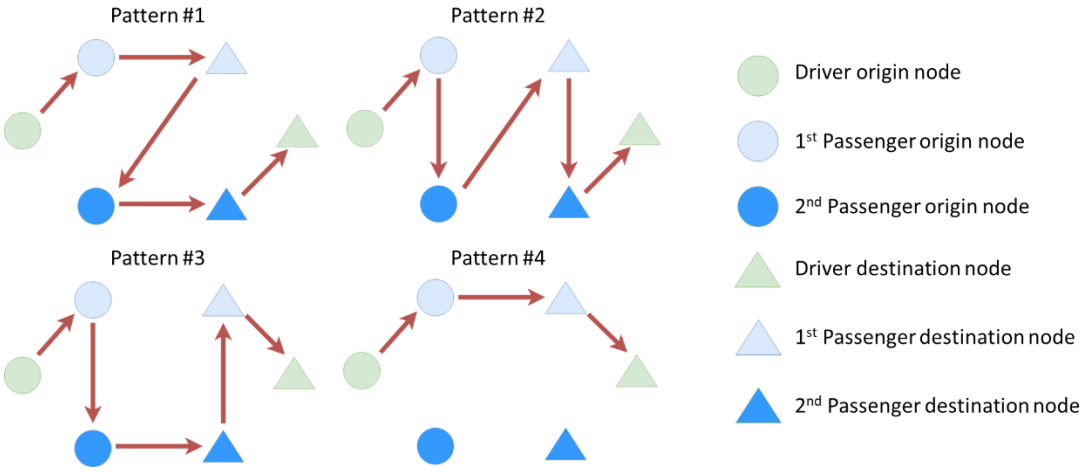

Figure 3.2 Single driver - Multiple passengers matching patterns.

Figure 3.2 shows that different matching patterns result in different ridesharing routes. The pattern number also differentiates each passenger's waiting time and in-vehicle time, ultimately affecting the total travel cost of each ridesharing user. The following Equation 3-5 demonstrates cost constraints in the single driver – multiple passenger matching problems.

Driver cost

Pattern #1 $\left[C_{(O_d,O_{p1})} + C_{(O_{p1},D_{p1})} + C_{(D_{p1},O_{p2})} + C_{(O_{p2},D_{p2})} + C_{(D_{p2},D_d)} - F^d_{d,p1,p2}\right] - C_{(O_d,D_d)} \leq 0$

Pattern #2 $\left[C_{(O_d,O_{p1})} + C_{(O_{p1},O_{p2})} + C_{(O_{p2},D_{p1})} + C_{(D_{p1},D_{p2})} + C_{(D_{p2},D_d)} - F^d_{d,p1,p2}\right] - C_{(O_d,D_d)} \leq 0$

Pattern #3 $\left[C_{(O_d,O_{p1})} + C_{(O_{p1},O_{p2})} + C_{(O_{p2},D_{p2})} + C_{(D_{p2},D_{p1})} + C_{(D_{p1},D_d)} - F^d_{d,p1,p2}\right] - C_{(O_d,D_d)} \leq 0$

1st passenger cost

Pattern #1 $\left[F^{p1}_{d,p1,p2} + C'_{(O_{p1},D_{p1})}\right] - C_{(O_{p1},D_{p1})} \leq 0$

Pattern #2 $\left[F^{p1}_{d,p1,p2} + C'_{(O_{p1},O_{p2})} + C'_{(O_{p2},D_{p1})}\right] - C_{(O_{p1},D_{p1})} \leq 0$

Pattern #3 $\left[F^{p1}_{d,p1,p2} + C'_{(O_{p1},O_{p2})} + C'_{(O_{p2},D_{p2})} + C'_{(D_{p2},D_{p1})}\right] - C_{(O_{p1},D_{p1})} \leq 0$

2nd passenger cost

Pattern #1 $\left[F^{p2}_{d,p1,p2} + C'_{(O_{p2},D_{p2})}\right] - C_{(O_{p2},D_{p2})} \leq 0$

Pattern #2 $\left[F^{p2}_{d,p1,p2} + C'_{(O_{p2},D_{p1})} + C'_{(D_{p1},D_{p2})}\right] - C_{(O_{p2},D_{p2})} \leq 0$

Pattern #3 $\left[F^{p2}_{d,p1,p2} + C'_{(O_{p2},D_{p2})}\right] - C_{(O_{p2},D_{p2})} \leq 0$

Equation 3-5 Cost constraints in the single driver – multiple passengers matching problems.

With matching pattern #1, the driver picks up and drops off the first passenger before navigating to the next passenger. This pattern is the most beneficial for passengers, as both travels in the ridesharing vehicle with their shortest route. From the passenger point of view, matching with this pattern is equivalent to the single driver-single passenger arrangement but at a lower fare. On the other hand, this pattern is least beneficial for drivers and the system, especially when passengers' route is similar. In this case, the algorithm should match users with the other two matching patterns (patterns #2 and #3). These patterns assign the driver to pick up both passengers before they start travelling to either destination. In the perfect scenario, where both passengers have the exact origin and destination, in other words, $O_{p1} = O_{p2}$ and $D_{p1} = D_{p2}$. This similarity makes passenger costs $C'_{(O_{p1},O_{p2})}$, $C'_{(D_{p1},D_{p2})}$ and $C'_{(D_{p2},D_{p1})}$ equal to zero. As a result, patterns #2 and #3 will most likely nominate by the algorithm. The entire ridesharing route will be the same as when it picks up only one passenger (pattern #4). Determining the most suitable matching pattern is heavily based on the users' origin and destination points. Ultimately, the algorithm selects the matching pattern that yields minimal ridesharing routes and generates fewer travel costs. In conclusion, the matching patterns are necessary for optimization as they will determine the total travel cost of each ridesharing user. An unsuitable pattern may result in the ridesharing becoming unworthy and unnecessarily increased travel costs.

Compared with single passenger matching, multiple matching increases the optimization scale substantially. This enlargement is due to the inclination of decision variables. An algorithm not only determines what matching patterns it also determines who is the first and the second passenger. This section provides a numerical example to clarify the complexity of the arrangement of the single

driver – multiple passengers. For instance, if there are 10 ODs for drivers and 15 ODs for passengers. The total matching possibilities in the single driver – single passenger arrangement are 10x15 = 150 possibilities. The total matching possibilities for the single driver – multiple passengers have been increased to (10x15x15x3) + (10x15) = 6,900 possibilities. This possibility increases exponentially when the number of ODs has risen. As a result, the optimization solver will suffer more computational costs, which will increase exponentially when the number of users increases.

The total matching possibilities increase exponentially when the number of Origin-Destination (OD) pairs has risen, as illustrated by the drastic numerical increase in complexity. As a result, the optimization solver will suffer severe computational costs, which prevents large-scale network application using the traditional optimisation methods reviewed in this chapter. Successfully applying multiple passenger matching in a macroscopic simulation environment thus requires a strategic, computationally efficient adaptation of the matching procedure. The design of this adapted framework, including the novel heuristic approach developed to manage these exponential computational constraints, is fully detailed and validated in Chapter 6 (Methodology) and Chapter 7 (Numerical Example). In summary, the single driver – multiple passengers' arrangement can further reduce users' travel costs and benefit system providers. However, the optimization problem will become more complex and cost more computational resources.

3.5 Matching algorithm for taxi sharing

Many taxis e-haling companies have implemented taxi sharing in their system. These companies included Uber and DidiChuxing, the two largest e-hailing service providers who applied the taxi sharing system called UberPool and ExpressPoll, respectively. Similar to the ridesharing system, the matching algorithm for taxi sharing is also an optimization problem. With minor tweaks, all previous algorithms for the ridesharing system mentioned in earlier sections can be applied to the taxi-sharing system. First, the main difference between a ridesharing system and taxi-sharing is the vehicle in which the system is operated. The ridesharing system uses a privately owned vehicle, and the driver is also the commuter. This vehicle ownership means the driver in such a system has their travelling demand with a fixed origin and destination. The ridesharing, therefore, must start and end at the corresponding locations. On the contrary, vehicles that operate the taxi-sharing system do not have a specific destination once released from the depot. Figure 3.3 describes the topology of matching patterns in taxi-sharing.

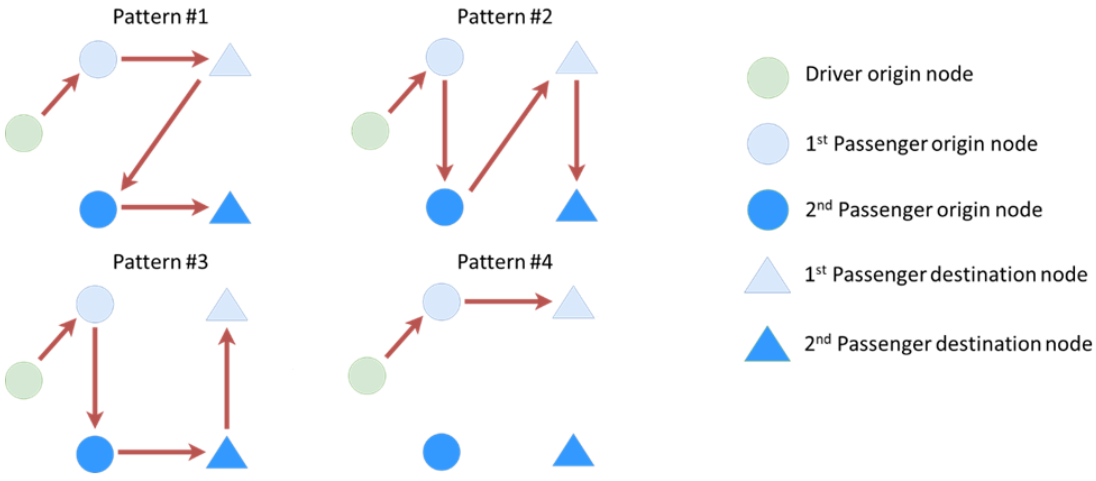

Figure 3.3 Topology of taxi-sharing system

Instead of matching drivers' and passengers' ODs, the matching algorithm for taxi-sharing finds a suitable matching pair with taxis' current location and passengers' ODs. After a taxi is matched, the vehicle departs to pick up the first passenger from its current location. The journey then transverse to all other nodes based on the matching pattern before reaching the final destination, the second passenger's destination node. Given the same network size and dataset, taxi-sharing has less matching possibility than ridesharing. This reduction in complexity is because the number of nodes used as a matching parameter in taxi sharing is typically less than the number of driver's ODs in the ridesharing, given the same network size. Therefore, the computational cost of the matching algorithm for taxi-sharing is less than the ridesharing algorithm. Cost constraints in the taxi-sharing system also require a minor adjustment. The driver's cost constraint in taxi sharing is changed into the following equation (Equation 3-6).

Driver $$C_{(N_t,O_{p1})} + C_{(O_{p1},D_{p1})} + C_{(D_{p1},O_{p2})} + C_{(O_{p2},D_{p2})} - F^{v}_{p1,p2,m} \leq 0$$

Equation 3-6 Cost constraint for Taxi Driver

Where, $N_v$is the taxi v current position and $F^{v}_{p1,p2,m}$ which is the ridesharing fare (incentive) collected from passenger $p1$ and $p2$. Although operators implement different functions to calculate the ridesharing fare for each customer, they share the same principle that the fare is distance-based. This scheme makes the matching pattern also play a crucial part in the optimization. Selecting a suitable pattern determines the distance of the detour, which cannot be charged to any passengers. The equation demonstrates that the matching is viable only if the driving cost to pick up the first passenger is less than the incentive collected from passengers' fares. If the distance between the taxi and passengers is relatively large compared with the passengers' travelling distance, they are more

likely to be unmatched. In other words, the algorithm ensures that all drivers make their profit on all matched trips. For the passenger, the constraints in taxi-sharing remain the same as in the ridesharing algorithm, where the total cost generated in the taxi-sharing must be less than the non-sharing mode.

It is also worth mentioning that in practice, taxi drivers may prefer to end their trips closer to certain desirable areas, such as taxi ranks, high-populated areas, or transport hubs, in order to improve their chances of finding the next passenger. In the framework used in this thesis, passenger assignments are determined by the optimisation model, so the driver's finish point is not directly chosen by the driver. Incorporating such preferences would require additional complexity, but this consideration is reflected later in the vacant taxi movement model (Section 3.5.1), where repositioning behaviour toward attractive areas is explicitly represented.

3.5.1 Vacant taxi movement model

The state of taxi-sharing vehicles can be separated into three stages: vacant, en route and occupied. The vacant stage is when the vehicle does not service any passengers. In this stage, the driver uses their own judgment to navigate the taxi vehicle and seek the customer. In some systems, the operator provides information about the location with high demand to help the driver. The enroute stage occurs once a driver matches a passenger(s). The system provides the driver's route to meet the passenger. Once the driver picks up the first passenger, it changes to an occupied mode.

The vehicle turns to the vacant stage where all passengers are served, or the vehicle does not have a new order provided by the system. The system regulates the navigation on both en route and occupied stage. However, most systems do not mandate but only provide recommended routes to the driver on the vacant stage.

During 1998 and 2010, Yang and Wong developed several models using taxi data in Hong Kong. They first created a macroscopic model to capture the taxi movement on the road network (Yang & Wong, 1998). They assumed that vacant taxi drivers chose the route which attempted to minimize the expected search time to find a customer. The probability of selecting a node was modelled from the logit model with a disutility function. The function consists of the shortest travelling time to the target node and waiting time until picking up a passenger. Once the vacant taxi moves to the target node, it must remain at the node until it picks up a passenger. The model was improved by (Wong, Wong, & Yang, 2001; Yang & Wong, 1998) to incorporate a congestion effect and passenger demand elasticity. However, these models have a weak assumption that a driver must stay at the chosen location until meeting a new passenger. In addition, these models have neglected the case where the vacant taxis travel to remote areas to pick up customers with a potentially high-profit return. For example, passengers at the airport in a remote location have a more extended trip than passengers travelling intracity.

Wong et al. (2005) considered a taxi's revenue in the utility function. The model included the effects of perceived revenues on the behaviour of vacant taxis movement. The drivers perceive the profit, which is then converted into a unit of time. This conversion allows the model to simulate the customer-search behaviour of vacant taxis incorporated with the expected profit that taxi drivers could receive by picking up passengers in particular nodes. Nevertheless, all models mentioned above are in a logit form and have not been calibrated and validated. Wong, Wong & Yang. (2014) calibrated and validated their multinomial logit models using the GPS data collected from taxis, predicting their travel choice for searching customers in peak and off-peak periods. Wong et al. (2015b) proposed two-stage modelling to predict vacant taxi movements. The first stage predicts the zone choice of vacant taxi drivers for finding a new passenger, and the second stage determines drivers' circulation time and distance within the zone. The model was calibrated using the GPS data of 460 Hong Kong urban taxis and separated the city into 18 zones. The vacant taxi movement model used in this thesis is developed based on this study. The study suggested that the mechanism of a vacant taxi driver's decision to stay in the current zone (node) or move to one of the adjacent zones (nodes) is based on the MNL model, which consists of four factors (Wong, Szeto & Wong, 2015a).

- Factor I Relative Passenger Demand E_j which is based on the total number of passengers being picked up at zone j compared with all other zones.
- Factor II Intra-zonal Circulation Distance D_j^c is defined as the average travel distance of all vacant trips for searching a passenger located within the zone boundary.
- Factor III Cross-zonal Travel Distance D_{ij}^t is represented by the average travel distance of all vacant taxi trips travelled from zone i to zone j.

- Factor IV Rate of Return R_{ij} is derived by the rate of profit in the unit of time. The rate of return for a vacant taxi travelling from zone i to zone j for seeking the next custom can be expressed by Equation 3-7.

$$R_{ij} = \frac{-C_{ij}^{t} - C_{j}^{c} + \bar{P}_{j}}{T_{ij}^{t} + T_{j}^{C} + \bar{T}_{j}^{t}}$$

Equation 3-7 Wong's Rate of Return

where C_{ij}^{t} and C_{j}^{c} are total travel cost when taxis moving from zone i and zone j, and intra-zonal circulation within zone j. $\bar{P}_{j}$ is the expected profit from a matched trip starting from zone j. T_{ij}^{t} and T_{j}^{C} are the associated time spent in travel between zone i and zone j, and intra-zonal circulation within zone j. $\bar{T}_{j}^{t}$ denotes the average travel time associated with all matched trip originated from zone j. Given the four factors, the zone choice model for vacant taxi vehicles is formulated (Equation 3-8).

$$P_{q}(j|i) = \begin{cases} 0 & ,if\ j \notin H_{i} \cup \{i\}; \\ \dfrac{\exp\left(\beta^{E}E_{j} + \beta^{t}D_{ij}^{t} + \beta^{c}D_{j}^{c} + \beta^{R}R_{ij}\right)}{\sum_{n \epsilon H_{i} \cup \{i\}} \exp\left(\beta^{E}E_{n} + \beta^{t}D_{in}^{t} + \beta^{c}D_{n}^{c} + \beta^{R}R_{in}\right)} & ,if\ j \in H_{i} \cup \{i\}, \end{cases}$$

Equation 3-8 Wong's Probabilistic function

where $P_{q}(j|i)$ is the probability of a vacant taxi driver navigate the vehicle from node i to node j. H_{i} denotes the set of zones that adjacent to zone i. β^{E}, β^{t}, β^{c}, β^{R} are coefficients associated with the relative passenger demand, the intra-zonal circulation distance, the cross-zonal travel distance, and the rate of return respectively. Zheng, Rasouli & Timmermans. (2018) reviewed numbers of taxi movement models and concluded that the model in Wong, Szeto & Wong. (2015b) is the most complete framework including different behaviours to date.

3.6 Discussion

Route choice models for ridesharing systems are called matching algorithms. Unlike other modes of transport, ridesharing drivers cannot select their route but are governed by a centralized system from ridesharing companies. A model used to determine drivers' routes is called a matching algorithm. The algorithm matches the driver with a suitable passenger based on some criterion and calculates routes that need to be taken. In most cases, the shortest route based on travel time or costs is selected. A matching algorithm is based on an optimization model consisting of three parts: objective function, constraints, and decision variables.

Depending on the service provider, the objective function of the optimization can be minimizing the overall cost of the network, minimizing travel time, or maximizing the number of users. Constraints can be separated into two types: time constraints and cost constraints. Time constraints are used in a model that considers the user's time preference. The constraint ensures that the matched trip does not start before and finish after the preferred time. Cost constraints are used to ensure that the generalized travel cost each ridesharing user pays, if they are matched, is reduced. Otherwise, the ridesharing system does not give any benefit to any users.

Decision variables dictate the type of optimization. All matching algorithms found in this literature are based on either binary or mixed integer programming, where the decision variable $x_{dp} = 1$ if driver d is matched with passenger p, otherwise $x_{dp} = 0$. In this thesis, we proposed an optimization problem based on

linear programming where the decision variable reflects the number of matched drivers and passengers. Furthermore, linear programming has less computational complexity, allowing us to model a ridesharing system on a more extensive network scale.

Matching a single driver with multiple passengers requires more computational costs. These increased computational costs are not only due to extra constraints that need to be considered but also an inclination of decision variables and matching possibilities. Apart from selecting a suitable driver and passengers, the algorithm also calculated the route of the ridesharing trips. In the case of matching one driver with up to two passengers, there are a total of four route patterns that need to be calculated its cost. We have shown that different patterns generated different costs among drivers and passengers. Although the matching of multiple passengers generates makes the algorithm more complicated.

The matching algorithm for taxi sharing is similar to the ridesharing system. The significant difference is that taxi vehicle does not have a fixed destination. Instead of matching drivers' and passengers' ODs, the matching algorithm for taxi-sharing finds a suitable matching pair with taxis' current location and passengers' ODs. Once it has been matched, the sharing trip is finished when the last passenger reaches his/her destination. This route shortening reduces the complexity of the driver's cost constraints in the optimization, making the algorithm easier to solve than the ridesharing system. Apart from the driver's constraint that needs to be adjusted, single driver – multiple passengers' arrangement can be applied in the same manner as the single passenger ridesharing system. The problem with modelling the taxi sharing system is that

after the trip is ended, the vehicle remains in the network. Hence, an additional model that determines the movement of a taxi when it is vacant is required. Vacant taxi models assume drivers chose the route and location, which attempted to minimize the expected search time to find a customer. Earlier models have a weak assumption that a driver must stay at the chosen location until meeting a new passenger. A more recent model proposed by Wong, Szeto & Wong. (2015b) considered multiple factors for the disutility function. The researchers then proposed a multinomial logit model that captures the mechanism of a vacant taxi driver's decision to stay in the current zone (node) or move to one of the adjacent zones (nodes).

In conclusion, this chapter has completed the review of microscopic matching algorithms, constraints, and optimization challenges, confirming the need for a highly efficient, large-scale approach. While traditional solutions rely on computationally expensive integer programming, the potential of adapting a less complex Linear Programming (LP) approach was introduced, alongside the necessity of modelling the complexity of single driver–multiple passenger arrangements and vacant taxi movements. This review directly informs the second objective of the thesis: to develop a macroscopic matching algorithm, based on Linear Programming, that is computationally tractable for large-scale networks, the detailed implementation of which will be described in Chapter 6.

CHAPTER 4 ROUTE CHOICE MODEL FOR PUBLIC TRANSPORT

This chapter continues the literature review by shifting focus to the mathematical models used for public transport assignment, commonly referred to as transit assignment models. It separates these approaches into frequency-based and schedule-based categories, arguing that the frequency-based approach is adopted in this research due to its computational efficiency and compatibility with network-wide analysis, aligning specifically with the overall research aim of integrating modes at a macroscopic scale. The review then details core concepts critical to the model design, including the challenge of the common lines problem, the utility of the hyper-path concept, and methods for incorporating congestion and crowding via soft capacity constraints.

The systematic review involved searching academic databases using key terms such as "transit assignment models," "frequency-based assignment," "schedule-based assignment," "common lines problem," and "hyper-path". Literature was included if it provided a formulation of passenger route choice under network conditions, emphasized computational efficiency for large networks, or addressed the common lines problem through established models such as the route section approach (Chriqui & Robillard, 1975). Special attention was given to literature that discussed the required assumption that passengers arrive randomly, without reference to the transit schedule. Purely schedule-based models focusing on low-frequency systems were generally excluded unless they offered foundational equilibrium concepts relevant to the generalized cost formulation.

4.1 Backgrounds

With an inclination of travel demand, public transport networks are congested and overcrowded, especially during rush hours. This disrupts the system's regularity and reliability, affecting users' travel behaviours. The behaviour is reflected in the user's perception of the performance in the transit system. Factors such as travel time, waiting time, transit time and crowdedness regulated passenger travelling routes. Mathematical models used to determine users' route choices in a public transport system are commonly called transit assignment models. This chapter presents a range of modelling frameworks to formulate transit assignments. Different users have their perceptions, which leads to different travel paths being chosen. Transport planners and researchers have developed transit assignment models which predict the number of users on various services connecting any OD pairs and validate the effectiveness of newly schemes implemented in the transit system. Transit assignment models are broadly separated into frequency-based (or headway-based) and schedule-based (or timetable-based) (Nuzzolo, 2002).

The frequency-based approach represents services in terms of headways or frequencies rather than exact vehicle timetables, thereby modelling the average distribution of passengers over time. It is less data-intensive, easier to implement, and computationally more efficient, which makes it especially attractive for large-scale urban networks (Cepeda, Cominetti & Florian, 2006; Spiess & Florian, 1989). It also provides a practical way to address the common-line problem when passengers arrive randomly at stops. However, the approach

has important limitations: it abstracts from timetables, overlooks schedule coordination, and cannot fully capture service irregularity or vehicle-specific crowding. Consequently, it may not be suitable for systems with long or irregular headways, or where transfer synchronisation plays a critical role (Schmöcker, 2006).

By contrast, schedule-based models represent the network at a disaggregate level, incorporating the precise arrival and departure times of individual runs. This enables passenger choices to be evaluated in real time, accounting for transfer coordination, variable waiting times, and the impacts of delays. Such detail provides greater realism in contehamdouchxts where reliability and timed connections are decisive (Hamdouch & Lawphongpanich, 2008; Nuzzolo, 2002). The drawback, however, is that schedule-based models require extensive operational and timetable data—often unavailable in the early stages of planning—and impose significant computational costs when applied to large networks (Cepeda, Cominetti & Florian, 2006).

The selection of modelling approach must therefore reflect both the research objectives and the characteristics of the transit system under study. In this thesis, the aim is to examine the interaction between shared mobility and public transport at the network-wide scale. Since the focus lies on overall accessibility patterns and modal interactions rather than on the punctuality of individual runs, a frequency-based approach is more appropriate and tractable. In addition, this choice is justified by several practical considerations. First, the method requires less detailed and more straightforward input data, which is particularly advantageous when modelling large and complex multimodal systems. Second,

the lower computational burden allows the analysis of extensive transit networks within reasonable time. Third, the frequency-based framework is more effective for representing situations in which passenger arrivals or vehicle departures have a random element, thereby offering a more straightforward solution to the common-line problem. Fourth, unlike most schedule-based approaches that assume a strict First-In-First-Out behaviour, the frequency-based method avoids unrealistic assumptions about boarding order, which is important in facilities such as long underground platforms where passengers may board simultaneously. Finally, because it is an aggregated representation, the frequency-based model is fully compatible with the other route choice models employed in this thesis, ensuring consistency across the overall simulation framework.

Nevertheless, the generalisability of results must be interpreted with regard to system characteristics. In dense urban bus or metro networks with high service frequency and multiple overlapping routes, the frequency-based approach captures essential passenger behaviour effectively. Conversely, in networks with sparse services or strong reliance on coordinated transfers, such as regional and intercity rail, a schedule-based approach would be more appropriate for capturing realistic user choices (Schmöcker, 2006; Spiess & Florian, 1989).

4.2 Initial approaches

The core of all transit assignment models is that users are always expected to choose the most cost-efficient path to complete their journey. In addition, an arrival time of a transit user at a station is commonly considered independent of others and does not correlate to any transit vehicle's arrival time. Hence,

passengers' arrival rate is assumed as uniformly random. This assumption was adopted in early transit assignment models, such as (Dial, 1967), developed in an era when many transit systems were less reliable and schedule adherence was weak. Without dependable timetables or real-time information, passengers typically turned up at stops without reference to the schedule, making the random arrival assumption both realistic and analytically convenient. While this assumption remains valid for high-frequency transit services today, it is less suitable for low-frequency routes, where passengers are more likely to coordinate their arrival with scheduled departures.

Building on this assumption, the first formal transit assignment model was developed by (Dial, 1967). In this model, the transit network was represented as a set of stations (nodes) connected by trunklines (links), each defined by attributes such as origin, destination, travel time, and serving lines. Within the same link, all lines were assumed to have equal travel times, simplifying the structure. Under random arrivals, if information about the remaining waiting time for a specific line was unavailable, passengers were assumed to board the first vehicle to arrive. The expected waiting time at a node could then be approximated using the following equation.

$$W_s = \frac{0.5}{\sum_{l \varepsilon L_s} f_l}$$

Equation 4-1 Expected waiting time under random arrivals

Where W_s is equal to average waiting time, or interchange penalty, which equals half of the headway (inverse of the service frequency f_l of all service $l \varepsilon L_s$ that leaves node s. This equation implies that the higher frequency of the service is, the more probability it would be for the first arriving vehicle. This equation

implies that the higher frequency of the service is, the more probability that it would be the first arriving vehicle. Hence it carries a greater chance of being chosen by the user. The waiting time until a user boards a transit vehicle is the most crucial factor that needs to be considered, especially in the frequency-based approach. As it directly affects the total travel cost. The service frequency or headways between transit lines are directly associated with users' waiting time at the station. In a case where there is only one service in the station, the average waiting time is equal to half of the headway of the service. In other cases, more service lines or service frequencies reduce users' waiting time at the station based on the equation. In more recent studies, the value 0.5 is commonly replaced by 1, which assumes an exponential distribution of interarrival times. The limitation of Dial's model is that it cannot solve a scenario where the travel time on each service line within the same link is different. Figure 4.1 illustrates the problem of Dial's model:

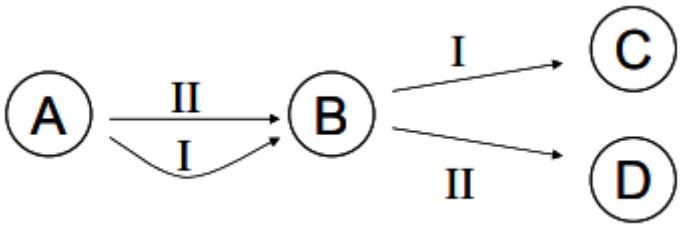

Figure 4.1 Simple transit network with common lines problems

In this figure, the travel time between nodes A and B using Line II is faster than Line I. Under Dial's model, all users travelling from A-C will be assigned to Line I. This might not reflect a realistic behaviour. Depending on the service frequency, travel time, and travel fare. Users may board service II between nodes A-B, then interchange at B for service I to reach C. This dilemma is commonly referred to as a *common line problem*. Le Clerq (1972) developed an

algorithm to simulate the common line problem. The researcher proposed an algorithm that finds the shortest path based on searching at all possible interchanges from the line that the user is currently boarding. Each link's travel and waiting times are embedded in the cost function. The model, however, still assigned all user's demands to the shortest route in terms of cost. Given an example in Fig. X, users with a destination at C will always take Line I if the average waiting time when transferring at B is larger than the additional travel time using line I. On the other hand, if the waiting time at B is shorter, all users will take Line II, transfer at B then board line I to C. To summarize, when the common line problem occurs, whether users board the next run, stay at the stop and wait for a better line, or transfer to other lines depends on the information users use in their decision-making process. The user may not be able to perceive the exact cost of the journey on each alternative but only estimate it and make a decisive decision within their choice.

4.3 Strategy approaches

Users can consider multiple lines and paths to their destination on a larger scale, where there are multiple service lines at the same stop. Lampkin & Saalmans (1967) explained that; in reality, users would only consider paths that are not noticeably bad. For example, two service lines travel between the user's origin and destination. If the travel time of a slower service line is greater than the travel time plus the average headway of the faster line, the user will always consider boarding only the faster line. Transit users can substantially reduce travel time and costs by considering multiple paths (Spiess & Florian, 1989). They denote that these considering paths are called strategies. After a set of

strategies have been chosen, the user will board the first transit vehicle that arrives. However, with this model, it is the first arriving vehicle that dictates the user's choice, not the user itself.

Nguyen & Pallottino (1988) demonstrated a similar approach by introducing a terminology called hyper-path. Similar to strategies, A hyper-path consists of a set of paths that the transit users are likely to choose. The probability of choosing a path is based on the service frequencies. This reduces the complexity of the transit assignment model in determining the shortest hyper-path, as the distribution of the user's demand is dictated by the frequency, which is a known variable. Cominetti & Correa (2001) introduced a shortest hyper-path algorithm based on Dijkstra's shortest-path algorithm for the road network. The model yields the same result as Spiess and Florian's model. However, the researcher claimed that their model is more flexible and has less computational cost in a more extensive transit network. They also suggested that the hyper-path should only contain a set of route S that minimizes the expected travel time, which can be calculated with the Equation 4-2:

$$.T_s = E(W_s) + \sum_{i \epsilon S} t_i H_i(S)$$

Equation 4-2 Expected Travel Time with a set of Route

where T_s is the expected travel time with a set of route S, $E(W_s)$ is the expected waiting time, t_i is the expected travel time of route i and $H_i(S)$ is the probability that route i is served first which is based on the service frequencies. Chriqui & Robillard (1975) suggested a heuristic algorithm to solve this problem. The algorithm sorts the route based on their total travel time in ascending order. The algorithm started from registering the shortest route with the lowest t_i into the

hyper-path. The algorithm then adds the second shortest route and checks the expected travel time T_s. If the expected travel time is reduced, the algorithm then added another route. The algorithm then added another route if the expected travel time is reduced. Given this heuristic algorithm, the problem still becomes difficult to solve where the calculation of all possible routes is required. (Spiess & Florian, 1989) proposed an algorithm based on linear programming to find the optimal strategy. The objective of the LP is to determine the strategy the minimizes the expected total travel time and waiting time for all users' flow from all origin nodes to a single destination node. However, the algorithm needs to be solved for each destination node separately. The result demonstrated that the model satisfies the user equilibrium condition.

4.4 Route section approach

The route sections approach was introduced by (Chriqui & Robillard, 1975). The model coded the transit network in terms of route sections. Compared to the previous approach, this approach creates additional links to reduce the complexity of the transit network. For example, a route between node A and C using service I without transit at B has a new link transverse directly from A to C, shown in Figure X. This additional ink is called the "route section". Hence each route section has only the origin and the destination node, and the waiting time value will only have to be considered at the origin node of each route section (de Cea & Fernandez, 1993). For illustration purposes, a transit network interpreted with the route section approach is presented in Figure 4.2 based on the simple network in Figure 4.1.

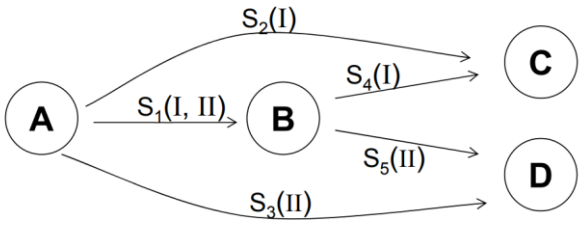

Figure 4.2 Transit network represented with route selection approach.

The researchers also proposed a travel time calculation for multiple service lines travelling between the same OD pair but with different travel times. For example, route section S_1 have two services which have different travel times between nod A and B. The expected travel time t_s can be calculated using the Equation 4-3 below, which is based on the proportion of the service frequency on each transit line.

$$t_S = \frac{\sum_{j \in S} f_j t_j}{\sum_{i \in S} f_i}$$

Equation 4-3 Expected Travel Time with multiple service frequency

In this approach, after route sections, travel times and waiting times are calculated, the model will assign passengers to the route section that minimizes the user's cost. If the route section contains more than one service line, the user's flow is assigned to each line according to its service frequency. Unlike the previous approach, the model does not consider strategies leading to a general suboptimal solution (Schmöcker, 2006). The merit of the strategy approach is that the optimal set of attractive routes is easily determined, but this generates more complexity in the transit network. On the other hand, the route section approach can reduce the number of required links in the network when the

number of common lines is high. Cea et al. (1988) explored the difference in results generated from the strategy and the route section approaches. The comparison was conducted on a large-scale experimental network in which most of the links have common lines. They concluded that their strategy approach leads to sparse and uniformly distributed travellers' flows compared to the route section approach. The difference in results between these two approaches is minimal, but the route section approach requires a higher computational cost. In reality, the difference between the two approaches is heavily based on the fare scheme. If the fare is charged every time the user board a new transit vehicle, the result will be almost no difference. If the fare is a fixed rate, the result differences will likely increase since the route section approach does not consider the alternative and feasible route.

4.5 Congestion in transit networks

The aforementioned transit assignment models are uncongested models which do not consider the user's congestion and crowding factor. In uncongested models, transit users can board a transit vehicle without extra delay or costs due to the crowdedness of the vehicle. In reality, transit users may experience these effects when travelling through the transit network in vehicle carriages or waiting at the platform. This is because the facility or service cannot supply the travel demand with a given operational service level. This situation typically occurs during rush hours, such as morning and evening. In terms of modelling, researchers consider the impact of the congestion in two ways.

The first impact is Fail-to-board passengers who cannot board the first coming vehicle due to limited space. This inaccessibility is due to the vehicle's capacity having been completely fulfilled, and it cannot board more passengers. Hence, those passengers must wait for the following vehicle. The second impact is the discomfort experienced by standing and sitting with other users. This negativity leads to different cost functions regarding in-vehicle travel costs and ultimately affects travellers' behaviour and travel strategies. While travelling in a crowded vehicle, the degree of discomfort increases when the user's flow increases.

de Cea & Fernandez (1993) developed a model where the user's waiting time increases when the user's volume increases. This volume-dependent waiting time is considered a congestion cost function related to the notion of effective frequency. The idea behind this is that the more transit vehicle arriving at full capacity, the more waiting time. This is because the vehicle becomes harder or fails to board. If the transit vehicle arrives empty, the effective frequency is equal to the nominal frequency. The following Equation 4-4 of the cost function of the route section s.

$$C_s = \bar{t}_s + \frac{\alpha}{f_l} + \varphi\left(\frac{\tilde{V}_s}{K_s}\right)$$

$$w_l^i = \frac{\alpha}{f_l} + \varphi\left(\frac{\tilde{V}_s}{K_s}\right)$$

Equation 4-4 De Cea's Cost functions

Where C_s is the total travel cost of route section $s, \bar{t}_s$ is expected travel time calculated with the previous equation. The waiting cost term consists of two parts where the first part is used to calculate the waiting time based on the

nominal frequency f_1. The second part is a function φ_s based on the number of users who already occupied the vehicle denoted as $\tilde{V}_s$, the route section flow V_s and the vehicle capacity K_s. The function φ_s takes explicitly into account of the effect of congestion on waiting times. De Cea and Fernández suggested that the function should be such that C_s is strictly monotone increasing with V_s. This property ensures that as passenger flow increases, the generalised cost also increases consistently, preventing flat or ambiguous cases. In practice, monotonicity guarantees that the assignment model produces stable and unique solutions. They also suggested a power function $\left(\frac{V_s+\tilde{V}_s}{K_s}\right)^n$ with n between 4 and 6 based on their experience for car network. Note that K_s is not an actual capacity of the vehicle but it is more likely a practical capacity but as n increases K_s acts more and more like a real capacity.

There are two approaches to represent capacity constraints. De Cea and Fernández's model is considered a soft capacity constraint approach, allowing a transit user to board a vehicle even when fully occupied by modelling extra waiting costs. The other approach is the hard capacity approach, which does not allow any passenger to board the fully occupied vehicle. In some scenarios, the hard capacity constraints approach is more realistic than the soft capacity constraints; for example, bus users in Hong Kong must be seated all the time when they are on board. However, a hard capacity constraints approach may not be able to provide a converged solution due to insufficient service capacity This leads to infeasible solution in large-scale models, since passengers may be left without any feasible path to their destination.

The advantage of the soft capacity constraint approach is that the solution must exist, given that the solution set is convex, and the mapping function is continuous (Jiang & Szeto, 2016). The disadvantage of the soft capacity is that it generates unrealistic link flows over the capacity. Nevertheless, De Cea and Fernández argued that their approach might be sufficient for strategic planning, where the focus is on aggregate demand patterns over a longer period rather than exact flows on individual services. In this thesis, the soft capacity approach is adopted for this reason: it reflects the research purpose of analysing system-wide and long-term dynamics, while ensuring feasibility in all scenarios. To reflect reality, the waiting cost is also set to increase sharply as flow approaches capacity, discouraging excessive loading and capturing the discomfort of overcrowding.

4.6 User Equilibrium and Stochastic assignment in transit network

The concept of user equilibrium (UE) in transit networks is adapted from road traffic assignment, first introduced by (Wardrop, 1952) . In road networks, equilibrium is reached when all used paths between an OD pair have equal and minimal travel times, reflecting an analogue of equilibrium in neoclassical economics. In this context, the generalised travel cost is strictly monotone with respect to flow, meaning that as flow increases, perceived cost consistently increases, which ensures stability of the solution. In public transport networks, however, additional elements influence the generalised cost beyond in-vehicle time. Passengers may experience extended waiting times due to denied boarding, reduced comfort when travelling in crowded vehicles, and increased schedule

unreliability. These factors complicate the equilibrium formulation and require modifications to the classical UE framework in order to represent traveller behaviour realistically (Liu, Bunker & Ferreira, 2010).

Several formulations of user equilibrium (UE) transit assignment models have been proposed, building on the work of (de Cea & Fernandez, 1993; Spiess & Florian, 1989; Wu, Florian and Marcotte, 1994). These models typically employ non-linear cost functions to capture the rise in perceived travel cost associated with overcrowding on transit links. (Bouzaiene-Ayari, Gendreau & Nguyen, 1995) highlighted limitations in Wu's formulation, noting that its assumptions regarding waiting time and in-vehicle travel time functions were unrealistic. To overcome these shortcomings, they reformulated the problem as a variational inequality, offering greater flexibility in representing passenger behaviour under varying operating conditions. Subsequently, (Cominetti & Correa, 2001) demonstrated that equilibrium properties could be obtained under relatively weak conditions, even when user-flow travel time functions are not strongly monotonic. In their formulation, waiting times were represented using a simplified bulk-queue model, providing a tractable yet behaviourally meaningful representation of passenger interactions under capacity constraints

Building on these equilibrium-based formulations, the next stage of development in transit assignment model is incorporating stochastic element. The stochastic assignment based on the random utility model was introduced in the 70s. Stochastic assignment in the transit network is also adopted from the traffic network. Further detail on the stochastic assignment will be discussed in the next section. (Hunt, 1990) proposed a logit model for transit route choice.

(Pursula & Weurlander, 1999) followed this and investigated the impact of the level of service factor in the transit system and route choices. (Douglas & Karpouzis, 2006) estimated the impact of overcrowding using a simple logit model and calibrated it with data from the Sydney transit system. Stochastic user equilibrium (SUE), introduced by (DAGANZO & SHEFFI, 1977) also has been applied to the transit assignment. (Lam, Cheung & Lam, 1999) proposed mathematical programming for SUE assignment based on a multinomial logit model. The researchers proved that when the link-capacity constraints are reached. Their mathematical programming yields an equilibrium result in the overcrowded network. (Nielsen, Otto Anker, 2000) presented another SUE transit assignment based on a multinomial probit model. (Nielsen, Otto & Frederiksen, 2006) developed an SUE transit assignment model using the nested logit model and solved it with the method of successive average which is a heuristic approach to solve the SUE problem.

4.7 Discussion

The route choice model for public transport is called the transit assignment model used to determine transit users' route choice. It can be classified into two types: schedule-based and frequency-based approaches. The schedule-based approach model at a disaggregated level. The approach requires detailed transit data, such as the vehicle's arrival and departure time. This allows the approach to modelled users' behaviour individually. However, as a result, it requires more parameters to be calibrated, which generates more computational costs. This computational problem makes the approach not suitable for a large-scale network. On the other hand, the frequency-based approach uses aggregated data.

The approach treats transit users as a group of travel demands with the same behaviours. The frequency-based approach requires less detail and uses a more straightforward input. This makes the approach consume less computation cost. Hence it is more suitable in a more extensive network.

In this thesis, we focus on the frequency-based transit assignment model. There are two unique parameters, service frequency and waiting time, in transit assignment that cannot be found in other modes. The service frequency is directly associated with the waiting time. (Dial, 1967) proposed a waiting time function based on service frequencies in early development.

A common lines problem occurs when multiple service lines have different travel times serving at the same transit node. The common line problem is a unique problem in transit that cannot be solved directly because the actual travel cost on each line cannot be determined. Two approaches are mainly used to tackle this problem. The strategy approach assigns transit users to predetermined routes called strategies or hyper-paths. These routes are generated based on expected waiting times and travel times. Once the strategy is determined, the demand is split based on service frequencies. The route section approach reduces the complexity of the transit network with multiple common lines by creating additional links (Chriqui & Robillard, 1975). After the new links or sections have been generated, the model assigns users to the section that minimizes travel costs. The solution from the two approaches is minimal, while the strategy approach requires less computational cost.

Congestion in transit networks is treated in two ways. First, the congestion makes some users unable to board the first arriving vehicle. (de Cea & Fernandez, 1993) proposed an effective frequency model where Fail-to-board users suffer a longer waiting time. This delay is equivalent to a decrease in the service frequency. The researchers modelled this from the system point of view, where the congestion reduces the actual Frequency of the service. Second, the congestion makes an on-broad crowdedness treated as an extra cost in transit. However, there are fewer user equilibrium assignment models for transit compared with the road traffic network (Cominetti & Correa, 2001). To the best of our knowledge, limited studies focus on UE (Bouzaiene-Ayari, Gendreau & Nguyen, 1995; Cominetti & Correa, 2001; Wu, Florian and Marcotte, 1994) and SUE (Lam, Cheung & Lam, 1999; Nielsen, Otto & Frederiksen, 2006; Nielsen, Otto Anker, 2000).

This chapter reviews successfully detailed the necessary components for implementing the public transport component within the multimodal framework. The discussion clarified the necessity of adopting the frequency-based assignment model over the schedule-based approach, justifying this choice based on the macroscopic research purpose and computational tractability, a key constraint for achieving the objectives of the entire integrated system. Concepts such as the route section approach and the application of soft capacity constraints, drawn from the work of De Cea and Fernández, provide the specific technical elements required to meet the first objective of the thesis: incorporating a fully featured public transport mode into the simulation framework.

CHAPTER 5 ROUTE CHOICE FOR PRIVATE TRANSPORT

This chapter completes the theoretical foundation by undertaking a literature review of route choice models specific to private transport, known as traffic assignment models. The review first distinguishes between Static Traffic Assignment (STA) and Dynamic Traffic Assignment (DTA), justifying the use of the STA approach—and its semi-dynamic adaptation detailed later—for computational efficiency in large networks. Key deterministic models are examined, ranging from the simplistic All or Nothing (AON) assignment to complex User Equilibrium (UE) models, supported by volume-delay functions like the well-known US Bureau of Public Roads (BPR) function. Finally, the chapter reviews heuristic solution techniques like the Frank-Wolfe algorithm and the core principles of Stochastic User Equilibrium (SUE) assignment.

The search strategy targeted concepts essential for establishing network equilibrium and simulating driver behaviour under congestion. Keywords included "Traffic Assignment," "User Equilibrium (UE)," "Stochastic User Equilibrium (SUE)," "BPR function," and "Volume-Delay functions". Key search engines and databases employed included scholarly platforms such as Imperial College Library, ScienceDirect and Google . A specific criterion for inclusion involved literature providing critique and detailed information about the US Bureau of Public Roads (BPR) function, examining its simplicity, computational efficiency, and documented limitations, as requested by the examiner. Purely empirical studies lacking mathematical model context were typically excluded.

5.1 Backgrounds

City growth is resulting in an inclination in travel demand and private vehicle ownership, especially in urban areas. However, existing infrastructure in most cities has failed to keep up with the increasing demand. This capacity restriction leads to traffic congestion, pollution, and accidents occurring in the road network. Moreover, traffic congestion gives commuters an extra delay and decreases the network's reliability (Saw, Joshi & Katti, 2015). The traffic assignment is the last stage of the infamous 4-step model of travel demand modelling. Note that the model is sometimes called route assignment or route choice model. This chapter will review different traffic assignment models for private transport vehicles. The main objectives of the traffic assignment model are (1) to estimate the volume of vehicle traffic on the network, (2) to estimate the travel cost between zone, (3) to analyse the travel pattern of each origin and destination pair and (4) to identify the congested links and other traffic data (Mathew & Rao, 2007).

The traffic assignment model can be coarsely classified into two categories: Static Traffic Assignment (STA) and Dynamic Traffic Assignment (DTA). STA models emphasize the aspect of transportation planning, while DTA models focus on traffic control and management. The theory of static traffic assignment assumes that the O-D demand is uniformly distributed to estimate the traffic pattern (Jayakrishnan, Tsai & Chen, 1995). Most models in static traffic assignment have low complexity and consume less computational cost compared with other types. Thus, it has been widely used for strategic transportation planning. On the other hand, dynamic traffic assignment models describe the

dynamic characteristic of the traffic flow and network. The model captures time-varying changes in traffic flow and time-dependent traffic demand (Zou et al., 2013). DTA uses historical and real-time data to estimate and forecast traffic conditions. However, DTA is more complex and requires higher computational resources. Hence, STA models are still widely used in large and complex networks. In this thesis, we aggregated both ridesharing and public transport demand. Our route choice models in other modes aggregated traveller's demand. Therefore, it is more logical to use STA for private transport demand. In addition, we introduced a semi-dynamic traffic assignment which captured some temporal effects of the travel demand and network. This semi-dynamic traffic assignment allows us to retain some benefits of the DTA models while not generating too much complexity to the simulation framework. More detail of the proposed semi-traffic assignment can be found in the next chapter.

5.2 All or Nothing assignment

The simplest traffic assignment model is the All or Nothing (AON) assignment. The method is based on the principle that all traffic is assigned to the route having the least travel resistance. The travel resistance is based either on travel cost, travel time or travel distance. However, the AON assignment does not consider the congestion effect. This assumption makes the travel resistance, especially the travel time, constant across all network segments. Another assumption is that all drivers consider, perceive, and weigh the factor the same way. Therefore, all drivers are assigned to the only route with the least resistance between each origin and destination pair, and no vehicle is assigned to other less

attractive routes. Figure below illustrates a simple traffic network with three routes that will be used for entire chapter,

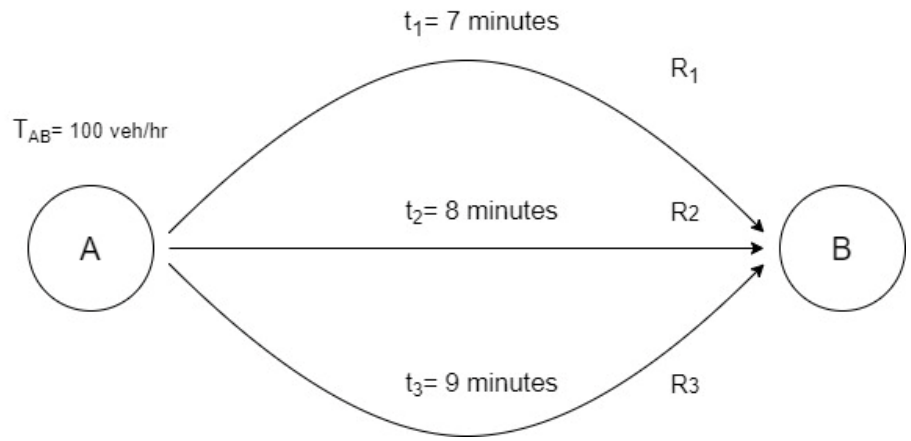

Figure 5.1 Simple traffic network with three routes

In Figure 5.1, the OD demand between nodes A and B: T_{AB} is equal to 100 vehicles per hour. There are three possible routes between nodes A and B. Each route has a travel time of 7, 8 and 9 minutes, respectively. Note that there is no congestion effect in this network. This means that the travel time is fixed and does not increase when the route flow is increased. The All or Nothing assignment's first step is finding the shortest route between each OD pair. In this example, we consider the travel time as the travel resistance. We found that route R_1has the least travel time $t_1 = 7$ minutes. When the traffic network is more complicated, shortest path algorithms such as Dijkstra's algorithm are commonly used to find the route with the least cost Next, after the shortest route is founded, we assign all the demand to the given route. Therefore, route flow $V_{r1} = 100$, $V_{r2} = 0$ and $V_{r3} = 0$. The assignment is finished at this step because there is only one OD demand in this network. If there is more than one OD Demand, the assignment continues until all demands have been assigned.

The AON model is the most unrealistic compared to other traffic assignment models. In reality, road travellers may not consider only the shortest route but also the reliability and safety of the route. Furthermore, some assumptions are unrealistic such as the travel time and cost do not vary with the link flow, and all travellers have full knowledge of the road network. This assumption is plausible in sparse and uncongested networks where the number of possible routes is limited, and the difference in cost is substantial. Due to a lack of realistic behaviour, the All or Nothing assignment is not very popular for transport planners. It may be used to determine a preferred route that drivers would use in the absence of congestion. However, this model is a building block for other types of assignments, such as User equilibrium and Stochastic assignments, which are more realistic and popular.

5.3 Volume-Delay functions

Consider the previous example; all traffic demand between A-B is assigned to a single route R_1. If the route's capacity is strictly limited, the travel time on the route will be increased. This may result in a higher travel time than the other two routes. Travellers who perceive the congestion in the route R_1 may alter their trip to other less congested routes. In traffic assignment that considered the congestion effect, the relationship between link travel time and link flow does exist. Based on a speed-flow relationship, vehicle speed tends to decrease from the free flow speed when link flow increases. This relationship consequently increases the travel time in the link. Traffic assignment models with congestion effect require a suitable function to describe this relationship reflecting link flow and link costs. Several researchers have suggested this cost-flow relationship

function. (Branston, 1976) reviewed several proposed functions, which are the following:

Smock (1962) proposed a link capacity function for use in the Detroit Area Transportation Study:

$$t = t_0 \, exp\left(\frac{V}{C}\right)$$

Equation 5-1 Smock's Volume-Delay function

where t is equal to link travel time per unit distance, t_0 is travel time per unit distance under the free flow traffic condition and C is the link's capacity. Smock estimated the value of C for each link from an average intersection's capacities at the end of the link. This makes their link's capacity less than ideal.

The UK Department of Transport published a large number of cost-flow for each link types such as urban, sub-urban and inter-urban roads. The general form of speed-flow functions is:

$$s(V) = \begin{cases} S_0 & V < F_1 \\ S_0 - \dfrac{S_0 - S_1}{F_2 - F_1}(V - F_1) & F_1 \le V \le F_2 \\ \dfrac{S_1}{\left[1 + \left(\frac{S_1}{8d}\right)\left(\frac{V}{F_2} - 1\right)\right]} & V > F_2 \end{cases}$$

Equation 5-2 The UK DoT Volume-Delay function

Where S_0 is the free flow speed, S_1 is the speed at flow capacity, F_1 is the maximum flow at which the speed remains at the free flow condition, and d is

the link's distance or length. Therefore, the time-flow function can be described as:

$$T(V) = \begin{cases} d/S_0 & V < F_1 \\ d/S(V) = \dfrac{d}{S_0 + SS_{01}F_1 - SS_{01}V} & F_1 \le V \le F_2 \\ d/S_1 + (V/F_2 - 1)/8 & V > F_2 \end{cases}$$

$$SS_{01} = \frac{S_0 - S_1}{F_1 - F_2}$$

Equation 5-3 The UK DoT Time-Flow function

Department of Transport (1985) also provided a table of recommended parameters S_0, S_1, F_1 and F_2 based on road types. For example, parameters S_0, S_1, F_1 and F_2 for 2 lanes Dual carriageway in rural area are 79 km/hr, 70 km/hr, 1600 pcu/h/lane and 2400 pcu/h/lane, respectively.

The US Bureau of Public Road (1964) proposed one of the best-known and the most widely used volume-delay functions in transport planning. The function is commonly referred to as the US BPR function. It defines the relationship between link travel time and the ratio of flow to capacity (V/C) through the following general form,

$$t = t_0 \left[1 + \alpha \left(\frac{V}{C}\right)^{\beta}\right]$$

Equation 5-4 The US BPR Volume-Delay function

where $1 + \alpha$ is the travel time ratio per use distance, α and β is parameters for calibration. The US BPR also suggested typical default values of 0.15 and 4 for α and β, although these values can be adjusted to reflect local conditions.

The main strength of the BPR function lies in its simplicity, computational efficiency, and ease of calibration. With only two parameters, it can be readily applied across large networks and is supported by extensive empirical evidence and widespread adoption in practice (Ortúzar & Willumsen, 2011). These characteristics make it particularly suitable for network-wide modelling where computational tractability is essential.

However, the function has also been criticised. It is essentially empirical and does not explicitly represent important operational features such as queue spillback, intersection delay, or capacity breakdown under oversaturated conditions (V/C > 1). As a result, it may underestimate travel times in heavily congested situations. Furthermore, its monotonic assumption—that travel time always increases smoothly with volume—does not fully capture the stochastic and unstable traffic regimes observed in real networks. Alternative formulations, such as (Akçelik, 1991), incorporate queuing delay and oversaturation effects but with at the cost of additional complexity.

Akcelik' s function. The functions mentioned above have two major drawbacks; (1) The function does not consider delay for junctions. (2) The function has a high error when the link flow reaches the capacity. This inaccuracy makes those models less desirable when using it with an urban area network where delays from junctions are dominated, and traffic is congested most of the time. Akçelik, (1991) proposed a link cost function based on earlier work by (Davidson, 1978):

$$t = t_o \left\{ 1 + 0.25 r_f \left[z + \sqrt{z^2 + 8\, J_D x / r_f} \right] \right\}$$

Equation 5-5 Akcelik's Volume-Delay function

where, J_D is a delay parameter, $z = x - 1$, x is a degree of saturation = V/C, r_f is a ratio of flow period to minimum travel time = T_f/t_0 and T_f is the flow modeling period which is typically one hour. The function is designed to approximate the delays due to queuing when the flow exceeds the capacity. Hence, there is no limit on the value x that can be used in the function.

Branston concluded that the calibration of these volume-delay functions is time-consuming and requires a large dataset, including values such as observation of travel times on links under different levels of link flow. Therefore, there is a limited study on these functions, and many countries have developed their functions. In this thesis the US BPR function has been selected. The BPR function is a commonly used method to model the relationship between traffic volume and travel time delay on a roadway network. It is preferred over other volume-delay functions because it accounts for the non-linear relationship between traffic volume and delay, is easy to calibrate with only two parameters, and has widely available data. Additionally, its extensive use in transportation research and practice has led to a large body of knowledge and experience in interpreting its results, making it a practical and effective tool for modelling traffic flow.

5.4 Wardrop's equilibrium

Wardrop (1952) formulated two principles of traffic assignment. The first principle is stated that under user equilibrium conditions traffic arranges itself in congested networks in such a way that no individual trip maker can reduce his path costs by switching routes. Therefore, Wardrop's first principle describes an

equilibrium state of the traffic assignment called user equilibrium. The User equilibrium (UE) is a user-driver traffic assignment where each traveller chooses the most convenient route in a selfishly way. This guarantees that all travellers are treated equally. When the UE is reached, all travellers that share the same origin and destination will experience the same travel time or cost. However, the main drawback of the UE is that the total travel time or cost of the system is not minimized. The second principle is stated that under social optimal conditions, traffic should be assigned to the network in such a way that the average (or total) travel cost is minimized. The System Optimum (SO) is an efficient system-driver traffic assignment in which travellers are assigned to a route on the network in such a way that the utilize the network and generates the minimum total travel cost. However, with SO, some travellers might experience a higher travel time than other travellers who have the same origin and the destination.

5.4.1 User Equilibrium

To illustrate the UE and SO assignment, we provide another example shown in Figure 5.1. In this example, there are two routes connecting Origin node A de A to Destination node B. The total OD demand from A to B equals to 8000 PCU/hr. As we consider the congestion effect in this network, travel time on each route is not fixed but under a link-cost function, which are:

$$t_1 = 15\left(1 + 0.15\left(\frac{v_1}{1000}\right)^4\right) \qquad (Ex.5.4.1-1)$$

$$t_2 = 20\left(1 + 0.15\left(\frac{v_2}{3000}\right)^4\right) \qquad (Ex.5.4.1-2)$$

Where t_1 and t_2 are travel cost in term of travel time of route 1 and 2 respectively, and v_1 and v_2 are their corresponding route flow. In order to implement the UE traffic assignment, we apply Wardrop's first principle. At the UE condition, all routes within the same OD pairs have equal cost. In other words, $t_1 = t_2$. By equating t_1 to t_2 it is now possible to find the UE solution as a function of the total flow $v_1 + v_2 = 8000$:

$$2.25\left(\frac{v_1}{1000}\right)^4 = 5 + 3\left(\frac{8000 - v_1}{3000}\right)^4 \qquad (Ex.\,5.4.1-3)$$

Based on above equations (Ex. 5.4.1 – 3), the equilibrium flows on each route are $v_1 = 2152.52$ and $v_2 = 5847.48$ PCU/hr. This generates route travel time $t_1 = t_2 = 63.30$ minutes which satisfy the UE condition. Next, we implement the SO traffic assignment with Wardrop's second principle. At the SO condition, total travel cost on all routes within the same OD pairs is minimized. The total cost functions can be written as (Ex. 5.4.1 – 4):

$$S(V) = V_1 t_1 + V_2 t_2 \qquad (5.4.1-4)$$

$$S(V) = v_1\left(15 + 2.25\left(\frac{v_1}{1000}\right)^4\right) + (8000 - v_1)\left(20 + 3\left(\frac{8000 - v_1}{3000}\right)^4\right) \quad (Ex.\,5.4.1-5)$$

Next, we use the first derivative to find the minimum point v_1.

$$\frac{d}{dv_1}\left(v_1\left(15 + 2.25\left(\frac{v_1}{1000}\right)^4\right) + (8000 - v_1)\left(20 + 3\left(\frac{8000 - v_1}{3000}\right)^4\right)\right) = 0\ (Ex.\,5.4.1-6)$$

Based on the above equation (Ex. 5.4.1 – 6) , route flows that satisfy the SO conditions are $v_1 = 2118.5$ and $v_2 = 5881.5$ PCU/hr, which generates route travel time $t_1 = 60.32$ minutes, and $t_2 = 64.32$ minutes. With SO, the total cost is 8434.68 PCU-hr., compared with total cost from UE, which is 8440 PCU-hr. Hence, the SO assignment saves the total travel time by 5.32 PCU-hr. These results confirm that the UE is not always the best traffic assignment form the system point of view and the result is suboptimal compared to the SO assignment. In order to quantify this suboptimality, the Price of Anarchy (POA) is used as a performance measure of UE and SO (Zhang, J. et al., 2016).

In a more complex network, the problem cannot be solved algebraically. Several techniques have been proposed to approximate Wardrop's equilibrium. While some techniques are simple heuristic approaches, the most noticeable technique use a mathematical programming framework. UE solution can be achieved with mathematical programming proposed by (Beckmann, McGuire & Winsten, 1956).

Minimize Z $= \sum_a \int_0^{x_a} t_a(x_a)dx,$

Subject to,

$$\sum_k f_k^{rs} = q_{rs}: \forall r, s$$

$$x_a = \sum_r \sum_s \sum_k \delta_{a,k}^{rs} f_k^{rs}: \forall a$$

$$f_k^{rs} \geq 0: \forall k, r, s$$

$$x_a \geq 0: a \in A$$

Equation 5-6 Beckmann's UE formulation

In the case of SO, the solution can be achieved by solving another mathematical programming proposed by Sheffi (1985).

$$\text{Minimize } Z = \sum_a x_a t_a(x_a) dx,$$

$$\text{Subject to, } \sum_k f_k^{rs} = q_{rs} : \forall r, s$$

$$x_a = \sum_r \sum_s \sum_k \delta_{a,k}^{rs} f_k^{rs} : \forall a$$

$$f_k^{rs} \geq 0 : \forall k, r, s$$

$$x_a \geq 0 : a \in A$$

Equation 5-7 Sheffi's SO formulation

where, t_a is the travel time on link a, x_a is equilibrium flows in link a, k is the route, f_k^{rs} is the flow on route k connecting OD pair $r-s$, q_{rs} is the total OD demand that travelling from r to s and $\delta_{a,k}^{rs}$ is a binary variable, which equal to 1 if link a is a part of route k and equal to 0 for other cases. The first constraint is the flow conversation constraint to ensure that a summation of route flow is equal to the demand. The second constraint is the link flow constraint which is used for loading route flow into corresponding links. The third and the fourth constraints are non-negativity constraints to ensure that links flow and route flow are positive values.

5.5 Heuristic Equilibration techniques

The heuristic approaches to the UE problem described in this section contain incremental assignment and Method of Successive Averages. The core of these methods is located in the network loading mechanism. Network loading is the

process that assigns the OD demands to the network. The process follows the route choice model, which underlies all traffic assignment models. These heuristic algorithms involve three steps process. (1) Calculate route travel time. In this step, travel time or travel costs on all used routes are calculated by the loaded traffic from the previous iteration. Most approaches used a free flow travel time for the first iteration as an initial condition. (2) Generate a set of shortest routes based on the current link travel time or costs. (3) Temporary load the traffic best on the criterion. If the stopping criteria are not met, the algorithm then proceeds back to step (1) until the condition is satisfied. Next, we provide another example of a traffic assignment problem. In this example (Figure 5.2), there are three routes connecting nodes A and B. The total OD demand between A and B is 100 PCU/hr. Following figure demonstrates the traffic network and corresponding link cost functions (Ex.5.5 – 1:3).

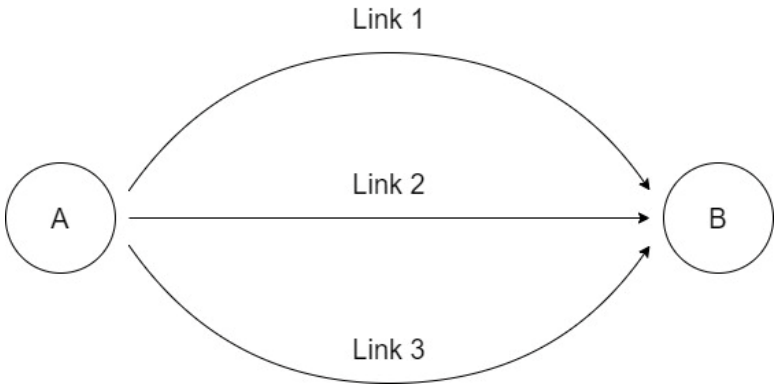

Figure 5.2 Traffic network and corresponding link cost functions

$$t_1 = 7\left(1 + 0.15\left(\frac{v_1}{30}\right)^4\right) \qquad (Ex.5.5-1)$$

$$t_2 = 10\left(1 + 0.15\left(\frac{v_2}{30}\right)^4\right) \qquad (Ex.5.5-2)$$

$$t_3 = 16\left(1 + 0.15\left(\frac{v_3}{30}\right)^4\right) \qquad (Ex.5.5-3)$$

5.5.1 Incremental assignment

The incremental assignment is a heuristic method for attaining the UE solution. The method assigns a fraction of the demand at each iteration. The travel time are then updated on each iteration. Before starting the incremental assignment, one must select a set of proportional factors P_n such that $\sum_n P_n = 1$. After a set of proportional factors is selected, a fraction of the demand is loaded incrementally on the shortest route calculated from the last accumulated flows. Typical values for P_n are: 0.4, 0.3, 0.2 and 0.1 The procedure of the incremental assignment is outlined below:

Select an initial condition of link costs, typically the free flow speed condition is used. Initialise all flows v_a=0.

Select a set of fractions P_n of the total OD demand V such that that $\sum_n P_n = 1$, that $n = 0$.

Find the shortest route x on each OD using the current travel time (or costs); make $n = n + 1$.

Load $T_n = P_n V$ using all or nothing assignment to the shorthest route, obtaining a set of auxiliary flow F_a ; accumulate flows on each link:

$$v_a^n = v_a^{n-1} + F_a$$

Equation 5-8 Accumulate flows

Update a new set of link travel time (or costs) based on the flows v_a^n; if all fractions of V have not yet been assigned, proceed to step 3; otherwise stop.

Consider Figure 5.2, we split the demand of 100 PCU/hr. into four increments of 0.4, 0.3, 0.2, 0.1 of this demand, in other words, 40, 30, 20, 10 PCU/hr. At each increment we calculate the new travel time using given equations. Appendix E1 demonstrates the calculation of this incremental assignment algorithm:

The calculated travel time on each route using the incremental assignments is $t_1 = 23.8$, $t_2 = 14.75$ and $t_3 = 16$ minutes. It can be seen that the algorithm does not converge. In this case, the user equilibrium condition does not meet. This is because the method cannot reduce the excess flow once it has already been loaded. Therefore, the flow on route 1 remains overestimated. The incremental assignment does not necessarily converge to Wardrop's user equilibrium even though the fraction number P_n is increased. The assignment has a significant limitation: once the flow has been loaded to the network, it cannot be removed or transferred to another route. Hence, if earlier iterations assign too much traffic on a link for equilibrium conditions to be met, the algorithm will not converge to the equilibrated solution. However, the incremental assignment is easy to program. Some researchers might use the result from this assignment to see a build-up in congestion during peak hours.

5.5.2 Method of Successive Averages

The Method of Successive Averages (MSA) was initially introduced by Robbins & Monro (1951). It is an iterative algorithm based on a predetermined move size along the descent direction (Sheffi & Powell, 1982), The method was developed to overcome the problem of allocating too much traffic to some links from the

incremental assignment. In the MSA algorithm, link flows are propagated as a linear combination of the flow on the previous iteration and auxiliary flow resulting from an AON assignment in the current iteration. The following steps can describe the algorithm of MSA:

Select an initial condition of link costs, typically the free flow speed condition is used. Initialise all flows $v_a = 0$, $n = 0$.

Find the shortest route x on each OD using the current travel time (or costs); make $n = n + 1$.

Load the whole of OD demand V using all or nothing assignment to the shortest route, obtaining a set of auxiliary flow F_a.

Calculate the current flow as:

$$v_a^n = (1 - \emptyset_n)v_a^{n-1} + \emptyset_n F_a$$

$$= v_a^{n-1} + \emptyset_n(F_a - v_a^{n-1})$$

$$with\, 0 \leq \emptyset_n \leq 1$$

Equation 5-9 MSA flow calculation

Update a new set of link travel time based on the flow v_a^n. If the flows (or link costs) have not changed significantly in two iterations, stop; otherwise process to stop 2.

According to the numerical results shown in Appendix E2, the standard value $\emptyset_n = 1/n$ proposed by Smock (1962) is used in the MSA assignment. At each

iteration, we update the new travel time and the assignment stop once route flows do not change significantly between iterations. After the 16th iteration, the travel time on each route is $t_1 = 15.10$, $t_2 = 16.78$ and $t_3 = 16.00$ minutes. Although the travel time on each route is not yet equal, but result is reaching an equilibrium on every iteration. It takes a number of iterations until the MSA algorithm approximates the right solution.

In a larger network, MSA requires more iterations until the equilibrium solution is founded. (Robbins & Monro, 1951) and (Blum, 1954) proved that if $\sum_n \emptyset_n = \infty$ and $\sum_n \emptyset_n{}^2 < \infty$, the the MSA algorithm is mostly converge. The conventional $\emptyset_n = 1/n$ is proved satisfying both conditions. The problem of the conventional $\emptyset_n$ is that at the very beginning iterations, $\emptyset_n$ could be too large. Hence, the equilibrium does not reach until a number of iterations. To avoid this drawback, some researchers have proposed an alternative sequence of value $\emptyset_n$. (Polyak, 1990) proposed $\emptyset_n = n^{-2/3}$, and (Zhang & Nagurney, 1996) proposed a sequence as:

$$\emptyset_n = \left\{1, \frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{n}, \frac{1}{n}, , \dots, \frac{1}{n}, \dots\right\}$$

Equation 5-10 Zhang's sequence

Both sequences satisfy conditions in (Robbins & Monro, 1951) and (Blum, 1954). (Bottom & Chabini, 2001) compared the Polyak's averaging method with the conventional $\emptyset_n = 1/n$ and found that Polyak's method outperforms MSA in their numerical experiments. It is clear that the MSA algorithm should not be stopped by the number of iterations but with reasonable criterion.

5.5.3 Frank-Wolfe algorithm

Frank-Wolfe algorithm (Frank & Wolfe, 1956) was initially suggested for solving quadratic programming problems. The algorithm is widespread, and it is one of the oldest methods for nonlinear constrained optimization. This includes the UE assignment problem. The advantage of the Frank-Wolfe algorithm is that it requires less memory and reaches convergence solutions within a few iterations. The algorithm tries to choose $\emptyset_n$ in a more systematic approach. In each iteration, the algorithm uses a linear approximation of the objective function in (Beckmann, McGuire & Winsten, 1956) and moves it towards a minimizer of the linear function. The Frank-Wolfe algorithm can be expressed as follow:

Select an initial condition of link costs, typically the free flow speed condition is used. Initialise all flows v_a^0=0, $n = 0$.

Calculates links travel time; Find the shortest route x on each OD using the current travel time (or costs).

Load the whole of OD demand V using all or nothing assignment to the shortest route; make $n = n + 1$.

Update the travel time on all routes t_a^1. Then perform the 2nd iteration with all or nothing assignment to the new shortest route.

Setting a linear function as follow:

$$u_a^n = v_a^{n-1} + \emptyset_{n,}(v_a^n - v_a^{n-1})$$

Equation 5-11 Franke-Wolfe linear function

Writing the objective function in terms of u_a^n:

Minimize Z $= \sum_a \int_0^{x_a} t_a(u_a^n) dx,$

Find $\emptyset_{n,}$ by taking $\frac{dz}{d\emptyset} = 0$

Calculates u_a^n with $\emptyset_{n,}$, then $v_a^{n+1} = u_a^n$.

If $|v_a^{n+1} - v_a^n| \leq \varepsilon \, \forall a$ then STOP, else set $n = n + 1$and go to 2nd step.

Equation 5-12 Franke-Wolfe algorithm

Next, we provide a numerical example with Frank-Wolfe aslgorithm shown in Appendix E3. As travel costs on each route are close to the equilibrium and route flow does not change significantly in the last iteration, we then stop the algorithm at the 4th iteration. In practice, the stopping criterion dictates the number of required iterations. For example, if the criteria If $|v_a^{n+1} - v_a^n| \leq \varepsilon$ is used, the number of the iteration is depended on the value ε.

Among the three heuristic approaches shown in this section, the Frank-Wolfe algorithm reaches the equilibrium solution faster than other algorithms. The frank-Wolfe algorithm is the most common algorithm to solve the UE assignment problem. The algorithm is favourable in large network problems because it can exploit the full advantage of the network's structure (Gartner, 1977). However, many researchers have not yet satisfied the convergence speed of the algorithm. Therefore, several researchers have proposed a modified version of the Frank Wolfe algorithm, such as (Fukushima, 1984), (Lee & Nie, 2001), (Gao et al., 2004) and (El Mouatasim & Ettahiri, 2019), which aim to increase the conveniently and efficiently.

5.6 Stochastic traffic assignment

Traffic assignment models in the previous sections are considered deterministic approaches. The approach assumes that the performance of the traffic network, including links' capacity, OD demands and travel times, are perfectly known by road users. In reality; however, such information is unknown and is subject to stochastic fluctuations. On the other hand, stochastic traffic assignment (SUE) models use the random utility in discrete choice theory to mitigate this problem. SUE assignment model assumes that the utility or costs that road users derive from their route have a stochastic part, which an observer cannot explain but under a known distribution. The road user's utility function consists of both deterministic and a stochastic part (Equation 5-13):

$$U_{ir} = V(-c_r) + \varepsilon_{ir}$$

Equation 5-13 Utility function with stochastic attribute

where U_{ir} is the utility that the user i derives from alternative (or route) r, which is consisted with a deterministic function V which is based on travel cost c_r and a stochastic term ε_{ir}. The deterministic function is independent of personal characteristic and the stochastic part captures the uncertainty that the road user faces. The stochastic part is also used to account for the unobserved attributes.

The utility function is assumed that each individual road user has their own perception of the route cost, and all users select the route that maximize its own utility. Based on the utility function the probability that user i selects route k is:

$$p_{i,k} = \Pr(U_{ir}(-c_r) \geq U_{ik}(-c_r), \forall k \neq r)$$

Equation 5-14 Probabilistic function

where $p_{i,k}$ is the probability that road user i selects route k over other routes. Several researchers proposed different techniques for solving the stochastic traffic assignment problem, but only two approaches have been widely accepted: (1) the Simulation-based approach and (2) the Proportional-based approach.

In the simulation-based approach, the probit model is used. The model assumes that each random variable ε_{ir} follows a multivariate normal distribution and does not have closed-form probabilities. Instead, numerical methods are required to determine the solution. The Monte-Carlo simulation is commonly used to solve the stochastic traffic assignment with the probit model. The major problem of using the Monte-Carlo simulation is that its final results depend on a series of random numbers, which can be mitigated by increasing the number of samplings.

In the proportional-based approach, the logit model is commonly used. The model assumes that each random variable ε_{ir} is Independently and Identically Distributed (IID), which follows the Gumbel distribution. The model has closed-dorm probabilities. The logit model has a well-known weakness: its inability to account for route overlapping or correlated routes properly. However, the approach is still commonly used in many modelling practices in transport. Due to its popularity, two common techniques, Dial's and Logit method, which are proportional-based approved, are reviewed in this thesis.

5.6.1 Dial's method

The method is based on a network loading algorithm that splits route flows at an origin node to all possible connected nodes. This is in the opposite of the all-or-nothing assignment method, which assigns all route flows to a single destination node. (Dial, 1971) proposed a double-pass algorithm which use a logit formulation to split trips from node i to node j among alternative routes r:

Forward pass (Calculates links and nodes weight):

Find the shortest path tree for the given network and process nodes in order of increasing minimum cost from the origin node to node n, $C_{o,n}$

Define the link weights w_{ij}that connected from node i to node j, and calculates node weights W_j at the exit node j using following equations.

$$w_{ij}\begin{cases} W_i e^{(-c_{ij}\theta)} & C_{o,i} < C_{o,j} \\ 0 & C_{o,i} < C_{o,j} \end{cases},$$

$$W_j = \sum_n w_{nj}$$

Equation 5-15 Dial's Forward pass

Compute for each link, starting from the origin node o with $W_o = 1$ and proceeding to other nodes in ascending order $C_{o,n}$.

Backward pass (Calculates link flows):

Initialize all links and nodes flow to zero: $v_i, v_{ij} = 0 \; \forall i \; \forall j$

Set the destination node as node i , and equal to the OD demand: $v_i = V_{od}$

For all links (n, i) that going to node i, set:

$$v_{ni} = v_i \frac{w_{ni}}{W_i}$$

Equation 5-16 Dial's Backward pass

Compute for each link and node from destination node in descending order $C_{o,n}$, until node i is the origin node.

If all the costs c_{ij} on all links across the network are not substantially different. The dial's algorithm will emulate minimum cost three from all links and routes. This makes the algorithm consume more computational cost. The algorithm also ignores the correlation and the similarity between routes. In practice, the algorithm assigns more traffic to a dense network area such as the city centre compared with spare areas such as bypass routes (Ortuzar & Willumsen, 2011).

5.6.2 Logit-based method

Logit-based stochastic assignment model assumed that different road users, who use the same link or route, perceive the travel costs differently. Due to its simplicity, the most widely used stochastic assignment model is based on the multinomial logit (MNL) formulation (Dial, 1971). Equation 5-17 describes the splitting of OD demand travelling from node i to node j among alternative routes r:

$$v_r = V_{ij} \frac{e^{-\theta t_r}}{\sum_{k \in R(i.j)} e^{-\theta t_k}}, \qquad \forall r \in R(i.j)$$

Equation 5-17 MNL function

where v_r equals to vehicle flow of route r within a set of routes $R(i.j)$ travelling from node i to node j , V_{ij} is the vehicle demand node i to node j and t_k is the travel time (or cost) of route, and θ>0 is the logit scaling parameter. If θ is high, it means that the differences between each user's perception of cost is small. In other words, users are less sensitive to the route cost, and the solution becomes sparser. The major drawback of this approach is its inability to account for overlapping among routes. Cascetta et al. (1996) proposed an approach for handling this overlapping problem. The researchers introduced a commonality factor. The functional form of the commonality factor can be expressed as Equation 5-18,

$$CF_r^{ij} = \beta ln \sum_{k \epsilon R(i.j)} \left(\frac{L_{rk}^{ij}}{\sqrt{L_r^{ij} \cdot L_k^{ij}}} \right)^{\gamma}, \qquad \forall r \in R(i.j)$$

Equation 5-18 Commonality function

where β and γ are the calibration parameters, L_{rk}^{ij} is the total length in common of routes r and k between OD pair $(i.j)$, L_r^{ij} and L_k^{ij} are the total length of route (sum of link lengths) r and route k respectively. The commonality logit model or C-Logit model proposed by the researchers can be written as:

$$v_r = V_{ij} \frac{e^{-\left(t_r - CF_r^{ij}\right)}}{\sum_{k \in R(i.j)} e^{-\left(t_k - CF_k^{ij}\right)}}, \qquad \forall r \in R(i.j)$$

Equation 5-19 C-Logit function

The researchers also denoted that the commonality factors can also be specified in other different ways. Noted that the factor is similar to the summation of correlation coefficient of perceived route costs with the specification of variances and covariances. There are also other researchers that proposed their modified logit model to tackle the overlapping problem. (Ben-Akiva, Moshe & Bierlaire, 1999) proposed the Path-Size Logit (PSL) model to alleviate the problem. The PSK model uses a logarithmic correction term determined by the length of links within a route and the relative lengths of paths that share a link.

5.7 Stochastic User Equilibrium

Combining a stochastic traffic assignment method described in the previous section with considering the congestion effect produces a stochastic user equilibrium (SUE) assignment model. In the SUE assignment, users are assumed to make errors in their route choice. They seek the minimum perceived travel time instead of the actual travel time in UE. Wadrop's first principle is still applied in the SUE with small modifications. At the equilibrium, road users cannot improve their perceived travel time by changing his/her route. (Fisk, 1980) proposed constrained mathematical programming for the SUE problem:

$$\text{Minimize } Z_{MNL} = \sum_{a \epsilon A} \int_0^{x_a = \sum_{ij \epsilon OD} \sum_{k \in R(i.j)} v_k^{ij} \delta_{ka}^{ij}} t_a(w) dw + \frac{1}{\theta} \sum_{ij \epsilon OD} \sum_{k \in R(i.j)} v_k^{ij} \ln v_k^{ij},$$

$$\text{Subject to } \sum_{k \in R(i.j)} v_k^{ij} = V_{ij}, \quad \forall ij \in OD,$$

$$v_k^{ij} \geq 0, \quad \forall k \in R(i.j), \forall ij \in OD$$

Equation 5-20 SUE optmisation problem

where A is the set of links, OD is the set of OD pairs, x_a is the flow on link a, $t_a(w)$ is the travel time on link a. (Fisk, 1980) demonstrated that solving the above mathematical programming comes up with the MNL function mentioned earlier. The advantage of the SUE model is that it has a closed-form logit probability expression that requires a less computational cost to achieve the equilibrium solution.

Solving an SUE assignment problem with a heuristic approach is similar to other UE problems. However, due to the more complicated objective function in the SUE assignment, Frank-Wolfe requires more computational resources, which makes the method less desirable. The most frequently used approach is the MSA algorithm, in which the predetermined step lengths $\emptyset_n$ are used. In the modified version of MSA for SUE assignment, the algorithm starts by solving a stochastic assignment and assigns the flow proportions to the selected routes using a logit model. This requires a route selection algorithm, whereas the MSA for UE does not. The most popular algorithm for route selection is a k-shortest path algorithm (Yen, 1971) which finds the least cost paths. However, it has the potential to select multiple routes that are overlapping. The next step is determining the optimal move size using a modified MSA function $v_a^{n-1} + \emptyset_n(v_a - v_a^{n-1})$. In this modified version, the value F_a from the all or nothing assignment is replaced with v_a from the logit model.

In order to demonstrate the performance of the SUE assignment with MSA algorithm, another numerical example from figure 5.2 is used. In this example, the classic MNL model (Dial, 1971) with $\theta = 0.5$ is used. Appendix E4 illustrates the calculation of this modified MSA algorithm:

Compared with the deterministic UE, the result from SUE requires more iterations until it reaches equilibrium. This is due to the step size of the MSA. In the case of the deterministic UE assignment, the step size $(F_a - v_a^{n-1})$ is always a solution from the all-or-nothing assignment, which is an extreme point into the solution space. Therefore, route flows reach the equilibrium solution as the iterations progress. In the case of the SUE assignment, the step size is $(v_a - v_a^{n-1})$ a solution from the logit model, which is relatively small and become less as n increases. It is common that SUE assignment gives a suboptimal solution and convergence is very slow compared with the deterministic UE. In addition, the travel time (or costs) on each route does not equal. This is because road users minimize their perceived route travel time not the actual one. Therefore, instead of travel time, differences route flows between each iteration are used as a stopping criterion.

5.8 Discussion

The traffic assignment is the last stage of the infamous 4-step model of travel demand modelling. Compared with other models reviewed in this literature, the traffic assignment model has the most significant number of studies. Similar to the route choice model in other modes of transport, the traffic assignments model can be separated into two categories: static and dynamic traffic assignments. However, the level of detail differentiates these in modelling. Static traffic assignment models aggregated demands in terms of vehicle units per hour. This makes the model less complex and requires less computational costs than dynamic traffic assignment models. However, the latter model considers the temporal effect of vehicle demands and networks. In this thesis,

we introduced a semi-dynamic traffic assignment model, which is a hybrid model between these two categories. The proposed model tries to capture the dynamic of demand and network characteristics while not impelling too much complexity. More details on our proposed model can be found in the next chapter. It is worth mentioning that the model is heavily based on the static assignment, which we have focused on in this chapter.

Traffic assignment models assign OD demands to a set of routes. The simplest method is the all-or-nothing assignment (AON), where all demands are assigned to that particular OD pair's shortest (or least cost) route. The method is eased to implement but lacks realistic behaviours. In reality, congestion might occur if all demands are assigned to the shortest route; hence, the travel time or cost might be higher than on other routes. In order to develop a traffic assignment model that considers the congestion, the model that describes the relationship between flow and congestion is required. This model type is called the volume-delay function in the transport research community. Several researchers and organizations have developed their functions, but the US Bureau of Public Road (BPR) function is the most widely used.

The two most used traffic assignment models considering the congestion effect are based on Wardrop's principles. The first principle states that at the equilibrium, all used routes between the same OD have the same travel cost, and users cannot further reduce their cost by changing to other routes. The second principle states that under social optimal conditions, traffic should be assigned to the network in such a way that the average (or total) travel cost is minimized. User equilibrium (UE) and Social Optimum (SO) assignment models are based

on the first and the second Wadrop's principle, respectively. Although, the mathematical programming function for these two has been founded (B Beckmann, McGuire & Winsten, 1956; Sheffi, 1985), it cannot be solved algebraically. Several heuristic approaches have been proposed to solve UE and SO problems. First, the incremental assignment algorithm divides OD demands into 40, 30, 20 and 10 per cent. The model increments a fraction of demand to the shortest route every iteration until all demands have been assigned. Second, the method of successive average (MSA) propagates the route flow as a linear combination of the flow on the previous iteration and auxiliary flow resulting from an AON assignment in the current iteration. Lastly, the Frank-Wolfe algorithm propagates the route flow in a more systematic approach. On each step, route flows are adjusted based on the gradient of the objective function. Based on a numerical example in this chapter, it is clear that the Frank-Wolfe algorithm reaches the UE solution faster than other algorithms.

Stochastic User Equilibrium (SUE) assumes that road users do not have full knowledge of traffic conditions and network characteristics. Hence, some random variations or errors occur while the user chooses their route. Users do not truly choose based on the actual cost but on their perception. SUE models are based on calculating the probability that users make their route choice. Based on how the random variation is assumed, SUE can be based on two approaches: probit and logit model. Probit models assume the random variable is under a multivariate normal distribution, which The Monte-Carlo simulation is used to solve the SUE problem. On the other hand, the logit model assumes that each variable is Independently and Identically Distributed (IID). The Multinomial logit function is commonly used. However, it suffers an overlapping route problem. (Cascetta et al., 1996) proposed an approach for handling this

overlapping problem by adding a commonality factor to the logit model. The mathematical programming function for SUE is founded and proved by (Fisk, 1980). The mathematical programming function is more complicated than the deterministic UE's function. Due to this complexity, the Frank-Wolfe algorithm cannot be used. A modified version of the MSA algorithm is commonly used to solve the SUE in a heuristic manner. However, it reaches the equilibrium solution slower than the deterministic UE.

The review of private transport assignment models established the principles required to model solo driving within the integrated framework. The adoption of the ubiquitous BPR function to model congestion effects, and the selection of the Stochastic User Equilibrium (SUE) principle realized via a logit model and the K-shortest path algorithm, are justified as crucial elements for capturing realistic user behaviour and achieving stable, converged flows. These elements provide the final technical foundation necessary to meet the first objective of the thesis: to develop a single, comprehensive framework that seamlessly integrates route choice models for all three primary transport modes (shared mobility, public transport, and private vehicles)

CHAPTER 6 METHODOLOGY

This chapter marks the shift from literature review to detailed framework development, fulfilling the primary purpose of constructing the integrated simulation framework. It begins by presenting a clear overview of the macroscopic, zonal simulation structure, which integrates the three primary modes (shared mobility, public transport, private transport) and incorporates dynamic variations across two scales: Within-Day (WDM) and Day-to-Day (D2DM). The chapter then details the sequential process, starting with the Mode Choice Model (MCM), adapted with crucial accessibility constraints (vehicle ownership and public transport access), before presenting the concrete mathematical implementation of all core Route Choice Models.

6.1 Overview

The simulation framework detailed in Chapter 6 is fundamentally a macroscopic, zonal model. It operates at an aggregated level, representing traffic demands as Origin-Destination (OD) flows between defined zones rather than tracking individual vehicles or travellers. This design, which includes a novel macroscopic matching algorithm based on linear programming, ensures computational efficiency for analysing large-scale networks and integrates both within-day and day-to-day dynamic traffic variations to capture temporal effects and user behavioural changes over time.

The classical transport model consists of four main steps: Trip generation, Trip distribution, Mode choice and Traffic assignment. Our simulation frameworks are located under the realm of the last two-step. The Mode Choice Model (MCM) is first employed to segment Origin-Destination (OD) demands into different modes of transport. Following this, Route Choice Models (RCMs) are used to determine specific traffic flows for each mode. In this thesis, we have developed two distinct simulation frameworks to incorporate shared mobility systems: ridesharing and taxi-sharing. Both frameworks consider three fundamental modes of transport: a shared mobility system, private transport, and public transport. The ridesharing framework assumes travellers own the sharing vehicle, which disappears from the network upon reaching its destination. In contrast, the taxi-sharing framework includes an additional Vacant Taxi Model to simulate taxi movements while searching for passengers, as taxi vehicles remain in the network.

The overall simulation framework is designed to be semi-dynamic, capturing both temporal variations within a single day and behavioural changes across multiple days. This is achieved through the integration of two crucial traffic variation models: the Within-Day Traffic Model (WDM) and the Day-to-Day Traffic Model (D2DM).

The within-day traffic model (WDM) is positioned between the current and subsequent time horizon's route choice models and the day-to-day traffic model. Its primary responsibilities include loading route flows from the various route choice models (e.g., matching algorithms for shared mobility, transit assignment for public transport, stochastic traffic assignment for private transport) onto the

road network. The WDM then calculates the actual travel time for each route and mode, accounting for prevailing network conditions (See section 6.4 for more detailed).

Specifically, this process involves several key steps

1. Converting Route Flows to Link Flows: Initial route flows f_{ijmr}^{td} are transformed into link-specific vehicle flows $x_k'^{t,d}$ by summing the demands for all routes and modes that traverse a particular link k.

2. Aggregating Total Link Flow: These newly calculated flows are combined with any existing traffic $\hat{x}_k^{t,d}$already present on the network from previous time horizons, yielding the total link flow $x_k^{t,d}$.

3. Calculating Actual Link Travel Time: The US Bureau of Public Roads (BPR) function is then applied to determine the actual travel time for each link $LTT_k^{t,d}$. This function models the non-linear relationship between traffic volume and delay, using the total link flow $x_k^{t,d}$, link capacity C_k, and free-flow travel time LTT_k^0, with standard parameters $\alpha = 0.15$ and $\beta = 4$.

4. Estimating Actual Route Travel Time: Finally, the actual travel cost for each route $RTT_r^{t,d}$ is determined by summing the actual link travel times along its constituent links

A crucial function of the WDM is to determine the duration each route flow lasts in the network, which is known as flow propagation. This is achieved using rolling horizon techniques. The model calculates the number of time horizons $n_r^{t^*,d}$ that a route flow will persist by dividing its actual route travel time $RTT_r^{t^*,d}$ by the defined length of each time horizon tLength. This mechanism enables the simulation to compute the existing link flow for the subsequent time horizon $\hat{x}_k^{t+1,d}$, thus dynamically influencing future network conditions. The level of detail in the simulation can be adjusted; a longer tLength makes the model resemble a static assignment, suitable for large-scale networks with lower computational costs, while a shorter tLength allows it to mimic a dynamic assignment, offering more detailed and realistic outcomes. The overall process of the WDM is visually represented in Figure 6.6

It is important to acknowledge a limitation of the current within-day traffic model: once the duration for a route flow is calculated based on instantaneous conditions at the start of a time horizon, this duration is fixed for future horizons. This means that if traffic conditions change significantly in subsequent horizons, the model does not dynamically readjust the actual travel times and flow durations from previous horizons. This absence of a continuous feedback mechanism for flow propagation can lead to potentially unrealistic actual route travel costs. While developing a model that could trace back and readjust travel times would enhance realism, it would also significantly increase the complexity of the simulation framework.

The second traffic variation model is the day-to-day traffic model (D2DM). This model is designed to capture traveller learning and adaptation by predicting changes in travel demand over multiple days. It collects the actual travel cost of each route and mode across every time horizon from the WDM, along with ridesharing costs and fees directly from the optimisation model, and a crowdedness factor for public transport. These actual costs are then transformed into perceived costs that travellers use for their decisions.

The D2DM operates by applying a weighted average learning operator, which updates travellers' perception of travel costs for each mode and OD pair, attributing greater importance to more recent experiences. Conceptually, the day-to-day traffic models function like a heuristic approach, similar to the Method of Successive Average (MSA), used for solving the user equilibrium problem. Based on these updated perceived costs, the perceived mode cost is then used to calculate the modal split for the following day, while the perceived route cost is used to determine route choice for private transport. The simulation framework is executed over multiple consecutive days (e.g., 15 days in our numerical examples) to capture this iterative learning process. The convergence result for the entire simulation is achieved when both the modal split and route flow become stable over successive days.

Figure 6.1 provides an overall illustration of this simulation framework, detailing the interplay between the Mode Choice Model (MCM), Route Choice Models (RCMs), Within-Day Traffic Model (WDM), and Day-to-Day Traffic Model (D2DM) across different time horizons and simulation days. A more

detailed description of each component in the simulation is provided in latter sections, along with mathematical representations

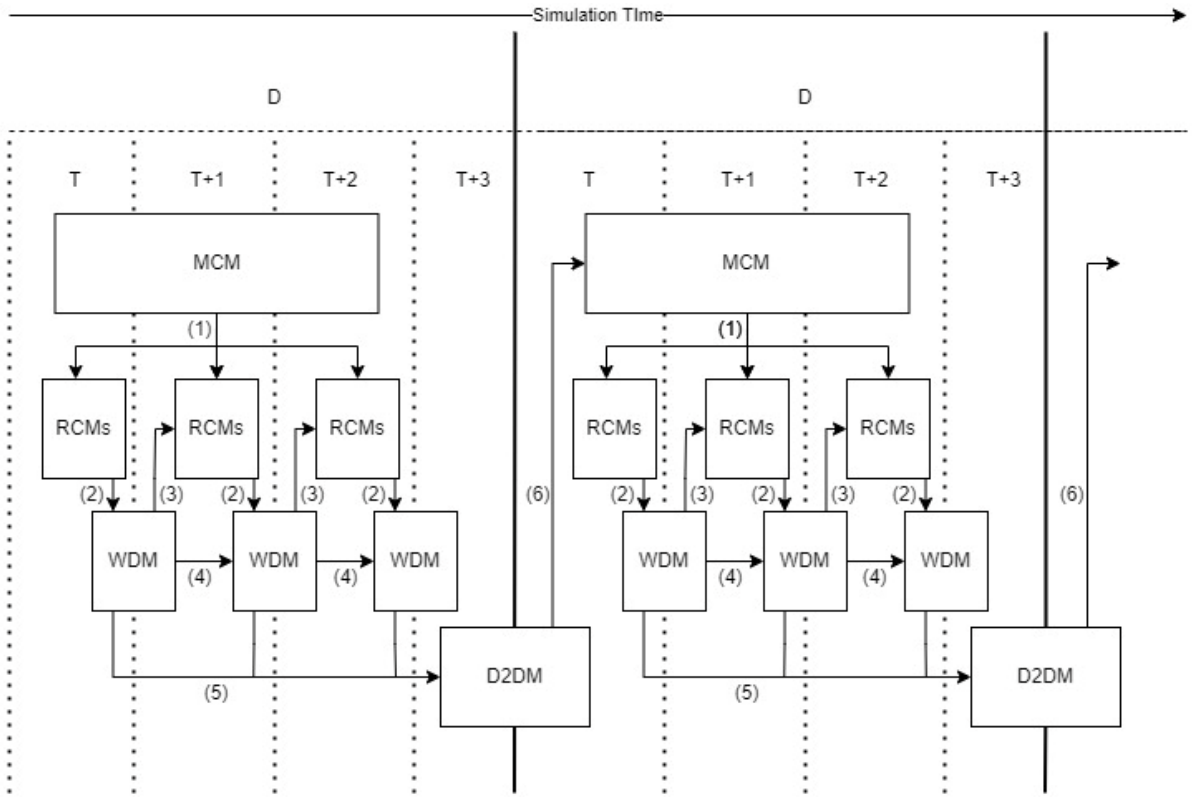

Figure 6.1 Overall simulation framework based on the simulation time.

6.2 Mode choice model

The mode choice model is the third step in the conventional four-step model. The mode choice model distributes an Origin-Destination trip matrix generated from a trip distribution model to the mode-specific trip table. This step is represented by a discrete choice model, which determines how travellers select one alternative from a set of mutually exclusive and collectively exhaustive options (Ben-Akiva, Moshe E. & Lerman, 1985). The model's result is the probability that a traveller will choose a particular mode of transport based on

some factors such as availability, vehicle ownership and, most importantly, the travel cost of each mode.

The origin of the discrete choice model began in the early studies of psychophysics in 1860. We found that the model has been implemented in transport contexts since 1972 when Daniel McFadden used a multinomial logit (MNL) model to forecast the numbers of users for the Bay Area Rapid Transit (BART) system (McFadden, 2001). MNL models are widely used in transport research. The MNL model remains one of the most widely applied approaches in transport analysis due to its simple functional form and computational efficiency (Train, 2009). The probability of individual n selecting mode i is expressed as:,

$$P_n(i) = \frac{exp(U_{in})}{\sum_{j=1}^{k} exp(U_{jn})},$$

Equation 6-1 MNL function

where U_{in} is the systematic utility of alternative i, and $k$is the number of available modes. Although widely adopted, the MNL model is limited by the independence of irrelevant alternatives (IIA) property, which can lead to unrealistic substitution among similar modes.

To address this, the Nested Logit (NL) model was introduced to account for correlation between groups of alternatives (McFadden, 1977). The probability of choosing mode iii within nest mmm is given by:

$$P_{in} = P_{mn}\, P_{i|m,n}, \qquad IV_{mn} = \ln\left(\sum_{j \epsilon m} \exp\left(\frac{V_{jn}}{\lambda_m}\right)\right)$$

Equation 6-2 Nested Logit function

where P_{mn} is the probability of choosing a nest m among n, IV_{mn} is the inclusive value (logsum) representing the expected maximum utility of nest m, and λ_m is the dissimilarity parameter. This structure provides more realistic substitution patterns and is especially relevant for public transport and shared modes.

Although this thesis does not employ the Nested Logit specification directly, its ability to capture correlation between alternatives has been influential in studies of shared mobility. Discrete choice models such as the Multinomial Logit (MNL), Nested Logit (NL), and Mixed Logit remain the standard for analysing traveller behaviour, with extensions that incorporate heterogeneity and service attributes such as waiting time, transfer time, and reliability (Ben-Akiva, Moshe E. & Lerman, 1985; Small, 1982; Train, 2009). These methods have been widely applied to shared mobility contexts, including ride-hailing, pooled trips, and on-demand services, where attributes like price, waiting, and matching reliability strongly affect adoption (Dias et al., 2017; Krueger, Rashidi & Rose, 2016).

Previous studies have recognised the importance of integrating mode choice with traffic assignment to capture the feedback between traveller decisions and network performance. Early work on combined models emphasised deterministic and stochastic frameworks that linked car and transit (Huang, Yang, & Bell, 2000). Other multimodal formulations have also been explored, particularly in the context of car–transit competition, using equilibrium approaches that balance modal split with route flows (Boyce, 1984; Florian & Spiess, 1983). These contributions highlight that integrating mode and route choice is essential to

obtain consistent solutions, as modal decisions are influenced by congestion and service attributes, which in turn depend on assignment results.

Building on these insights, and within this broader domain huang, ridesharing has received particular attention as both a complement and substitute to private and public transport. Studies have shown that adoption is shaped by factors such as fuel prices, congestion, socio-demographic characteristics, and attitudes toward technology and the environment (Alemi et al., 2019; Huang, Yang, & Bell, 2000). More recent work highlights its potential to support first- and last-mile connections to public transport (Kumar & Khani, 2021). though outcomes vary depending on integration and pricing policies (Thao, Imhof & von Arx, 2021) Among these, (Bahat & Bekhor, 2016) developed a joint mode choice–traffic assignment model that explicitly incorporated ridesharing, accounting for waiting time and fares as determinants of mode utility. Their framework considered two key determinants: waiting time (depending on driver availability) and the fare paid to drivers. They concluded that the number of ridesharing drivers was a critical parameter for a successful system and that ridesharing could reduce overall in-vehicle times across the network. However, their model implicitly assumed that all travellers had access to all modes, regardless of vehicle ownership or public transport accessibility. This assumption limits its behavioural realism and may lead to infeasible outcomes.

To address these weaknesses, the mode choice model in this thesis has been revised and extended by introducing two accessibility constraints: vehicle ownership and public transport accessibility. These constraints determine the feasible choice set for each traveller. Travellers who own a vehicle can choose

from a broader set of modes, including private car, ridesharing as driver, ridesharing as passenger, and public transport (if available). In contrast, travellers without a vehicle have a feasible set limited to ridesharing as passengers and public transport, provided their OD pair is accessible by the public transport network.

By combining these two conditions, four traveller categories are summarised, as shown in Tables 6.1 and 6.2. This classification ensures that mode choice reflects realistic availability conditions. A modelling issue arises for travellers with neither vehicle ownership nor public transport accessibility, as they are restricted solely to ridesharing as passengers. This results in unrealistic behaviour and non-convergent solutions in the simulation. In practice, it is uncommon for travellers who lack PT access also to lack vehicle ownership. To resolve this, the framework assumes that all travellers in OD pairs without PT access are vehicle owners. This assumption eliminates infeasible categories and guarantees convergence.

The traffic network is represented by a directed graph $G = \{\mathbb{N}, \mathcal{L}\}$ that includes a set of nodes $\mathbb{N}$ and a set of links $\mathcal{L}$, where $i, j \in \mathbb{N}$, $l \in \mathcal{L}$, $t \in T$ and $d \in D$ which are sets of origins i and destination j at time t and day d. $f_{ijm}^{t,d}$ is the number of trips traveling from node i to node j at time t and day d with mode m. Three general modes of transport are considered: shared mobility systems, public transport, and private transport. In the ridesharing scenario, four feasible modes are included: (A) ridesharing as driver, (B) ridesharing as passenger, (C) public transport, and (D) private car. In the taxi-sharing scenario, three modes are available: (B) ridesharing as passenger, (C) public transport, and (D) private

car. In this case, travellers cannot act as drivers because the vehicle fleet is managed externally.

To enhance realism and properly reflect vehicle fleet constraints, the model explicitly incorporates the two accessibility conditions described above. Vehicle ownership determines whether a traveller can drive or act as a ridesharing driver, while public transport accessibility depends on whether the OD pair is connected to the PT network. Together, these two dimensions define the four traveller categories summarised in Tables 6.1 and 6.2.

Table 6.1 Modes of transport that travellers can be accessed in ridesharing scenario.

Ridesharing system scenario		Origin and destination can be reached with public transport	
		Yes	No
Vehicle Ownership	Yes	A, B, C, D	A, B, D
	No	B, C	B*

Table 6.2 Modes of transport that travellers can be accessed in taxi-sharing scenario.

Taxi-sharing system scenario		Origin and destination can be reached with public transport	
		Yes	No
Vehicle Ownership	Yes	B, C, D	B, D
	No	B, C	B*

*Not modelled in this thesis

As shown in the table, a modelling issue arises for travellers who lack both vehicle ownership and public transport accessibility. Such travellers would be restricted solely to ridesharing as passengers, which produces unrealistic

behaviour and prevents the simulation framework from reaching convergence. Since it is uncommon in practice for individuals without public transport access also to lack a vehicle, this traveller type was excluded from the model. Instead, the framework assumes that all OD pairs without public transport access are associated with vehicle ownership.

After the definition of all modes of transport and its accessibility to each type of traveller have been clearly explained, the next stage is to allocate proportional of travellers to each mode. This allocation is governed by the utility functions within the multinomial logit (MNL) model. While the structure of the model follows the framework of (Bahat & Bekhor, 2016), the utility specifications have been adapted in this study to incorporate the accessibility constraints introduced earlier. The revised utility and MNL functions are expressed as follows:

$$\begin{pmatrix} f_{i,j,1}^{t,d} \\ f_{i,j,2}^{t,d} \\ f_{i,j,3}^{t,d} \\ f_{i,j,4}^{t,d} \end{pmatrix} = \begin{pmatrix} \exp\left(-\theta U_{i,j,1}^{t,d}\right) \\ \exp\left(-\theta U_{i,j,2}^{t,d}\right) \\ \exp\left(-\theta U_{i,j,3}^{t,d}\right) \\ \exp\left(-\theta U_{i,j,4}^{t,d}\right) \end{pmatrix} \bullet \frac{f_{i,j}^{t,d}}{\sum_m \exp\left(-\theta U_{i,j,m}^{t,d}\right)}$$

Equation 6-3 Revised MNL function

where $f_{i,j,m}^{t,d}$ is the number of trips travelling from origin node i to destination node j with mode m at time horizon t and day d. $m = 1$ for ridesharing as a driver, 2 for ridesharing (or taxi-sharing) as a passenger, 3 public transport and 4 for self-driving (private transport). θ is the scale of the Gumbel distribution. Finally, $U_{i,j,m}^{t,d}$ is a utility function of mode m consist of cost related variable that occur when travelling with the corresponding mode of transport. The utility function of each mode in this thesis can be written as follow (Equation 6-4).

Ridesharing as a driver: $U_{i,j,1}^{t,d} = \alpha + \beta_f D_{ijtd}^{D} + \beta_{VOT} TT_{ijtd}^{D} - f_{P(1)} - f_{P(2)} + f_{fee}$

Ridesharing/Taxi-sharing as a passenger: $U_{i,j,2}^{t,d} = \beta_{VOT}(WT_{ijtd}^{P} + TT_{ijtd}^{P}) + f_p + f_{fee}$

Public Transport: $U_{i,j,3}^{t,d} = \beta_{VOT}(WK_i + WT_b^B + TT_{ijtd}^{B}) + f_b + C_{ijtdb}$

Self-driving (Private Transport): $U_{i,j,4}^{t,d} == \alpha + \beta_f D_{ijtd}^{S} + \beta_{VOT} TT_{ijtd}^{s}$

Equation 6-4 Proposed mode cost functions

Self-driving travellers who travel from node i to node j at time horizon t on day d have the utility function $U_{i,j,4}^{t,d}$ generated from vehicle fixed cost $\propto$, fuel cost β_f multiplied by Actual travel distance D_{ijtd}^{S} and Value of Time (VOT), β_{VOT} multiplied by actual travel time TT_{ijtd}^{D}. Ridesharing drivers have the same utility function as self-driving travellers plus the ridesharing fee f_{fee}, which is charged by the shared mobility service provider. In most cases, fuel cost and travel time in this mode are higher than Solo driver, given that travellers have the same Origin-Destination, i.e., $TT_{ijtd}^{D} \geq TT_{ijtd}^{s}$. This due to a detour imposed from pick up and drop off passengers at their origins and destinations. However, the Ridesharing driver have a reduction in travel cost from passenger's fares f_p. Different service providers have their own fare structure. Note that this fare can be either fixed or varied by factors such as trip length. In some cases, the fare is varied by passenger's demand compared with available drivers. This is referred as surge pricing, which will be discussed in the next chapter. The utility function for Ridesharing or Taxi-sharing passengers does not contain vehicle cost, but passengers have an extra waiting time WT_{ijtd}^{P} for the designated driver to pick

up plus fare f_p. In the case of public transport users, travellers have extra walking time WK_i from node i to the bus stop and bus waiting time WT_b^B. In this framework, we assumed that the traffic does not affect bus services and the bus frequency is uniformly distributed. Thus, $WT_b^B = BF_b^{-1}$, where BF_b^{-1} is a public transport service frequency of service line b. f_b is a public transport fare charged in a constant rate. Finally, $C_{i\,j\,t\,d\,b}$ is a cost from public transport crowdedness which increased exponentially when the service is over-capacity. Figure 6.2 illustrates the factors affecting the mode choice model.

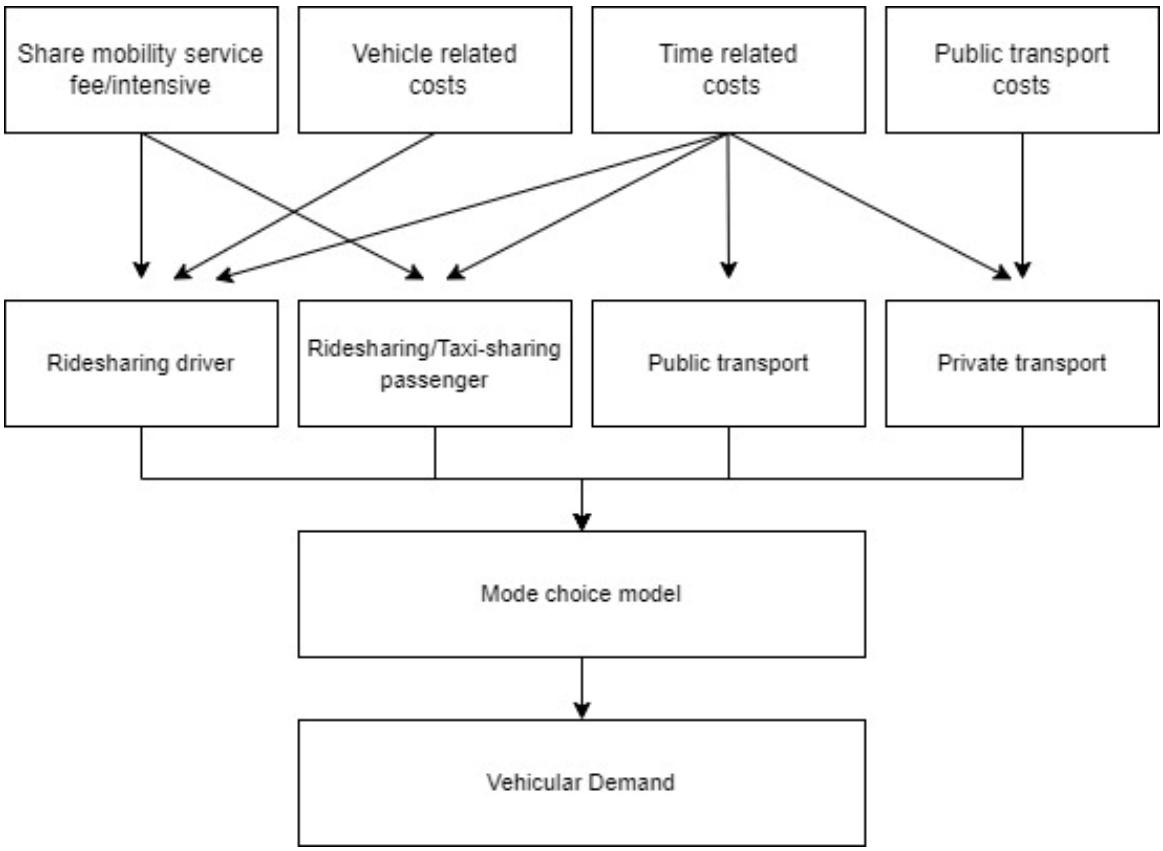

Figure 6.2 Factors affecting the mode choice model.

After vehicular demands on each mode of transport are determined, the ridesharing matching model is used to match numbers of passengers and drivers given the objective function and constrains, which will be discussed in the next

section. Note that it is not always that case that all drivers and passengers are matched. This matching inability is due to some constrains such as a lack of users in the area or the ridesharing trip is not cost efficient. In such case both users will change to other available mode of transport. If there are more than one mode of transport that is left available, unmatched users will be split to every possible mode equally. Note that such users will be an extra unmatched cost $f_{unmateched}$, which makes the mode less desirable for them in their future decision. The next section will describe route choice models on each mode of transport applied in this thesis.

6.3 Route choice models

The route choice models (RCMs) for each transport mode, previously introduced in Chapters 3, 4, and 5, are integrated here to determine demand flows. The ridesharing system uses a matching algorithm, the taxi-sharing system uses the vacant taxi model, public transport employs the transit assignment model, and private vehicles utilize the traffic assignment model. A significant challenge of this thesis is modelling the seamless interaction between each of these modes and assessing their collective effect on the road network. This process is addressed in this section: once all routes and their flows are determined, the within-day traffic model (WDM) loads this vehicular demand onto the network, propagates the flow, and calculates updated road network conditions (links travel time). These updated conditions are subsequently used by the RCMs in future iterations. Figure 6.3 demonstrates this detailed model framework.

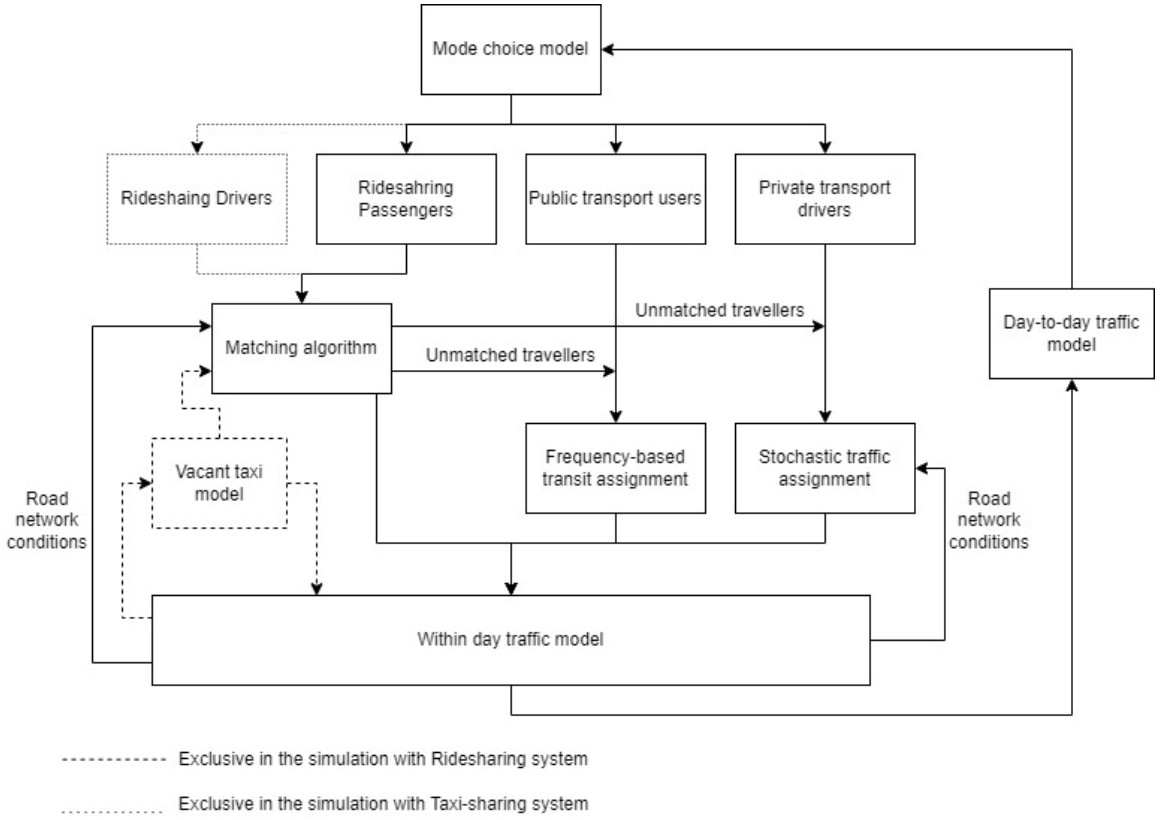

Figure 6.3 Detailed framework of the simulation

6.3.1 Matching algorithms

The matching algorithm is a route choice model for ridesharing and taxi-sharing system. In this thesis, linear programming which is an efficient optimization model is used to match ridesharing passengers and drivers, or in taxi-sharing case, vacant taxis. The input of the matching algorithm are traveller's demands $f_{i,j,m}^{t,d}$ for $m = 1,2$ resulted from the mode choice model explained earlier, and road network conditions which represented in link's travel time $\widehat{tt}_l^{t,d}$. In taxi sharing case, $\dot{f}_l^{t,d}$ which is a number of vacant taxis located at node i on time horizon t and day d. Although there are only few inputs are required, there are several steps in the algorithm until the output is generated. The first step of the matching

algorithm starts with generated all matching possibilities which can be expressed as Equation 6-5:

$$sd^{t,d} = \begin{Bmatrix} f_{1,1,1}^{t,d} & \cdots & f_{i,1,1}^{t,d} \\ \cdots & \cdots & \cdots \\ f_{1,j,1}^{t,d} & \cdots & f_{i,j,1}^{t,d} \end{Bmatrix}, sp1^{t,d} = \begin{Bmatrix} f_{1,1,2}^{t,d} & \cdots & f_{i,1,2}^{t,d} \\ \cdots & \cdots & \cdots \\ f_{1,j,2}^{t,d} & \cdots & f_{i,j,2}^{t,d} \end{Bmatrix}, sp2^{t,d} = \begin{Bmatrix} f_{1,1,2}^{t,d} & \cdots & f_{i,1,2}^{t,d} \\ \cdots & \cdots & \cdots \\ f_{1,j,2}^{t,d} & \cdots & f_{i,j,2}^{t,d} \end{Bmatrix}$$

Equation 6-5 Matching possibility matrices

Where $sd^{t,d}, sp1^{t,d}, sp2^{t,d}$ are matrices of the number ridesharing driver, first passenger and second passenger, respectively. Each matrix has a size of (i, j). The total number of matching possibilities are $sd^{t,d}$x $sp1^{t,d}$x , $sp2^{t,d}$x p where p is the number of matching patterns. For each matching possibility, calculates the travel cost that each participant bare. The calculation is based on the equations shown in Chapter 3, Figure 3.2. Next, the cost for each participant in ridesharing system is then compared with their non-ridesharing alternatives; if the latter prevail, the matching becomes unattractive, and the drivers and passengers may resort to other modes. To evaluate these possibilities, a Linear Programming (LP) optimization is proposed. The standard form of linear programming is given as:

$$\begin{aligned} \textbf{given} \quad & x_{dp} \in R \\ \textbf{minimize} \quad & f(x_1, x_2, \ldots x_n) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\ \textbf{subject to} \quad & a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \leq b_1 \\ & a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \leq b_2 \\ & \cdots \\ & a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \leq b_m \\ & x_n \geq 0 \quad \forall\, n \in N \end{aligned}$$

Equation 6-6 Linear Programming

Based on this standard form (Equation 6-6), the objective function and constrain functions are all linear equations, and all decision variable $x_1, x_2, \dots x_n$ are non-negative real numbers with $c_1, c_2, \dots, c_n$ as objective function coefficients, $a_{11}, a_{12}, \dots, a_{mn}$ are constraints' coefficients and $b_1, b_2, \dots, b_m$ are constant numbers which known as the right-hand sight of equations.

To the best of our knowledge, we have not found a single study that used linear programming as a matching algorithm in shared mobility systems. All existing matching algorithms modelled the ridesharing system at a microscopic level that treated users individually. Hence, decision variables are limited to binary (0 for unmatched and 1 for matched) or integer numbers (number of matched seats). Instead, we proposed a matching algorithm based on linear programming with decision variables representing a number of users' demands that are matched. The decision variable does not necessarily need to be an integer value. Full detail of our proposed algorithm will be provided in the later chapter of this thesis. The main benefit of linear programming is that the approach can be solved in polynomial time, which is classified as a P class in the computational complexity theory (Hofman, 2006). This P class complexity means linear programming is less complex and faster to be solved than NP-hard optimization, such as integer programming. Therefore, the linear programming embedded in our framework allows us to simulate shared mobility systems on a much larger scale.

The proposed LP is inspired from the binary integer programming approach in (Nourinejad & Roorda, 2016), which is on the microscopic level while ours deals with the macroscopic quantities. In their framework, each driver and passenger is represented individually, and the matching process assigns specific

passengers to specific drivers. While this approach provides a detailed and behaviourally rich representation of ridesharing, it is computationally intensive and becomes impractical for large-scale applications.

In this thesis, their approach is adapted to the macroscopic level. Instead of modelling each traveller individually, trips are represented as aggregated flows in the form of OD matrices. Driver trips $sd^{t,d}$ and passenger trips $sp1^{t,d}, sp2^{t,d}$ are expressed as flows between zones, and the matching process determines the number of trips that can be paired rather than the identities of specific individuals. The decision variable $X_{ijk_1l_1k_2\ l_2p}$ therefore denotes the number of matched trips between a driver OD pair and passenger OD pairs under a given matching pattern.

The optimisation objective remains consistent with Nourinejad and Roorda's work—maximising vehicle-kilometres travelled savings (VKTS)—but it is reformulated as a linear program suitable for macroscopic analysis. This aggregation allows the matching problem to be integrated with the traffic assignment model, enabling the simulation of shared mobility systems at the network scale while keeping computational costs manageable.

$$\text{Objective function} \quad \max \sum_{i,j} \sum_{k,l} \sum_{p} X_{ijk_1l_1k_2\ l_2p} VKTS_{ijk_1l_1k_2\ l_2p}$$

$$\text{Subject to} \quad \sum_{(i,j)\in D} X_{ijk_1l_1k_2\ l_2p} \leq F_{(k,l)}\ \forall (k,l)\ \in P$$

$$\sum_{(k,l)\in P} X_{ijk_1l_1k_2\ l_2p} \leq F_{(i,j)}\ \forall (i,j)\ \in D$$

$$X_{ijk_1l_1k_2\ l_2p}\ \geq 0\ \forall (i,j) \in D\ ,\ \forall (k,l) \in P$$

Equation 6-7 Proposed optimisation problem for Ridesharing

From above equations VKTS is a difference between the total distance in non-ridesharing modes for all users and the newly assigned ridesharing trips. The first two constraints ensure that both number of matched ridesharing drivers and ridesharing passengers do not excess their origin OD demand $F_{(i,j)}$, $F_{(k,l)}$ for drivers travelling from node i to node j and passengers travelling from node k to node l. P and D are sets of driver and passengers ODs, respectively. Finally, Equation 10 is a non-negativity constraint. The output of this algorithm is the decision variable $X_{ijk_1l_1k_2\ l_2p}$ that determine the number of matched trips the route. In our model, the matched ridesharing route always using the shortest route between nodes based on an instantaneous link's travel time, using Dijkstra's algorithm. As a result of this, the route flow for ridesharing users $f_{i,j,m}^{t,d}$ for $m = 1,2$ can be calculated. For taxi sharing system, the driver's cost constraint is changed (See section 3.4) and the parameter $sd^{t,d}$ are replaced by $\dot{f}^{t,d}$ and the LP can be written as followed:

$$\text{Objective function} \quad \max \sum_{i,j}\sum_{k,l}\sum_{p} X_{ik_1l_1k_2\ l_2p} VKTS_{ik_1l_1k_2\ l_2p}$$

$$\text{Subject to} \quad \sum_{(i,j)\in D} X_{ik_1l_1k_2\ l_2p} \le F_{(k,l)}\ \forall (k,l)\ \in P$$

$$\sum_{(k,l)\in P} X_{ik_1l_1k_2\ l_2p} \le \dot{f}_i\ \ \forall i\ \in T$$

$$X_{ik_1l_1k_2\ l_2p}\ \ge 0\ \forall i\ \in T\ ,\ \forall (k,l) \in P$$

Equation 6-8 Proposed optimisation problem for Taxi-sharing

Where, $\dot{f}_i$ is the number of taxis located at node i and T is a set of taxi's location. Thus, the ridesharing and taxi-sharing route and its demand are determined. For travellers who are unmatched, they are assumed to change to an alternative mode. This assumption is to ensure that all travellers are loaded in the road network. We assume that all unmatched drivers are switched to solo drivers while unmatched

passengers are switched to private transport or public, based on their vehicle possession and access to the public transport.

With the linear programming in our matching algorithm, the problem is still noticeable in a large-scale network. To mitigate this problem, we proposed a clustering method which is a heuristic approach that can reduce the computational time substantially while yielding a satisfactory outcome. This heuristic approach will be further discussed in the later chapter.

In summary, shifting from a microscopic to a macroscopic formulation introduces potential limitations. The model no longer captures individual-level behaviours, preferences, or stochastic variations, and aggregation may mask mismatches that would occur in reality. As a result, the model may overstate matching efficiency, since fractional flows can be matched more smoothly than individual trips. These trade-offs are acknowledged; however, the framework incorporates a rolling-horizon design that allows the temporal resolution to be adjusted. With small time horizons, the algorithm operates in a more microscopic manner, closely resembling agent-level matching, while larger horizons aggregate flows to produce macroscopic outcomes. When sufficient data are available, smaller horizons can be applied to increase behavioural detail, whereas in data-scarce contexts, larger horizons ensure tractability. This flexibility mitigates some of the drawbacks of pure aggregation and justifies the approach, as it enables the ridesharing matching model to be applied consistently across scales while remaining computationally efficient for large-scale simulation.

6.3.2 Vacant taxi model

The status of a taxi can be divided into three stages: matching, en route, and vacant. The matching stage is when the taxi is paired with a customer in the system and travels to their location. Once the first customer has been picked up, the taxi enters the en route stage and follows the designated route until all passengers have been served. The matching algorithm regulates the route choice for these two stages. However, once all passengers have been dropped off and before the next match has been made, the taxi enters the vacant mode, during which it navigates the network and searches for the next customer. This is where the vacant taxi model is used. A vacant taxi model is a mathematical tool designed to predict the likelihood of a vacant taxi movement in the road network. In this thesis, we adopted the vacant taxi model based on MNL model explained in section 3.5.1, where passenger demand, travel distance, and rate of return are considered.

Wong, Szeto & Wong. (2015b) proposed a two-stage multinomial logit (MNL) model to represent the movement of vacant taxis. In their formulation, the first stage predicts whether a taxi remains in its current zone or relocates to an adjacent zone, while the second stage estimates intra-zonal circulation distance and time before the next trip is assigned. Their model was calibrated using detailed GPS data from Hong Kong and provides a fine-grained representation of vacant taxi behaviour.

In this thesis, the model is adapted rather than applied in its original form. The intra-zonal component is deliberately excluded because the present framework is macroscopic and based on inter-zonal OD flows. At this level of aggregation, intra-zonal circulation is secondary to inter-zonal relocation, and including it would increase model complexity without substantially improving behavioral realism. While more detailed modelling of intra-zonal behavior could provide additional insights, it is beyond the scope of a macroscopic framework whose focus is large-scale system interactions.

The adaptation preserves Wong et al.'s core multinomial logit structure while modifying its operationalisation to suit the shared mobility context (Equation 6-9). Demand and travel times are made time- and day-dependent, passenger demand is estimated from previous-day observations, and the rate of return parameter is reformulated as the ratio of driver cost to travel time. The coefficients themselves are not recalibrated in this thesis; however, the original values from Wong, Szeto & Wong. (2015b) provide a suitable baseline, for future studies should calibration to local data be required. Accordingly, $E_j \rightarrow E_j^{t,d}$, $D_{ij}^{t} \rightarrow D_{ij}^{t,d}$ and $R_{ij} \rightarrow R_{ij}^{t,d}$. The passenger demand can be calculated as a summation of ridesharing passengers at the same time t in the previous day or $d-1$, i.e. $E_j^{t,d}=\sum_k f_{j,k,4}^{t,d-1}$. The travel distance between node i and j is the instantaneous travel time of the shortest route between the two nodes. The rate of return parameter or $R_{ij}^{t,d}$ is replaced by the cost constraint function of the taxi driver (see section 3.5.1) divided by travel time between two nodes $TT_{i\,j\,t\,d}^{D}$, i.e.

$$R_{ij}^{t,d} = \frac{C_{(N_t,O_{p1})} + C_{(O_{p1},D_{p1})} + C_{(D_{p1},O_{p2})} + C_{(O_{p2},D_{p2})} - F_{p1,p2,m}^{v}}{TT_{i\,j\,t\,d}^{D}}$$

Equation 6-9 Proposed Rate of Return function

This thesis does not adopt the intra-zonal travel distance and travel time formulation illustrated in Wong, Szeto & Wong. (2015b). Instead, intra-zonal costs and travel times are neglected within the present framework. Nevertheless, the conceptual basis of Wong, Szeto & Wong. (2015b) is retained, although parameter definitions and coefficients differ and therefore require future recalibration. Three vacant taxi models are proposed to represent distinct repositioning strategies: ATT Revenue, CLOSE Revenue, and CLOSE Pax. Each model reflects different assumptions regarding driver or system behavior.

ATT Revenue Model (System-Optimized Approach) This model represents a centrally coordinated repositioning strategy, mimicking a scenario where a platform operator uses predictive analytics to guide vacant taxis toward optimal locations. It is the most sophisticated of the three models in this thesis.

From the limited choice set of adjacent nodes $j \in H_i \cup \{i\}$, the model identifies the top three most attractive alternatives. Given the three parameters, the zone choice model for vacant taxi vehicles is formulated as a following logit model:

$$P_q(j|i) = \frac{\exp\left(\beta^E E_j^{t,d} + \beta^t TT_{i\,j\,t\,d}^D + \beta^R R_{ij}^{t,d}\right)}{\sum_{n \epsilon H_i \cup \{i\}} \left(\beta^E E_n^{t,d} + \beta^t TT_{i\,n\,t\,d}^D + \beta^R R_{in}^{t,d}\right)} \qquad j \in H_i \cup \{i\}$$

Equation 6-10 ATT Revenue probabilistic function

where $P_q(j|i)$ is the probability of a vacant taxi driver navigate the vehicle from node i to node j. H_i denotes the set of zones that adjacent to zone i β^E, β^t, β^R are coefficients associated with the relative passenger demand, the intra-zonal circulation distance, the cross-zonal travel distance, and the rate of return respectively. I may also worth mentioning that the parameters might be different

when comparing it with Wong, Szeto & Wong,. (2015b). Therefore, it is necessary to recalibrate the coefficients.

CLOSE Revenue Model (Driver Heuristic: Profit-Seeking) This model simulates the behavior of an experienced driver who relies on practical knowledge to maximize income. It represents a profit-seeking heuristic, where the primary goal is to secure the most lucrative fare based on past experience. The model operates deterministically. For a vacant taxi at node i, it evaluates all adjacent nodes $j \in H_i \cup \{i\}$ and identifies the top three with the highest expected fare revenue, $F^{v}_{p1,p2,m}$ estimated using data from the previous day $d-1$. The vacant taxis at node i are then distributed evenly among these three high-profit destinations. This strategy mimics a driver using historical knowledge and thinking, *"Where can I go nearby to get the best fare, based on how yesterday was?"*

CLOSE Pax Model (Driver Heuristic: Demand-Seeking) This model represents an alternative driver heuristic focused on minimizing idle time. It simulates the strategy of a driver who prioritizes finding any passenger quickly based on their memory of high-activity zones. Similar to the CLOSE Revenue model, this approach is deterministic and relies on historical data. It evaluates the adjacent nodes $j \in H_i \cup \{i\}$ and identifies the top three with the highest total originating passenger demand, $E_j^{t,d}$ as observed during the same time horizon on the previous day $d-1$. Vacant taxis at node i are subsequently split evenly among these three high-demand locations. This strategy mimics a driver thinking, *"Where were the most people waiting for a ride at this time yesterday?"*

For all three models, once a vacant taxi is dispatched to a new destination, it is considered unavailable for matching until it reaches that node. By comparing the network-level impacts of these three distinct strategies, the framework can provide valuable insights into the trade-offs between system optimization and driver-led decision-making based on historical knowledge.

6.3.3 K-shortest path algorithm

K-shortest path algorithm are used in the route generating stage on both transit assignment and traffic assignment in this thesis. In large networks, the number of possible paths between two nodes can be very large, making it difficult to find the best path for a given traffic demand. The K shortest path algorithm addresses this problem by finding not just the shortest path, but the K shortest paths between two nodes which can be set by the user. The K shortest path algorithm is a fundamental graph algorithm used to find the K shortest paths between origin and destination node in a weighted directed graph. There are several variations of the K shortest path algorithm, each with its own strengths and weaknesses. In this section, we will discuss two popular variations of the K shortest path algorithm: Yen's algorithm and the K star algorithm.

Yen's algorithm is one of the most well-known and widely used variations of the K shortest path algorithm. This algorithm was first introduced by Yen in 1971 and has since been widely studied and used in various applications, such as network routing, traffic engineering, and logistics planning. Yen's algorithm works by iteratively modifying the shortest path and adding non-shortest paths to the set of candidates. Specifically, it starts by computing the shortest path using

a standard shortest path algorithm, such as Dijkstra's algorithm. Then, it generates a set of candidate paths by iteratively modifying the shortest path and adding non-shortest paths to the set of candidates. This process is repeated until K shortest paths are found. The K shortest paths are then found by selecting the K paths with the lowest total weight from the set of candidates.

The K star algorithm is another variation of the K shortest path algorithm that was introduced by Aljazzar & Leue (2011). This algorithm works by using a heuristic search to explore the space of potential candidate paths. Specifically, it starts by computing the shortest path using a standard shortest path algorithm. Then, it generates a set of candidate paths by iteratively modifying the shortest path and adding non-shortest paths to the set of candidates. However, instead of exploring all possible candidate paths, the K star algorithm uses a heuristic search to explore a subset of the candidate paths. This heuristic search is based on an estimate of the distance from the current path to the Kth shortest path. By using this heuristic search, the K star algorithm is often able to find K shortest paths more quickly than Yen's algorithm.

While both Yen’s algorithm and the K star algorithm are effective approaches to the K-shortest path problem, a key distinction lies in their computational demands. Yen’s algorithm systematically explores all candidate paths, ensuring that the true KKK-shortest loopless paths are identified, whereas K star uses a heuristic to explore only a subset. This makes Yen’s algorithm more computationally expensive, particularly in large networks, and the resulting alternatives may overlap heavily, offering limited behavioural diversity. In the context of this thesis, these limitations imply that the route choice set may be

simplified and that scalability could become an issue in very large networks. However, the trade-off is acceptable because Yen's algorithm provides robustness and consistency in path generation, which are essential for the frequency-based transit assignment and the stochastic traffic assignment models embedded here.

6.3.4 Transit assignment model

In this thesis, a frequency-based transit assignment model is used to determine transit route choice for travellers who choose public transport as a mean of travel i.e., $f_{i,j,3}^{t,d}$. In order to incorporate with the transit assignment model in the simulation framework, first the network representation of the transit network needs to be modified. The route section approach proposed by (Chriqui & Robillard, 1975) has been chosen, due to its ability to minimise the complexity of a large transit network. Figures 6.4 and 6.5 example the original and modified transit network representation, respectively. In the current framework, first- and last-mile connections are simplified by assigning travellers to their nearest transit node via centroid connectors. However, the framework is flexible: smaller nodes could be introduced to explicitly represent pickup and drop-off points at the beginning and end of trips. These nodes would allow walking connections, feeder services, or shared micromobility options to be represented more realistically. While not implemented in this study to maintain tractability, future work could extend the framework in this direction. In particular, as outlined in Section 9.3.4, shared mobility can serve as a complementary mode to bridge the gap between residential areas or final destinations and transit hubs, thereby strengthening the treatment of first- and last-mile connectivity.

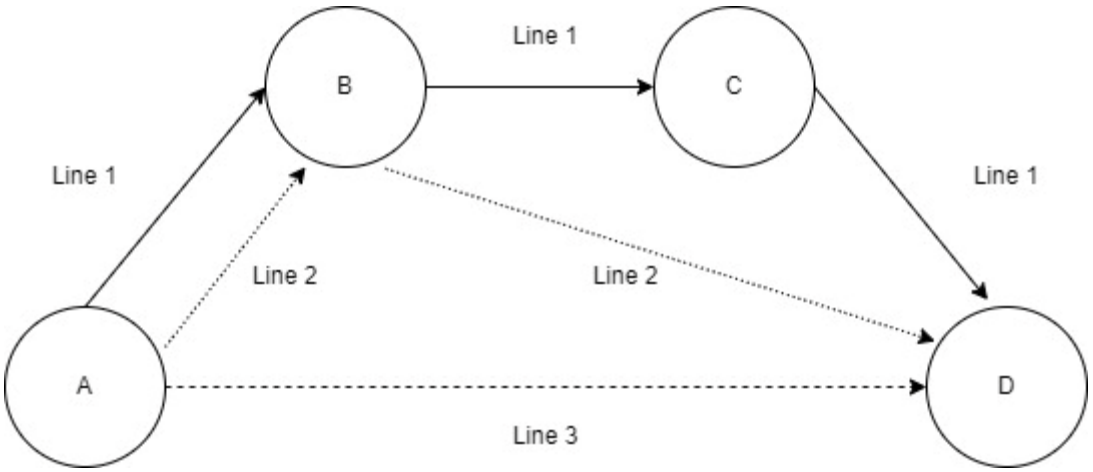

Figure 6.4 Example of transit network (Original)

Figure 6.4 describes a transit network representation with 4 transit nodes and 3 service lines. Line 1 is a regular service which stop at every node between transit node A and D. Line 2 is a semi-rapid service which stop fewer node and Line 3 is an express service which is a direct service from node A to node D without any stop. Based on the original transit network representation, it is difficult to calculates the travel cost that transit user perceive. This is due to a common line problem mentioned in Chapter 4. We then modify the transit network with route section approach by adding more imaginary nodes. The concept behind this is to increases these imaginary nodes until it has only one service line connected. Figure 6.5 demonstrates the transit network after the modification.

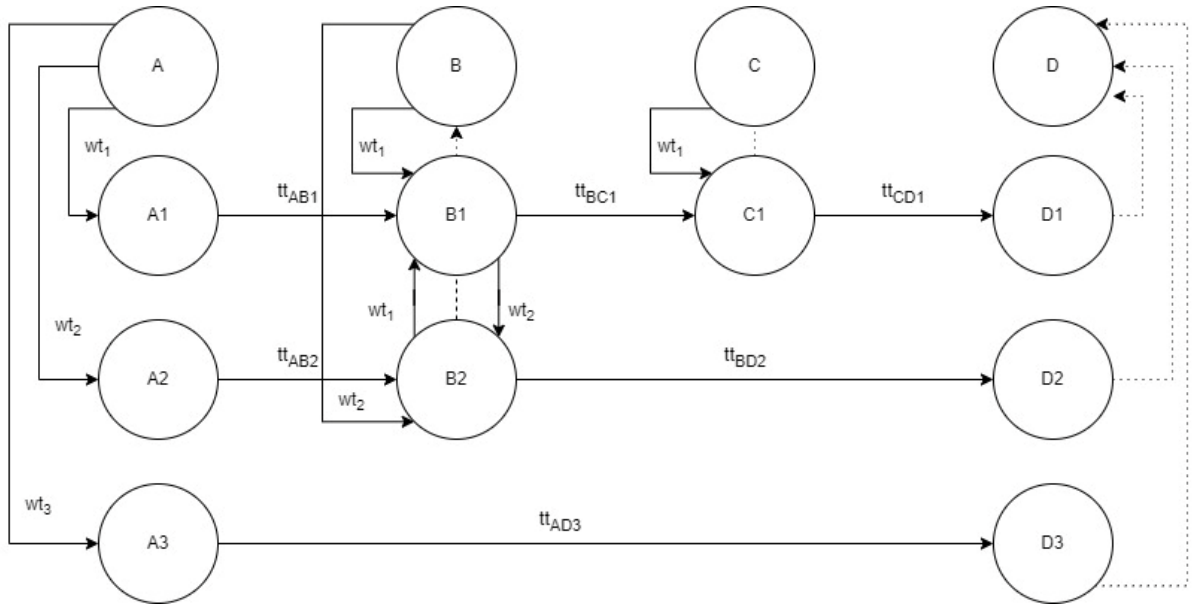

Figure 6.5 Example of transit network (Modified)

Compared to the original version of the transit network, the modified network has more nodes and links. Although the modified network is visually complex, it requires less computational cost compared with other network modification approach. Hence, the route section approach is the most popular approach in transit assignment modelling. In this figure, node A connected with three imaginary nodes; A1, A2, A3 which can be seen as a boarding/alighting node to service line 1, 2 and 3, respectively. The link cost has from A to A1, A2 and 3 are waiting cost wt_1, wt_2, wt_3 , where $wt_l = 0.5f_l^{-1}$, these costs represent the waiting time that passengers needs to wait until the service arrives the station which is a half of a service's headway. The other type of link costs is the travel time of the service between each transit node. If the transit network is used for system such as undergrounds or trains, then these travel times are fixed. However, for a bus system that travelling on the same road network as other modes, travel time is depended on the link travel of the vehicle network. In a

scenario where High Occupancy Vehicle Lane is applied to the network, the bus travel time can be calculated separately to the vehicle traffic.

After the transit network has been modified, the next step is generating the hyper-path which is a set of paths that the transit user is likely to choose. Before creating the hyper-path, first we need to find a set of shortest paths between each OD, this is called K-shortest path. An algorithm that determines the set of paths are required in both frequency-based transit assignment model and the stochastic traffic assignment model. There are several algorithms that are used to determine K-shortest paths. In this thesis, Yen's algorithm has been chosen due to its popularity and ease of implementation. Further details on the algorithm can be found in the next section (Yen, 1972). Once K-shortest paths are determined, we implemented (Chriqui & Robillard, 1975) heuristic algorithm solving hyper-path. The algorithm started from registering the shortest route with the lowest travel time or cost into the hyper-path. Then adding the route in ascending order from K-shortest paths until expected travel time is not reduced. The expected travel time is calculated with Equation 6-11.

$$T_s = E(W_s) + \sum_{r \epsilon S} rtt_r H_r(S)$$

Equation 6-11 Proposed expected travel time

Where T_s the expected travel time of hyperpath S, $E(W_s)$ is the expected waiting time of the hyperpath, rtt_r is a route travel time of route $r \in S$ and $H_r(S)$ is a probability that route r arrive first based on the service frequency. Next, we provide a numerical example for this algorithm based on Figure 6.5. Following table 6.3 provide the transit data used as the numerical example (Ex. 6.3 - 1:5):

Table 6.3 The transit data used as the numerical example.

Line	Service frequency (veh/hr)	Link travel time tt_{ij}			Total travel time
1	15	AB=3 mins	BC=5 mins	BC= 3mins	11 mins
2	10	AB=3 mins	BD=6 mins		9 mins
3	4	AD=7 mins			7 mins

For travellers from A to D, there are total five possible transit route which are:

$r_1 = A \rightarrow A1 \rightarrow \mathrm{B1} \rightarrow \mathrm{C1} \rightarrow \mathrm{D1} \rightarrow \mathrm{D}$

$r_2 = A \rightarrow A1 \rightarrow \mathrm{B1} \rightarrow \mathrm{B2} \rightarrow \mathrm{D2} \rightarrow \mathrm{D}$

$r_3 = A \rightarrow A2 \rightarrow \mathrm{B2} \rightarrow \mathrm{B1} \rightarrow \mathrm{C1} \rightarrow \mathrm{D1} \rightarrow \mathrm{D}$

$r_4 = A \rightarrow A2 \rightarrow B2 \rightarrow D2 \rightarrow D$

$r_5 = A \rightarrow A3 \rightarrow D3 \rightarrow D$

The total route travel time of each route can be calculated as follow:

$$rtt_1 = wt_1 + tt_{AB1} + tt_{BC1} + tt_{CD1} = 0.5\left(\frac{60}{15}\right) + 3 + 5 + 3 = 13\ mins \qquad (Ex.6.3-1)$$

$$rtt_2 = wt_1 + tt_{AB1} + wt_2 + tt_{BD2} = 0.5\left(\frac{60}{15}\right) + 3 + 0.5\left(\frac{60}{10}\right) + 6 = 14\ mins \qquad (Ex.6.3-2)$$

$$rtt_3 = wt_2 + tt_{AB2} + wt_1 + tt_{BC1} + tt_{CD1} = 0.5\left(\frac{60}{10}\right) + 3 + 0.5\left(\frac{60}{15}\right) + 5 + 3 = 16\ mins \ (Ex.6.3-3)$$

$$rtt_4 = wt_2 + tt_{AB2} + tt_{BD2} = 0.5\left(\frac{60}{10}\right) + 3 + 6 = 12\ mins \qquad (Ex.6.3-4)$$

$$rtt_5 = wt_3 + tt_{AD3} = 0.5\left(\frac{60}{4}\right) + 7 = 14.5\ mins \qquad (Ex.6.3-5)$$

Then based on (Chriqui & Robillard, 1975) the heuristic starts with calculate the expected travel time of route 4 which is the shortest route which have a value of 12 minutes. Next, we add the second least travel time route which is route 1 then the expected travel time is calculated as $0.5\frac{60}{15+10} + 9(0.6) + 11(0.4) = 11$ minutes which is lower than the previous iteration. Then we add route 2 to the hyperpath and the expected travel time is $0.5\frac{60}{15+10} + 9(0.6) + 11(0.4 \times 0.4) + 12(0.4 \times 0.6) = 11.24$ minutes which is greater than the previous iteration. The algotihm then stop at this point and hence the hyper-path consists of two routes r_4 and r_1. Note that the assumption behind this heuristic approach is based on travellers does not have a full knowledge of the waiting time and trying to consider paths that minimize the expected travel time.

Once the hyper-path is generated, the transit assignment model allocates traveller's demand from the service frequency of each route in the hyper-path. In this example the probability for travellers choosing route 4 and route 1 are 60 and 40 percent respectively. After the route and its demand of public transport are determined the next step is to load these demands into the transit network. At this stage we only modelled the route choice behaviour before they use the transit system. The waiting time and travel time are not only costs that are considered, but also a crowdedness factor which will be determine after all the transit demand have been loaded at the transit network. With the additional factor, the actual travel costs which reflects the attractiveness of the public transport are then determine. The process of determining the actual cost of travel will be discuss at the day-to-day traffic model which will be explain later in this chapter.

6.3.5 Stochastic traffic assignment

This thesis utilizes stochastic traffic assignment with logit model to determine route choice for private transport users i.e.,$f_{i,j,4}^{t,d}$. The assignment model commences after determining K shortest paths for private transport within the road network. It should be noted that the paths for private transport differ from those of public transport due to network topology. The approach is a widely used method in transportation engineering for modelling the behaviour of travellers when choosing routes under uncertain conditions (Cascetta et al., 1996). The logit model is a popular tool in transportation research and has been extensively applied in stochastic traffic assignment. The model assumes that travellers choose routes based on a utility function that considers the perceived cost and benefits of each option. The probability of selecting a particular route is then derived from the ratio of the utility of that route to the sum of the utilities of all available routes. Equations 6-12 describe the stochastic traffic assignment model used in this thesis:

$$f_{i,j,4,r}^{t,d} = f_{i,j,4}^{t,d} \frac{e^{-\emptyset RTT'^{t,d}_r}}{\sum_{k \in R(i.j)} e^{-\emptyset RTT'^{t,d}_k}}, \qquad \forall r \in R(i.j)$$

$$RTT'^{t,d}_r = \sum_l \delta_{l,r}^{t,d} LTT'^{t,d}_l \qquad \forall l \in r$$

Equation 6-12 Stochastic traffic assignment

Where $f_{i,j,4,r}^{t,d}$ is a public transport vehicle flow using route r, $\emptyset$ Is the scale of Gambel distriction at route choice level $RTT'^{t,d}_r, LTT'^{t,d}_l$ are travel time of route r and link l on time horizon t and day d, respectively. $\delta_{l,r}^{t,d}$ is a binary variable which will equal to 1 when link l is a member of route r, i.e., $l \in r$. Note that, on

each OD pairs, number of routes and its member are determined from the K-Shortest path algorithm.

Using the stochastic model with K-shortest path algorithm, this approach can efficiently handle large-scale networks with a high level of complexity. The K-shortest paths algorithm can efficiently identify the shortest paths between each origin-destination pair, which is critical in large networks where there may be a high number of routes to consider. The logit model can then estimate the probability of selecting each route based on the various attributes, such as travel time, cost, and convenience. This provides a more comprehensive understanding of the travel patterns and traffic flow in the network (Amini et al., 2016). With this model combination, the route flow for private vehicle mode can then be determined.

6.4 Within day traffic model

The within-day traffic model is situated between two route choice models (current time horizon t and the subsequent time horizon t+1) and the day-to-day traffic model. This model is responsible for processing route flows $f_{i,j,m,r}^{t,d}$ from all modes of transportation, as generated by the route choice models discussed in the preceding section. The within-day traffic model subsequently incorporates these flows into the road network and computes the actual travel time for each route and mode. These actual travel times are then relayed to the day-to-day traffic model, which utilizes them to determine mode choices. The other function of the within-day traffic model is to determine the number of time horizons t during which the route flow remains on the road network. This determination is based

on the current network condition and the actual route travel time. Figure 6.6 illustrates the process of the within-day traffic model.

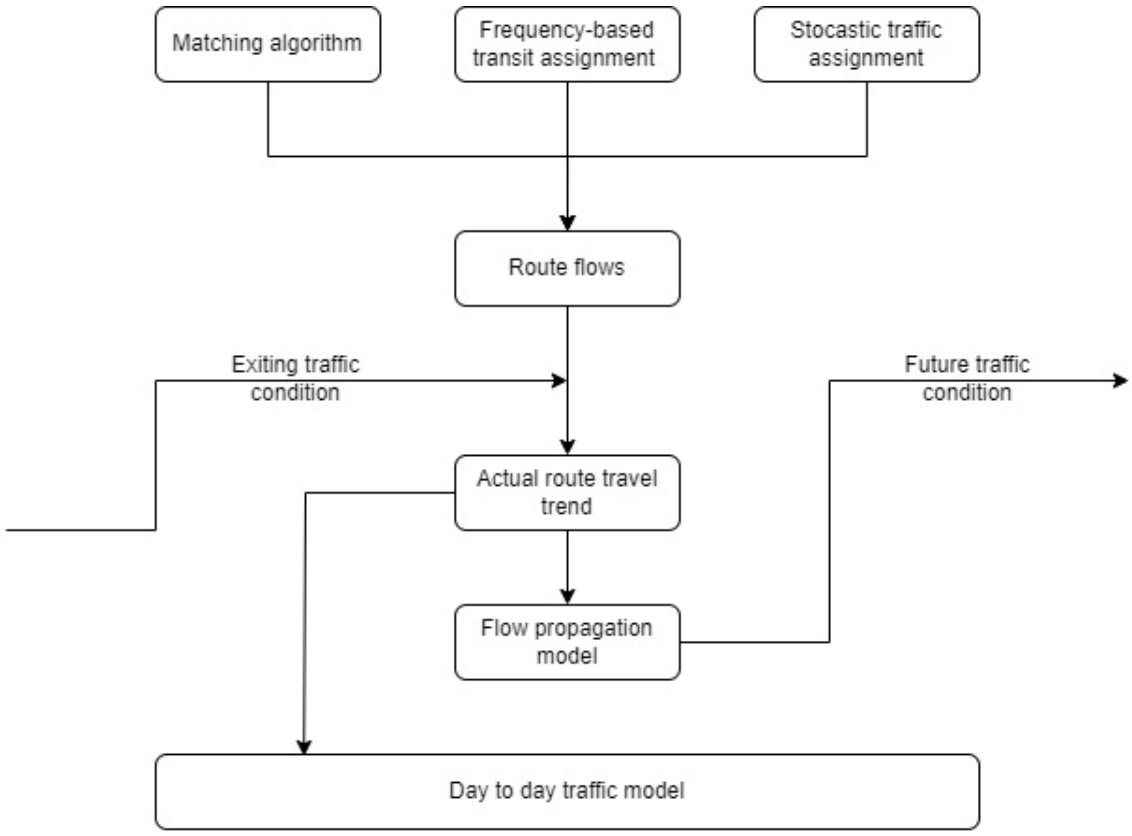

Figure 6.6 The within-day traffic model

6.4.1 Actual route travel time

Once all route flows $f_{i,j,m,r}^{t,d}$ for all OD demands originate from node i to node j with all modes of transport m and routes r are determined, the first process of the within-day traffic model is to convert these route flows into the road network. This can be initiated by converting route flows into links flows using Equation 6-13.

$$x'^{t,d}_{k} = \sum_{i}\sum_{j}\sum_{m}\sum_{r} f^{t,d}_{i,j,m,r}\,\delta^{t,d}_{k,i.j.m.r}$$

Equation 6-13 Link flow conservation

Where $x'^{t,d}_{k}$ is vehicle flows on link k, $\delta^{t,d}_{k,i.j.m.r}$ is a binary variable which equal to 1 if link k is on route r with mode m between O-D pair (i,j), otherwise equal to 0. The binary variable is to ensure that only route flows that pass the given link are considered. Once all links flow is determined, the next stage is combining the flow with existing traffic and calculates the travel time using the volume-delay function. Hence, Equation 6-14 are used in this stage.

$$x^{t,d}_{k} = x'^{t,d}_{k} + \hat{x}^{t,d}_{k}$$

$$LTT'^{t,d}_{k} = LTT^{o}_{k}\left[1 + 0.15\left(\frac{\hat{x}^{t,d}_{k}}{C_k}\right)^{4}\right]$$

$$LTT^{t,d}_{k} = LTT^{o}_{k}\left[1 + 0.15\left(\frac{x^{t,d}_{k}}{C_k}\right)^{4}\right]$$

Equation 6-14 Link Travel Time calculation

Where $\hat{x}^{t,d}_{k}$ is an existed link flow in the road network at time t prior the route assignment. $x^{t,d}_{k}$ is the total link flow which will be used for calculating the actual link travel time. C_k is the capacity of link k. Note that the volume-delay function in the second equation represents the US Bureau of Public Roads (BPR) function with recommended values of parameters (α=0.15 and β=4). LTT^{o}_{k} is the free flow travel time of link k. Finally, $LTT'^{t,d}_{k}, LTT^{t,d}_{k}$ are the link k travel times at time t and day d before and after route flow are loaded into the network. After

calculating the travel times for all links in the network, the next step is to estimate the travel time for each route. The estimated route travel time represents the actual travel time that would be experienced by users who select the given route. The calculation of route travel time involves summing the travel times for all the links that make up the route. The route travel time is estimated by summing the link travel times along the chosen path (Equation 6-15), following standard formulations in traffic assignment literature (Ortúzar & Willumsen, 2011; Sheffi, 1985) This approach is used in both static and dynamic traffic assignment models to represent the total cost experienced by a traveller along a route:

$$RTT_r^{t,d} = \sum_k \delta_{k,r}^{t,d} LTT_k^{t,d} \qquad \forall k \in r, \forall r \in R(i.j)$$

Equation 6-15 Actual Route Travel Time

Where $RTT_r^{t,d}$ is an actual Route Travel Time for route r at a time t and day d. $\delta_{l,r}^{t,d}$ is a binary variable which equal to 1 if link k is on path of route r. After computing the actual travel times for all existing routes, these values are utilized in two ways. Firstly, they serve as input for the flow propagation model, which calculates the duration of demand on the transportation network. Secondly, they are transmitted to the day-to-day traffic model, which computes the perceived route travel time and cost for each origin-destination pair on the following day.

6.4.2 Flow propagation model

The flow propagation model based on the rolling horizon technique is a dynamic and iterative approach for modelling traffic flow in transportation networks. Its aim is to predict and manage the flow of traffic over a series of time intervals,

known as horizons, by dividing the analysis of the network into smaller time periods. The rolling horizon technique involves estimating the traffic flow for each horizon based on the data available at the beginning of that period. The estimates are then updated continuously as new data becomes available, enabling accurate predictions of traffic flow.

This study employs the flow propagation model to estimate the duration of demands on the route flow, specifically in the road network. To determine the length of time the route flow persists in the network, one approach is to calculate the maximum number of horizons for which the flow will remain on the route. This can be achieved by dividing the route travel time by the length of each horizon and taking the ceiling of the result using Equation 6-16.

$$n_r^{t^*,d} = \left\lceil \frac{RTT_r^{t^*,d}}{t_{Length}} \right\rceil \qquad \forall r \in R(i.j)$$

Equation 6-16 Number of horizons

Where, $n_r^{t^*,d}$ is the number of horizons that the demand of route r originated at time t^* and day d remains in the road network (including horizon t^*th). The variable t_{Length} represents the length of the horizon, which remains constant throughout the simulation period and is defined by the user. For example, suppose we are analyzing a network with a total time period of 24 hours, and we have divided the time period into 96 15-minute horizons ($t_{Length} = 15$ minutes). If a route r originated at time t^* has an actual travel time of $RTT_r^{t^*,d} = 46$ minutes, we can estimate that the flow will remain on the route for a maximum of 4 horizons or until the end of $t^* + 3$ horizon. After the number of iterations that demand on each route last, we can then compute the existed link flow or $\hat{x}_k^{t+1,d}$ with Equation 6-17.

$$\hat{f}_{i,j,m,r}^{t+1,d} = \begin{cases} f_{i,j,m,r}^{t^*,d} & \text{, if } t+1 \leq (t^* + n_r^{t^*,d} - 1) \\ 0 & \text{, if } t+1 > (t^* + n_r^{t^*,d} - 1) \end{cases}$$

$$\hat{x}_k^{t+1,d} = \sum_i \sum_j \sum_m \sum_r \hat{f}_{i,j,m,r}^{t+1,d} \delta_{k,i.j.m.r}^{t+1,d}$$

Equation 6-17 Flow Propagation model

Where $\hat{f}_{i,j,m,r}^{t+1,d}$ is remaining flow in the network at time $t+1$. $\hat{x}_k^{t+1,d}$ is the existing flow of link k at time $t+1$.

In order to have a better visualization of how the flow propagation in this thesis works, we introduce a numerical example. Assume that at time horizon t with $t_{Length} = 15$ minutes there are total of three routes that pases link k Route 1 has an actual route travel time $RTT_1^{t^*,d} = 46$ mins, $RTT_2^{t^*,d} = 29$ mins and $RTT_3^{t^*,d} = 14$ mins, with a route demand of 15, 20 and 30 vehicles, respectively. Table 6.4 shows the result of the flow propagation model:

Table 6.4 Numerical example of flow propagation (vehicle per 15 minutes)

	t	t+1	t+2	t+3	t+4
Route Flow I $f_{i,j,m,1}^{t^*,d}$	15	15	15	15	0
Route Flow II $f_{i,j,m,2}^{t^*,d}$	20	20	0	0	0
Route Flow III $f_{i,j,m,3}^{t^*,d}$	30	0	0	0	0
Existed link flow $\hat{x}_k^{t+1,d}$		$\Sigma = 35$			

In this example, $t^* = 1$ and $t = 1$, Hence the number of horizons that the demand of route I, II, III existed in the network are $n_1^{t^*,d} = 4$, $n_2^{t^*,d} = 2$ and $n_3^{t^*,d} = 1$, In the next time horizon $t+1$, the existed flow of link k, or $\hat{x}_k^{t+1,d}$ will be equal to

15+20 = 35 vehicle, The demand of route 3 is finished at time horizon t, This is because its actual travel time $RTT_3^{t,d} \leq t_{Length}$. It is worth taking note that we cannot yet compute $\hat{x}_k^{t+2,d}$. This is because, it still missing the route flows from assignment models at time $t+1$.

Based on this proposed flow propagation model. The simulation framework can capture the temporal effects of traffic conditions in the road network. With this rolling horizon approach, the detail level of the simulation can be tailored for each study. If t_{Length} is large, the model can be performed similar to static assignment model which have less computational cost and suitable in a large scale of network. On the other hand, if t_{Length} is small the model mimics the dynamic assignment which is more detail and realistic.

6.5 Day-to-day traffic model.

The day-to-day traffic model focuses on predicting changes in travel demand over multiple days. By considering their previous experiences in term of travel cost, the model then converts these values into perceived travel costs. Within the model, a learning process function updates the traveller's perception of travel costs for each mode of transport, given the same origin and destination. Notably, the model attributes greater importance to the fresher experiences of the later day, as they have a stronger impact on the travellers' perception compared to experiences from previous days.

There are several studies that developed simulation frameworks to model day-to-day traffic assignment in traditional macroscopic traffic problem. For

example. (Zhou, B. et al., 2017) integrates day-to-day traffic model to predict the travel demand under a predefined market penetration and Advance Travelers Information System (ATIS). (Yu, Han & Ochieng, 2020) proposed a day-to-day traffic assignment with imperfect information on a large-scale network such as Anaheim network. (Sun, M., 2023) developed a new day-to-day dynamic model for autonomous vehicle. Although there are several studies on day-to-day traffic, we found only one study that related to ridesharing system. (Wei et al., 2020) proposed a double dynamic traffic models (within-day and day-to-day traffic model) that incorporates with ridesharing system. The model is used to investigate the performance of each ridesharing pricing schemes and compared it with its impact to the traffic congestion.

In the day-to-day traffic model, the actual route travel time for each route calculated from the within-day traffic model are weighted-average (based on its corresponding route flow) for each mode of transport and OD pairs. These aggregated values represent the average travel time when travellers travel from Node i to j with mode m at time t and day d or $TT_{i\,j\,t\,d}^{m}$. In addition to the travel time, other values that needs to be summarized and aggregated in this day-to-day traffic model are ridesharing costs and fees obtained from the optimization model. These latter values can be directly aggregated once the matching results are generated and does not require to be processed in the within-day traffic model. This is due to the assumption that ridesharing users and drivers recognize their ridesharing costs and fees prior the matching and they compared their total cost with non-ridesharing mode. Hence, the ridesharing cost and fee are fixed once both users are accepted prior the trip is initiated. It is worth to mention that, although ridesharing users their costs with the other mode, the actual travel cost at the end of the trip may not equal from what they are expected. This is due to

the traffic conditions that based on other route flows which are unforeseen in the matching algorithm. Despite this, the total actual cost will ultimately reflect the number of ridesharing users in following simulation days.

For public transport users, the crowdedness factor can be determined after the transit assignment model. The public transport system within this thesis is assumed to be travelling on dedicated links and not affect to the road networks. Once the total demands of public transport users on each links and OD pairs are loaded into the transit network where its capacity C_b for service line b capacity are determined from the service frequency f_b and the transit vehicle capacity c_b. We allow the public transport can take passengers more than its capacity, however the cost from the crowdedness is increased exponentially when its over-capacity based on Equation 6-18.

$$C_b = f_b \cdot c_b$$

$$\omega_{i\,j\,t\,d\,b} = \frac{\max\{\ddot{x}_k^{t,d}\}}{C_b}$$

$$C_{i\,j\,t\,d\,b} = \beta_{PT} exp^{\omega_{i\,j\,t\,d\,b}}$$

Equation 6-18 Crowdedness factor

$\omega_{i\,j\,t\,d\,b}$ is the maximum capacity ratio of bus users who travelling between node i and j , $\ddot{x}_k^{t,d}$ is a user flow on a public transport link k. β_{PT} is a public transport crowdedness coefficient. $C_{i\,j\,t\,d\,b}$ is the cost from public transport crowdedness. Once all travel costs related parameters for all modes of transport are determined. The latter stages of to recalculate the total travel cost on each mode based on the utility function shown in the mode choice model and converts it to the perceived mode cost for future references. The calculation of the perceived mode cost

(Equation 6-19) is based on the weighted average learning operator (Cascetta 1989; Yu and Han 2019).

$$U_{i,j,m}^{t,(d)} = \frac{1}{\sum_{i=1}^{k} \lambda^{i-1}} \left(U_{i,j,m}^{t,d-1} + \lambda U_{i,j,m}^{t,d-2} + \lambda^2 U_{i,j,m}^{t,d-3} \dots \lambda^{k-1} U_{i,j,m}^{t,d-k} \right)$$

Equation 6-19 Perceived mode cost

The parameter $\lambda \in (0,1)$ represents the weight of past days' experienced costs. k is the number of previous days that influenced the present day's perceived cost. $U_{i,j,m}^{t,(d)}$ is the perceived mode cost of mode m travelling between node i and j at time horizon t and day d. Next, we defined perceived cost in the equation a disutility value. This perceived mode cost is then sent back to the mode choice model in the next simulation day. Based on previous days costs, the perceived mode cost is used to calculate the mode choice behaviour on following days. This enables us to track changes of traveller's choice in terms of mode of transportation and its impact to the road network. Model convergence is achieved when modal splits reach stability. In this thesis, stability is defined as the point where successive iterations show only negligible changes in these values, consistent with the approaches discussed by (Cantarella & Cascetta, 1995; Ortuzar & Willumsen, 2011; Sheffi, 1985) Once convergence is reached, sensitivity analysis can be conducted by examining the resulting modal splits under variations in key parameters.

6.6 Discussion

In this thesis, we proposed a macroscopic traffic simulation model with multimodal traffic network, including ridesharing, taxi-sharing, public transport, and private vehicles. The availability for each mode of transportation is based on traveller's vehicle ownership. The simulation breaks the dynamic of traffic in two level; The first level is separated by time horizon t. In the beginning of each time horizon the mode choice model is used to calculates the mode split or travellers. The second level is separated by day d or simulation iterations. This allows us to capture the road network with dynamic traffic conditions when t are changes and an evolution of user's behaviour when d are changed. The simulation initiates at t=1 and d=1 with a predefined modal split parameter. Once the number of travellers on each mode are determined, the next stage is to determine the route and its flow. We further introduce route choice model for each mode of transport, which works seamlessly.

For ridesharing and taxi-sharing systems, we introduce an optimization model based on linear programme. The optimization model is commonly used to find the best matching pairs of ridesharing drivers and passengers, using an objective function and a set of constraints. With the linear programming, a scale of matching problem can be increased significantly but with smaller computational cost. We further develop this model to incorporate single driver – multiple passengers matching mechanism which enhance the benefit of such systems. For taxi-sharing system, an additional model to determine vacant taxi movements is also implemented.

For public transport system, a frequency-based transit assignment is used to generate hyper paths and calculates its flow. The public transport system has a unique assignment problem called the common lines problem. The common lines problem arises when multiple transit lines share common nodes and links along their routes with different service level such as frequency and operational speed. This situation can create complexities in the assignment process because passengers may have multiple options to travel between their origin and destination using different lines. To solve this problem, we modified the transit network with route section approach to reduce the network complexity before determining hyper paths, A heuristic algorithm by (Chriqui & Robillard, 1975) is used to find a set of possible transit routes or hyper-paths. Once the hyper-path for each OD is generated, flows on each hyper-path is proportionate based on the service frequency.

For private transport, a stochastic traffic assignment with multinomial logit model is used to calculate traveller's route choice. The first stage of this route choice model is to generate a set of possible routes between an OD pair. K-shortest path based on Yen's algorithm is used in this thesis. Once routes are determined, the flow of each route are calculated based on the logit model. Once the route and its flow for all mode of transport are calculated, the next stage of the simulation framework is a within-day traffic model.

The within-day traffic model starts from calculating the actual route travel time after all the flows are loaded in the network, which might not equal in the route choice model level. After the actual travel time for all routes and modes are determined, the model used these values in two folds. First, the model determines

the duration of each route flows in the network and propagate them accordingly. This flow propagation model is used to calculate link's flow for route choice models in future time horizons. Second, actual travel times are sent to the day-to-day to traffic model. The day-to-day traffic model is responsible for summarizing all travel cost related values into average cost of travel for all modes of transports and OD pairs. These values are then converted into perceived mode costs for next days d using the weighted average learning operator. In the following day, the simulation restarts with these perceived values to calculate to modal split and followed my route choice models for the corresponding mode of transport. The simulation framework then continues with these processes until it reaches the last simulation day d or the convergence of the model is found.

In summarizing this crucial chapter, the first and second objectives of the thesis have been structurally addressed through the development of the integrated methodological framework. The chapter detailed the specific implementation of the macroscopic mode choice model, linking modal utility functions directly to the performance of the road network (WDM) and traveller learning processes (D2DM). Most notably, the development of the novel Linear Programming (LP) based macroscopic matching algorithm directly fulfills the second objective, establishing a computationally efficient method for shared mobility assignment essential for achieving tractability in large-scale urban analysis. The subsequent chapters proceed to validate the mathematical coherence of this complex integrated framework and demonstrate its capability as a robust policy tool.

CHAPTER 7 NUMERICAL EXAMPLE

This chapter provides a numerical example of our proposed simulation framework, marking the essential transition from theoretical framework development (Chapter 6) to robust practical application and fulfilling the initial requirements of the Third Objective of this thesis: to evaluate the capability of the proposed model against timely research questions. The overarching purpose of this chapter is twofold: first, to perform internal validation by demonstrating the mathematical soundness and numerical convergence of the integrated multimodal model; and second, to conduct computational feasibility analysis to ensure the framework can scale effectively for larger policy evaluations. For this purpose, the Sioux Falls network is used, a small, well-established benchmark model for transportation problems. Using this tractable network allows our simulation's equilibrium results to be compared with other related studies in the field, providing a reliable proof-of-concept before implementing larger datasets in Chapter 8.

As part of our internal model validation, we incorporate sensitivity analysis to assess the reasonableness of results by systematically varying key parameters, such as fares and the Value of Time (VoT), and observing their impact on modal splits and outputs. This iterative process strengthens our confidence in the proposed simulation framework and its behavioral realism. Finally, a stress test is performed to determine the simulation's computational limits, which identifies the matching algorithm—specifically, the generation of all possible matching combinations—as the major bottleneck. This critical analysis allows us to identify and propose a clustering heuristic approach to reduce or eliminate these

issues, ensuring computational efficiency and tractability for implementing the full spectrum of demand and policy scenarios that will be performed in the next chapter.

7.1 Sioux Fall network problem

In this section, we focus on evaluating the performance of a proposed simulation framework using the Sioux Falls network as an illustrative example (see Figure 7.1). Originally the network was developed by (LeBlanc, Morlok & Pierskalla, 1975), this 24-node, 76-link dataset has become a benchmark for transport modelling, widely used to test equilibrium algorithms and network design by (LeBlanc, 1979; Sheffi, 1985). The network covers a total link length of approximately 300 km, with an average link distance of around 3–4 km. This scale is representative of a mid-sized US city and provides a tractable yet realistic basis for testing simulation and optimisation methods. The Sioux Fall network sustained popularity stems from its transparent specification of topology, capacities, free-flow times, and O–D demand, and from its availability through the Transportation Networks for Research (TNTP) archive (Transportation Networks for Research Core Team, 2023) and related repositories (MATSim, 2020). This accessibility enables reproducibility and facilitates cross-study comparison. The network has been widely used to evaluate equilibrium algorithms, network design formulations, and route choice models (Chakirov & Fourie, 2014; LeBlanc & Abdulaal, 1979).

More recently, Sioux Falls has also been employed in shared mobility and ridesharing research. (Pang et al., 2020) used it to evaluate coupled taxi e-hailing and ridesharing services, showing reductions in vehicle miles travelled (VMT) and driver numbers compared to decoupled models. (Zhou, Z. & Roncoli, 2022) benchmarked scalable vehicle assignment strategies for real-time e-hailing services on Sioux Falls, while (Liu, Z. et al., 2018) applied it to examine shared autonomous taxi systems and their utilisation of road capacity. These studies confirm Sioux Falls as a credible and widely recognised test environment, motivating its use here to evaluate our proposed framework.

The simulation is implemented over a 15-day horizon. This value was determined through several internal trials, which guaranteed that the system stabilises by around this point in terms of mode split and route flows. Extending the horizon further did not materially change the outcomes but significantly increased computation time. This choice is also consistent with the computational limitations identified in Section 7.7, where the exponential growth of matching possibilities was shown to severely increase runtime. Extending the day-to-day horizon substantially beyond 15 days would multiply these computational costs without providing further meaningful changes in system convergence. To further analyse network dynamics within each day, the one-hour simulation period is divided into four distinct time horizons of 15 minutes each. Hence, d = 15, t = 4 and t_{Length} = 15 minutes.

In this example (Figure 7.1), bus operations are incorporated into the Sioux Falls network to provide a network with public transport test environment alongside shared mobility. Two service lines are included, both operating initially at a

fixed frequency of 8 buses per hour (f_b = 8 buses/hour), and each with a capacity of 30 passengers per vehicle (c_b= 30 passengers per vehicle). This setting represents a simplified but tractable approximation of mid-sized urban buses commonly used in small and medium-sized US cities, ensuring that passenger behavioural assignment can be investigated without creating excessive computational burden.

It should be noted that these service lines are first introduced here as a controlled baseline. In Chapter 8, the bus routes and service frequencies are modified in response to congestion patterns identified from the Sioux Falls traffic simulation. One line is adjusted to represent a high-frequency service in areas of greater congestion, while the other is set to a lower frequency in less congested areas. This design enables the framework to examine how ride-sharing interacts differently with public transport under varying service conditions. The strategy of enriching the Sioux Falls dataset with synthetic transit services is consistent with previous research (Chakirov & Fourie, 2014), which provides precedent for extending this benchmark network to multimodal analysis.

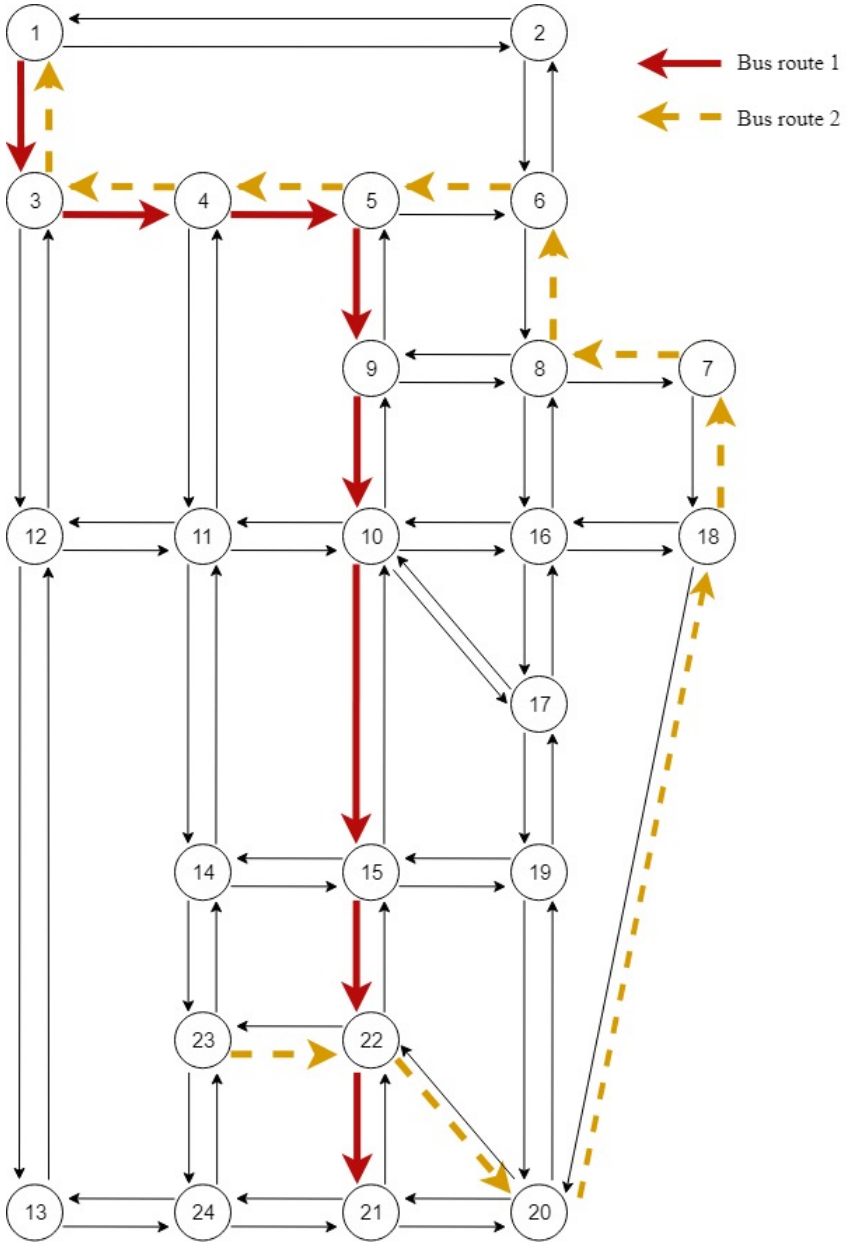

Figure 7.1 The Sioux network. With two bus lines

Overall, this setup establishes a multimodal test environment that integrates shared mobility with public transport. While simplified, the configuration provides a practical foundation for assessing traveller responses under different service conditions and for demonstrating the capability of the proposed framework. Importantly, the subsequent sections distinguish clearly between the

two shared mobility modes: Section 7.2 examines ridesharing system, where private drivers carry passengers while travelling to their own destinations, whereas Section 7.3 focuses on taxi-sharing system, where travellers access a dedicated fleet of shared taxis that continue circulating in the network. Each mode is modelled with its own demand patterns, initial mode split, and simulation parameters, ensuring that results are evaluated separately and interpreted within the correct operational context.

7.2 Numerical Convergence: Ridesharing system

In this section, we describe the simulation setup used to examine the interaction between ridesharing and public transport within the test network. In the ridesharing setting, drivers own their vehicles and have predefined destinations; once a driver reaches their destination, the vehicle is removed from the network. The setup involves pre-defining essential data, and specifying the key parameters that govern traveller behaviour and system performance. The purpose is to provide a consistent basis for evaluating the numerical results and assessing the stability of the proposed framework.

To this end, a total of 20 origin–destination (OD) pairs demands, among these, 8 ODs have access to the public transport system, while the remaining 12 ODs are restricted from such access. In this study, it is assumed that all travellers located at nodes without public transport services are car owners. This modelling choice simplifies demand allocation and avoids the unrealistic case of travellers without access to cars being left without any feasible travel option in the network. It is acknowledged that in reality such situations can occur, as some households

without private vehicles may still reside in areas with limited or no public transport. However, the assumption is made here to maintain tractability and to concentrate the analysis on the interaction between ride-sharing and scheduled public transport services. Future extensions of the framework could relax this assumption by explicitly incorporating non-car users and assigning them to alternative modes such as other demand-responsive services or walking connections to adjacent nodes. Following Table 7.1 shows an initial mode split on $d = 1$, for time horizon $t = 1{:}4$. The total demand of this table remains the same throughout the simulation.

Table 7.1 An initial mode split on $d = 1$, for time horizon $t = 1{:}4$

		Car Ownership							
		Yes					No		
Origin Node	Destination Node	Demand (trips/hr)	% Solo	% Pax	% Driver	% Bus	Demand (trips/hr)	% Pax	% Bus
20	13	1,470	0	90	10	0	0		
22	17	2,100	70	30	0	0	0		
4	1	1,680	40	30	20	10	720	50	50
22	19	210	70	0	30	0	0		
16	2	630	0	10	90	0	0		
3	9	1,890	60	10	20	10	810	50	50
7	16	2,100	0	40	60	0	0		
13	24	840	80	20	0	0	0		
23	15	420	50	20	30	0	0		
23	18	1,470	60	30	0	10	630	50	50
4	18	630	70	20	10	0	0		
23	20	210	10	30	50	10	90	50	50
23	11	2,100	30	40	30	0	0		
12	9	420	80	0	20	0	0		
19	17	1,050	70	30	0	0	0		
4	9	2,100	50	0	40	10	900	50	50
10	21	210	20	60	10	10	90	50	50
22	7	420	70	0	20	10	180	50	50
19	10	2,100	30	10	60	0	0		
23	6	420	60	10	20	10	180	50	50

Constant values that use in this simulation are; $\lambda = 0.3, \theta = 0.1, \emptyset = 0.1, \alpha = 4$, $\beta_f = 1\ \beta_{vot} = 40$ per hour , $f_{fee} = 1, WT_i = 3, f_b = 5, f_p = 0.1D_p$ where D_p is the shortest route distance (non-ridesharing) for passenger p. This assume that

passenger's fare is based on their original distance between their origin and destination, $f_{unmatched} = 0.5$ and $\beta_{PT} = 4e^{-1}$. With these parameters defined, the next step is to ensure that the simulation achieves numerical convergence.

Convergence is essential in a macroscopic traffic model as it ensures the stability of mode split proportions over time. By achieving convergence, the model can represent and predict the distribution of travellers across different modes of transportation. This capability is crucial for effective transportation planning, as it enables traffic engineers or researchers to make informed decisions regarding the impact of ridesharing systems and represent key performance index (KPI) of such system. Ultimately, the ability of a traffic model to find convergence enhances its reliability and usefulness in addressing real-world traffic challenges.

In this section, we examine the long-term behaviour of travellers' mode choice, specifically focusing on the empirical convergence of the modal split (Figure 7.2). The legend labels Solo, Pax, Driver, and Bus refer to solo drivers, ridesharing passengers, ridesharing drivers, and bus users, respectively, while V and NV indicate travellers with and without vehicle ownership. Convergence in the simulation can be assessed through two complementary indicators. Figures 7.2a and 7.2b illustrate the stability of the modal split where successive simulation days show negligible changes in the distribution of trips across modes for travellers with and without vehicle ownership. Figure 7.2c is the absolute relative difference in perceived mode costs between consecutive days, which reflects whether travellers continue to adjust their choices in response to cost variations. Larger values indicate that costs and choices are still evolving, whereas values approaching zero suggest stabilisation. Because these differences diminish rapidly as convergence progresses, they can become very small and difficult to distinguish on a linear scale. For this reason, Figure 7.2c presents the results on a logarithmic scale, which highlights the exponential reduction of cost

differences over successive days. When both the modal split stabilises and the relative differences in mode costs approach zero, the system can be considered converged, consistent with established practices in traffic assignment, where equilibrium is characterised by stability in flows and costs (Sheffi, 1985a).

Figure 7.2a Modal split for Travellers with Vehicle Ownership

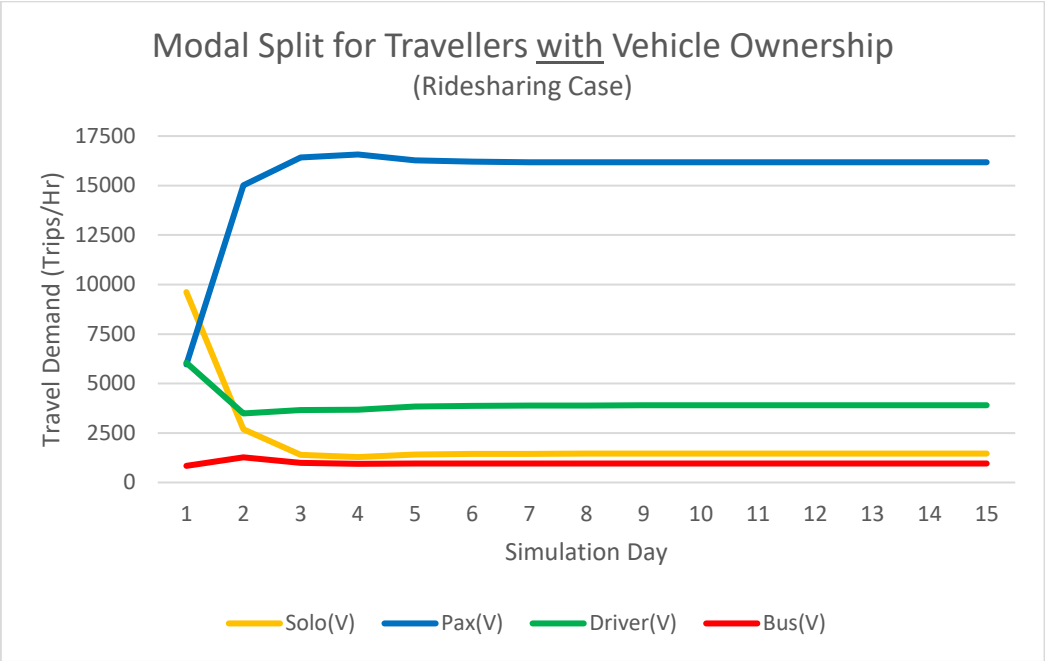

Figure 7.2b Modal Split for Travellers without Vehicle Ownership

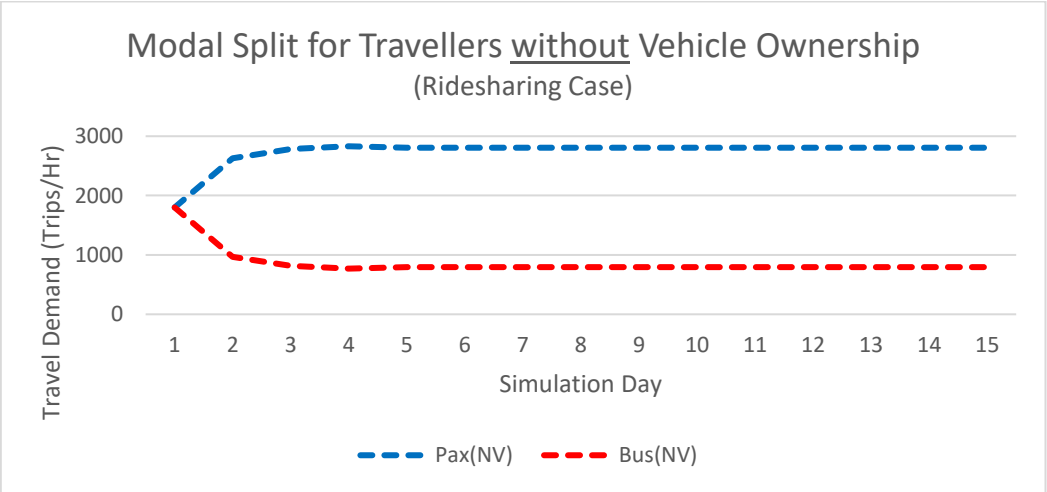

Figure 7.2c Absolute Relative different in mode costs

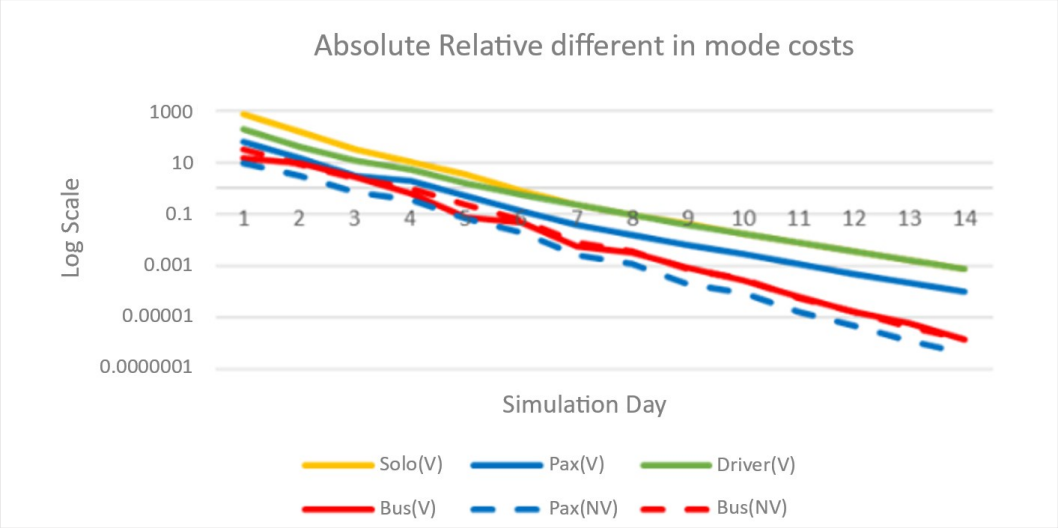

Figure 7.2 The long-term behaviour of traveller's mode choice with the empirical convergence of model split

As shown in Figure 7.2a for travellers with vehicle ownership (V), ridesharing passenger mode, Pax(V), is the most popular choice, with approximately 17,000 trips at equilibrium. Notably, the converged ridesharing passenger count, shown in the figure, does not represent the maximum demand value observed across the simulation days but represent travellers who initially choose the model at the given day. The peak occurred on the 3rd and 4th simulation, followed by a slight decline before steady. The increase in ridesharing passenger count is primarily influenced by the number of solo drivers, Solo(V), transitioning into this mode. The relatively low cost of ridesharing passengers compared to solo drivers likely contributes to this mode shift. For the number of bus users Bus(V) among travellers remains relatively stable throughout the simulation.

However, for travellers without vehicle ownership (NV) which have accessed only two modes; Bus(NV) and Pax(NV) (Figure 7.2b), there is a noticeable

decrease in the number of bus users, Bus(NV). This reduction can be attributed to the increasing popularity of ridesharing passengers Pax(NV) as the primary alternative mode for NV travellers. Overall, the analysis highlights the convergence of modal split in the simulation, indicating that the mode choice proportions stabilize over time. These findings illustrate evolving dynamics of transportation mode preferences and provide insights for effective transportation planning or research,

Although the converged result is found in the simulation, another objective of this study lies in proposing a simulation framework that effectively incorporates ridesharing to other modes of transport. Additionally, the study places significant emphasis on benchmarking the proposed matching algorithm. In early stages of the simulation, ridesharing drivers were observed to pick up passengers from different origin-destination (OD) pairs. This can be observed from the presence of residual ridesharing drivers and passengers. However, as the simulation progressed through multiple iterations, the matching result is changed into matching drivers and passengers within the same OD pair. This shift occurred due to the growing number of ridesharing passengers within the same OD pairs and was influenced by the objective function of the algorithm, that tries to maximize VKTS. Matching a driver with passengers sharing the same OD pair resulted in the algorithm achieving the maximum value of VKTS, which corresponds to saving approximately 2/3 of the total non-ridesharing trips.

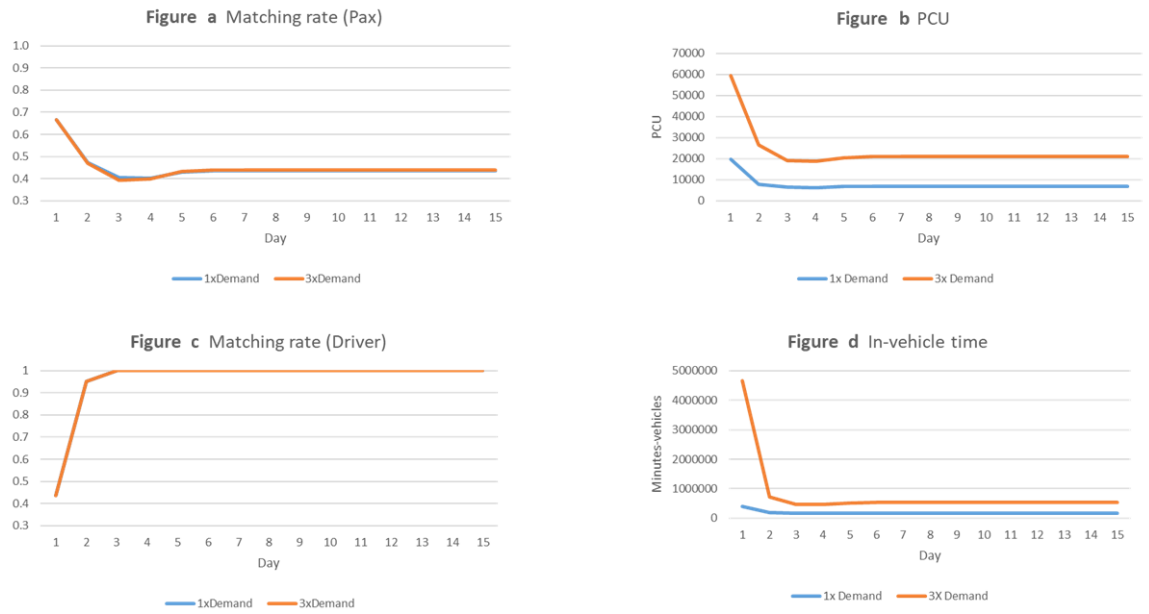

Figure 7.3 The performance of the matching algorithm under different demand scenarios

The purpose of Figure 7.3, visualises performance of the matching algorithm under different demand scenarios, is to assess the algorithm's effectiveness across varying levels of network demand. This assessment contrasts the system's performance under original demand (1x Demand), representing conditions of low traffic congestion, against its performance under a scenario with three times the original demand (3x Demand), which simulates high levels of congestion.

The results shown in the figure consistently indicate that the ridesharing system generates substantial benefits for the road network. These benefits are quantified primarily by the reduction in the total Passenger Car Unit equivalent (PCU) (visible in Figure 7.3b) and the decrease in the overall in-vehicle time (shown in Figure 7.3d).

Crucially, the simulation demonstrates that the ridesharing system yields greater benefits in highly congested networks compared to those that are less congested. This improved efficiency is tied to the successful operation of the proposed matching algorithm, which allows for a maximum of two passengers per driver, resulting in a reduction of approximately two-thirds of the total PCU. It is important to note that the initial PCU values observed on the first day of iteration do not necessarily depict the maximum worst-case scenario, as they are initially influenced by the predetermined modal split and the proportion of travelers with and without vehicle ownership

In terms of the matching rates observed in Figures 7.3a and 7.3c, it is shown that at the point of convergence, all drivers are matched, while approximately 45 percent of passengers are successfully matched. This rate suggests that there is an oversupply in the number of passengers within the ridesharing system. Notably, the number of passengers spikes sharply, whereas the number of ridesharing drivers remains steady, indicating that the generalized cost for ridesharing passengers is considerably low. Despite the presence of unmatched passengers who incur an additional cost, the ridesharing mode remains attractive enough for travellers, leading to their willingness to risk being rejected by the matching algorithm. This implies that the benefits derived from entering a ridesharing system continue to outweigh the potential extra cost for unmatching results, promoting this mode as a preferred choice for travellers.

The numerical analysis conducted in Section 7.2 confirms the substantial benefits of the ridesharing system, showing a significant reduction in both the total Passenger Car Unit equivalent (PCU) and overall in-vehicle time, providing

essential insights into how shared mobility contributes to a more sustainable and cost-effective transportation network, particularly in strained traffic conditions. Building on this, Chapter 8, "Impact of Shared Mobility Systems", provides the large-scale visual evidence of congestion relief by applying the full simulation framework, showcasing the resultant dramatic reduction in link flow (comparing Figure 8.2 to Figure 8.4) and the measured improvement in average network speed (Figure 8.7) across a complex road network.

7.3 Numerical Convergence: Taxi sharing system.

Next, the numerical example for taxi-sharing system is undertaken in this section. In this simulation there is no ridesharing driver mode, only ridesharing passengers (who use a taxi-sharing system) are defined. The same set of 24 origin-destination (OD) pairs and their corresponding demands, as used in the ridesharing system, are applied. However, the initial modal split is adjusted accordingly. Furthermore, 100 taxis are assigned to each node, evenly distributed, resulting in a total of 2,400 taxi-sharing vehicles for this simulation. This relatively large fleet size is chosen to ensure sufficient vehicle availability across the network, avoiding artificial supply shortages and allowing the analysis to focus on traveller behaviour and system interactions rather than fleet scarcity. Table 7.2 displays the total traveller's OD demands used in the taxi-sharing simulation.

Table 7.2 The total traveller's origin-destination demands used in the taxi-sharing simulation.

		Car Ownership						
		Yes				No		
Origin Node	**Destination Node**	**Demand (trips/hr)**	**% Solo**	**% Pax**	**% Bus**	**Demand (trips/hr)**	**% Pax**	**% Bus**
20	13	1,470	50	50	0	0		
22	17	2,100	50	50	0	0		
4	1	1,680	50	40	10	720	50	50
22	19	210	50	50	0	0		
16	2	630	50	50	0	0		
3	9	1,890	50	40	10	810	50	50
7	16	2,100	50	50	0	0		
13	24	840	50	50	0	0		
23	15	420	50	50	0	0		
23	18	1,470	50	40	10	630	50	50
4	18	630	50	50	0	0		
23	20	210	50	40	10	90	50	50
23	11	2,100	50	50	0	0		
12	9	420	50	50	0	0		
19	17	1,050	50	50	0	0		
4	9	2,100	50	40	10	900	50	50
10	21	210	50	40	10	90	50	50
22	7	420	50	40	10	180	50	50
19	10	2,100	50	50	0	0		
23	6	420	50	40	10	180	50	50

All constant parameters defined in the ridesharing system is remain the same in this simulation; $\lambda = 0.3, \theta = 0.1, \emptyset = 0.1, \alpha = 4,\ \beta_f = 1\ \beta_{vot} = 40$ per hour , $f_{fee} = 1, WT_i = 3, f_b = 5, f_p = 0.1D_p$ where D_p is the shortest route distance (non-ridesharing) for passenger p. This assume that passenger's fare is based on their original distance between their origin and destination, $f_{unmatched} = 0.5$ and $\beta_{PT} = 4e^{-1}$.

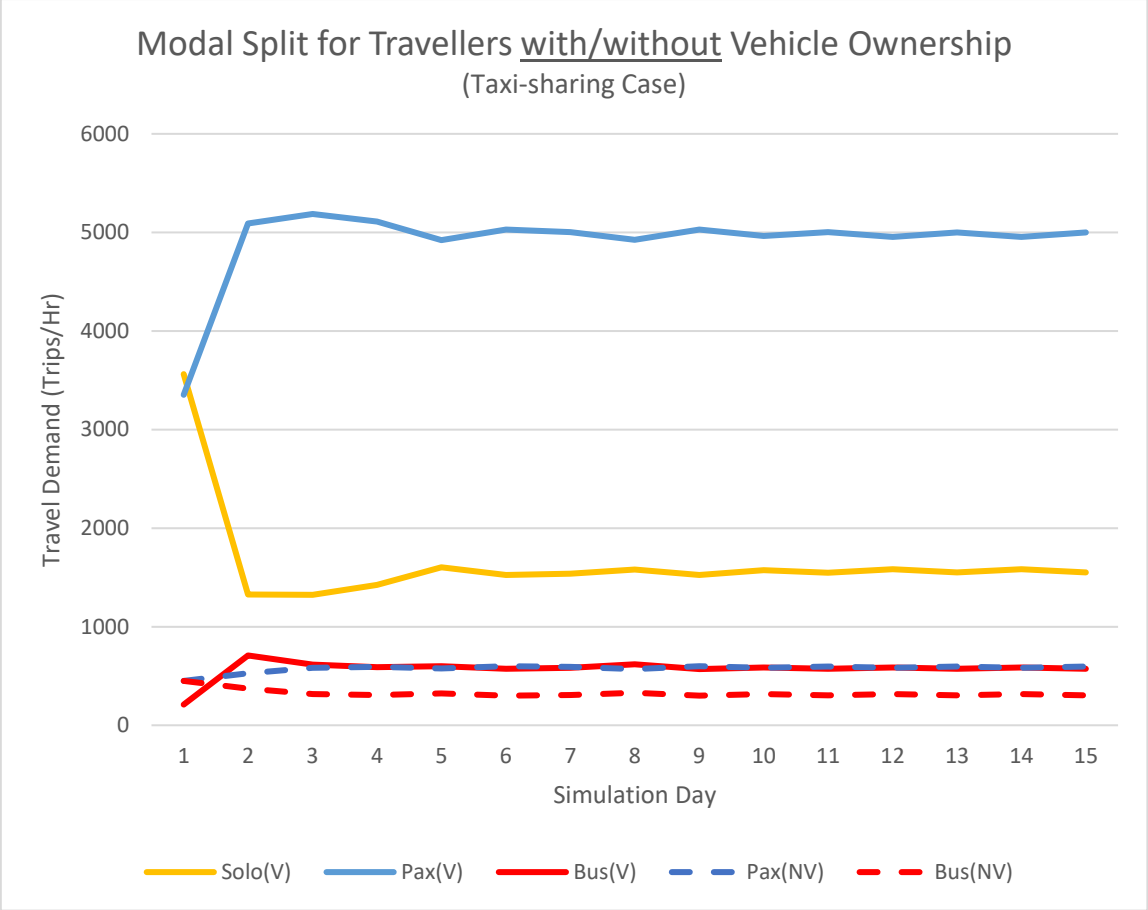

Figure 7.4 A convergence result of the taxi-sharing simulation

Similar to the ridesharing system, the taxi-sharing simulation also demonstrates a convergence result, as illustrated in Figure 7.4. The ridesharing passenger mode remains the most popular choice, with approximately 5000 trips at equilibrium. The peak in usage occurred during the initial days, followed by a slight decrease, and subsequently stabilizing at steady values. Importantly, the fluctuations in the number of Solo(V), representing solo driver from travellers with vehicle ownership correspond inversely to the number of taxi-sharing passengers, Pax(V), indicating a direct competitive relationship between these two modes. In contrast, bus users, Bus(V) and Bus(NV), remain comparatively unaffected throughout the iterations. This outcome reflects the cost-driven substitution in the model: car owners shift from solo driving to taxi-sharing due

to its lower generalized cost, whereas bus demand shows limited sensitivity under the current assumptions. Nonetheless, it is possible that bus usage could respond more strongly if the generalized cost of public transport were higher or if travellers exhibited stronger aversion to crowding. Overall, the analysis reveals the convergence of modal splits in the simulation, indicating that the modal splits stabilize over time.

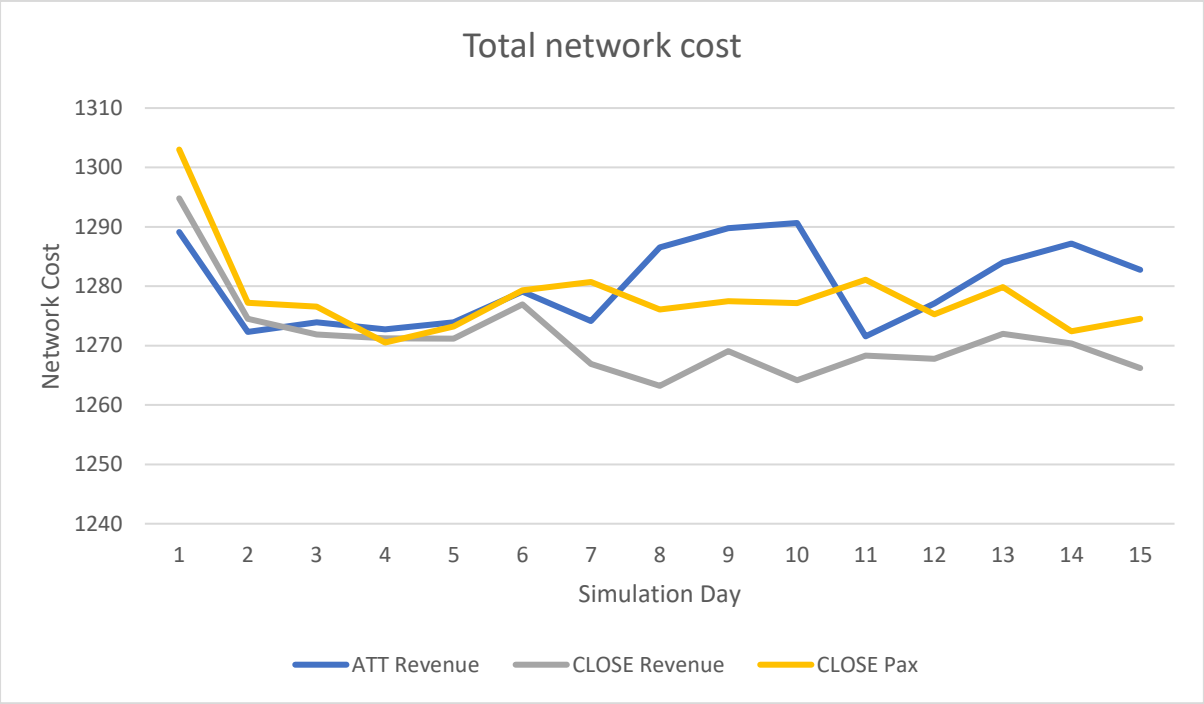

Figure 7.5 The total network cost based on different vacant taxi models.

In addition to the ridesharing system, the simulation for the taxi-sharing system incorporates an extra route choice model. This model, known as the vacant taxi model, governs the movement of taxi vehicles when they do not have passengers on board and seeking customers. Its primary objective is to optimize the search for new customers. In this section, three vacant taxi models introduced in Section 6.3.2 are implemented: ATT Revenue, CLOSE Revenue, and CLOSE Pax.

The underlying assumption behind these vacant taxi models is to replicate the behaviour of vacant taxis seeking customers. By incorporating these models, the simulation can more accurately simulate the taxi-sharing system. From the values shown in the Figure 7.5, it is observed that the choice of vacant taxi model has a relatively small impact on the total network cost. The total network cost represents the cumulative cost borne by each traveller when traversing the network. In comparison to the ridesharing system, the taxi-sharing system only achieves a marginal reduction in the network cost.

In the subsequent chapter, our focus will be on investigating the efficiency of the taxi-sharing system, exploring methods and strategies to enhance its performance and effectiveness. It is important to note that in this chapter, we provide only a numerical example, not as a policy evaluation, but as a way to validate the realism and coherence of the simulation framework. This validation is based on applying the model to a small dataset that allows full verification of the code and mathematical correctness, demonstrating that the results converge to stable modal splits and that the sensitivity analysis produces plausible responses. Building on this foundation, subsequent investigations aim to identify ways to optimize the taxi-sharing system, reduce costs, and improve overall efficiency. In doing so, this research contributes to the development of more effective and sustainable transportation solutions.

7.4 Sensitivity analysis Framework

A sensitivity analysis was conducted to assess the model's robustness and to understand how key exogenous parameters influence traveller behaviour and system-level outcomes, such as modal splits and network congestion. The analysis systematically varies critical input parameters and observes the resulting changes in the converged equilibrium state of the simulation.

7.4.1 Rationale for Parameter Selection

The following three variables were identified as primary drivers of the trade-offs between transport modes and were therefore selected for analysis. The ranges for each were strategically determined through preliminary testing to explore the model's behavior under diverse and plausible real-world conditions.

- Ridesharing Fare (f_p): As a direct monetary cost to passengers and a component of revenue for the system, the fare is a primary determinant of the attractiveness of shared mobility. Varying this parameter is essential for understanding the economic equilibrium of the ridesharing system and its competitiveness against other modes. The selected range simulates a spectrum of operator pricing strategies, from a highly competitive, low-cost model designed to attract market share, to a premium-service model. This allows the analysis to identify the critical tipping point at which cost outweighs convenience for the average user.
- Value of Time (β_{vot}): This parameter is fundamental to the mode choice model, as it governs how travellers trade off longer travel times (often

associated with shared mobility detours) against lower monetary costs. The tested range is intended to capture the heterogeneity of traveller preferences. A lower VoT is characteristic of leisure trips or lower-income individuals, while a higher VoT reflects business travel or higher-income users who are more sensitive to delays. Analyzing its impact is crucial for assessing how ridesharing's viability changes for different socioeconomic segments and trip purposes..

- Bus Fare (f_b): This variable represents the cost of the main public transport alternative. Adjusting the bus fare allows for a direct investigation of the substitution effects and the competitive relationship between shared mobility and the existing public transport network. The tested range is designed to simulate various policy scenarios, such as significant fare hikes or the reduction of public transport subsidies, allowing for a robust examination of the threshold at which travellers might abandon public transport.

7.4.2 Specification of Parameter Ranges

The ranges for each parameter in the sensitivity analysis were not chosen arbitrarily but were strategically determined through preliminary model testing. The goal was to identify a spectrum of values that produces meaningful and interpretable changes in model outputs, ensuring the analysis captures key inflection points and equilibrium shifts. This approach allows for a robust exploration of how the system behaves under different economic conditions and policy scenarios. The tested values and their rationale are specified as follows:

- Ridesharing Fare (f_p): The baseline fare was defined as a function of travel distance, f_p=0.1D_p, where D_p is the non-ridesharing travel distance for a passenger. For the sensitivity analysis in the ridesharing simulation, this base fare was scaled by a multiplier tested at; 1x (baseline), 2x, 4x, 6x, 8x, and 10x.
- Value of Time (β_{vot}): The baseline VoT was initially set at 40 (currency units/hour). The model was tested for increasing sensitivity to travel time across the set of {40, 45, 50, 55, 60}.
- Bus Fare (f_b): The baseline fare was initially set at 5 (currency units). The analysis explored the effect of increasing public transport costs across the set of {5, 10, 15, 20}.

7.5 Sensitivity analysis: Ridesharing system

In this section, we conducted a sensitivity analysis test on our simulation, the results of which are presented in Figure 7.4, illustrating results at the converged modal splits obtained from the analysis. The primary variables influencing the cost incurred by passengers appear to be the Value of Time (VoT) and the Ridesharing Fare. These are crucial parameters that, when adjusted, could potentially increase the number of drivers, and improve the matching rate for all ridesharing users. Thus, a careful evaluation of these variables might be instrumental in achieving a more balanced ridesharing system, enhancing its overall efficiency and user satisfaction.

Initially, we assessed the impact of ridesharing fare on the system. Previous figures indicated an oversupply of ridesharing passengers, which was attributed to the low generalized cost. This oversupply is largely due to the low original fare ($f_p = 0.1D_p$) that passengers pay drivers. To reduce passenger numbers and increase driver count, we varied the value of f_p to 1, 2, 4, 6, 8, and 10 times of its original fare. As shown in Figure 7.6a, this adjustment significantly influenced the modal split, indicating its sensitivity to the fare variable. An increase in the ridesharing fare made bus usage a more attractive option, once again emphasizing the competitive impact between ridesharing and bus systems.

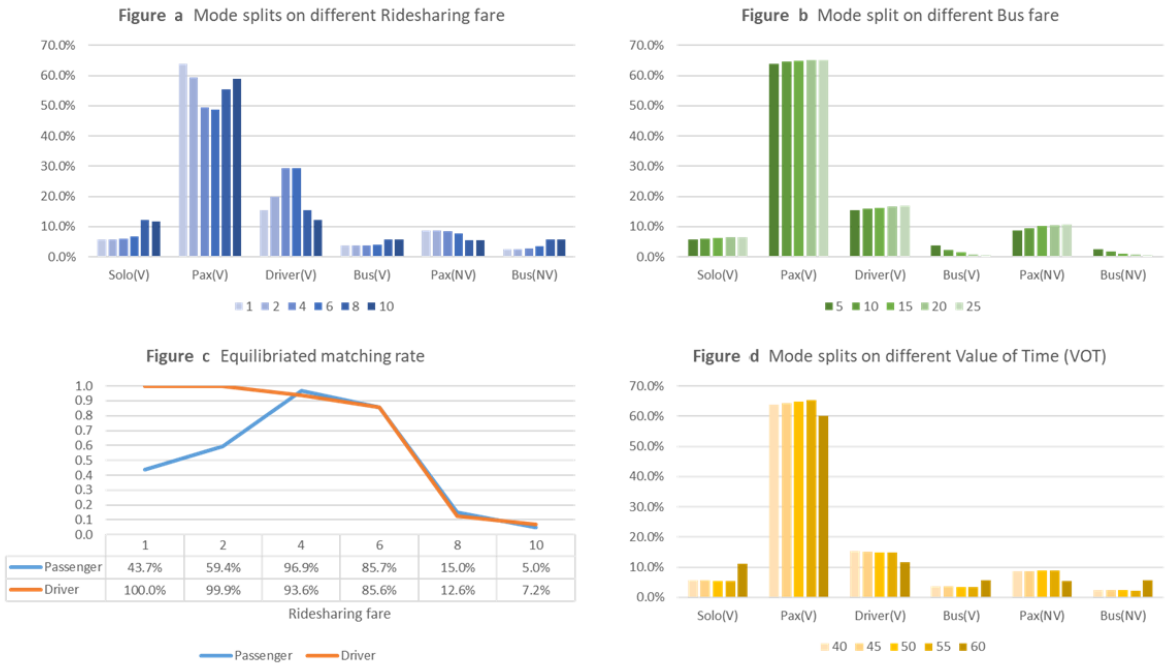

Figure 7.6 A sensitivity analysis test on ridesharing system

In the early stage of the ridesharing fare increment, the number of passengers reduced, while the number of drivers increased. This fare alteration encourages travellers to become ridesharing drivers and discourages potential passengers.

Consequently, the matching rate of ridesharing passengers peaked at $f_p = 0.4D_p$,where the ridesharing system reached its most efficiency, boasting a successful matching rate of 96.9% for passengers and 93.6% for drivers (Figure 7.6c). However, when the fare rose beyond this rate, it led to a marginal decrease in the matching rate, as the escalated cost in ridesharing caused both types of ridesharing users to switch to other modes, indicating an incline in non-ridesharing mode usage. Another insight found in these figures is a rebound number of ridesharing passengers at higher fare rates, which could be explained by a more significant decrease in ridesharing drivers.

The final parameters conducted in this sensitivity analysis were bus fare (f_b) and the Value of Time (β_{vot}), as seen in Figures 7.6b and 7.6d. An increase in the bus fare led to a decrease in bus users, while other modes experienced a small increasing in users. Subsequently, we evaluated the impact of the Value of Time (VoT), a variable integrated into the cost functions of all modes. As the VoT increased, travellers' mode decisions became more sensitive to travel time. To probe this further, we increased the value of β_{vot} to 40, 45, 50, 55, and 60 per hour. While low VoT values had minimal impact on travellers' decisions, at high values, such as β_{vot}= 60 per hour, there was a steep decline in the number of travellers using both ridesharing modes, primarily due to the longer travel time associated with these modes compared to solo driving in most scenarios.

7.6 Sensitivity analysis: Taxi-sharing system

In this section, a sensitivity analysis test was conducted to assess the performance of the Taxi-sharing system. The results of this analysis are presented in the following figures, showcasing the outcomes obtained at the converged modal splits. One of the crucial parameters influencing the performance and profitability of the Taxi-sharing system is the fare paid by passengers to taxi drivers and operators. Increasing this fare value has the potential to increase profitability while reducing the number of passengers. Therefore, conducting a sensitivity analysis on this parameter is essential for analysing its impact on other variables such as matching rate, total revenue, and modal splits.

The sensitivity analysis provides valuable insights into how variations in the fare parameter can influence the overall performance and outcomes of the taxi-sharing system. By systematically exploring different fare values, researchers and practitioners can determine the optimal fare setting that balances profitability and passenger demand. This analysis aids in optimizing the system's efficiency, revenue generation, and customer satisfaction.

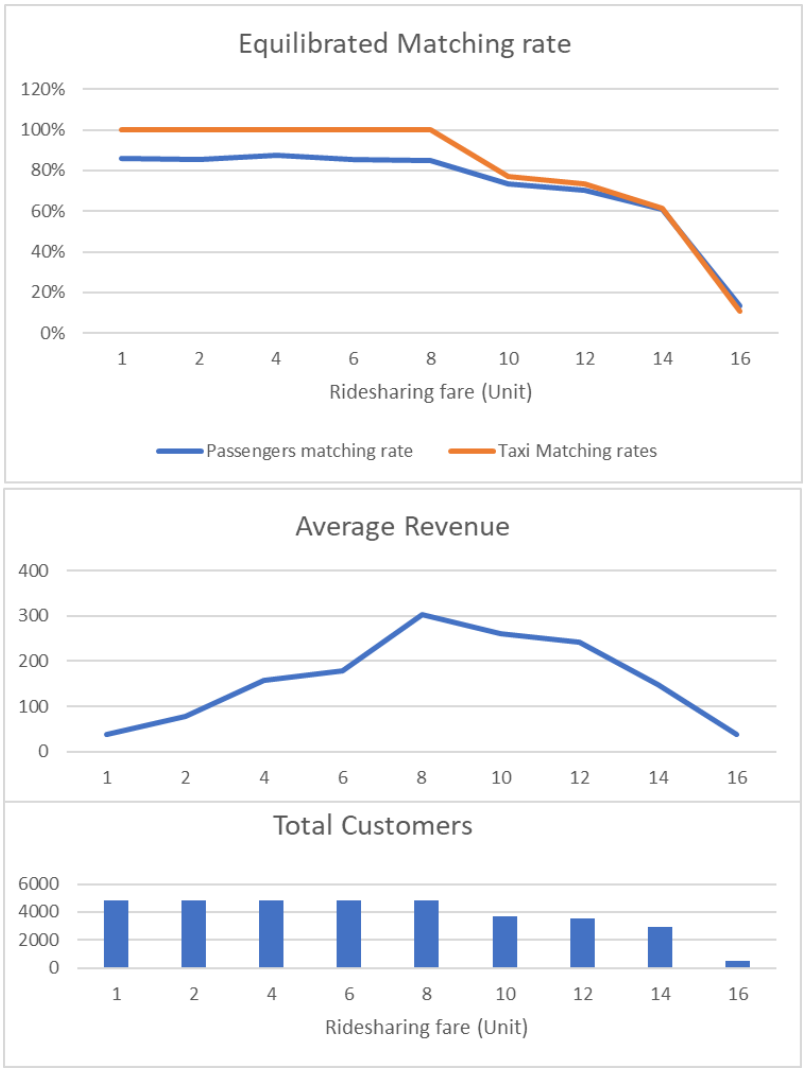

Figure 7.7 A sensitivity analysis test for taxi-sharing system

In this sensitivity analysis, the ridesharing fare was varied, starting from 1x of the original fare $f_p = 0.1D_p$, and increasing up to 16x of the original fare. Initially, when the ridesharing fare is increased, the matching rate for taxi drivers remains consistently at 100%, while the matching rate for passengers stabilizes around 80%. This suggests that the taxi-sharing system remains attractive to passengers, indicating that it is still a cost-effective mode of transportation compared to other alternatives. However, the matching rate for passengers does not reach

100%, indicating that the number of available taxi vehicles is insufficient to accommodate the demand.

In terms of revenue, increasing the fare leads to an increase in the average revenue, as the revenue per passenger increases. However, once the fare surpasses its peak point at 8x of the original fare, despite the increase in revenue per passenger, the number of successful matches reduces significantly, resulting in a decrease in average revenue. This highlights the trade-off between fare rates, passenger demand, and overall revenue in the taxi-sharing system.

Once the ridesharing fare surpasses 8x of the original fare, which corresponds to the maximum revenue of the taxi-sharing system, the matching rate for both taxi drivers and passengers decreases exponentially, along with a reduction in the number of passengers. Simultaneously, there is an increase in the number of solo drivers. This indicates that travellers with vehicle ownership (V) shift from using taxi services to self-driving, suggesting that the taxi-sharing system becomes relatively expensive for them compared to solo driving. For travellers without vehicle ownership (NV), there is a competitive relationship between the taxi-sharing system and buses, as evidenced by the passengers switching to buses when the fare is high.

The findings from this sensitivity analysis demonstrate the usefulness of the simulation in analysing various aspects of shared mobility. By examining the model splits, the effect of the taxi-sharing system on the public transport system can be analysed. Moreover, by modifying the fare structure or increasing the number of taxi vehicles, the network's response can be observed in Figure 7.8.

These outputs of the simulation provide valuable insights for studying topics such as surge pricing and optimizing fleet size in shared mobility systems.

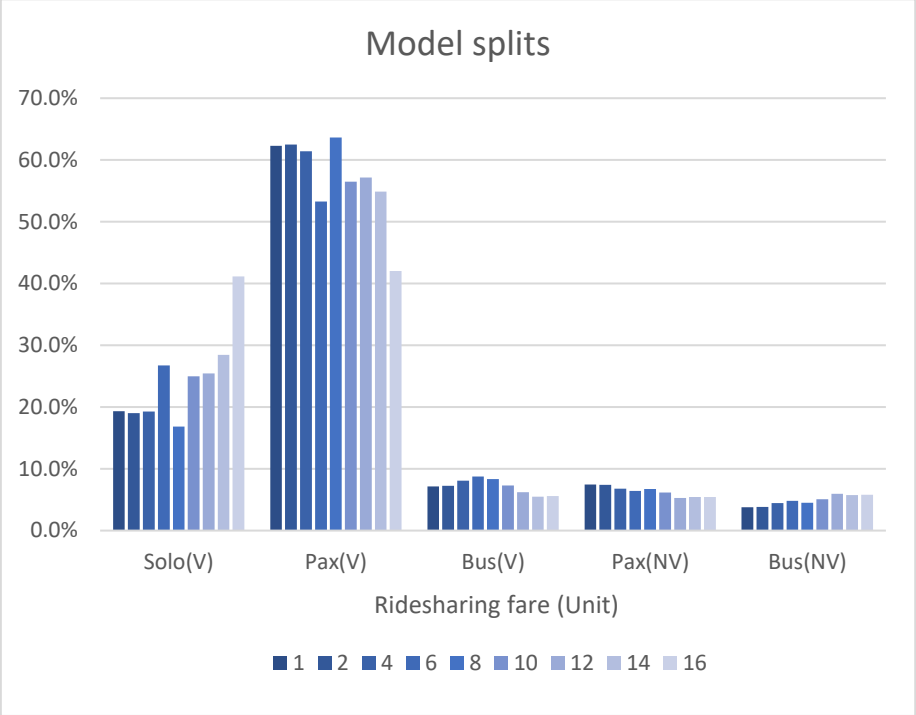

Figure 7.8 The sensitivity analysis of modal split when increasing fare.

7.7 Stress test

A stress test is a technique used to evaluate the performance, stability, and resilience of a system or model under extreme or challenging conditions. By subjecting the system to different stress scenarios, testers can evaluate its robustness, reliability, and stability under heavy loads. This process is essential to pinpoint the system's weak points, potential bottlenecks, or areas where it may fail or yield incorrect outputs. In the context of macroscopic traffic simulation, stress testing assumes a critical role in evaluating the performance of the simulation. By pinpointing the specific bottleneck of the model, it becomes possible to find solutions to address the identified issue.

In our analysis of the numerical examples presented earlier, we discovered that the matching algorithm model utilized for matching shared mobility users represents the bottleneck of our simulation. Interestingly, we observed that increasing the flow of travellers per hour did not noticeably escalate the computational cost. This can be attributed to the matching algorithm model employed in our simulation, which relies on linear programming techniques to match passengers and drivers as groups, generating a set of matched passengers and drivers for each Origin-Destination (OD) pair as outputs. This differs from conventional approaches that employ binary programming to match passengers and drivers individually, resulting a higher computational cost and not suitable for running in a macroscopic level.

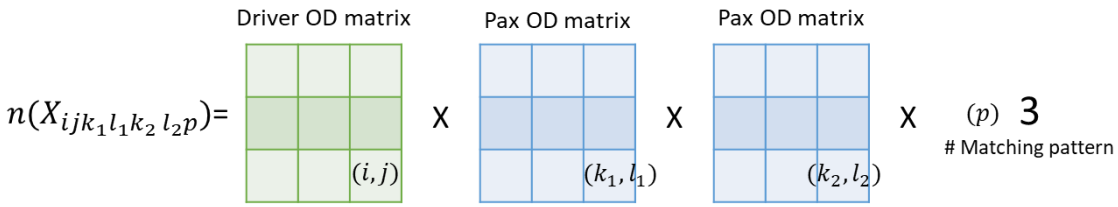

Figure 7.9 Matching possibilities

However, it was observed that an increase in the number of OD pairs led to a significant rise in computational cost within the matching algorithm. Specifically, this bottleneck is found from a step in the algorithm's initial phase, where all the matching possibilities $X_{ijk_1l_1k_2\,l_2p}$ are generated. As illustrated in Figure 7.9, the number of matching possibilities in a ridesharing system can be calculated by multiplying the number of driver OD pairs, passenger OD pairs, and the matching pattern. For instance, if there are 20 driver OD pairs and 30 passenger OD pairs, with a total matching pattern of 3 for the single driver-multiple passenger scheme, the resulting number of matching possibilities would be 20 x 30 x 30 x 3 = 54,000 possibilities. Subsequently, the algorithm proceeds

to check each matching possibility to ascertain whether the ridesharing cost for all ridesharing users is lower compared to non-ridesharing alternatives. For taxi sharing system, the equation remains the same, but driver OD pairs are replaced by the number of nodes that taxis are located. Typically, the number of matching possibilities in a taxi sharing system is lower than in a ridesharing system since the number of OD pairs tends to be greater than the number of nodes in the network.

In summary, the increase in the number of OD pairs for shared mobility users leads to a significant exponential increase in computational time. This poses a limitation on our simulation framework, as it becomes challenging to simulate larger network problems with a high number of OD pairs. To further investigate this, we conducted simulations of a ridesharing system using the same network data and constant parameters. However, in this iteration, we systematically increased the number of OD pairs for ridesharing users, including both drivers and passengers. Our objective was to observe the computational time (for matching algorithm) required to compute for a single time horizon. To obtain the total calculation time, we multiplied this computational time by the number of time horizons t, and the number of simulation day d.

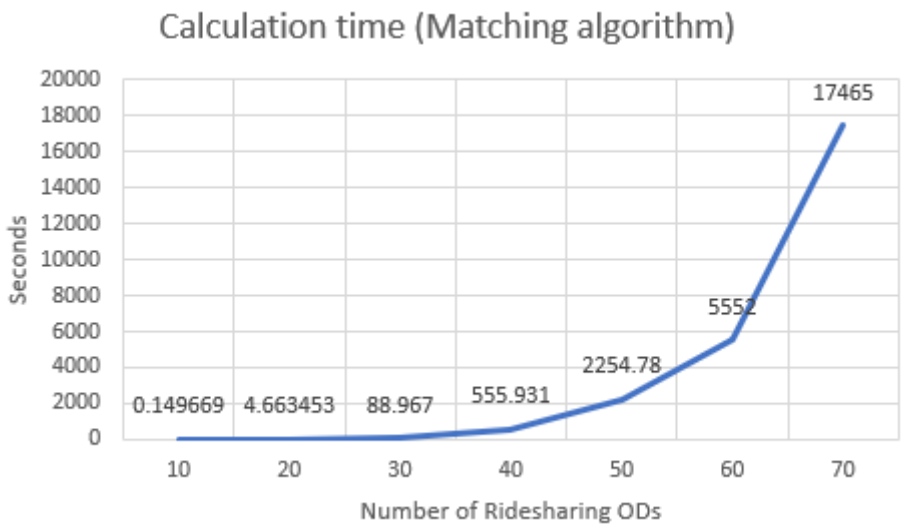

Figure 7.10 The computational time required by the matching algorithm.

Figure 7.10 presents the computational time required by the matching algorithm to pair ridesharing drivers with passengers. The computational procedure is implemented using Matlab, executed on a quad-core desktop computer operating at 3.6 GHz with 16GB of RAM. The figure illustrates that as the number of Origin-Destination (OD) pairs increases, while the computational time also increases exponentially. At the lower limit of the stress test, with 10 OD pairs, the algorithm required 0.149 seconds. However, at the upper limit with 70 OD pairs, the computation time escalated exponentially to 17,465 seconds (4.85 hours). Given the result of this stress test, it is essential to maintain computational time per time horizon below 60 seconds (1 minute) for the algorithm to function optimally. Consequently, this limitation implies that the matching algorithm can effectively handle only 20 OD pairs for ridesharing drivers and passengers within the stipulated computational time.

Given the current computational limitation, the simulation can only efficiently run with 20 OD pairs, which inadequate for simulate in large datasets. In the next section, we will introduce a heuristic approach designed to manage this issue. This method aims to increase the simulation's capacity to handle larger datasets while minimally impacting the efficiency of the matching algorithm.

7.8 Heuristic approach to reducing computational costs.

In this section, we focus on an innovative approach aimed enhancing our simulation's capacity to manage larger datasets, whilst still maintaining an efficient computation time. Recognizing the limitations, explained in previous section, imposed by the exponential increase in computational time with an increasing number of OD pairs, we propose the application of a clustering method which a part of machine learning method. There are two common types of clustering methods: K-means clustering and Hierarchical clustering.

K-means Clustering: K-means clustering is a popular partitioning-based clustering algorithm. It aims to divide a dataset into a number K clusters, where K is a predefined parameter by user. The algorithm starts by randomly initializing K cluster centroids. It then iteratively assigns each data point to the nearest centroid and updates the centroids based on the mean of the data points assigned to each cluster. This process continues until convergence, where there are only small changes in centroids, or a maximum number of iterations is reached. K-means clustering is efficient and works well when the clusters are spherical and have similar sizes.

Hierarchical Clustering: Hierarchical clustering is a technique that builds a hierarchy of clusters, also known as a dendrogram. It does not require a pre-specified number of clusters and can be agglomerative or divisive. Agglomerative hierarchical clustering starts with each data point as a separate cluster and then iteratively merges the closest pairs of clusters based on a chosen distance measure, such as Euclidean distance. Divisive hierarchical clustering starts with the entire dataset as a single cluster and then recursively splits it into smaller clusters until each data point is in its own cluster. The resulting dendrogram can be cut at a desired similarity threshold to obtain the final clusters. Hierarchical clustering is more flexible but can be computationally expensive for large datasets.

To mitigate the computational cost based on increasing number of Origin-Destination (OD) pairs, we propose a clustering approach where OD pairs are grouped, and our matching algorithm is subsequently applied independently within each cluster. Based on our existing computational time data, running the matching algorithm for a total of 60 OD pairs per iteration takes approximately 5,552 seconds. If, however, we cluster this into groups of 20 OD pairs, the computational time per iteration reduces dramatically to around 4.66 seconds for each group, resulting in a total of approximately 13.98 seconds. Consequently, we aim to cap the maximum number of OD pairs per cluster at 20, as a strategy to control computational demand.

K-means clustering is not ideal for this application as it necessitates defining centroids, which is challenging in this context, and pre-specifying the number of clusters (K), which we cannot ascertain upfront. In contrast, our primary

requirement is to limit the maximum members, or the number of OD pairs, within each cluster rather than predetermine the number of clusters. Given these constraints, hierarchical clustering is selected as the more suitable method for our scenario.

7.8.1 Similarity measures

The definition of similarity measures is vital in shaping the clusters within our model. In hierarchical clustering, a cluster is defined as a group of members that are similar or have a small value of similarity measures. As our focus is on clustering OD pairs, we first need to define the parameters that establish similarity between these pairs. Figure 7.11 illustrates five sample OD pairs from the Sioux Falls dataset. In this figure, rectangles represent origin nodes, triangles represent destination nodes, and each colour corresponds to a different OD pair. The figure highlights that OD pairs in the Sioux Falls network are relatively sparse and widely distributed, which makes it difficult to visually discern appropriate clusters. This illustrates the need for a more systematic definition of similarity measures.

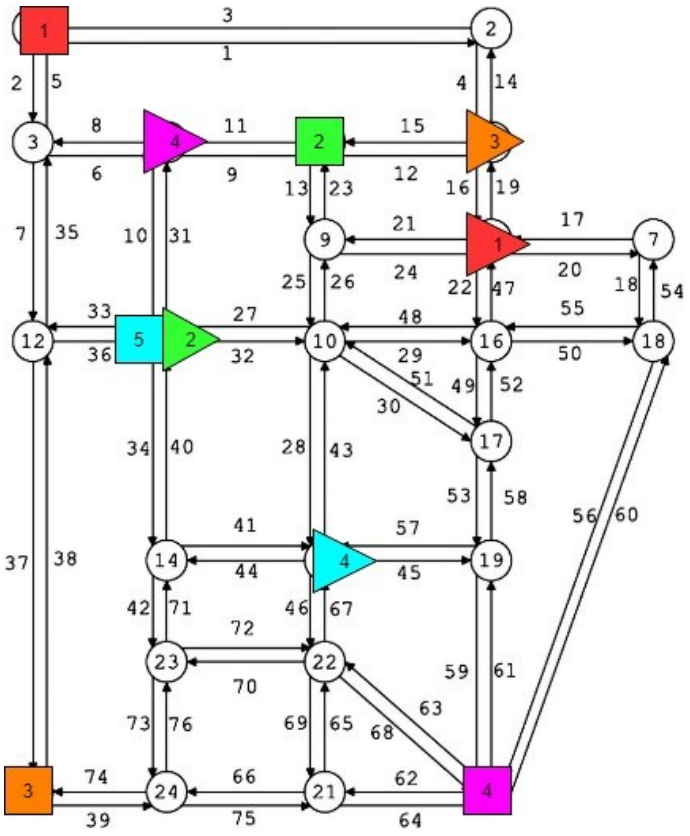

Figure 7.11 Sioux Fall network with mixed OD demands
(rectangles = origins, triangles = destinations, colours = different OD pairs)

To address this, we propose two similarity measures: OD distance and Saved distance. The first similarity measure we consider is the OD distance. This can be computed as the sum of Euclidean distances between corresponding origin nodes and between destination nodes. The distance between two nodes is calculated as the shortest path between them, using Dijkstra's algorithm. This measure is under the assumption that the driver's OD pair should not be substantially distanced from the passenger's OD pair. However, this similarity measure might lead to clusters of OD pairs that are proximal and have comparable journey lengths, but it could exclude scenarios where the passenger's OD pair is located en route the driver's planned trajectory. For instance, in the Sioux Falls network, if a driver's OD pair is 1→20 and a passenger's OD pair is

9→10, although the OD distances between the two users are considerable, the shortest routes for both users overlap. Thus, the driver incurs no additional distance to pick up this passenger. However, based solely on the similarity measure of OD distances, these pairs might not be grouped within the same cluster.

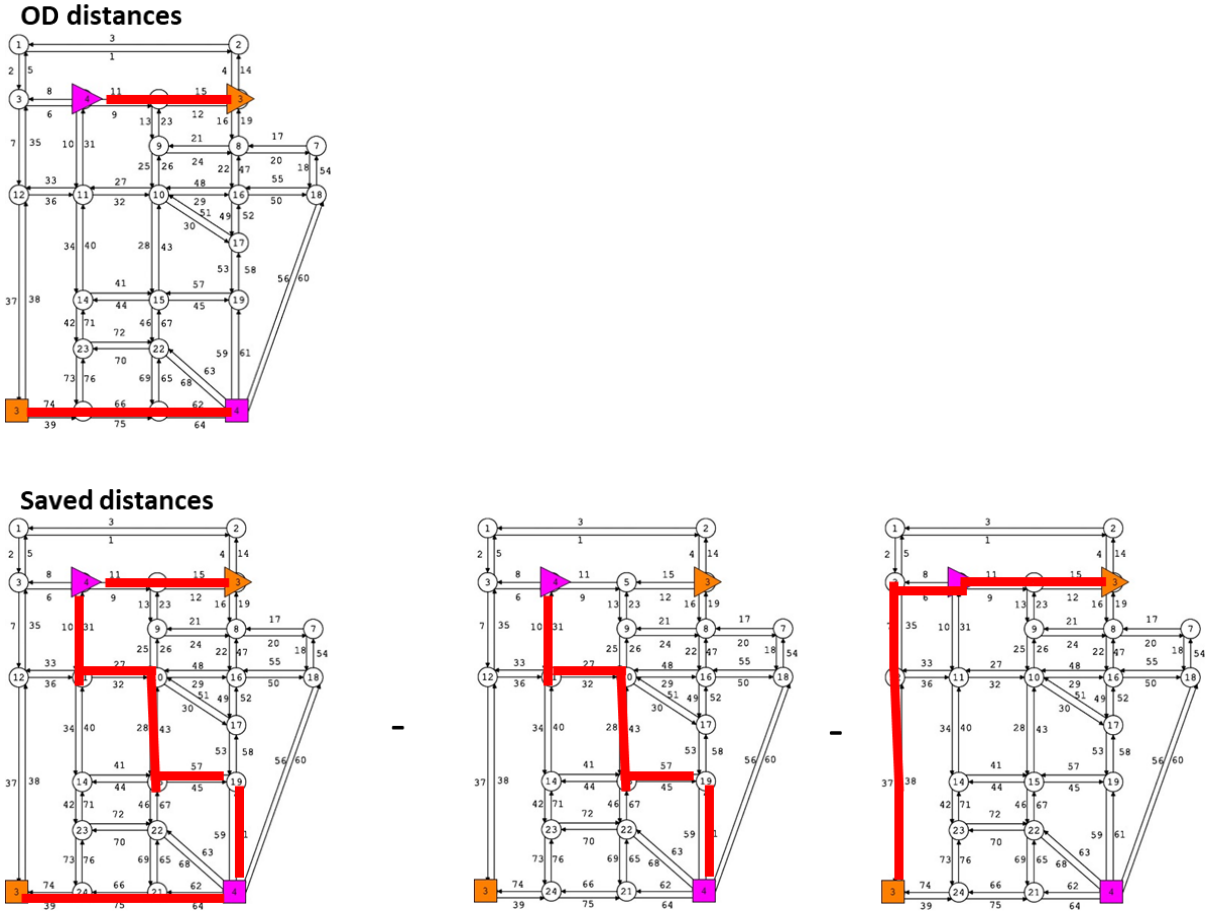

Figure 7.12 Similarity measures; OD and Saved Distances

To address such cases, we consider the Saved Distance as the second measure; This measure quantifies the total distance that both ridesharing users can save if they utilize the ridesharing service, compared to if they travelled independently. Computationally, this measure is more demanding as it involves multiple

shortest path calculations. For example, to compute the Saved Distance for OD pairs 3 and 4, one begins by finding the shortest ridesharing distance, subtracting the original shortest non-ridesharing distances for both OD pairs. This measure captures the total potential distance savings offered by ridesharing compared to non-ridesharing modes. More negative values indicate a more suitable matching pair that should be clustered together based on this measure. Although computationally intensive, this similarity measure aligns directly with the objective function of our matching algorithm, which is to maximize Vehicle Kilometres Travelled Saved (VKTS).

7.8.2 Clustering Performance

Based on these two-similarity measure that have been proposed in the previous section, we then try to compute the performance of using the clustering analysis that group ridesharing user's ODs into some clusters and perform the matching algorithm separately. We start the evaluation by used all available OD demands of 528 ODs, in the Sioux fall network. Without clustering, we cannot perform the matching algorithm for this high number of ODs. We then start the evaluation by, using the clustering analysis with using OD distance and Saved distance as a similarity measures. We cluster ODs until we have a satisfy number of the maximum ODs member in any cluster. The performance test start by the maximum of 0 member which mean the matching algorithm unable to match any ODs. This means the objective function which is the VKTS yield a zero value. Then we test again with the maximum ODs member in any cluster at 10, 20, 30, 40 and 50. Note that with a maximum ODs member in any cluster at 50 members, it will require up to 2,254 seconds to run the matching algorithm for

each cluster. We are unable to perform the test in the bigger cluster size due to the computational time is too extremely high.

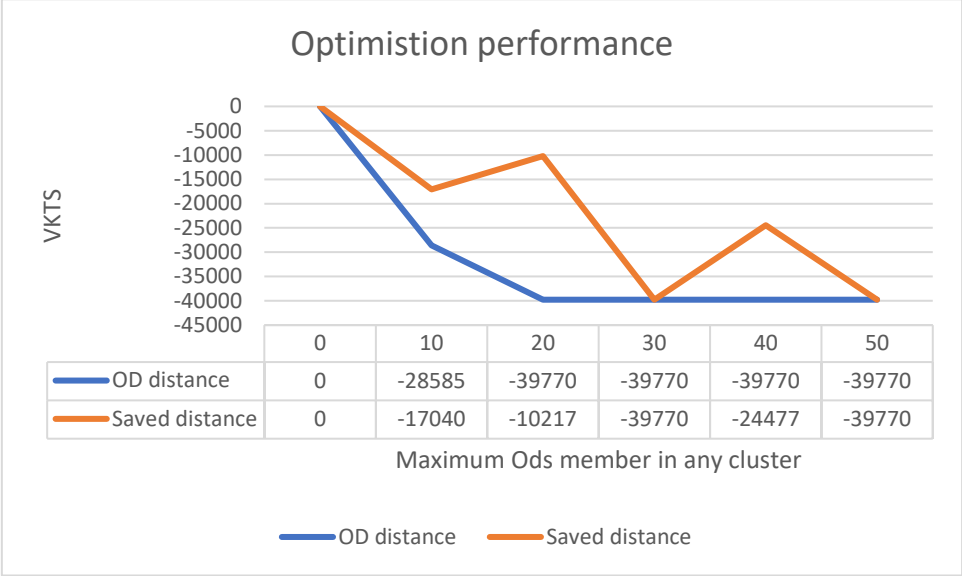

Figure 7.13 The performance in different size of cluster

When the maximum ODs member in any cluster increases, the total number of clusters is reduced. From a theoretical perspective, larger clusters or fewer clusters should yield superior objective function values, equivalent to a scenario that running the algorithm without any clustering. In this analysis, we aim to identify the optimal cluster size, understanding that larger sizes might offer improved results but at the cost of increased computational time, while smaller sizes might lead to faster computation but a potential compromise in performance. Another crucial aspect of this analysis is to determine which similarity measure, OD distance or Saved Distance, results in better performance. The formation of clusters is fundamentally shaped by the chosen similarity measure. Therefore, given a specific cluster size, a more appropriate similarity measure should enhance performance, as reflected in higher Vehicle

Kilometres Travelled Saved (VKTS) values - the objective function of our matching algorithm.

Our results indicate that Saved Distance consistently outperforms OD distance across all runs. The VKTS values, representing distance savings, are more negative when using Saved Distance as the similarity measure, implying superior performance. It's important to note that VKTS is computed as the sum of VKTS across all clusters. Regarding the optimum cluster size, the data reveals a sharp drop in VKTS values when the maximum cluster size increases from 0 to 10 ODs. However, the optimal performance is achieved at a cluster size of 20, beyond which there is no noticeable improvement in performance. Considering the additional computational time required for larger clusters, we conclude that a cluster size of 20 OD pairs is optimal. Therefore, Saved Distance will be used as the similarity measure, and the maximum number of OD pairs in any cluster will be capped at 20. With a heuristic approach using clustering method with this setting, it allows our model to run in a larger scale of network or demands, while scarify only small performance.

Before progressing to more practical simulations, it was necessary to conduct a stress test to identify the computational bottleneck of the proposed simulation. The test revealed that the matching algorithm posed a significant challenge, particularly as the number of OD pairs for ridesharing users increased. This difficulty arose from the exponential growth in computation time associated with generating all possible matching combinations. To address this issue, a heuristic based on hierarchical clustering was introduced, allowing OD pairs with high similarity to be grouped and matched independently within each

cluster. This approach substantially reduced the calculation time per iteration. A comparison of similarity measures between OD distances and Saved distances indicated that the latter yielded superior performance with respect to the matching algorithm's objective function. Based on these tests, the optimum maximum cluster size was selected as 20 OD pairs, which provided a balance between reducing computation time and maintaining solution quality. With the integration of this heuristic, the simulation framework is sufficiently robust to be applied to more complex dataset and practical analysis in the following chapters.

7.9 Discussion

In conclusion, this chapter presented a numerical example using our proposed simulation framework, utilizing the Sioux Falls network, a popular testing network for macroscopic traffic simulation. The simulation demonstrated the ability to calculate converged results, which are crucial for reliable and stable solutions in the study area. Converged solutions in modal splits provide valuable insights for transportation planning and decision-making, enabling a long-term understanding of traveller behaviour over the network.

Furthermore, we conducted an analysis of the impact of matching algorithms, vacant taxi models, and fare adjustments on system performance, considering factors such as PCU reduction, in-vehicle time improvement, and network cost reduction. Sensitivity analysis of fares in the taxi-sharing system provided insights into the trade-off between profitability and passenger demand. We observed how varying fare rates influenced matching rates, modal splits, and revenue, shedding light on the competitive dynamics between the taxi-sharing system and other transportation modes. These simulations demonstrate the

capability of our proposed simulation framework to investigate the impacts of ridesharing systems and address topics of interest in the field of shared mobility systems, such as the impact on the road network, pricing schemes, optimal fleet size, or impact on the public transport system. These findings lay the groundwork for the subsequent chapter, where we will further explore these topics using our proposed simulation framework.

Although Sioux Falls is a widely used benchmark network, its realism is limited when compared to large urban settings. With only 24 nodes and 76 links, the dataset is highly suitable for testing algorithms and validating methodological soundness, but it does not capture the density, multimodal integration, and congestion effects that are characteristic of real metropolitan areas. Recent studies have noted that the Sioux Falls network exhibits limited path diversity and only weak alignment with empirical routing behaviour when assessed against higher-order trajectory data (Hoogendoorn-Lanser, Nes & Bovy, 2005). Similarly, comparative work using network-in-network approaches demonstrates that while Sioux Falls serves well for method development, it fails to represent the spatial-temporal and topological complexities of larger networks (Lin & Ban, 2017) In contrast, a city such as London features orders of magnitude more OD pairs, stronger demand intensity, and much more heterogeneous travel patterns, implying that clustering and matching behaviours may differ substantially in practice. As such, the present results should be interpreted as illustrative rather than representative of real-world systems, and future applications should extend the framework to larger, more complex datasets, for example Chicago-Regional, Munich, Sydney or Philadelphia from the (Transportation Networks for Research Core Team, 2023) — to better assess robustness and scalability.

This chapter successfully validated the mathematical and numerical coherence of the integrated simulation framework. The analysis demonstrated clear convergence in modal splits for both shared mobility systems (ridesharing and taxi-sharing) and utilized sensitivity analysis to confirm plausible behavioral responses to cost variations (ridesharing fare, VoT, bus fare). Furthermore, the stress test revealed the exponential computational bottleneck posed by generating matching possibilities, leading to the necessary implementation of a hierarchical clustering heuristic using Saved Distance measures, thereby addressing the tractability requirements inherent in the framework's design. This robust validation and optimization confirm that the model is now computationally prepared to meet the third objective of the thesis: evaluating its capability against complex policy scenarios in Chapter 8. With the integration of this heuristic approach, the simulation framework is now prepared to be applied in more complex networks and practical analysis. The next chapter will demonstrate the application of the simulation in different scenarios to outline the contributions of our proposed model to the main topics of study in shared mobility systems.

CHAPTER 8 IMPACT OF SHARED MOBILITY SYSTEMS

This chapter serves as the practical evaluation stage of the thesis, employing the comprehensively developed and validated simulation framework (Chapters 6 and 7) to address the timely research questions identified in the literature review (Chapter 2). The primary purpose is to demonstrate the model's capability as an advanced, comprehensive policy tool by simulating seven key areas, including network impact, algorithm objective comparison, HOV lane efficiency, and public transport interaction. Beginning with a baseline scenario (solo drivers only), the chapter proceeds to show how the integrated, macroscopic approach captures complex dynamic effects, such as induced demand, that isolated analytical models overlook.

A deeper understanding of these areas necessitates the formation of more comprehensive models. Consequently, our research question emerges, focusing on the creation of a simulation framework that is both realistic and computationally efficient. In the next section, we will employ our proposed macroscopic traffic simulation to a more realistic and practical scenario. This will demonstrate the practical utility and necessity of our model. We aim to confirm the supposition that our model is essential for generating outcomes that more closely resemble reality.

8.1 Baseline Road Network (No Shared Mobility)

Before applying the shared mobility framework introduced in Chapter 6, it is necessary to establish a baseline scenario for comparison. In this base case, the Sioux Falls network is simulated under the assumption that only solo drivers are present. Ridesharing and taxi-sharing modes are excluded, meaning that no passenger–driver matching or clustering procedures are applied at this stage. This setup allows us to assess the performance of the traffic assignment component of the framework in isolation, and provides a reference point for subsequent analysis once shared mobility systems are introduced.

In this baseline run, all 528 OD pairs of the Sioux Falls dataset are included. Route choice for each OD is restricted to the five K-shortest paths, generated using Yen's algorithm, and traffic is assigned through the stochastic day-to-day and within-day assignment process described in Chapter 6. The simulation is executed until convergence, defined by stable modal and route flows across iterations. To validate this simplified framework, the resulting equilibrium link flows are compared against the widely reported *best-known solution* for Sioux Falls, obtained by minimizing the Average Excess Cost (normalized gap) to 3.9E-15 (Transportation Networks for Research Core Team, 2023). Figure 8.1 shows the benchmark link flows, while Figure 8.2 depicts the equilibrium link flows generated by the baseline simulation.

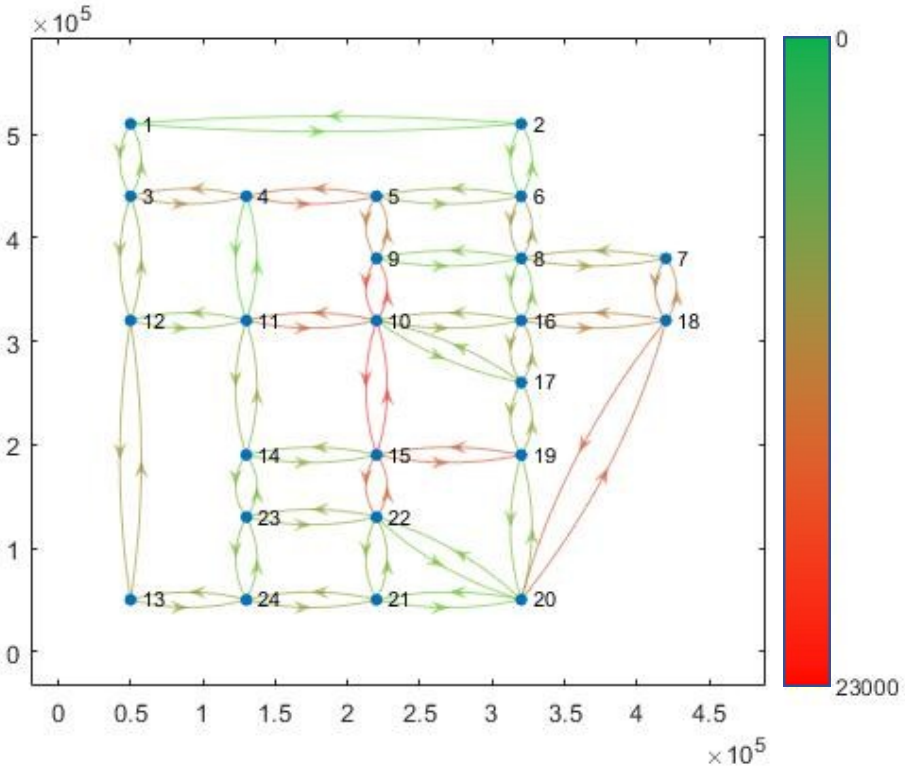

Figure 8.1 Network Link flow (the best well knowns)

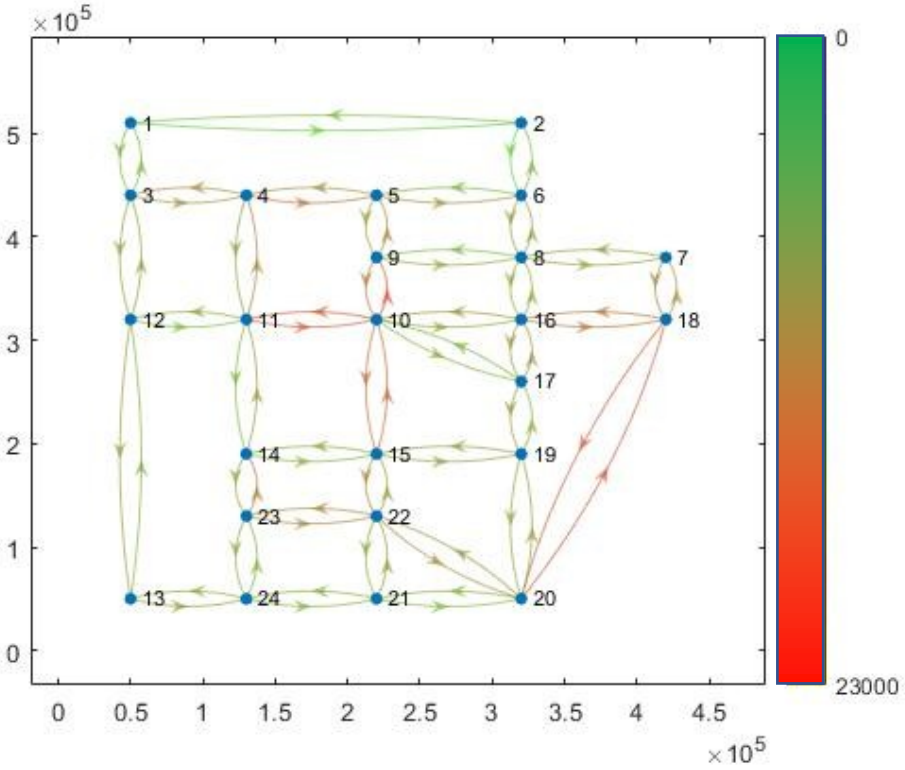

Figure 8.2 Network Link flow (based scenario - no ridesharing)

The link flows in Figure 8.2 are at equilibrium. The colour gradient from green to red denotes congestion, with green indicating low traffic (lower link flow) and red indicating high traffic (higher link flow). The highest traffic volume is observed on Link 26, connecting node 10 to node 9, with a total of 21,938 vehicles. Areas of heavy congestion appear predominantly between nodes 9, 10, 11, and 15 at the centre of the Sioux Falls network, as well as between nodes 18 and 20 on the eastern side of the network, which indicates a shortcut route. The total number of vehicles in this simulation is slightly lower than in the benchmark because some travellers are allocated to public transport in our framework, which is not considered in the benchmark data. Despite these differences, both models highlight similar congestion hot-spots, underscoring the validity of the baseline scenario as a point of reference.

Although the average absolute error between the two solutions is about 22%, this difference arises primarily from the methodological choice to restrict route choice to a fixed set of K-shortest paths per OD, rather than enumerating all feasible paths and minimizing the average excess cost until full convergence. As noted in the literature, such path-set restrictions reduce computational burden but can introduce non-negligible deviations in link flows (Mousavi, Yazdiani & Shafahi, 2022; Nikolic & Teodorovic, 2021) Increasing the number of candidate paths would reduce the error, but at the expense of substantially higher computation times, which becomes impractical for larger networks.

Importantly, such levels of deviation are not unusual in practice. The National Guidelines for Transport System Management (Australia) state that for hourly flows on individual links, a tolerance of ±20% error is considered acceptable,

while aggregated measures across screenlines. which capture broader congestion patterns are expected to be within ±10%. Similarly, the FHWA Traffic Analysis Toolbox emphasizes that calibration should prioritise reproducing congestion behaviour and bottleneck locations over perfect link-flow matching, and acknowledges that tolerances must balance against computational feasibility and data availability (Dowling, Skabardonis & Alexiadis, 2004).

8.2 Impact on the road network.

Having established the baseline scenario in Section 8.1, we now begin exploring the effect of introducing shared mobility. At this stage, however, we do not apply the full framework described in Chapter 6, which allowed for Single Driver – Multiple Passenger matching yet. Instead, we deliberately simplify the model to examine the most basic configuration: single driver – single passenger ridesharing.

This simplification means that additional passengers are neglected, reducing the number of feasible matches compared to the full framework. In practical terms, only one matching pattern remains applicable (Pattern #4 in Figure 3.2) and the matching possibilities are generated solely from driver–passenger OD pair combinations. The objective function is therefore adjusted accordingly (Equation 8-1).

Objective function

$$\max \sum_{i,j} \sum_{k,l} \sum_{p} X_{ijk_1l_1} VKTS_{ijk_1l_1}$$

Subject to

$$\sum_{(i,j)\in D} X_{ijk_1l_1} \leq F_{(k,l)} \; \forall (k,l) \in P$$

$$\sum_{(k,l)\in P} X_{ijk_1l_1} \leq F_{(i,j)} \; \forall (i,j) \in D$$

$$X_{ijk_1l_1} \geq 0 \; \forall (i,j) \in D \, , \; \forall (k,l) \in P$$

Equation 8-1 Optimisation problem for Single Driver - Single Passenger matching

Appendix F presents the results from the first iteration (Day 1) of the ridesharing matching algorithm. These outputs show the OD nodes of matched users, their generalized travel costs, the corresponding matching pattern, and the total number of matches. At this stage, the algorithm identifies 21,488 driver–passenger pairs, implying an equivalent reduction in vehicles. However, this figure is an overestimation, as it does not account for the day-to-day traffic dynamics that provide feedback between congestion levels and travellers' mode choices.This phenomenon, driven by induced demand not captured by standalone matching systems (as commonly found in literature), underscores the necessity of our integrated macroscopic approach. Without this iterative adjustment, the results cannot represent realistic network conditions. Only through the full day-to-day simulation, which continues until convergence, can a credible estimate of ridesharing uptake and its network effects be obtained.

For comparison, when the framework is extended to the single driver–multiple passenger scheme (as originally proposed in Chapter 6), the matching opportunities become more constrained. Under this configuration, the model yields 10,682 drivers and 21,364 passengers. This reduction compared to the single-passenger case highlights how modelling assumptions influence the

magnitude of potential benefits, and reinforces that the initial Day-1 figures should be interpreted only as an upper bound until the full day-to-day simulation is applied.

Without the incorporation of a route choice model and a day-to-day traffic model, the number of vehicles potentially reduced in the network due to the ridesharing system does not truly reflect real-world circumstances. In reality, a reduction in the number of vehicles on the road network or other changes could lessen the cost of travel due to reduced congestion. This reduced cost may trigger a phenomenon known as 'induced demand', enticing more travellers to purchase or utilize more vehicles due to the lower associated cost of driving. Figure 8.3 shows how the number of matched drivers evolves throughout the simulation period.

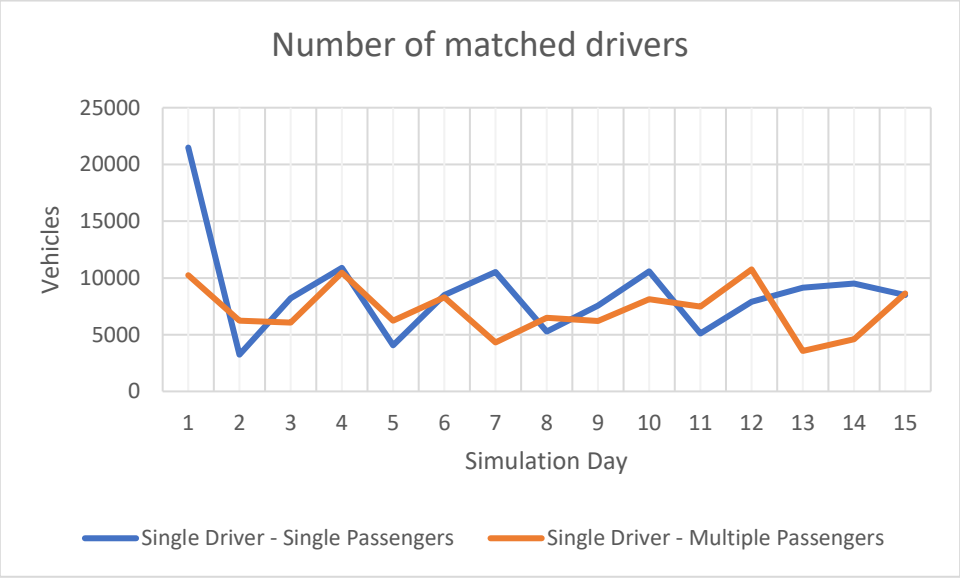

Figure 8.3 Number of matched drivers in the ridesharing system

The number of matched drivers showed a significant reduction from the first day of the simulation, a consequence of travellers' mode choice behaviour. Following the decline, the number of matched drivers for both single-driver-single-passenger and single-driver-multiple-passengers schemes fluctuated between 5,000 to 10,000 vehicles. This confirms that the number of matched travellers can be overestimated without a simulation framework in place, as it fails to account for induced demand. In practical scenarios, the number of matched travellers tends to be lower than the value generated by the matching algorithm. In this thesis, the number of matched drivers for the ridesharing system for both matching schemes approximates around 8,500 vehicles.

Even though the number of actual ridesharing users is less than the number derived solely from the matching algorithm, the primary benefit of the ridesharing system to the road network—namely the reduction in the number of vehicles—has been confirmed. The succeeding figures will depict the link flows at the 15th day of the simulation.

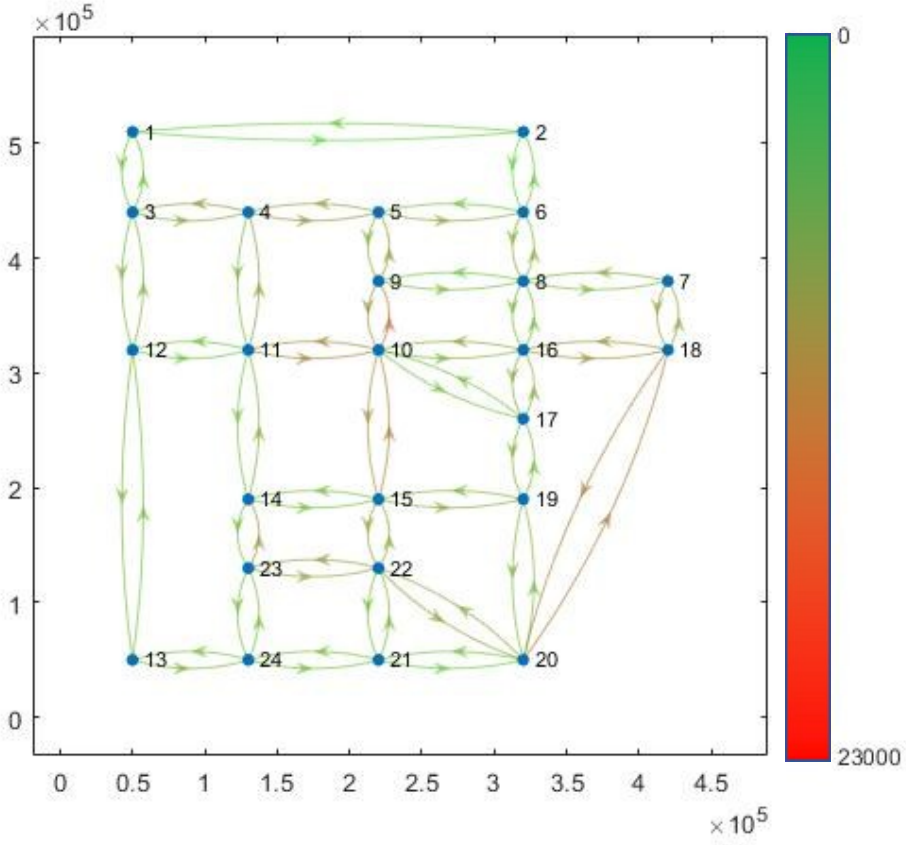

Figure 8.4 Network Link flow (single driver-single passenger)

Comparing the link flow with the scenario where this is no ridesharing system (Figure 8.2), the links flow and traffic congestion reduced dramatically (Figure 8.4). The highest traffic volume is still located on Link 26 , but now with a total of 21,370 vehicles. For overall network, the links flow is reduced on average by 17.22 percent. This proves that that shared mobility systems indeed impact the road network by reducing traffic congestion, achieved through a decrease in vehicular demand for the network.

8.3 Development of matching algorithm

Given the numerical evidence demonstrated in the previous section, it's impossible to deny that the matching algorithm plays a vital role in the successful operation of shared mobility systems. Different configurations within the matching algorithm can yield different outcomes. For instance, the objective function of our current optimization in the matching algorithm is to maximize the Vehicle Kilometres Travelled Saved (VKTS). This objective primarily benefits the road network, which might not necessarily align with real-life scenarios. Shared mobility companies such as Uber, being private entities, aim to maximize their profit. Consequently, the optimisation problem (Equation 8-2) could be adjusted from maximizing VKTS to maximizing profit $F^{d}_{d,p1,p2}$. Without a simulation framework, transportation planners lack a tool to thoroughly investigate the system's impact.

$$\text{Objective function} \quad \max \sum_{i,j} \sum_{k,l} \sum_{p} X_{ijk_1l_1k_2\,l_2p} F^{d}_{d,p1,p2}$$

$$\text{Subject to} \quad \sum_{(i,j)\in D} X_{ijk_1l_1k_2\,l_2p} \leq F_{(k,l)}\ \forall (k,l)\ \in P$$

$$\sum_{(k,l)\in P} X_{ijk_1l_1k_2\,l_2p} \leq F_{(i,j)}\ \forall (i,j)\ \in D$$

$$X_{ijk_1l_1k_2\,l_2p}\ \geq 0\ \forall (i,j) \in D\ ,\ \forall (k,l) \in P$$

Equation 8-2 Profit maximisation problem

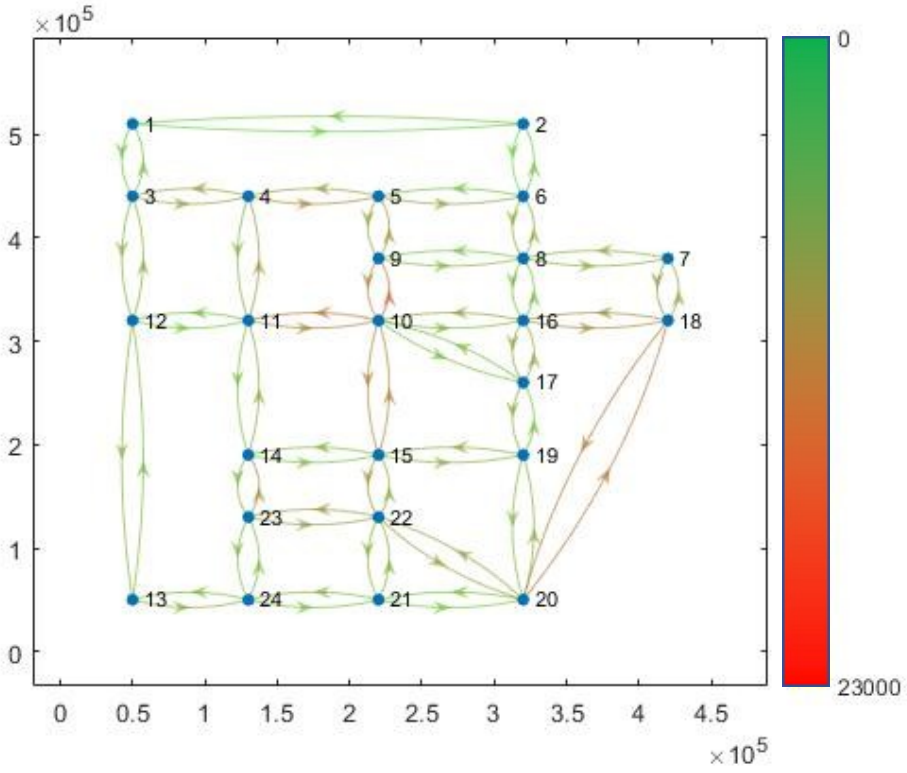

Figure 8.5 Network Link flow (profit maximization)

In this section, we elucidate the implications of modifying the configuration of the matching algorithm, specifically by adjusting the objective function to prioritize profit maximization. In this scenario, the matching algorithm aims to pair ridesharing passengers who offer the highest fares before allocating vehicles to other passengers. Figure 8.5 presents the equilibrium link flows on the 15th simulation day under this profit-driven objective. The results show that, even under a profit-maximization objective, there remains a measurable network benefit: overall link flows decrease by 15.9% compared with the baseline, slightly less than the 17.22% reduction achieved when maximizing VKTS. The most congested link (from node 10 to node 9) carries 21,440 vehicles, which is still below the baseline value but marginally higher than in the VKTS-maximization scenario. In other words, congestion relief persists, but at a

reduced magnitude compared with the socially optimal case of maximizing VKTS described in Section 8.2.

From a public transportation perspective, shifting the objective to profit maximization could alter mode dynamics. A profit-driven matching scheme tends to prioritise high-paying OD pairs, which may overlap less with bus routes and schedules. This can reduce the competitive pressure on public transport, since some lower-fare passengers are left unserved by ridesharing and continue to rely on buses. Conversely, if surge pricing is introduced, passengers with higher willingness-to-pay may switch away from buses during peak times, potentially eroding bus ridership. The overall impact depends on the fare structure and the relative attractiveness of ridesharing versus public transport, and is therefore more complex than in the VKTS-maximization case.

8.4 Environmental Impact

In previous sections, we highlighted the significance of integrating the matching algorithm with route choice models to be more accurately simulate the behaviour of travellers' mode and route choices. Through this integration, the simulation yields outputs that are more reflective of real-world conditions as opposed to models developed for isolated purposes shown in the literature review. Most studies related to the environmental impact of shared mobility system were based on empirical data. In contrast, out model is an analytical model which is a different approach. In a macroscopic traffic simulation model, vehicle emissions can be estimated by incorporating emission factors and traffic flow data.

Our proposed simulation framework is capable of generate several factors that can be employed to calculate vehicle emissions at a macroscopic level. The first factor, link flow, which we have calculated previously, represents the number of vehicles traversing each link in the network. A model based solely on the matching algorithm is unable to calculate link flows; its only output is the number of matched travellers, which does not account for the duration or distance that a vehicle remains in the network. The next factor, vehicle speed, can be computed as the average speed of vehicles on each link. Typically, higher speeds contribute to greater emissions when compared to lower speeds, the latter of which often leads to more efficient driving and consequently, reduced emissions. Another factor is Vehicle Kilometres Travelled (VKT), calculated by summing the product of link flow and link length.

In this section, we demonstrate the outputs of the proposed simulation using the aforementioned factors. As illustrated in Figure 8.4, the total link flow, or the total volume of vehicles on all links in the network at equilibrium with the inclusion of a ridesharing system is 836,598 vehicles. Comparing with, in the absence of a ridesharing system, the total link flow of the network is 1,014,668 vehicles. Thus, it can be inferred that the introduction of the ridesharing system decreases the total link flow of the network by approximately 17.5 percent.

Table 8.1 Example of Sioux Falls link's properties

Link number	Origin Node	Destination node	Link Capacity	Length	Freeflow traveltime	Alpha	Beta	Speed limit	Toll	Type
1	1	2	25900.2006	6	6	0.15	4	0	0	1
2	1	3	23403.4732	4	4	0.15	4	0	0	1
3	2	1	25900.2006	6	6	0.15	4	0	0	1
4	2	6	4958.18093	5	5	0.15	4	0	0	1
5	3	1	23403.4732	4	4	0.15	4	0	0	1
6	3	4	17110.5237	4	4	0.15	4	0	0	1
7	3	12	23403.4732	4	4	0.15	4	0	0	1
8	4	3	17110.5237	4	4	0.15	4	0	0	1
9	4	5	17782.7941	2	2	0.15	4	0	0	1
10	4	11	4908.82673	6	6	0.15	4	0	0	1

Next, we discuss the factor of vehicle speed. In the macroscopic simulation context, vehicle speed is represented by the average link speed, which is derived from dividing the link's travel time by its length. The link travel time is based on the Volume Delay Function (VDF) presented in Chapter 5. This approach, which dynamically generates vehicle speed and flow data based on current traffic equilibrium, contrasts sharply with older methodologies, such as those relying solely on static or predefined external emission factors (e.g., COPERT models cited in existing literature), thereby enabling a more robust investigation of real-time environmental externalities. The Sioux Falls network dataset, which is widely used in many transportation studies including our thesis is embedded with link specification data for BRP function. The table 8.1 example of the Sioux Fall link’s properties that is used to calculates the average vehicle speed in the link. Next, we perform a comparison in the average speed of each link in the network.

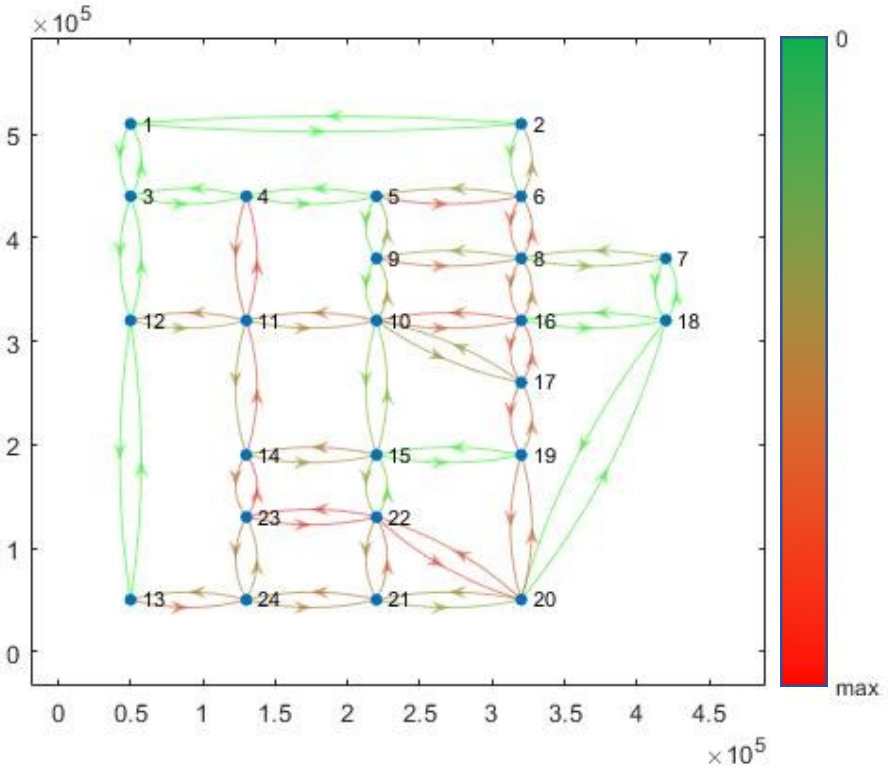

Figure 8.6 Network Average speed (no ridesharing)

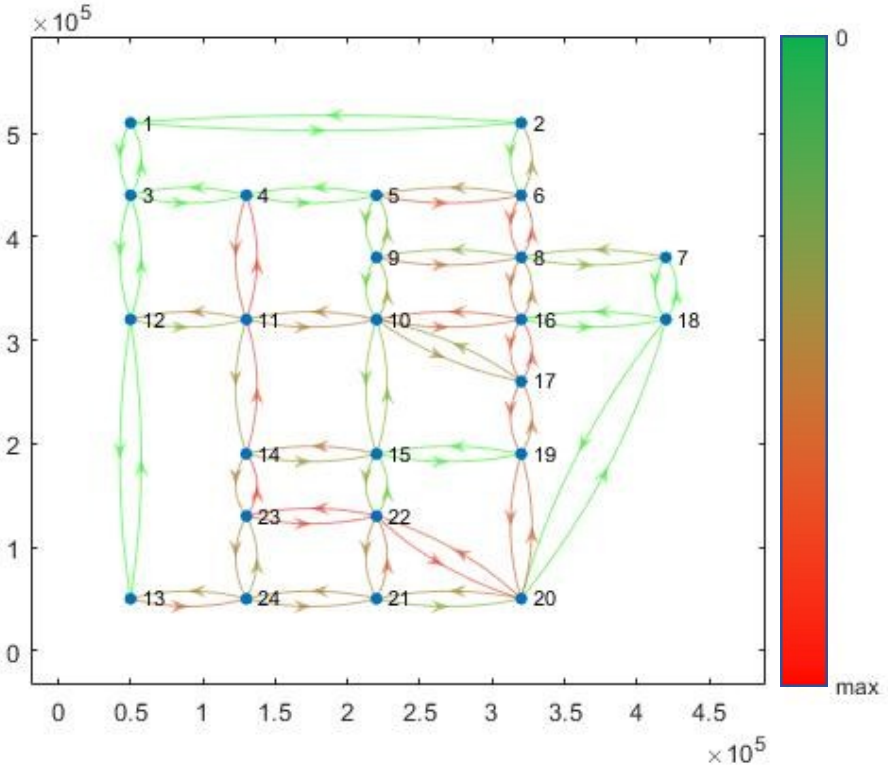

Figure 8.7 Network Average speed (with ridesharing)

Our current network configuration, accompanied by a total demand of 528 ODs, is represented in Figure 8.6 (no ridesharing), showcasing the average vehicle speed for each network link while red area indicates links with low speed and green area indicates a higher speed. Interestingly, despite the high link flow in the network's centre between links 9-10, and the shortcut road between links 18-20 - as previously observed in Figure 8.2 - the average speed does not proportionately increase, due to road characteristics. Areas demonstrating slower travel speeds are dispersed across the network, with the slowest average speed found at Link 74, connecting nodes 23 to 14, registering an average speed of merely 4.24 miles per hour. The average speed for this scenario is approximately 32.32 miles per hour.

In contrast, the scenario incorporating ridesharing into the network is depicted in Figure 8.7. Visually it may not reveal a substantial difference between the two scenarios in terms of an increase in the average speed. However, the numerical data tell a different story: the introduction of a ridesharing system elevates the network's average speed to 33.66 miles per hour, marking a 4.14 percent improvement. The link maintaining the lowest average speed remains Link 74, though its average speed now improves to 4.77 miles per hour (reflecting a 12.5 percent increase). The most significant improvement can be observed at Link 42, extending from Node 14 to Node 23, with its average speed surging from 18.02 miles per hour to 25.47 miles per hour - a substantial increase of 41.3 percent.

The calculation for Vehicle Kilometres Travelled (VKT) can be calculated from the link flow data obtained from section 8.1, which is then multiplied by the respective link length, as provided in the dataset (refer to Table 8.2). In the baseline scenario, wherein no ridesharing system is incorporated in the network, the total Vehicle Kilometres Travelled equates to 3,550,881 vehicle-miles. With the introduction of a ridesharing system, the VKT is expected to diminish due to the reduced necessity for individual vehicles within the network.

However, the VKT under the ridesharing scenario can be subdivided based on the objective function employed in the matching algorithm. When the matching algorithm is maximizing revenue, the VKT amounts to 3,464,557 vehicle-miles. On the other hand, when the objective function of the matching algorithm is change to maximizing Vehicle Kilometres Travelled Saved (VKTS) - thereby aiming to minimize VKT as much as possible - the VKT drops to 3,390,918 vehicle-miles, which is further reduced compared with the other objective function.

Our findings found that the integration of a ridesharing system into the network can yield notable environmental benefits, such as reduced link flow and improved average vehicle speed, leading to lower VKTs. These improvements can be further optimized by tailoring the objective function of the matching algorithm. In essence, this simulation framework can serve as an instrumental tool in evaluating and fine-tuning ridesharing systems for maximum environmental sustainability.

Environmental effects of ridesharing remain empirical-based contested in the literature. (Cai et al., 2019) found measurable fuel and CO_2 reductions from taxi ride-sharing in Beijing, while (Wang, Zixuan et al., 2022) observed air quality improvements in Shenzhen. Conversely, systematic reviews (Sheldon & Dua, 2024) and empirical U.S. studies (Clewlow & Mishra, 2017) show that ride-hailing often increases overall VMT and GHG emissions when substituting for public transport or active modes.

Our proposed simulation framework offers a more accessible method for generating emission-related factors for analysing the environmental impacts of shared mobility systems. This model allows an adjustment to a wide range of parameters such as vehicle ownership rates and maximum speed limits, thereby offering deeper insights into the effects of ridesharing systems on the environment. Unlike empirical models that rely on fixed, pre-existing data, our analytical framework offers greater flexibility, facilitating comprehensive investigations of potential impacts based on variable inputs. Key factors such as link flow, vehicle speed, and Vehicle Kilometres Travelled (VKT) can be derived from the simulation, showing their interaction and effects on vehicular emissions.

8.5 Efficiency of High-Occupancy Vehicle lanes

High-Occupancy Vehicle (HOV) lanes which sometimes called carpool lanes. Is a highway lane that reserved for vehicles with carrying more than one passenger ad restricted other vehicles from the lane. Make this type of lane exclusive for vehicle with a higher occupancy. The primary reason for these lanes is to reduce

traffic congestion and promote vehicle occupancy in the city. Ridesharing vehicles typically qualify to use HOV lanes as they usually meet the passenger requirements. This allows ridesharing operators such as Uber and Lyft take advantage of such infrastructure in many urban areas to facilitate their passengers reach their destinations more quickly.

In terms of transportation modelling, our computational simulation framework incorporates the ability to modify the properties of links to include the integration of High-Occupancy Vehicle (HOV) lanes. This augmentation lends a more comprehensive perspective on traffic dynamics, particularly regarding the effect of ridesharing vehicles on traffic flow and congestion. There are two primary methodologies for integrating HOV lanes within our framework:

The first approach involves the addition of discrete, supplementary links that are exclusively utilized by ridesharing vehicles. This method serves to depict the influence of dedicated HOV routes on the broader transportation network, offering a more granular understanding of how ridesharing services might impact traffic distribution and route selection.

The second methodology reserves a specific lane within an existing link solely for the use of ridesharing systems. By earmarking a segment of the pre-existing transportation infrastructure for ridesharing vehicles, this strategy facilitates the study of HOV lanes within a multi-modal transportation context, elucidating their effect on overall traffic flow and the dynamics of lane usage.

In this empirical investigation, we proceed with the implementation of our second methodological strategy, wherein one lane from the aggregate number of lanes on the highway is explicitly allocated for a ridesharing system. This approach is favoured as it is more prevalent and cost-effective for highway agencies. Implementing this adjustment within our simulation framework involves two steps. Initially, we diminish the capacity of the link that is intended to accommodate the High Occupancy Vehicle (HOV) lanes. Subsequently, we create an additional link with identical characteristics and equip it with the same volume of capacity that was previously reduced. To prohibit solo drivers from utilizing the HOV lanes within the simulation, we strategically remove any route containing the additional HOV link when calculating the K-shortest path for solo drivers. Notably, this measure does not impact ridesharing vehicles, as their route calculation is based on the Dijkstra algorithm, which independently determines the shortest path without considering these removed routes. This approach ensures the realistic representation of traffic regulations pertaining to HOV lanes within our simulation.

Table 8.2 Links that incorporate High-Occupancy Vehicle lanes.

Link number	Origin Node	Destination node	Link Capacity	New Link Capacity (70%)	Length	Freeflow traveltime	Alpha	Beta
42	14	23	4924.79	3447.35	4	4	0.15	4
63	20	22	5075.70	3552.99	5	5	0.15	4
68	22	20	5075.70	3552.99	5	5	0.15	4
70	22	23	5000.00	3500.00	4	4	0.15	4
71	23	14	4924.79	3447.35	4	4	0.15	4
72	23	22	5000.00	3500.00	4	4	0.15	4

Table 8.3 Auxiliary links exclusively for High-Occupancy Vehicle lanes.

Link number	Origin Node	Destination node	Link Capacity	Length	Freeflow traveltime	Alpha	Beta
77	14	23	1477.44	4	4	0.15	4
78	20	22	1522.71	5	5	0.15	4
79	22	20	1522.71	5	5	0.15	4
80	22	23	1500.00	4	4	0.15	4
81	23	14	1477.44	4	4	0.15	4
82	23	22	1500.00	4	4	0.15	4

Typically, HOV lanes in real-world scenarios are implemented on roads with high traffic volumes, where congestion poses a significant issue. An examination of the average link speed across our network, as depicted in Figure 8.7 indicates that the southern region of the Sioux Fall network is the most congested area. Consequently, this area has been selected for the implementation of HOV lanes.

More specifically, the links connecting nodes 20-22-23-14, encompassing a total of six links (numbers 42, 63, 68, 70, 71, and 72), have been chosen for HOV designation. We will reduce the link capacity of these links by 30 percent, emulating the shift in lane allocation to reserve HOV (Table 8.2). Correspondingly, we will add six auxiliary links (numbers 77,78,79,80,81, and 82), each mirroring the properties of the corresponding original link but have the 30 percent vehicle capacity that was initially subtracted (Table 8.3).

As a further baseline scenario, we model the network with the introduction of an HOV lane but without any ridesharing activity. This case is directly comparable to the baseline in Section 8.1, but now incorporates the HOV lane configuration. Under the assumption that all drivers travel alone and no shared mobility services are available, no vehicles qualify to use the reserved lane. As a result, capacity in the general-purpose lanes is reduced, particularly in already congested areas, while the HOV lane itself remains underutilized. The value of this scenario lies in establishing a worst-case benchmark: it isolates the impact of reallocating capacity to HOV use without any offsetting benefits, showing how congestion could worsen if participation in carpooling or ridesharing is low. This provides a necessary reference point for later scenarios, where the introduction of shared mobility users into the HOV lane demonstrates the marginal benefits of enabling access.

The results in Figure 8.8 confirm that withdrawing capacity from the congested corridor increases traffic density and travel times along the restricted links. To fully understand the implications of HOV lane policies, however, it is also important to examine the behaviour of nearby non-HOV links, since drivers excluded from the restricted lane may attempt to reroute through adjacent corridors. Such spillover effects can either amplify or mitigate congestion depending on network structure. In this case, we focus on the auxiliary connections between nodes 19–20 and nodes 14–15, which serve trips between nodes 14 and 20 but are not subject to HOV restrictions. Evaluating these parallel routes provides insight into how displaced solo drivers redistribute within the network, offering a more complete picture of the system-wide consequences of HOV lane implementation.

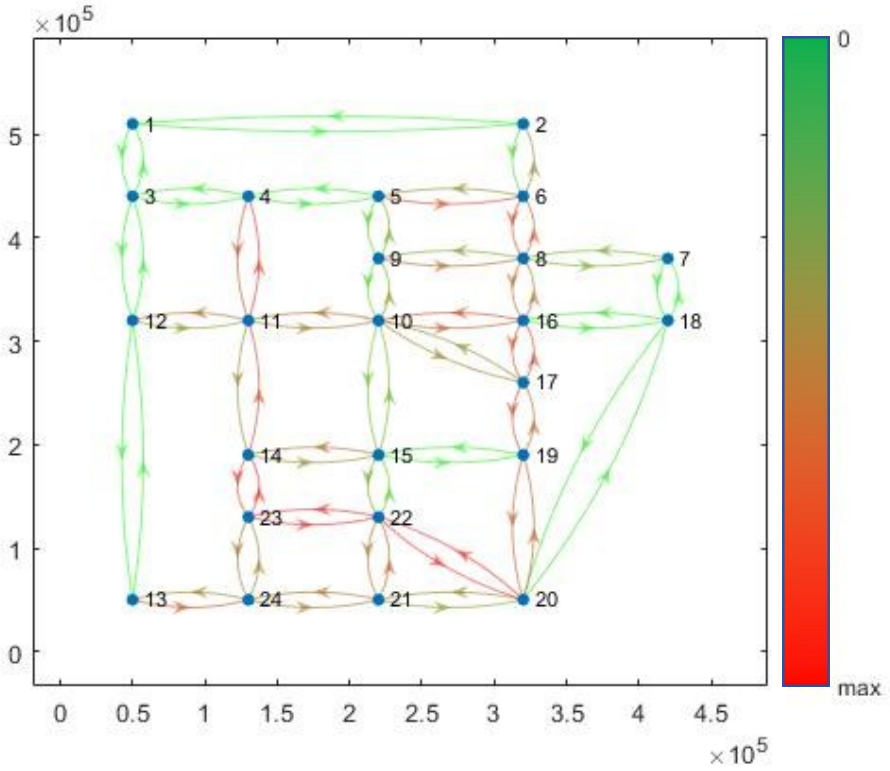

Figure 8.8 Network Average speed (with High-Occupancy Vehicle lanes)

Table 8.4 Implementing High-Occupancy Vehicle lanes without ridesharing system.

Origin Node	Destination Node	Link number	Avg speed (no HOV)	Avg speed (HOV)	Change in %
Links with HOV lanes					
14	23	42	18.02	5.61	-69%
20	22	63	10.74	2.99	-72%
22	20	68	7.54	2.00	-73%
22	23	70	6.03	1.57	-74%
23	14	71	4.24	1.08	-75%
23	22	72	6.82	1.79	-74%
Nearby Links					
14	15	41	29.02	29.02	0
15	14	44	20.09	20.09	0
19	20	59	14.57	14.57	0
20	19	61	15.41	15.41	0

Table 8.4 directly highlights the impact of introducing HOV lanes in the congestion-heavy region. The data show that traffic conditions worsen substantially, as evidenced by a marked decline in average link speed. The column labelled “Average Speed (no HOV)” reports the average link speed in the baseline scenario shown in Figure 8.2, which assumes the absence of both ridesharing and HOV lanes. In contrast, the “Average Speed (HOV lane)” column corresponds to the same links under the HOV-lane scenario presented in Figure 8.8. The withdrawal of 30 percent of the link's capacity results in an approximate 73 percent reduction in the average speed. This extreme outcome, serving as a crucial worst-case benchmark, highlights a limitation in many existing HOV studies that tend to assume positive social welfare outcomes. Our model demonstrates that isolated analysis fails to capture the significant capacity reallocation effects and resulting congestion risks for non-HOV users, underscoring the necessity of integrated traffic equilibrium modelling. It is noteworthy that the adjacent links, which we initially hypothesized might experience a negative impact due to alterations in travellers’ route selection behaviour, minimizing their travel cost (spillover effect). However, we found that the average speed in nearby links remains the same, this is due to the extra travel times is still less than using a detour routes.

Table 8.5 Implementing High-Occupancy Vehicle lanes with ridesharing system (30 percent capacity reduction)

Origin Node	Destination Node	Link number	Link flow (no HOV)	Link flow HOV lane	Link flow Normal Lane	Avg speed (no HOV)	Avg speed HOV lane	Avg speed Normal lane
Links with HOV lanes								
14	23	42	9,776	83.33	9,747	18.02	60.00	5.67
20	22	63	11,934	128.33	11,657	10.74	60.00	3.26
22	20	68	13,244	0.00	12,896	7.54	60.00	2.22
22	23	70	13,897	376.67	13,471	6.03	59.96	1.77
23	14	71	15,067	0.00	14,647	4.24	60.00	1.20
23	22	72	13,427	0.00	13,408	6.82	60.00	1.80
Nearby Links								
14	15	41	8,375	-	8,175	29.02	-	30.47
15	14	44	9,782	-	9,900	20.09	-	19.45
19	20	59	10,683	-	10,491	14.57	-	15.38
20	19	61	10,484	-	10,225	15.41	-	16.58

Next, we integrate a ridesharing system into the network with HOV lanes. Table 8.5 showcases the link flow and average speed, comparing scenarios with and without the ridesharing system. The numerical results suggest that the introduction of HOV lanes in areas of high congestion leads to a small decrease in vehicular flow in the standard lanes. However, the average speed in these normal lanes experiences a marginal decrease due to the substantial reduction in link capacity to accommodate HOV lanes.

In this modelling setup, ridesharing vehicles are assumed to use the HOV lane whenever it is available along their chosen route. The relatively low flow observed in the HOV facility therefore reflects the limited number of ridesharing vehicles travelling through this corridor, rather than a preference for the general-purpose lanes. As shown in Table 8.6, only a small fraction of overall traffic utilises the HOV lane, which allows it to operate close to free-flow conditions with an average speed of around 60 percent. These findings suggest that

withdrawing 30 percent of link capacity for HOV use may be overly restrictive under the current demand conditions. With respect to neighbouring non-HOV links, only negligible changes in flow are observed, indicating that the introduction of the HOV lane did not significantly alter traffic distribution outside the restricted corridor.

Table 8.6 Implementing High-Occupancy Vehicle lanes with ridesharing system (10 percent capacity reduction)

Origin Node	Destination Node	Link number	Link flow (no HOV)	Link flow HOV lane	Link flow Normal Lane	Avg speed (no HOV)	Avg speed HOV lane	Avg speed Normal lane
Links with HOV lanes								
14	23	42	9,776	116.67	9,616	18.02	60.00	13.88
20	22	63	11,934	0.00	11,530	10.74	60.00	8.47
22	20	68	13,244	281.67	12,806	7.54	60.00	5.85
22	23	70	13,897	0.00	13,232	6.03	59.96	4.91
23	14	71	15,067	0.00	14,308	4.24	60.00	3.47
23	22	72	13,427	161.67	13,280	6.82	60.00	4.85
Nearby Links								
14	15	41	8,375	-	8,293	29.02	-	29.61
15	14	44	9,782	-	9,972	20.09	-	19.07
19	20	59	10,683	-	10,753	14.57	-	14.28
20	19	61	10,484	-	9,942	15.41	-	17.96

Our findings indicate that a reduction of 30 percent in link capacity may be overly drastic, leading to a noticeable increase in travel time for non-ridesharing vehicles. Consequently, we have modified this reduction to 10 percent. The outcomes of this adjustment are presented in Table 8.6, revealing more favourable results.

The average speed in the normal lane experiences a less significant decrease compared with the previous 30 percent reduction scenario. For the HOV lane, the speed remains remarkably close to the free-flow speed, with a slight decrease in link flow. These findings suggest that the presence of HOV lanes can encourage the use of shared mobility systems. However, it is noteworthy that the ridesharing origin-destination (OD) pattern changes in response to the capacity reduction modification. In the 30 percent reduction scenario, the majority of ridesharing vehicles traversed the 20-22-23 route, whereas in the 10 percent scenario, the vehicles predominantly followed the 23-22-20 route. The underlying factors driving this phenomenon remain undiscovered within the scope of this thesis.

Nonetheless, our comprehensive model provides a basis for future investigations to discern the reasons behind this observed pattern change. This study underscores that modifications to the road network such as the implementation of HOV lanes can significantly alter traffic behaviour. Although the HOV lane scenario without qualifying vehicles does not yield optimal outcomes for the ridesharing system, it serves as a valuable benchmark, isolating the worst-case effects of capacity reallocation when uptake is absent. Importantly, the results demonstrate that the proposed simulation framework is both robust and adaptable, requiring only minor modifications to incorporate HOV lanes, and capable of supporting more advanced analyses.

With this simulation, we can further probe other related parameters to potentially enhance the efficiency of HOV lanes. The model allows us to experiment with implementing HOV lanes on various types of roads, based on our assumptions.

Such experiments can illuminate the impact of HOV lanes on additional external factors such as public transportation usage or vehicular emissions. At present, we assume that all solo drivers outside the ridesharing system travel alone and are therefore ineligible for HOV use. While this simplifies the modelling, it may not fully reflect real-world conditions. A more realistic extension would be to allow some solo drivers to be reclassified as higher-occupancy vehicles and thus eligible for HOV lanes. Although this assumption was not applied here, our framework is capable of supporting such segmentation. Exploring the effects of varying vehicle occupancy levels would provide valuable insights into the interaction between ridesharing participation and HOV infrastructure, and offer transportation planners a clearer picture of the system-wide benefits of enabling shared mobility access to reserved lanes.

8.6 Impact to the Public transport

In some cases, ridesharing services have been found to decrease the usage of public transportation, particularly in areas where public transportation is inconvenient or inadequate. The ease and convenience of ridesharing services can draw users away from traditional public transport options. Conversely, ridesharing services can also complement public transport by serving areas not well covered by public transit, and during times when public transport services are less frequent (late nights, early mornings, etc.). They can provide a solution for the "last-mile" problem, bridging the gap between transit stops and people's final destinations. This is often the case in suburban or rural areas where public transportation networks are less dense.

In chapter 7, we evidenced that ridesharing systems potentially pose negative implications for public transportation modes such as buses. The data indicated that when ridesharing fares are reduced, there is a consequential increase in ridesharing users and a proportional decrease in the usage of public transport. Conversely, when bus fares are amplified, we observe a decrease in bus usage and an increase in the use of alternative modes of transportation. This competitive effect across distinct modes of transportation is predicated on our stochastic mode choice model, which splits travellers into various modes based on the average cost of travel. However, we can still measure the amount of impact of ridesharing effect to public transport system.

In this chapter, we aim to provide an illustration of the impact of shared mobility systems, such as ridesharing, on public transportation. We observe notable differences in congestion between the eastern and western regions as shown in Figure 8.7. To capture the unique traffic dynamics of each region, we propose two distinctive public transport lines (Figure 8.9).

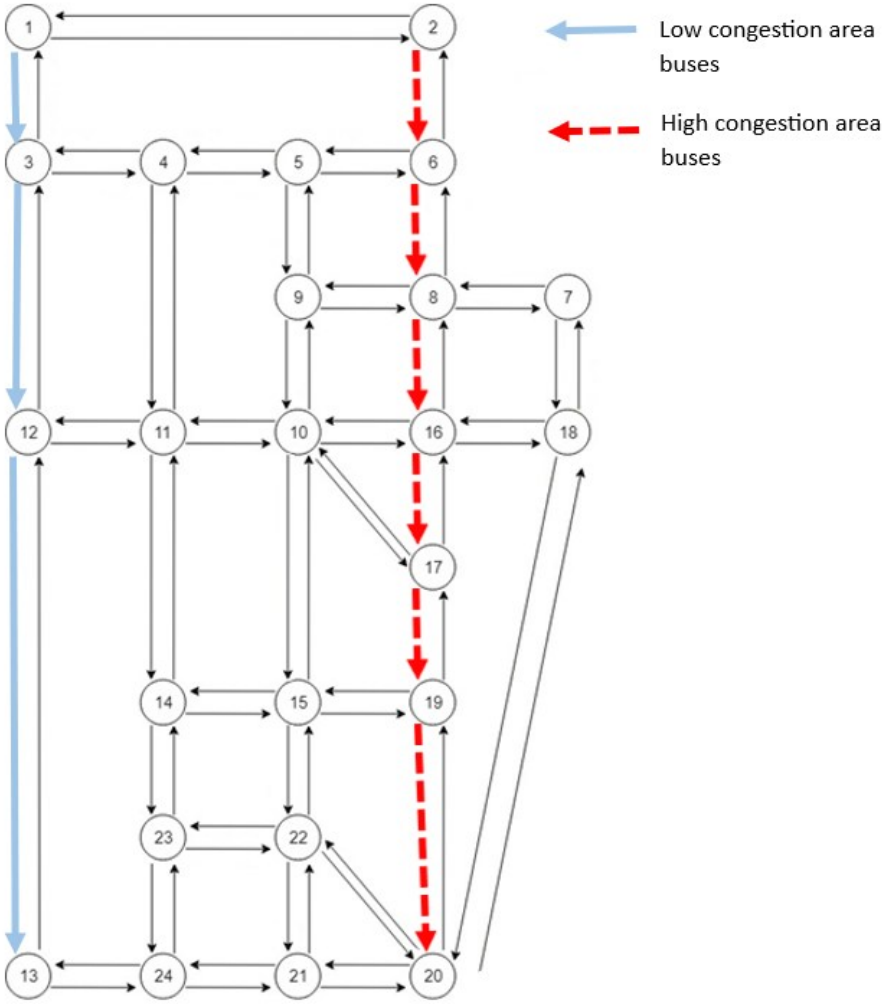

Figure 8.9 Network with two types of bus operations

The first line is designed to operate in the more urban area on the east side of Sioux Falls, characterized by higher vehicle traffic congestion. In this line, the bus route originates from node 2 and travels to nodes 6, 8, 16, 17, 19, and 20. Considering the urban context and the associated traffic densities, we designate a bus frequency of 20 buses per hour for this route. In contrast, the second line operates in the more rural, west side of Sioux Falls where traffic congestion is significantly lower. This bus route originates from node 1, traveling to nodes 3, 12, and 13. Given the rural context and consequent lower traffic densities, the

bus frequency for this line is reduced to 10 buses per hour. In both cases, we assume that each bus has a capacity of up to 30 passengers per vehicle. By implementing these distinct bus lines within the simulation, we aim to highlight the differential impacts of shared mobility systems in varying urban and rural contexts. Following tables compare the performance of buses system in a scenario with and without ridesharing system.

Table 8.7 The performance of buses system (High congestion area buses)

High congestion area buses						
Origin-Destination nodes	2-6	6-8	8-16	16-17	17-19	19-20
Link number	4	16	22	49	53	59
Without Ridesharing	1216	2440	4864	6400	6352	3848
With Ridesharing	1006	1894	4318	6224	6176	3672
Percent Change	**-17.3%**	**-22.4%**	**-11.2%**	**-2.8%**	**-2.8%**	**-4.6%**

Table 8.8 The performance of buses system (Low congestion area buses)

Low congestion area buses			
Origin-Destination nodes	1-3	3-12	12-13
Link number	2	7	37
Without Ridesharing	400	640	1744
With Ridesharing	240	300	570
Percent Change	**-40.0%**	**-53.1%**	**-67.3%**

In the absence of a ridesharing system, the average number of bus users in high congestion areas is approximately 4,186, peaking at 6,400 users traveling on link 49 between nodes 16 and 17. Meanwhile, in low congestion areas, the average number of bus users is significantly lower, at 928 users, with the maximum number of users reaching 1,744 on link 37 between nodes 12 and 13. Upon the implementation of a ridesharing system, it becomes evident that such shared mobility systems compete directly with public transport, particularly

buses. An observable decline in the number of bus users follows the introduction of the ridesharing system, impacting all areas where buses operate. This finding helps resolve the ambiguity often found in previous literature regarding whether the impact of shared mobility is complementary or competitive to public transport, confirming a generally competitive relationship that is significantly amplified in lower congestion environments where auto travel time is less penalized. However, the degree of reduction varies significantly based on the level of congestion.

In low congestion areas, the decrease in bus users is more pronounced, with reductions of up to 67.3 percent. By comparison, the decline in bus users in high congestion areas is less drastic, at approximately 22.4 percent. This differential impact implies that the ridesharing system exerts a negative effect on bus operations, although the severity of this effect is contingent upon the local congestion level. In urban areas with high levels of congestion, buses remain a viable alternative, retaining their user base to a greater extent. This contrasts with less congested areas, where longer bus waiting times render the service less attractive than ridesharing alternatives.

In the current implementation, bus frequency is held constant and not directly varied across scenarios. As such, the framework is not explicitly tested for sensitivity to frequency changes. However, the generalized cost formulation for public transport is readily adaptable: waiting time is inversely related to service frequency, so higher bus frequencies would reduce waiting time and lower generalized costs, thereby increasing the attractiveness of the bus mode. Conversely, lower frequencies would increase waiting time, raising costs and

reducing bus ridership. Incorporating variable bus frequencies into the framework would therefore provide a straightforward extension for future analysis, enabling a richer understanding of how service supply interacts with ridesharing uptake and overall network performance.

Without our multimodal traffic simulation model, a comprehensive investigation into the impacts of a ridesharing system on public transport would not be feasible. Our model is uniquely designed to accommodate the implementation of shared mobility systems alongside multiple public transport service lines. Moreover, we have engineered our model with flexibility in mind, allowing for modifications and adaptations to various parameters related to both public transport and ridesharing. Such adaptability permits thorough examinations of how alterations to different factors may influence the interplay between these two modes of transport.

A noteworthy feature of our model is its ability to handle multiple service lines passing through the same links - a scenario that is known as the "common line problem". With our simulation model, we can study this issue and its broader impacts extensively, with minimal complications. Therefore, our multimodal traffic simulation model provides an invaluable tool for probing the complexities of transportation networks, especially with the emergence of shared mobility systems. Its flexibility and adaptability allow for rich explorations of different scenarios and their corresponding impacts on various forms of transport.

8.7 Pricing scheme

In taxi sharing systems which was operated from e-hailing companies such as Uber, Didi, and Lyft, the pricing mechanism plays a pivotal role in profit maximization for the firm. As shown in the preceding chapter, a sensitivity analysis was conducted concerning the number of ride-sharing system users with respect to the fare charged from customers and paid to the drivers. The analysis revealed that an escalation in the fare in the early stages resulted in an inclination in total revenue. This increase can be attributed to an amplified profit margin per customer. This positive correlation persisted until it reached an apex, signifying the point of maximum profit. Beyond this optimal fare, a downward trajectory in revenue was observed, due to a decrease in user engagement and the matching rate. Therefore, a careful calibration of the fare structure emerges as a critical determinant in ensuring the successful operation of ride-sharing platforms.

The operators have an algorithm that calculates price in a real-time fashion by observing the supply of drivers and the demand from passengers in a particular area at a particular time. This dynamic pricing model is known as "surge pricing." Surge pricing is essentially a form of price discrimination, where fares increase during periods of high demand or when the supply of drivers is low. The surge price can vary based on the level of demand, so the same route can cost differently at different times or on different days. The aim of surge pricing is to incentivize more drivers to get on the road and meet the increased demand. This results in more rides being available, albeit at a higher price.

In this dissertation, the surge pricing scheme can readily be integrated into our matching algorithm via modifications to the fare calculation function. Given that the simulation is macroscopic in nature, key variables such as the number of taxi vehicles (representing the demand side) and the number of passengers (representing the supply side) are predefined as simulation inputs. These variables, initially defined at the outset of the simulation, evolve, and dynamically change throughout the simulation period.

We initially carry out the taxi-sharing simulation under the same parameters as explained in Chapter 7. Due to the introduction of a clustering method, the computational time is reduced, and the simulation is able to run with a more complex network. Consequently, we implement the full spectrum of 528 Origin-Destination (OD) pairs within the Sioux Falls network like our simulation within this chapter. At this stage we focus on the evaluation of link flow values, indicative of network congestion, and total revenue, representative of operators' earnings. It is worth noting that the current setting, akin to Chapter 7, charges passengers based on their original travel distance if they were to operate their own vehicle.

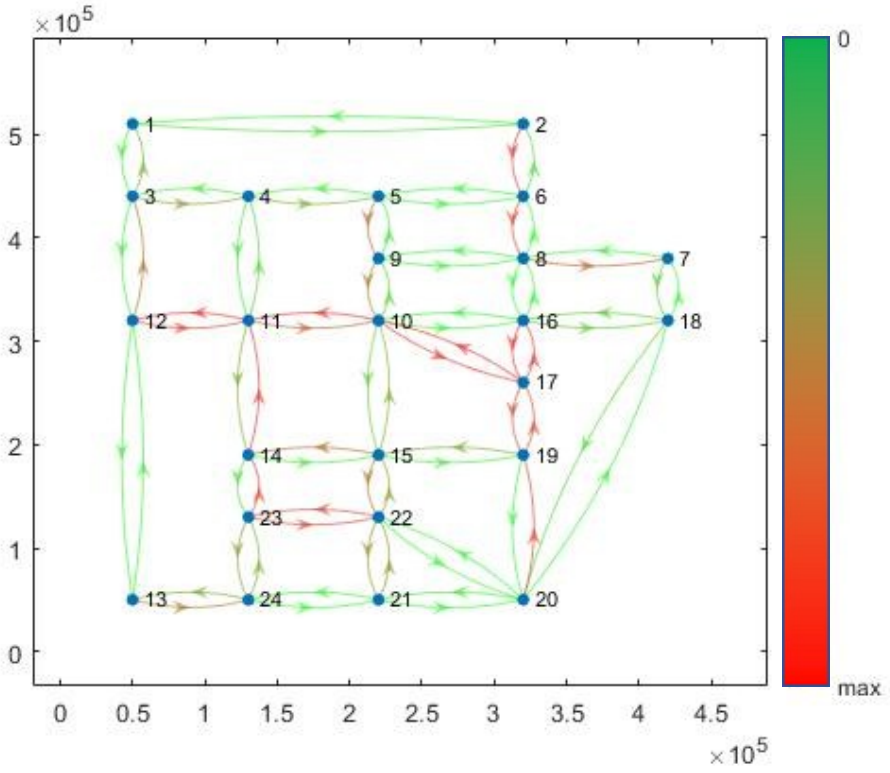

Figure 8.10 Network Average speed (with taxi-sharing)

The figure above illustrates the average network speed within a taxi-sharing system. The average speed of this network is reported at 38.02 miles per hour. This suggests that if the number of taxi vehicles in the system is adequate, it can yield more benefits for the network than a ridesharing system. This is partly because taxi vehicles, as compared to private vehicles, are easier to match given that they don't need to account for additional costs incurred from detour routes. Next, we introduce a new pricing scheme, a model of surge pricing, which increases the fare when the number of passengers is relatively high compared to the number of available taxis. The adjusted equation is shown in Equation 8-3:

$$Adjusted\ fare = Original\ fare \times \rho \frac{n(Pax)}{n(Taxi)}$$

Equation 8-3 Surge Pricing

In this surge pricing function, the original fare (based on passenger distance) is multiplied by the ratio of passengers originating from a given Origin-Destination (OD) pair divided by the number of vacant taxis on that OD. The parameter ρ represents the surging ratio. If ρ is high, it implies that the price is more sensitive to the ratio. In this section, we simulate two scenarios with surge pricing scheme where the parameter ρ=0.25 and ρ=0.5 respectively. As a result of this, Figure 8.11 illustrates the network average speed with the surge pricing function.

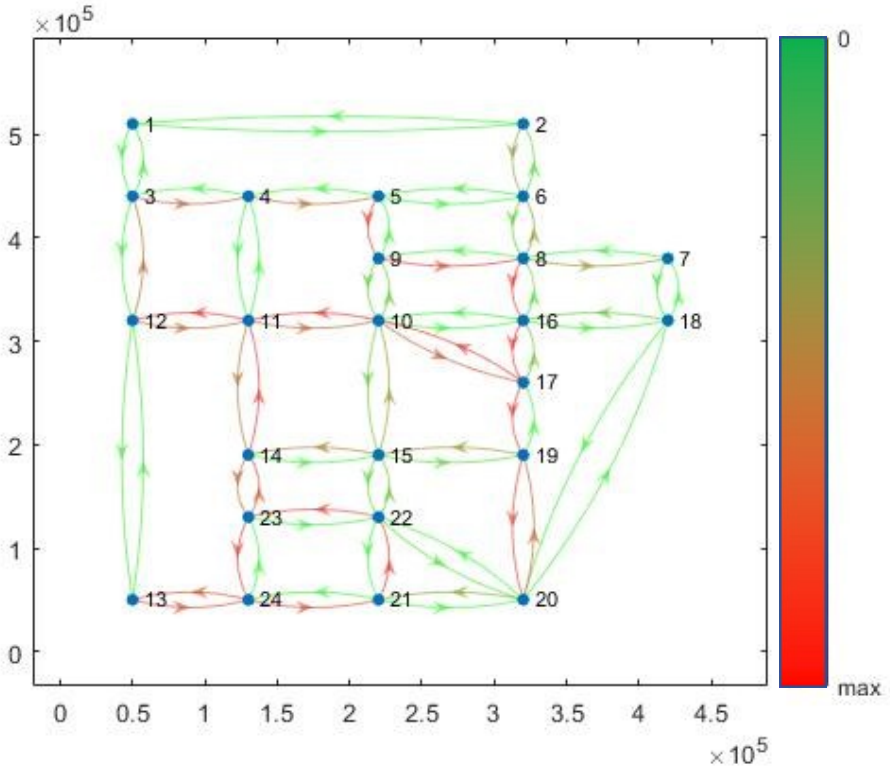

Figure 8.11 Network Average speed (with taxi-sharing with surge pricing ρ = 0.25)

Implementing surge pricing resulted in higher network congestion, reducing the average link speed to 37.1 miles per hour at ρ=0.25. The traffic conditions further deteriorated with an increased surge pricing strength at ρ=0.5, thereby reducing the average link speed to 34.8 miles per hour. From these findings, it can be inferred that surge pricing impacts the road network by increasing traffic

congestion. This is attributed to passengers shifting to solo driving due to the additional costs associated with the taxi-sharing system.

Concerning revenue, surge pricing elevated the revenue collected by taxi-sharing operators. However, as evidenced in Chapter 7, excessively raising the ridesharing fare could potentially reduce the number of matched passengers, leading to a decrease in revenue eventually.

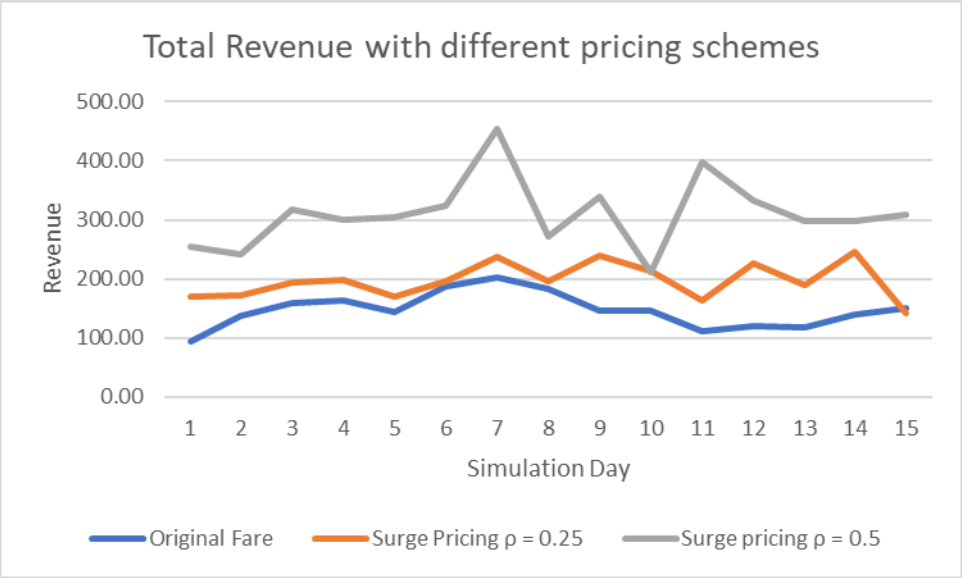

Figure 8.12 Impact of surge pricing on the road network

In conclusion, the simulation framework we proposed is capable of determine the impact of surge pricing on the road network. This is facilitated by our combined matching algorithm, which serves as a route choice model for shared mobility systems, and a stochastic traffic assignment model for solo drivers. Without this combination, studies may lack a realistic behaviour model, such as equilibrated traffic flow and modal splits, both critical to investigating the long-term effects of any implementation. This finding contrasts sharply with conventional pricing models found in the literature that focus primarily on

optimizing platform utility or profitability, demonstrating that dynamic network equilibrium modeling is essential to capture the negative externalities (e.g., increased network congestion and reduced average speed) resulting from modal shifts triggered by profit-focused pricing schemes. At this stage, we only demonstrate our model's capability to investigate diverse research topics. For a deeper investigation and robust conclusions, model adjustments, such as implementing a more realistic surge pricing model used in real-world practices, are essential.

8.8 Optimum fleet size

The concept of the optimum fleet size pertains to identifying an ideal number of vehicles, in this context, taxis, that best serves a specific objective. Although fleet sizing is a widely explored topic in logistics and operations research literature, most transportation studies focus on minimizing the required fleet size based purely on matching demand. Our methodology deviates by employing a network-centric objective, focusing on determining the fleet size that maximizes overall network efficiency (e.g., average link speed) at traffic equilibrium. The determination of this optimum value depends on the objectives that the stakeholder aims to achieve.

From the perspective of taxi operators, the optimization of fleet size is typically subject to considerations of cost efficiency and service commitments. Operating a larger fleet incurs significant costs, including those related to the purchase or leasing of vehicles, fuel or energy consumption, maintenance, insurance, and driver remuneration. In this regard, profitability can be achieved by maintaining

an equilibrium where revenue outweighs the expenditure. Furthermore, if the operators commit to a particular Service Level Agreement (SLA), such as a guaranteed maximum response time (for instance, a taxi's arrival within 10 minutes of booking), the fleet size must be sufficient to fulfil this commitment under all typical conditions.

On the other hand, a transportation planner may strive to find an optimum fleet size based on the volume of demand. This necessitates a fleet size that large enough to accommodate the demand during peak hours without any delays. One approach to measure this is by incrementally increasing the number of taxis in the network until it begins to exert a negative impact on the overall efficiency of the network.

In this section, we will conduct a simulation aimed at determining the optimum fleet size for a taxi-sharing system. We will initiate the simulation by incrementally increasing the number of taxi vehicles in the network. At the initial stages, this should lead to a reduction in traffic congestion across the network. We will then continue to raise the number of taxis until we identify the optimum fleet size. Beyond this optimum point, any further increase will potentially generate adverse impacts on the network. It's important to note that we will set the initial number of taxis at 2,400 vehicles, a quantity that has already been proven adequate for the system in question.

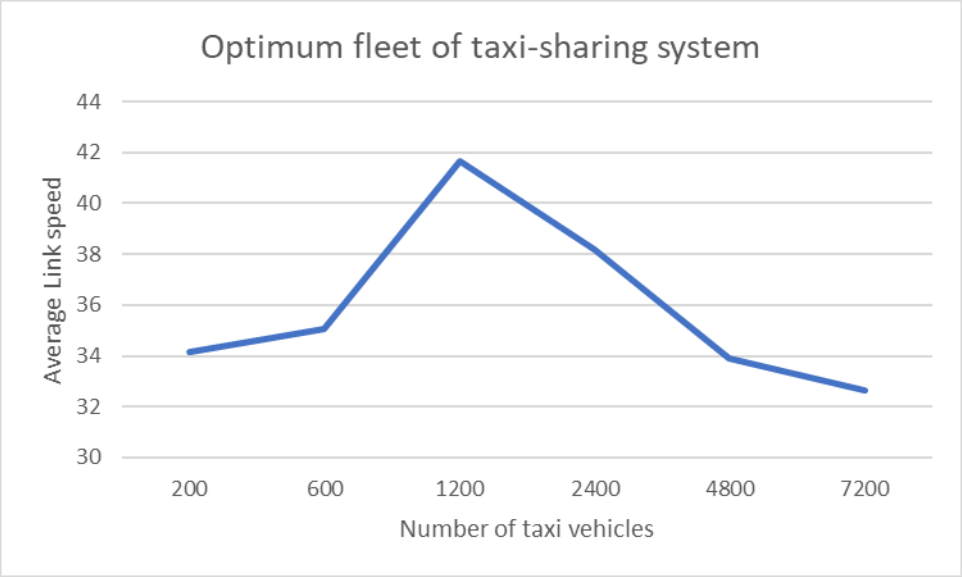

Figure 8.13 The initial effects of incrementally adding taxi vehicles to the taxi-sharing system.

The figure demonstrates the initial effects of incrementally adding taxi vehicles to the network. During the initial phase, the introduction of a taxi-sharing system appears to alleviate network congestion and enhance the average network link speed. The speed continues to rise until reaching an optimum point at a fleet size of 1,200 taxis. This is the point at which the addition of taxis is most beneficial to the road network. However, once this threshold is exceeded, the additional taxis contribute to an oversupply, which results in an increased overall congestion and subsequently leads to a decline in average speed.

From this example, it is evident that the optimal fleet size for taxis in the network, aimed at maximizing the average link speed given the input parameters, can be determined. The simulation framework facilitates the identification of the optimum fleet size according to various criteria with minimal effort. This

demonstrates the utility and efficiency of our simulation tool in the study of shared mobility systems.

8.9 Discussion

In this chapter, we present a practical numerical example demonstrating the capabilities of our simulation framework as an advanced comprehensive tool to help researchers studying shared mobility systems more extensively. Through a comprehensive literature review, we identified seven main research topics in the shared mobility domain: Impact on the road network, Development of matching algorithms, Environmental Impact, Pricing Schemes, Efficiency of High Occupancy Vehicle (HOV) Lanes, Public Transport, and Optimum Fleet Size. Through repeated runs of our simulation, modifying objectives and minor variables, we successfully performed simulations providing basic insights into these topics. Crucially, the numerical findings consistently confirmed the inherent limitations of isolated analytical models prevalent in the existing literature. Specifically, the simulation demonstrated that initial matching estimations, typical of isolated models, are subject to overestimation [Section 8.2], that non-optimized HOV lane implementation can generate severe congestion externalities contrary to common assumptions [Section 8.4], and that pricing strategies optimized solely for platform profit maximization can detrimentally impact overall road network performance [Section 8.6]. By capturing these complex interdependencies and dynamic feedback loops, our integrated framework reliably generates results that overcome the static assumptions of traditional shared mobility research.

Our proposed simulation framework can execute multi-modal traffic simulations in a macroscopic model, which includes an essential route choice model for shared mobility systems. While investigating the impact on the road network, we established that our combined route choice model and matching algorithm is necessary, as results solely based on the matching algorithm model were overestimated. This is mainly due to the reduction in the number of vehicles in the network, which led to a rebound effect in traffic-induced demand. For the development of matching algorithms, we compared different objective functions and mechanisms, studying how network changes affect congestion levels.

Regarding environmental impact assessment, our model successfully generated emission-related factors such as average speed and Vehicle Kilometres Travelled (VKT). According to the values derived from these factors, shared mobility systems like ridesharing can significantly reduce vehicle emissions. We also implemented a surge pricing scheme, which increases fare when passenger demand exceeds the number of available taxi vehicles. The results suggest that while surge pricing could potentially increase revenue, it might negatively impact the road network by increasing congestion levels.

Additionally, we examined the impact of ridesharing on the public transport system by comparing the reduction in bus users in two areas with varying congestion levels. The results show that ridesharing has a more pronounced negative impact on bus lines in areas with low congestion. Regarding HOV lanes, we manipulated the link capacity by remove a fraction and added an auxiliary link with equivalent capacity, representing a partial conversion of the road into a HOV lane. The results indicate the necessity of our simulation model

in determining the effectiveness of HOV lanes. We discovered that an excess of HOV lanes could generate extra congestion. Lastly, we demonstrated that our simulation could determine the optimum fleet size for a taxi-sharing system, given specific objectives.

Although we have shown our simulation framework's ability to answer fundamental research questions in the field of shared mobility systems Comprehensive investigations in the mentioned area of studies are still require an external validation. This can be achieved through the using more realistic data sets and calibration with actual traffic flow. Furthermore, the model requires minor adjustments to get deeper into each topic. In the next chapter, we will summarize our thesis and discuss potential future works to enhance the reliability and applicability of our proposed simulation in the transportation research community.

In conclusion, this extensive numerical demonstration successfully met the third and final objective of the thesis: evaluating the proposed framework’s capability as a research tool for complex shared mobility topics. The simulations confirmed the intrinsic limitations of isolated models, showing that initial matching overestimations are corrected by dynamic feedback, and demonstrated the model's capacity to derive critical policy insights regarding the trade-offs between profit maximization and congestion relief (matching objectives), capacity allocation in HOV lanes, and the competitive relationship with public transport in different congestion contexts. These findings provide compelling evidence for the necessity and utility of this integrated macroscopic simulation approach in urban transportation planning.

CHAPTER 9 CONCLUSION AND FUTURE WORKS

The world's major cities are witnessing increased urban growth, leading to intensified traffic congestion. Although public transport systems exist, they often fall short in providing efficient service due to the "last-mile" problem - the gap between travellers' residences/destinations and the nearest public transport hub. Urban constraints hold these issues; introducing new high-capacity transport systems such as rail or underground is exceedingly expensive in densely populated areas. These challenges have given rise to an innovative transport paradigm known as shared mobility.

Shared mobility, a concept encompassing both shared journeys and shared vehicle usage, introduces an alternative means of navigating urban landscapes. This dissertation primarily focuses on two shared mobility models: ridesharing and taxi sharing systems. These contemporary transport modes are gaining popularity but are considerably new. Therefore, the understanding of their impacts on the transportation network remains limited, unlike more traditional public transport systems that have been studied extensively over time. Compared to other transport modes, shared mobility systems are still lacking a comprehensive framework to evaluate their impacts and interactions with other external factors.

In this dissertation, we introduce a simulation framework that integrates shared mobility systems with various modes of transport. The model is consisting with two parts: mode choice model and rout assignment model, allowing a comprehensive understanding of shared mobility's influence on road networks

in a macroscopic level. This chapter revisits our research objectives and contributions outlined earlier in the dissertation and provides an overview of our proposed framework's implementation. The final sections discuss future research directions.

9.1 Revisiting Research Objectives

This section revisits the aims and objectives of this thesis, as stated in Section 1.2, and provides a summary of the research contributions achieved.

9.1.1 First Objective

The aims and objectives of this thesis were *"to develop a comprehensive traffic simulation encompassing two shared mobility systems: ridesharing and taxi-sharing."* These objectives have been successfully accomplished through the development of the proposed simulation frameworks, as summarized in Chapter 6. Additionally, the objectives specified certain requirements for the traffic simulation, which are as follows:

- *"Multiple modes of transport: The proposed simulation frameworks incorporate ridesharing, taxi-sharing, private transport, and public transport."* To address the diverse modes of transport, two simulation frameworks were introduced. The first framework includes ridesharing, private and public transport, while the second framework focuses on taxi-sharing, private and public transport. These frameworks were developed

after conducting a literature review on route choice models for different modes of transport (Chapter 3, 4, and 5) and are detailed in Chapter 6.

- *"Multiple time scales: The proposed model incorporates both within-day and day-to-day dynamic variations."* This doubly dynamic model consists of two components: the within-day model and the day-to-day traffic model. The within-day model combines route flows from different models, determines the actual time that flows remain in the network, and propagates the flows throughout the time windows (Section 6.4). The day-to-day traffic model functions as a travellers' learning model of mode choice, adjusting the mode split from day to day (simulation day) based on perceived average mode cost (Section 6.5).

- *"Mode switching: The simulation accounts for mode switching, allowing ridesharing drivers and passengers to switch their modes if they are unmatched."* Unlike many previous studies that neglect unmatched travellers, this thesis classifies travellers into two groups: those with vehicle possession (V) and those without vehicle possession (NV). Table 6.1 and 6.2 illustrated the modes of transport accessible to each group of travellers. In the case of unmatched ridesharing users, they are automatically switched to an alternative mode of transport based on their accessibility (Section 6.2).

- *"Macroscopic model: Both simulation frameworks employ a macroscopic model, which reduces computational costs when simulating traffic."* Our simulation framework aggregates the number of travellers and other

values into average values over time (Chapter 6). The nature of the macroscopic model enables it to perform well in large and complex networks, making it suitable for investigating the impact of shared mobility systems in a macroscopic level.

- *"Multiple route assignments: The thesis introduces exclusive route assignment models for each mode of transportation."* After reviewing the development of route choice models for each mode (Chapter 3, 4, and 5) and determining the appropriate models, a new matching algorithm based on linear programming was introduced for shared mobility systems, allowing for the matching of a single driver with up to two passengers. For public transport, a frequency-based transit assignment with a route section approach was proposed. Lastly, a stochastic traffic assignment based on the logit function was selected for private transport. The integration of these mode-specific route choice models is explained in Chapter 6.

9.1.2 Second objective

The second main objective of this research was "*to develop a matching algorithm that determines the route choice model for shared mobility vehicles at a macroscopic level."* The existing matching algorithm models for shared mobility vehicles found in this thesis (Chapter 2 and 3) operate at a microscopic level, where a driver is matched with passenger and treated as a single unit, rather than considering a group of travellers' demand in our macroscopic approach. To address this, we proposed a linear programming-based optimization

model to tackle the problem of route choice for shared mobility systems in a macroscopic context (Chapter 6). This objective of developing a matching algorithm references two essential features.

- *"Multiple passengers: The matching algorithm allows for the matching of up to two passengers with different origin-destination (OD) pairs simultaneously*" Both simulation frameworks presented in this thesis have the ability to match a single driver (or taxi) with up to two passengers. Chapter 3 provides a detailed explanation of the four possible matching patterns that can occur. However, this increased the computational complexity, as discussed in Section 7.6. To address this issue, we implemented a heuristic approach based on clustering methods to reduce the complexity of the matching problem (Section 7.7).

- *"Realistic matching algorithm based on Linear Programming*:" The matching algorithms developed for both ridesharing and taxi-sharing are based on linear programming techniques. The matching algorithm begins by generating all possible matching combinations among a set of drivers and passengers' OD demands. Subsequently, unsuitable matching pairs are excluded before entering the optimization process. The criterion used in this thesis, states that if the cost of ridesharing for any user exceeds the cost of their non-ridesharing mode, it is considered unsuitable. Thus, such pairs are excluded from the optimization process (Chapter 3 and 6).

9.1.3 Third Objective

The third objective of this research is "*to evaluate the capability of our proposed model as a research tool for topics related to the shared mobility system."* In Chapter 2, we reviewed popular research questions related to the shared mobility system, which can be categorized into several areas: impact on the road network, development of matching algorithms, environmental impact, impact on public transport, efficiency of high-occupancy vehicle (HOV) lanes, pricing schemes and optimum fleet size. We provided numerical examples to simulate and demonstrate the usefulness of our proposed models in addressing each of these topics.

- Impact on the road network: In Section 8.1, we conducted simulations on the Sioux Fall network using all 529 ODs data. We compared the results of our simulation in a scenario without ridesharing (referred to as the base scenario) with the best well known values from the data source, which represent the most acceptable as a standard acceptant value for the network. The results indicated that the absolute percentage error of link flow from the best well knowns value was only 22 percent, which is relatively low for this type of simulation. Next, we implemented the ridesharing system in the network and found that it had the potential to reduce traffic congestion.

- Development of matching algorithms: In Section 8.2, we further analysed the benefits of our ridesharing system by comparing matching scenario of one driver with one passenger only and one driver with multiple

passengers. We also highlighted the impact of the objective function on the road network. By modifying the objective function from maximizing Vehicle Kilometre Travelled Saved (VKTS) to maximizing revenue, we assessed its influence on the road network. The results demonstrated that even when the ridesharing operator aimed to maximize profit rather than network benefit, there was still a reduction in network congestion, although it was less pronounced compared to using VKTS as the objective function. This provides insight and confirms our initial hypothesis that the multiple passenger scheme could be more beneficial for the road network. It also shows that our simulation framework is not limited to a single configuration, allowing future researchers to use it more broadly.

- Environmental impact: Section 8.3 provided proof that our simulation can generate emission-related factors such as link flow, link speed, and Vehicle Kilometre Travelled (VKT) through another numerical example. We demonstrated that these factors are readily generated and showed that ridesharing systems could lead to a reduction in emissions. However, in this thesis, we only determined how each related factor is reduced, without providing specific emission values such as CO2 and PM2.5. Incorporating these emission values would require additional models, such as COPERT which requires additional vehicle-related factors that beyond the scope of this research. Nevertheless, we have shown that our simulation is ready to be integrated with such emission models.

- Efficiency of HOV lanes: Assessing the efficiency of HOV lanes required modifying the network link properties. We reduced the capacity of certain links and added auxiliary links with equivalent capacity to replicate the concept of HOV lanes, which reserve specific lanes for shared mobility vehicles. The results, as shown in Section 8.4, indicated that HOV lanes encourage travellers to use the ridesharing system. However, implementing too many HOV lanes could potentially increase congestion for other modes of transport. We also provided evidence that implementing HOV lanes is feasible within our simulation framework.

- Impact on public transport: To demonstrate the impact of shared mobility systems, such as ridesharing, on the public transport system, we created two new public transport lines in Section 8.5 to represent low-congestion and high-congestion areas. The low-congestion area had a longer link topology, representing suburban areas with greater distances between bus stops. The results showed that the ridesharing system competes with public transport, but the reduction in the number of bus users was more significant in the low-congestion area.

- Pricing schemes: Surge pricing, a form of price discrimination that increases fares when the number of vehicles or drivers is relatively low compared to passenger demand, was examined in Section 8.6. We proposed a pricing function for the taxi-sharing system that replicates surge pricing. We simulated the environment and assessed the impact of different strength factors multiply by the ratio between the number of passengers and drivers at a given location. The results showed that surge

pricing increased the revenue of taxi-sharing operators, but it also led to increased traffic congestion. Notably, when a small strength factor was used, the increase in congestion was negligible.

- Optimum fleet size: The ability to calculate the optimum fleet size of a shared mobility system was demonstrated in Section 8.7. We performed multiple iterations of the simulation, varying the fleet size of the taxi-sharing system and comparing it with an objective value, such as average link speed. The optimum fleet size was determined by finding the point that minimized congestion or maximized travel speed on the network. However, our simulation framework is not limited to this objective value. A range of objective parameters can be used to determine the optimum fleet size, such as maximizing revenue for operators or minimizing in-vehicle time.

By conducting these evaluations and providing concrete examples, we have established the capability of our proposed model as a versatile research tool for investigating various aspects of the shared mobility system. The simulations conducted in this thesis contribute to the understanding and advancement of shared mobility research, providing valuable insights for future researchers in the field.

9.2 Contributions in model development

9.2.1 Macroscopic matching algorithms

In this section, we would like to emphasize our contributions in terms of modelling. To the best of our knowledge, our proposed simulation framework is the first of its kind to integrate a matching algorithm, which is a core model for shared mobility system research, with a dynamic traffic variation model that can achieve a user equilibrium solution. Previous research has typically treated these two models separately due to their different levels of modelling. The matching algorithm are modelled at a microscopic level, while the traffic assignment model is at a macroscopic level. However, our adaptation, as demonstrated in Chapter 6, allows the matching algorithm to perform at a macroscopic level by using linear programming, while maintaining computational efficiency. Without this integration of two different models, the matching results would be overestimated due to induced demand, as shown in Chapters 7 and 8. Thanks to this integration, our model can now handle multiple tasks and be considered a simulation-level model.

9.2.2 Rolling horizon approach

Although our matching algorithm operates at a macroscopic level, with the within-day and day-to-day traffic variation model embedded in the framework, we can separate temporal variation into time window called time horizon into smaller frames, making our model similar to dynamic traffic assignment model.

The matching algorithm exhibits will behaviour similar to those microscopic level models. On the other hand, if highly detailed data are not available, we can perform the simulation with wider time windows, resembling a more static traffic simulation that can handle larger network datasets.

9.2.3 Hierarchical clustering

The final modelling contribution we would like to emphasize in this section is our clustering technique, which aims to reduce computational costs. While we utilize a linear programming method, known for its efficiency compared to other models, the bottleneck in our simulation framework lies in the matching algorithm when generating all possible matching combinations (Section 7.6). We observed that computational costs increased exponentially as the number of origin-destination (OD) pairs increased. To address this issue, we implemented a hierarchical clustering method that separates OD pairs into clusters and performs the matching algorithm for each group independently. Furthermore, to ensure manageable calculation times, we set a maximum number of OD pairs per cluster.

In addition, we explored multiple similarity measures to compare the performance of clustering, using the objective function in the matching algorithm as a benchmark. Our findings revealed that the Saved distances measure yielded the best performance when the maximum number of OD pairs per cluster was set at 20-30 ODs. This clustering technique allows our proposed model to simulate the matching algorithm in a large network with minimal computational cost, a capability unmatched by any other existing research.

This modelling contribution is crucial in overcoming the computational challenges associated with generating all matching possibilities. By employing hierarchical clustering and selecting an optimal similarity measure, we significantly reduce the computational burden while maintaining the performance of the matching algorithm. This advancement enables our simulation framework to handle very large networks with minimal computational resources, providing a unique and supplement tool for researchers in the field.

9.3 Policy and Practical Implications

The comprehensive simulation framework developed in this thesis, designed for the "SIMULATION AND OPTIMISATION OF DYNAMIC MULTIMODAL TRAFFIC NETWORK WITH SHARED MOBILITY SYSTEMS", offers significant policy and practical implications for transport planners, policymakers, and researchers. By providing a sophisticated yet efficient tool for investigating the complex interplay between shared mobility and traditional transport modes, this research addresses critical needs in contemporary urban planning.

Transport planners and policymakers would find this framework particularly valuable for informing strategic planning and decision-making, as it simulates the impacts of shared mobility systems on traffic congestion, emissions, public transport usage, and the efficiency of HOV lanes. This allows for more informed decisions regarding system development and integration. Unlike empirical models reliant on fixed data, this analytical framework offers greater flexibility for comprehensive investigations based on variable inputs and scenarios. The framework can effectively assess congestion management strategies by

evaluating how different matching algorithms or operational parameters influence link flows and average network speeds, thereby understanding network-level effects. Furthermore, by generating emission-related factors such as average speed and Vehicle Kilometres Travelled (VKT), the model enables policymakers to quantify potential emission reductions from shared mobility, assisting in sustainable urban transport policy formulation. It also facilitates optimising pricing schemes and fleet management, as it can model the impact of dynamic pricing on operator revenues and congestion, and identify optimum fleet sizes for taxi-sharing to minimize congestion or maximize travel speed. The model's design also enables robust investigation into the competitive or complementary relationships between shared mobility and existing public transport, providing crucial insights into how shared mobility impacts bus ridership and its potential as a first-mile or last-mile solution.

For the research community, the framework serves as an advanced comprehensive tool that provides a realistic and computationally efficient simulation framework capable of addressing a wide range of shared mobility research questions that previously required disparate models. This macroscopic approach, integrating a matching algorithm with dynamic traffic variation, bridges the gap between microscopic and macroscopic modelling. The framework contributes to an enhanced understanding of user behaviour through its dynamic multi-modal nature, including mode switching for unmatched travellers and day-to-day behavioural changes in response to perceived costs. Methodologically, the development of macroscopic matching algorithms using linear programming and a hierarchical clustering technique to manage computational complexity significantly advances the capabilities of simulating large-scale shared mobility systems, allowing for unprecedented scalability in

network analysis. Ultimately, by clearly delineating its capabilities, as well as its limitations, the framework lays a solid groundwork for future research, including extensions towards dynamic ridesharing, external validation with real-world data, more realistic traffic propagation models, and fully integrated multimodal transportation analysis.

In essence, this research provides a powerful and adaptable instrument for both the academic community and practitioners, enabling a deeper, more integrated understanding of shared mobility systems and their profound influence on modern urban transportation.

9.4 Limitations of the Framework

Despite the comprehensive nature and advanced capabilities of the proposed simulation framework, it is essential to acknowledge several inherent limitations that shape the scope and generalisability of its findings. These limitations, arising from behavioural representations, modelling simplifications, network scope, and validation considerations, are outlined below, providing avenues for future research and refinement.

9.4.1 Traveller Behaviour and mode choices

A primary limitation of the present framework resides in its behavioural representation of traveller mode choice. While mode choice is predominantly modelled as a function of fare and travel time, studies of car ownership and car

use consistently demonstrate that individuals also value a range of additional, often intangible, attributes associated with private car travel, such as flexibility, privacy, comfort, reliability, and the autonomy of driving alone. As these benefits are not explicitly captured, the perceived willingness of some travellers to shift from private car use to ridesharing may be overstated. Furthermore, specific traveller time preferences, such as earliest departure or latest arrival times, are neglected in favour of representing time as a generalised cost.

Another behavioural limitation arises from the potential for the actual travel costs experienced during a trip to deviate from initial expectations, primarily due to unforeseen dynamic traffic conditions influenced by other network flows. The model's inability to dynamically adjust expected costs based on evolving network states represents a simplification in capturing real-world behavioural adaptations to uncertainty. Additionally, travellers are generally modelled as committing to a single mode of transport for their entire journey once a choice is made, with the exception of unmatched shared mobility users who are re-routed to an alternative mode. This restricts the model's capacity to fully capture complementary benefits, such as shared mobility serving as first-mile or last-mile solutions integrating with public transport hubs.

9.4.2 Public Transport Modelling

A notable simplification in public transport representation is the omission of institutional and pricing structures prevalent in real-world contexts. While the generalised cost function incorporates fare, travel time, and a crowding factor, it does not explicitly reflect the effects of season tickets or subsidised passes, which can substantially lower the marginal cost of individual public transport

trips. This simplification may lead to underestimating the persistence of bus ridership under certain policy or pricing regimes. Similarly, bus frequency is fixed across all scenarios. Consequently, the dynamics of waiting times and service reliability, which are directly influenced by frequency and significantly shape public transport's attractiveness, are not captured, providing only a partial view of how bus services interact with shared mobility systems.

The framework employs a "soft capacity" constraint approach for modelling crowdedness in public transport links. While this method ensures a converged solution, it risks generating unrealistic link flows that exceed the actual physical capacity of vehicles, not fully reflecting real-world phenomena like passengers failing to board overcrowded vehicles. Moreover, the public transport system is assumed to operate on dedicated links, meaning it does not interact with or affect the general road network shared by private vehicles and shared mobility services, thereby overlooking potential mutual congestion impacts.

9.4.3 Shared Mobility Modelling

The treatment of ridesharing in the current implementation is constrained by static assumptions, where drivers and passengers are matched once at the beginning of the journey, and routes are fixed thereafter. This approach fails to capture the inherent uncertainty and flexibility of real-world dynamic services, where passengers might accept a trip without knowing in advance if they will ride alone or with others, and where fares are adjusted accordingly. This is further simplified by the fixed charges for ridesharing fares, which do not reflect the complex dynamic pricing strategies (e.g., surge multipliers, discounts) frequently employed by real-world operators. These factors can significantly

influence demand, profitability, and equity outcomes but were beyond the scope of this analysis.

The vacant taxi model employed in this thesis makes certain simplifications by not fully adopting intra-zonal travel distance and time parameters as described in some literature, neglecting these factors in the prediction of vacant taxi movements. Additionally, in HOV lane scenarios, all ridesharing vehicles were assumed to be eligible for and to use HOV facilities whenever present on their route, while all solo drivers not utilising the ridesharing system were assumed to be driving alone and thus ineligible. In practice, compliance with HOV regulations may vary, and some ridesharing vehicles might still use general-purpose lanes, or solo drivers might carpool informally, representing a simplification of real-world behaviour and vehicle segmentation

.

9.4.4 General Modelling and Scope

The experiments primarily rely on the Sioux Falls test network, a benchmark widely used in transport modelling research. However, this network is relatively small compared with real metropolitan networks (e.g., London or Bangkok). Consequently, the findings of this study should be regarded as illustrative of potential interactions and dynamics rather than directly predictive of real-world outcomes. The framework also incorporates a K-shortest path approach, restricted to five paths per Origin-Destination (OD) pair, to balance computational feasibility with accuracy. While this significantly reduces the computational burden, it introduces an average link flow error of approximately 22% relative to the benchmark solution, highlighting an inherent trade-off between computational tractability and precision.

A significant drawback of the within-day traffic model lies in its flow propagation process. The model determines the duration each route flow remains in the network based on instantaneous route conditions at the start of a specific time horizon. Once calculated, this duration is fixed for future horizons. This absence of dynamic readjustment of actual travel times and flow durations as traffic conditions evolve across subsequent horizons can lead to potentially unrealistic actual route travel costs, as a continuous feedback mechanism for flow propagation is not incorporated. Lastly, to ensure model convergence and reflect a more realistic behavioural context, the simulation framework explicitly neglected a specific category of travellers: those without vehicle possession who also lack access to the public transport system. This omission was implemented by assuming that all OD pairs not reachable by public transport included vehicle-owning travellers, avoiding scenarios where individuals would be left without any viable transport mode if unmatched by shared mobility, which would pose issues for model stability and realism.

9.4.5 Validation and Applicability

While the model's internal performance was validated through numerical convergence analysis, sensitivity tests, and comparisons against predefined parameters and accepted academic benchmarks, external validation using real-world traffic and shared mobility data was not undertaken in this thesis. This is primarily attributable to the significant challenges in obtaining comprehensive and disaggregated real-world data for relatively new shared mobility systems, which are often operated by private firms with inherent data privacy constraints. Such external validation would be highly beneficial for confirming the model's applicability to actual urban scenarios.

Despite these limitations, the framework provides a robust basis for exploring the complex interactions between shared mobility and traditional transport systems. The simplifications adopted were necessary to ensure tractability and transparency, enabling clear insights into the network-level effects of ridesharing and taxi-sharing systems. Rather than undermining the contribution of this work, these limitations are viewed as opportunities for future refinement, where richer behavioural representations and more complex network structures can be incorporated to improve realism and policy relevance

9.5 Future works

In this section, we will discuss potential future improvements for our simulation framework, addressing both the limitations of our models and potential extensions to our thesis.

9.5.1 Dynamic ridesharing

The ridesharing system proposed in this thesis allows a onetime match for each driver and passenger which is still suboptimal. To increase the benefit of the system, this section will describe a more advanced matching algorithm called dynamic matching. The proposed matching algorithms are categorized as static matching. It is performed using all requested data gathered ahead of time, returning route information and pick up – drop off sequence to matched drivers. Once the data is given, the driver must navigate the vehicle accordingly. The matching and its results are generated only once throughout the ridesharing

journey. This one-time matching sometimes leads to a suboptimal solution as there is a vacant seat at some part of the journey. After one of the two passengers has been dropped off, a space for one passenger in that vehicle has become available again. This vacancy allows the vehicle to be matched with another potential passenger. This temporary vacancy is where dynamic matching supplements the ridesharing system. To maximize the utility of the ridesharing service, if the scenario occurs, the vehicle should be nominated in the next iteration of the matching algorithm...

Dynamic matching supplements this system by allowing new passengers to be assigned whenever a temporary vacancy arises. For instance, once one passenger has alighted, the algorithm can nominate the vehicle again in the next iteration to serve another traveller. This increases seat utilisation, reduces the number of vehicles needed, and can lower costs for passengers. In practice, such systems are used in real-world services like UberPool, where passengers accept a lower fare in exchange for the possibility of sharing the vehicle with others who join dynamically.

The drawback of dynamic matching is that it occasionally generates a longer travel time and distance for some ridesharing users. By adding another passenger on the ridesharing trip, ridesharing divers make a further detour to pick up and drop off another passenger. The extra passenger may also increase the waiting time and travelling time for the second passenger; in other words, it may interrupt the existing pick-up and drop-off process. Although accepting the new passenger increases the travel cost, the ridesharing fare charged from the additional passenger can help reduce the travel cost to affected users. The driver

could receive some of the passenger's fare as an extra incentive ($F^{d}_{d,p1,p2,p3} > F^{d}_{d,p1,p2}$). If the incentive is substantial enough, the travel cost from the extra detour could be nullified, or the driver may have a lower travel cost than the static matching. For the affected passenger, the additional fare charged from the additional passenger can help reduce the ridesharing fare ($F^{p2}_{d,p1,p2,p3} < F^{p2}_{d,p1,p2}$). If the additional passenger is relatively too far from the existing ridesharing route, the new matching can become unworthy to any traveller. To avoid this scenario, the dynamic matching algorithm must ensure that the cost on each affected user is still less than the non-ridesharing trip. The dynamic matching algorithm with a single driver – multiple passengers is the most complex in this thesis. This complexity is because more factors are involved in the travel cost calculation. The combination of previous and current matching patterns and the vehicle stage in which the new matching occurs affected the travel cost for each user differently. Some combinations may result in the second passenger's longer travel time and cost. Note that the dynamic matching does not increase the travel cost of the 1st passenger because the new matching can only occur after the 1st passenger has dropped off. Hence, the 1st passenger will not have any reduced fare from the additional passenger. There are only three stages (picking up the 2nd passenger, dropping off the 2nd passenger and reaching the driver's destination) that the ridesharing vehicle can facilitate an additional passenger. The vehicle can accommodate only one extra passenger in the first and second stages, while the last stage can accept up to two passengers. Following figure illustrates an example of a dynamic matching algorithm:

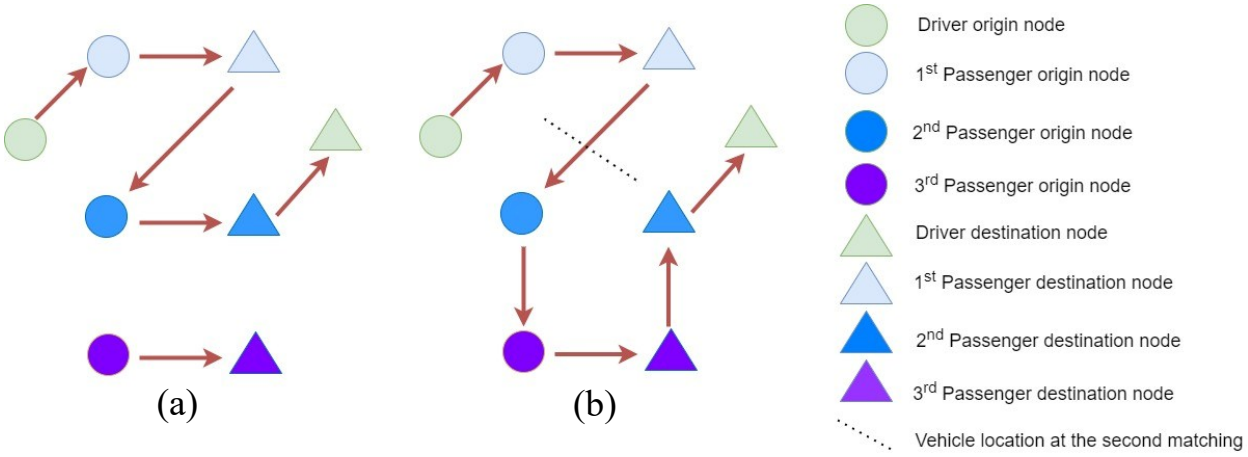

Figure 9.1 Matching algorithm results (a) without dynamic matching (b) with dynamic matching

Without the dynamic matching algorithm (Figure 9.1a), the system matches the driver with two passengers using matching pattern # 1. Corresponding with the matching pattern, the ridesharing route departed at the driver's location O_d, followed by the first passenger origin O_{p1} and destination D_{p1} nodes. After the first passenger is dropped off at their destination, After the first passenger is dropped off at their destination, the vehicle navigates to pick up and drop off the second passenger at O_{p2}, D_{p2} before arriving at its final destination at the driver destination node D_d. The third passenger could not participate in the ridesharing system as there was no available vehicle in the system. With the dynamic matching algorithm, the third passenger has a better chance to be matched in the ridesharing system. Next, we assume that the new matching happened after the first passenger had been dropped off. At this stage, the vehicle has a vacancy for an extra passenger. Figure 9.1b demonstrates the case where the third passenger is matched with pattern # 3. This changes the ridesharing route from $O_{p2} \rightarrow D_{p2} \rightarrow D_d$ into $O_{p2} \rightarrow O_{p3} \rightarrow D_{p3} \rightarrow D_{p2} \rightarrow D_d$. In other words, the vehicle picks up and drops off the new passenger before dropping off the second passenger. The new

ridesharing makes a detour that generates extra travel time and cost for the second passenger and the driver.

Although dynamic ride-sharing has primarily been explored in microscopic models that follow individual vehicles and passengers, extending the concept to a macroscopic framework offers clear planning advantages. By incorporating dynamic matching, the framework can represent higher seat utilisation and a further reduction in the number of vehicles required, leading to measurable decreases in congestion and vehicle-kilometres travelled. It also allows the effect of fare adjustments to be reflected, since dynamic pooling often offers passengers lower prices if they accept longer detours or waiting times. At the operator level, dynamic ride-sharing increases the opportunity to generate additional revenue by serving more passengers per trip. Capturing these aggregate impacts within a macroscopic model enables analysis at the scale of an entire network, where the trade-offs between congestion relief, fare policy, operator profit, and public transport competition become visible. From a planning perspective, this extension is valuable because it provides a practical way to test how real-time matching could amplify the benefits of shared mobility when applied across a metropolitan system, without the computational burden of full microscopic simulation.

9.5.2 External validations

One area for future work is the external validation of our simulation framework. Validating a large-scale network simulation requires a significant amount of data, which can be particularly challenging for shared mobility systems like ridesharing or taxi-sharing. As these modes of transportation are relatively new and often controlled by private firms, obtaining the necessary data for validation purposes can be difficult. Additionally, certain aspects of the matching algorithm used in the real world may be classified for business purposes, further complicating the validation process.

While this thesis validated the model's performance by analysing the results against predefined parameters and comparing them to the best-known data accepted in academic research, validating the model with real-world data, and investigating the impact of shared mobility in real-world scenarios would be highly beneficial. However, validating such a large-scale network can be challenging due to data availability and privacy constraints.

Although dynamic ride-sharing has primarily been explored in microscopic models that follow individual vehicles and passengers, extending the concept to a macroscopic framework offers clear planning advantages. By incorporating dynamic matching, the framework can represent higher seat utilisation and a further reduction in the number of vehicles required, leading to measurable decreases in congestion and vehicle-kilometres travelled. It also allows the effect of fare adjustments to be reflected, since dynamic pooling often offers passengers lower prices if they accept longer detours or waiting times. At the

operator level, dynamic ride-sharing increases the opportunity to generate additional revenue by serving more passengers per trip. Capturing these aggregate impacts within a macroscopic model enables analysis at the scale of an entire network, where the trade-offs between congestion relief, fare policy, operator profit, and public transport competition become visible. From a planning perspective, this extension is valuable because it provides a practical way to test how real-time matching could amplify the benefits of shared mobility when applied across a metropolitan system, without the computational burden of full microscopic simulation.

9.5.3 A more realistic traffic propagation model.

The within-day traffic model in our simulation framework has a significant drawback related to the traffic propagation process. The current model calculates the actual travel time for each route flow and determines the duration of the route flow based on the value and the length of the time horizon. However, this calculation solely relies on the instantaneous route conditions within that specific time horizon. Once the duration is calculated it will be fixed for future horizons. As we transition to the next time horizon and the traffic conditions change based on the new set of route flow, the actual travel time, and the duration of the route flow from the previous horizon should be adjusted accordingly. This discrepancy results in unrealistic actual route travel costs, as we lack a feedback model for the flow propagation process.

One potential improvement is to develop a within-day traffic model that can trace back and readjust the travel time of each route based on the evolving traffic conditions. However, implementing such a model would significantly increase the complexity of the simulation framework. Given the limitations of the research period, we did not have sufficient time to develop a model that addresses this issue.

9.5.4 Multimodal transportation

In this thesis, although travellers are presented with multiple mode choices for traveling from their origin to their destination, once a decision is made, they are typically required to use only that mode of transport throughout their journey, with the exceptional of unmatched travellers from shared mobility modes who can switch to another available mode. This approach may not fully reflect reality and may limit the utilization of the benefits offered by shared mobility systems.

One significant advantage of shared mobility systems, compared to traditional public transport, is their accessibility, which can help address the last-mile problem associated with public transport networks. In this thesis, we simulated the impact of ridesharing systems on the public transport system and observed a competitive effect where shared mobility systems compete for users. However, if a multimodal transportation model were developed that allows users to utilize multiple modes of transport for their journey, different results might be observed. Shared mobility systems, such as ridesharing or taxi-sharing, could serve as a complementary mode of transport, bridging the gap between travellers' residences/destinations and public transport hubs (Figure 9.2). Travelers could

choose shared mobility options for shorter journeys and then connect to public transport, such as rail, for longer journeys, which are often more cost-effective compared to other modes of transport. Thus, the potential synergistic effects between shared mobility and public transport can be observed.

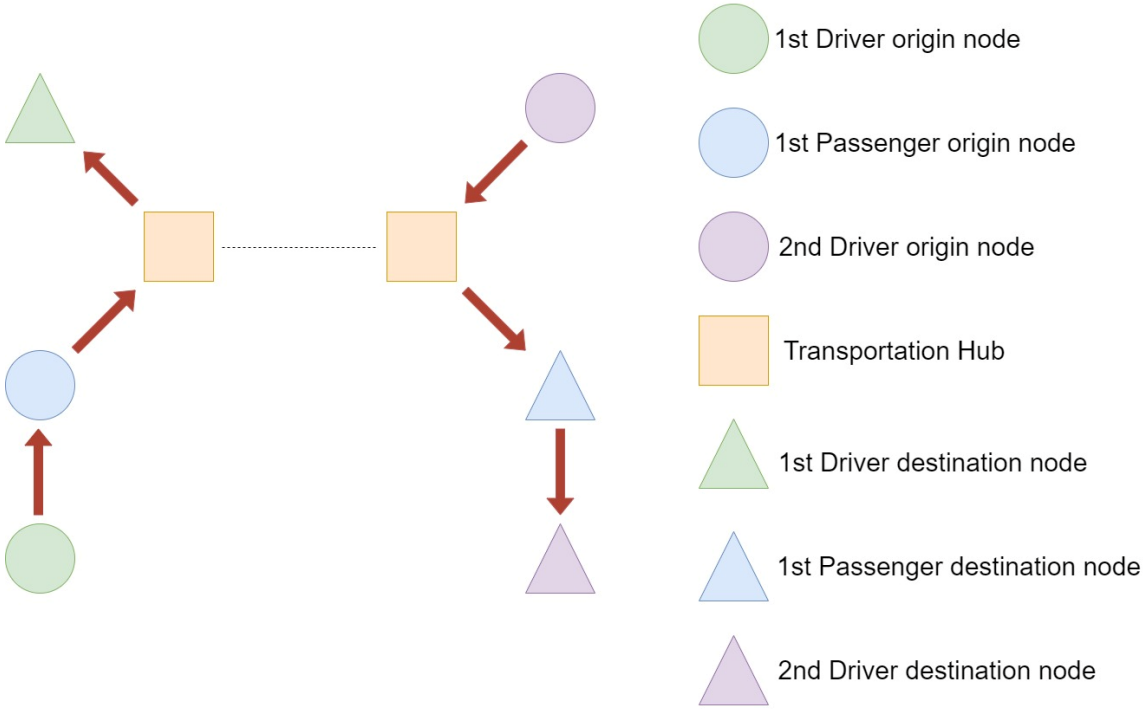

Figure 9.2 An example of using shared mobility for last-mile problem.

By developing a multimodal transportation model that incorporates shared mobility systems as part of the overall transportation network, we can gain a deeper understanding of the interplay between shared mobility and public transport. This could shed light on the potential benefits and opportunities for enhancing transportation efficiency, accessibility, and overall passenger experience. Further research in this area could explore the implications of integrating shared mobility and public transport, offering insights into policymaking, infrastructure planning, and improving urban mobility systems.

In summing up the research, the thesis successfully achieved its primary objective by developing and validating a comprehensive, dynamic, macroscopic multimodal traffic simulation framework. The detailed sections reviewing research objectives, outlining contributions, and addressing limitations collectively confirm the thesis provides transport planners and policymakers with an advanced tool for investigating the long-term impacts of shared mobility. Future research, guided by the limitations discussed (e.g., incorporating non-monetary values, season tickets, and more realistic dynamic traffic propagation models), is essential for external validation and enhancing the model's applicability to complex, real-world metropolitan systems, thereby maximizing its utility in designing data-driven urban transport strategies.

REFERENCES

Agatz, N.A.H., Erera, A.L., Savelsbergh, M.W.P. & Wang, X. (2011) Dynamic ride-sharing: A simulation study in metro Atlanta. *Transportation Research Part B: Methodological.* 45 (9), 1450–1464. https://doi.org/10.1016/j.trb.2011.05.017

Agatz, N., Erera, A., Savelsbergh, M. & Wang, X. (2012) Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research.* 223 (2), 295–303. https://doi.org/10.1016/j.ejor.2012.05.028

Akçelik, R. (1991) Travel time functions for transport planning purposes: Davidson's function, its time-dependent form and an alternative travel time function. *Australian Road Research.* 21.

Alemi, F., Circella, G., Mokhtarian, P. and Handy, S. (2019) What drives the use of ridehailing in California? Ordered probit models of the usage frequency of Uber and Lyft, *Transportation Research Part C: Emerging Technologies*, 102, 233–248. https://doi.org/10.1016/j.trc.2018.12.016

Aljazzar, H. and Leue, S. (2011) K*: A heuristic search algorithm for finding the k shortest paths, *Artificial Intelligence*, 175(18), 2129–2154. https://doi.org/10.1016/j.artint.2011.07.003

AlKheder, S. (2021) Taxi ride sharing in Kuwait: Econ–enviro study, *Energy*, 225, 120269. https://doi.org/10.1016/j.energy.2021.120269

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E. and Rus, D. (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment, *Proceedings of the National National Academy of Sciences*, 114(3), 462–467. https://doi.org/10.1073/pnas.1611675114

Amey, A. (2011) A proposed methodology for estimating rideshare viability within an organization, applied to the MIT community, in *TRB Annual Meeting Proceedings*, 1–16.

Amini, B., Peiravian, F., Mojarradi, M. and Derrible, S. (2016) Comparative analysis of traffic performance of urban transportation systems, *Transportation Research Record*, 2594(1), 159–168. https://doi.org/10.3141/2594-19

Angelelli, E., Morandi, V., Savelsbergh, M. and Speranza, M.G. (2021) System optimal routing of traffic flows with user constraints using linear programming, *European Journal of Operational Research*, 293(3), 863–879.

Armant, V. and Brown, K.N. (2014) Minimizing the driving distance in ride sharing systems, in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, 568–575.

Ashraf Javid, M. and Al-Khayyat, M.A. (2021) Factors affecting the student's intentions to choose carpooling: A case study in Oman, *Journal of the Chinese Institute of Engineers*, 44(4), 332–341.
Aydin, O.F., Gokasar, I. and Kalan, O. (2020) Matching algorithm for improving ride-sharing by incorporating route splits and social factors, *PLOS ONE*, 15(3), e0229674. https://doi.org/10.1371/journal.pone.0229674
Babar, Y. and Burtch, G. (2020) Examining the heterogeneous impact of ridehailing services on public transit use, *SSRN*. https://doi.org/10.2139/ssrn.3042805
Bahat, O. and Bekhor, S. (2016) Incorporating ridesharing in the static traffic assignment model, *Networks and Spatial Economics*, 16, 1125–1149.
Baldacci, R., Maniezzo, V. and Mingozzi, A. (2004) An exact method for the car pooling problem based on Lagrangean column generation, *Operations Research*, 52(3), 422–439.
Beckmann, M., McGuire, C.B. and Winsten, C.B. (1956) *Studies in the economics of transportation*. New Haven, CT, Yale University Press.
Ben-Akiva, M.E. and Lerman, S.R. (1985) *Discrete choice analysis: Theory and application to travel demand*. Cambridge, MA, MIT Press.
Ben-Akiva, M. and Bierlaire, M. (1999) Discrete choice methods and their applications to short term travel decisions, in Hall, R.W. (ed.) *Handbook of transportation science*. Boston, Springer US, 5–33. https://doi.org/10.1007/978-1-4615-5203-1_2
Blum, J.R. (1954) Multidimensional stochastic approximation methods, *The Annals of Mathematical Statistics*, 25(4), 737–744. https://doi.org/10.1214/aoms/1177728659
Bottom, J. and Chabini, I. (2001) Accelerated averaging methods for fixed point problems in transportation analysis and planning, in *TRISTAN IV – Triennial Symposium on Transportation Analysis*,. 69–74.
Bouzaiene-Ayari, B., Gendreau, M. and Nguyen, S. (1995) On the modelling of bus stops in transit networks, Part I: A survey of literature and new formulations. *Centre de Recherche sur les Transports*.
Boyce, D.E. (1984) Urban transportation network-equilibrium and design models: Recent achievements and future prospects, *Environment and Planning A*, 16(11), 1445–1474.
Braekers, K., Caris, A. and Janssens, G. (2014) Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots, *Transportation Research Part B: Methodological*, 67, 166–186. https://doi.org/10.1016/j.trb.2014.05.007
Branston, D. (1976) Link capacity functions: A review, *Transportation Research*, 10, 223–236.

Bučková, M., Krajčovič, M. and Edl, M. (2017) Computer simulation and optimization of transport distances of order picking processes, *Procedia Engineering*, 192, 69–74. https://doi.org/10.1016/j.proeng.2017.06.012

Bureau of Public Roads. (1964) *Traffic assignment manual for application with a large, high speed computer*. U.S. Department of Commerce, Bureau of Public Roads, Office of Planning, Urban Planning Division.

Cai, H., Wang, X., Adriaens, P. and Xu, M. (2019) Environmental benefits of taxi ride sharing in Beijing, *Energy*, 174, 503–508. https://doi.org/10.1016/j.energy.2019.02.166

Cantarella, G.E. and Cascetta, E. (1995) Dynamic processes and equilibrium in transportation networks: Towards a unifying theory, *Transportation Science*, 29(4), 305–329.

Cascetta, E., Nuzzolo, A., Russo, F. and Vitetta, A. (1996) A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks.

Cea, J., Zubieta, L., Bunster, J. and Florian, M. (1988) Optimal strategies and optimal routes in public transit assignment models: An empirical comparison, *Traffic Engineering and Control*, 29. Available at: https://search.proquest.com/docview/24974383

Cepeda, M., Cominetti, R. and Florian, M. (2006) A frequency-based assignment model for congested transit networks with strict capacity constraints: Characterization and computation of equilibria, *Transportation Research Part B: Methodological*, 40(6), 437–459. https://doi.org/10.1016/j.trb.2005.05.006

Chakirov, A. and Fourie, P.J. (2014) *Enriched Sioux Falls scenario with dynamic and disaggregate demand*. Arbeitsberichte Verkehrs- und Raumplanung, 978.

Chan, N.D. and Shaheen, S.A. (2012) Ridesharing in North America: Past, present, and future, *Transport Reviews*, 32(1), 93–112.

Chen, X. and Di, X. (2021) Ridesharing user equilibrium with nodal matching cost and its implications for congestion tolling and platform pricing, *Transportation Research Part C: Emerging Technologies*, 129, 103233. https://doi.org/10.1016/j.trc.2021.103233

Chen, Z., Liu, X.C. and Wei, R. (2019) Agent-based approach to analyzing the effects of dynamic ridesharing in a multimodal network, *Computers, Environment and Urban Systems*, 74, 126–135. https://doi.org/10.1016/j.compenvurbsys.2018.10.004

Chriqui, C. and Robillard, P. (1975) Common bus lines, *Transportation Science*, 9(2), 115–121. https://doi.org/10.1287/trsc.9.2.115

Clewlow, R.R. and Mishra, G.S. (2017) *Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the United States.* University of California, Davis, Institute of Transportation Studies.
Cominetti, R. and Correa, J. (2001) Common-lines and passenger assignment in congested transit networks, *Transportation Science*, 35(3), 250–267. https://doi.org/10.1287/trsc.35.3.250.10154
Concas, S. and Winters, P.L. (2007) Impact of carpooling on trip-chaining behavior and emission reductions, *Transportation Research Record*, 2010(1), 83–91.
Coogan, M.A. (2008) *Ground access to major airports by public transportation*. Washington, D.C., Transportation Research Board.
Cordeau, J. and Laporte, G. (2007) The dial-a-ride problem: Models and algorithms, *Annals of Operations Research*, 153(1), 29–46. https://doi.org/10.1007/s10479-007-0170-8
Daganzo, C.F. and Sheffi, Y. (1977) On stochastic models of traffic assignment, *Transportation Science*, 11(3), 253–274. Available at: http://www.jstor.org/stable/25767878
Davidson, K.B. (1978) The theoretical basis of a flow–travel time relationship for use in transportation planning, *Australian Road Research*, 8(1).
de Cea, J. and Fernandez, E. (1993) Transit assignment for congested public transport systems: An equilibrium model, *Transportation Science*, 27(2), 133–147. https://doi.org/10.1287/trsc.27.2.133
Department of Transport. (1985) *Traffic appraisal manual (TAM)*. London, His Majesty's Stationery Office (HMSO).
Di Febbraro, A., Gattorna, E. and Sacco, N. (2013) Optimization of dynamic ridesharing systems, *Transportation Research Record*, 2359(1), 44–50. https://doi.org/10.3141/2359-06
Dial, R.B. (1967) Transit pathfinder algorithm, *Highway Research Record*, 205.
Dial, R.B. (1971) A probabilistic multipath traffic assignment model which obviates path enumeration, *Transportation Research*, 5(2), 83–111. https://doi.org/10.1016/0041-1647(71)90012-8
Dias, F.F., Lavieri, P., Garikapati, V., Astroza, S., Pendyala, R. and Bhat, C. (2017) A behavioral choice model of the use of car-sharing and ride-sourcing services, *Transportation*, 44, 1307–1323. https://doi.org/10.1007/s11116-017-9797-8
Dong, Z. and Leng, M. (2021) Managing on-demand ridesharing operations: Optimal pricing decisions for a ridesharing platform, *International Journal of Production Economics*, 232, 107958.
Douglas, N. and Karpouzis, G. (2006) Estimating the passenger cost of train overcrowding.

Dowling, R., Skabardonis, A. and Alexiadis, V. (2004) *Traffic analysis toolbox, volume III: Guidelines for applying traffic microsimulation modeling software*. Federal Highway Administration (FHWA).

El Mouatasim, A. and Ettahiri, A. (2019) Accelerated Frank–Wolfe algorithm for convex optimization. [Conference paper / Working paper].

Farhan, J. and Chen, T.D. (2018) Impact of ridesharing on operational efficiency of shared autonomous electric vehicles, in *Transportation Research Board 97th Annual Meeting*. Available at: https://trid.trb.org/view/1496974

Ferguson, E. (1991) Ridesharing, firm size, and urban form, *Journal of Planning Education and Research*, 10(2), 131–142. https://doi.org/10.1177/0739456X9101000210

Ferguson, E. (1997) The rise and fall of the American carpool: 1970–1990, *Transportation*, 24(4), 349–376.

Fisk, C. (1980) Some developments in equilibrium traffic assignment, *Transportation Research Part B: Methodological*, 14(3), 243–255. https://doi.org/10.1016/0191-2615(80)90004-1

Florian, M. and Spiess, H. (1983) On binary mode choice/assignment models, *Transportation Science*, 17(1), 32–47. https://doi.org/10.1287/trsc.17.1.32

Frank, M. and Wolfe, P. (1956) An algorithm for quadratic programming, *Naval Research Logistics Quarterly*, 3(1–2), 95–110. https://doi.org/10.1002/nav.3800030109

Frohardt-Lane, S. (2012) Promoting a culture of driving: Rationing, car sharing, and propaganda in World War II, *Journal of American Studies*, 46(2), 337–355. https://doi.org/10.1017/S0021875812000072

Fukushima, M. (1984) A modified Frank–Wolfe algorithm for solving the traffic assignment problem, *Transportation Research Part B: Methodological*, 18(2), 169–177. https://doi.org/10.1016/0191-2615(84)90029-8

Gao, Z., Lam, W.H.K., Wong, S.C. and Yang, H. (2004) The convergence of equilibrium algorithms with non-monotone line search technique, *Applied Mathematics and Computation*, 148(1), 1–13. https://doi.org/10.1016/S0096-3003(02)00821-4

Gartner, N. (1977) Analysis and control of transportation networks by Frank–Wolfe decomposition, in [Book title unavailable], 591–623.

Geisberger, R., Luxen, D., Neubauer, S., Sanders, P. and Volker, L. (2009) Fast detour computation for ride sharing, *arXiv*. arXiv:0907.5269.

Ghoseiri, K., Haghani, A. and Hamedi, M. (2011) Real-time rideshare matching problem. University of Maryland, Department of Civil and Environmental Engineering.

Hamdouch, Y. and Lawphongpanich, S. (2008) Schedule-based transit assignment model with travel strategies and capacity constraints, *Transportation Research Part B: Methodological*, 42, 663–684. https://doi.org/10.1016/j.trb.2007.11.005

Hang, C., Liu, Z., Wang, Y., Hu, C., Su, Y. and Dong, Z. (2019) Sharing diseconomy: Impact of the subsidy war of ride-sharing companies on urban congestion, *Transportation Letters* , 22(5), 491–500. https://doi.org/10.1080/13675567.2019.1619677

Hao, Y. (2018) An integer programming model for dynamic taxi-sharing considering provider profit. *PhD thesis*. Available at: https://doi.org/10.13016/M2W37M00V

Henao, A. (2017) Impacts of ridesourcing – Lyft and Uber – on transportation including VMT, mode replacement, parking, and travel behavior. *PhD thesis*. Available at: https://www.proquest.com/dissertations-theses/impacts-ridesourcing-lyft-uber-on-transportation/docview/1899208739

Herbawi, W. and Weber, M. (2012) The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm, in *2012 IEEE Congress on Evolutionary Computation*, 1–8.

Hoogendoorn-Lanser, S., Nes, R. and Bovy, P. (2005) Path size and overlap in multimodal transport networks, in [Book title unavailable], 63–84. https://doi.org/10.1016/B978-008044680-6/50006-4

Hosni, H., Naoum-Sawaya, J. and Artail, H. (2014) The shared-taxi problem: Formulation and solution methods, *Transportation Research Part B: Methodological*, 70, 303–318.

Hou, L., Li, D. and Zhang, D. (2018) Ride-matching and routing optimisation: Models and a large neighbourhood search heuristic, *Transportation Research Part E: Logistics and Transportation Review*, 118, 143–162. https://doi.org/10.1016/j.tre.2018.07.003

Huang, H., Yang, H. and Bell, M.G.H. (2000) The models and economics of carpools, *Annals of Regional Science*, 34, 55–68.

Hunt, J.D. (1990) A logit model of public transport route choice, *ITE Journal*. Available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.435.4684

Javid, R., Nejat, A. and Salari, M. (2016) The environmental impacts of carpooling in the United States, in *Proceedings of the Transportation, Land and Air Quality Conference*.

Jayakrishnan, R., Tsai, W.K. and Chen, A. (1995) A dynamic traffic assignment model with traffic-flow relationships, *Transportation Research Part C: Emerging Technologies*, 3(1), 51–72. https://doi.org/10.1016/0968-090X(94)00015-W

Jia, Y., Xu, W. and Liu, X. (2017) An optimization framework for online ride-sharing markets, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 826–835.
Jiang, Y. and Szeto, W.Y. (2016) Reliability-based stochastic transit assignment: Formulations and capacity paradox, *Transportation Research Part B: Methodological*, 93, 181–206.
Kelly, K. (2007) Casual carpooling-enhanced, *Journal of Public Transportation*, 10(4), 119–130. https://doi.org/10.5038/2375-0901.10.4.6
Kleiner, A., Nebel, B. and Ziparo, V.A. (2011) A mechanism for dynamic ride sharing based on parallel auctions, in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*.
Krueger, R., Rashidi, T.H. and Rose, J.M. (2016) Preferences for shared autonomous vehicles, *Transportation Research Part C: Emerging Technologies*, 69, 343–355. https://doi.org/10.1016/j.trc.2016.06.015
Kumar, P. and Khani, A. (2021) An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access, *Transportation Research Part C: Emerging Technologies*, 122, 102891. https://doi.org/10.1016/j.trc.2020.102891
Kowshik, R.R., Kitamura, R. and Jovanis, P.P. (1996) *Evaluation of the Sacramento-area real-time rideshare matching field operational test: Final report*. Institute of Transportation Studies, University of California, Davis.
Lalos, P., Korres, A., Datsikas, C., Tombras, G. and Peppas, K. (2009) A framework for dynamic car and taxi pools with the use of positioning systems, in *Computation World 2009*. https://doi.org/10.1109/ComputationWorld.2009.55
Lam, W.H.K., Cheung, C. and Lam, C.F. (1999) A study of crowding effects at the Hong Kong light rail transit stations, *Transportation Research Part A: Policy and Practice*, 33(5), 401–415. https://doi.org/10.1016/S0965-8564(98)00050-0
Lampkin, W. and Saalmans, P.D. (1967) The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study, *Journal of the Operational Research Society*, 18(4), 375–397. https://doi.org/10.1057/jors.1967.70
Le Clerq, F. (1972) A public transport assignment method, 91–96.
LeBlanc, L.J. and Abdulaal, M. (1979) An efficient dual approach to the urban road network design problem, *Computers & Mathematics with Applications*, 5(1), 11–19.

LeBlanc, L.J., Morlok, E.K. and Pierskalla, W.P. (1975) An efficient approach to solving the road network equilibrium traffic assignment problem, *Transportation Research*, 9(5), 309–318.
Lee, D. and Nie, Y. (2001) Accelerating strategies and computational studies of the Frank–Wolfe algorithm for the traffic assignment problem, *Transportation Research Record*, 1771, 97–105. https://doi.org/10.3141/1771-13
Li, L., Negenborn, R.R. and De Schutter, B. (2013) A sequential linear programming approach for flow assignment in intermodal freight transport, in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 1224–1230.
Li, M., Di, X., Liu, H.X. and Huang, H. (2020) A restricted path-based ridesharing user equilibrium, *Journal of Intelligent Transportation Systems*, 24(4), 383–403. https://doi.org/10.1080/15472450.2019.1658525
Li, W., Huang, H. and Shang, H. (2020) Dynamic equilibrium commuting in a multilane system with ridesharing, *Physica A: Statistical Mechanics and its Applications*, 557, 124860.
Li, Y. and Chung, S.H. (2020) Ride-sharing under travel time uncertainty: Robust optimization and clustering approaches, *Computers & Industrial Engineering*, 149, 106601. https://doi.org/10.1016/j.cie.2020.106601
Lin, J. and Ban, Y. (2017) Comparative analysis on topological structures of urban street networks, *ISPRS International Journal of Geo-Information*, 6, 295. https://doi.org/10.3390/ijgi6100295
Liu, Y., Bunker, J. and Ferreira, L. (2010) Transit users route-choice modelling in transit assignment: A review, *Transport Reviews*, 30(6), 753–769. https://doi.org/10.1080/01441641003744261
Liu, Z., Miwa, T., Zeng, W., Bell, M.G.H. and Morikawa, T. (2018) Shared autonomous taxi system and utilization of collected travel-time information, *Journal of Advanced Transportation*, 2018, Article ID 8919721.
Ma, S., Zheng, Y. and Wolfson, O. (2013) T-share: A large-scale dynamic taxi ridesharing service, in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 410–421.
Masoud, N. and Jayakrishnan, R. (2017a) A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system, *Transportation Research Part B: Methodological*, 106, 218–236.
Masoud, N. and Jayakrishnan, R. (2017b) A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem, *Transportation Research Part B: Methodological*, 99, 1–29. https://doi.org/10.1016/j.trb.2017.01.004

Masson, R., Lehuédé, F. and Péton, O. (2014) The dial-a-ride problem with transfers, *Computers & Operations Research*, 41, 12–23. https://doi.org/10.1016/j.cor.2013.07.020

Mathew, T.V. and Rao, K.K. (2007) Fundamental parameters of traffic flow, in *Introduction to Transportation Engineering*, National Program on Technical Education and Learning (NPTEL), Mumbai, India, 1–8.

MATSim (2020) *Sioux Falls, SD, USA scenario*. Available at: https://matsim.org/gallery/sioux-falls/

McFadden, D. (1977) Modelling the choice of residential location, *Cowles Foundation for Research in Economics, Yale University*. Available at: https://econpapers.repec.org/RePEc:cwl:cwldpp:477

McFadden, D. (2001) Economic choices, *American Economic Review*, 91(3), 351–378.

Minett, P. and Pearce, J. (2011) Estimating the energy consumption impact of casual carpooling, *Energies*, 4(1), 126–139. https://doi.org/10.3390/en4010126

Mousavi, S.R., Yazdiani, A. and Shafahi, Y. (2022) Stochastic multi-class traffic assignment for autonomous and regular vehicles in a transportation network, *arXiv preprint*, arXiv:2212.05495.

Najmi, A., Rashidi, T.H. and Liu, W. (2020) Ridesharing in the era of Mobility as a Service (MaaS): An activity-based approach with multimodality and intermodality, *arXiv preprint*, arXiv:2002.11712.

Nguyen, S. and Pallottino, S. (1988) Equilibrium traffic assignment for large-scale transit networks, *European Journal of Operational Research*, 37(2), 176–186. https://doi.org/10.1016/0377-2217(88)90327-X

Nielsen, O.A. (2000) A stochastic transit assignment model considering differences in passengers utility functions, *Transportation Research Part B: Methodological*, 34(5), 377–402. https://doi.org/10.1016/S0191-2615(99)00029-6

Nielsen, O.A. and Frederiksen, R.D. (2006) Optimisation of timetable-based, stochastic transit assignment models based on MSA, *Annals of Operations Research*, 144, 263–285. https://doi.org/10.1007/s10479-006-0012-0

Nikolic, M. and Teodorovic, D. (2021) User equilibrium traffic assignment: k paths subtracting–adding algorithm, *arXiv preprint*, arXiv:2107.04018.

Nourinejad, M. and Roorda, M.J. (2016) Agent-based model for dynamic ridesharing, *Transportation Research Part C: Emerging Technologies*, 64, 117–132.

Ntziachristos, L., Samaras, Z., Eggleston, S., Gorissen, N., Hassel, D. and Hickman, A. (2000) *COPERT III: Computer programme to calculate emissions from road transport – Methodology and emission factors (Version 2.1)*. Copenhagen, European Environment Agency (EEA).

Nuzzolo, A. (2002) Schedule-based transit assignment models, in Lam, W.H.K. and Bell, M.G.H. (eds.) *Advanced modeling for transit operations and service planning*. Bingley, Emerald Group Publishing Limited.

Oliphant, M., Greenberg, A., Boenau, R. & Raw, J. (2013) Fill those empty seats! Public Roads. 77 (2).

Ortúzar, J. de D. and Willumsen, L. (2011) *Modelling transport*. 4th ed. Chichester, Wiley. https://doi.org/10.1002/9781119993308.fmatter

Pang, J., Zhang, M., Dessouky, M.M. and Gu, W. (2020) Modeling e-hailing and car-pooling services in a coupled morning–evening commute framework. [Unpublished manuscript].

Papoutsis, P., Fennia, S., Bridon, C. and Duong, T. (2021) Relaxing door-to-door matching reduces passenger waiting times: A workflow for the analysis of driver GPS traces in a stochastic carpooling service, *Transportation Engineering*, 4, 100061. https://doi.org/10.1016/j.treng.2021.100061

Polyak, B. (1990) New method of stochastic approximation type, *Automation and Remote Control*. [Details unavailable].

Prifti, V., Dervishi, I., Dhoska, K., Markja, I. and Pramono, A. (2020) Minimization of transport costs in an industrial company through linear programming, in *IOP Conference Series: Materials Science and Engineering*, 909(1), 012040.

Pursula, M. and Weurlander, M. (1999) Modeling level-of-service factors in public transportation route choice, *Transportation Research Record*, 1669(1), 30–37. https://doi.org/10.3141/1669-04

Rayle, L., Dai, D., Chan, N., Cervero, R. and Shaheen, S. (2016) Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco, *Transport Policy*, 45, 168–178. https://doi.org/10.1016/j.tranpol.2015.10.004

Robbins, H. and Monro, S. (1951) A stochastic approximation method, *The Annals of Mathematical Statistics*, 22(3), 400–407. https://doi.org/10.1214/aoms/1177729586

Savelsbergh, M.W.P. and Sol, M.M. (1995) The general pickup and delivery problem, *Transportation Science*, 29, 17–29.

Saw, K., Joshi, G. and Katti, B.K. (2015) Measures of travel time variability and reliability and its improvement strategies, *Urban Transport Journal*, 3.

Schaller, B. (2017) *Unsustainable? The growth of app-based ride services and traffic, travel and the future of New York City*. Schaller Consulting.
Schmöcker, J. (2006) Dynamic capacity constrained transit assignment. *Thesis*.
Sheffi, Y. (1985) *Urban transportation networks: Equilibrium analysis with mathematical programming methods*. Englewood Cliffs, NJ, Prentice-Hall.
Sheffi, Y. and Powell, W.B. (1982) An algorithm for the equilibrium assignment problem with random link times, *Networks*, 12(2), 191–207. https://doi.org/10.1002/net.3230120209
Sheldon, T.L. and Dua, R. (2024) Impacts of ride-hailing on energy and the environment: A systematic review, *Environmental Research Letters*, 19(4), 043004.
Small, K. (1982) The scheduling of consumer activities: Work trips, *American Economic Review*, 72, 467–479.
Smock, R. (1962) An iterative assignment approach to capacity restraint on arterial networks, in [Conference name unspecified], p. 347.
Song, F., Li, R., Zhou, H., Zhang, H. and You, I. (2014) A case study for potential carpooling in Beijing, in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 172–177.
Spiess, H. and Florian, M. (1989) Optimal strategies: A new assignment model for transit networks, *Transportation Research Part B: Methodological*, 23(2), 83–102. https://doi.org/10.1016/0191-2615(89)90034-9
Sun, M. (2023) A day-to-day dynamic model for mixed traffic flow of autonomous vehicles and inertial human-driven vehicles, *Transportation Research Part E: Logistics and Transportation Review*, 173, 103113.
Sun, S. and Szeto, W.Y. (2021) Multi-class stochastic user equilibrium assignment model with ridesharing: Formulation and policy implications, *Transportation Research Part A: Policy and Practice*, 145, 203–227. https://doi.org/10.1016/j.tra.2020.12.011
Tamannaei, M. and Irandoost, I. (2019) Carpooling problem: A new mathematical model, branch-and-bound, and heuristic beam search algorithm, *Journal of Intelligent Transportation Systems*, 23(3), 203–215. https://doi.org/10.1080/15472450.2018.1484739
Tao, C. (2007) Dynamic taxi-sharing service using intelligent transportation system technologies, in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 3209–3212.
Thao, V.T., Imhof, S. and von Arx, W. (2021) Integration of ridesharing with public transport in rural Switzerland: Practice and outcomes, *Transportation Research Interdisciplinary Perspectives*, 10, 100340.

Tikoudis, I., Martinez, L., Farrow, K., Bouyssou, C.G., Petrik, O. and Oueslati, W. (2021) Ridesharing services and urban transport CO_2 emissions: Simulation-based evidence from 247 cities, *Transportation Research Part D: Transport and Environment*, 97, 102923.

Train, K.E. (2009) *Discrete choice methods with simulation*. Cambridge, Cambridge University Press.

Transportation Networks for Research Core Team (2023) *Transportation networks for research*. Available at: https://github.com/bstabler/TransportationNetworks

Villagran, A. (2019) *Microtransit: An overview on pilot projects and lessons learned*. Minneapolis, MN. Available at: https://sharedusemobilitycenter.org/wp-content/uploads/2019/11/NovConvening_Villagran.pdf

Wang, X., Dessouky, M. and Ordonez, F. (2016) A pickup and delivery problem for ridesharing considering congestion, *Journal of Transportation Safety & Security*, 8(5), 259–269. https://doi.org/10.1179/1942787515Y.0000000023

Wang, X., Wang, J., Guo, L., Liu, W. and Zhang, X. (2021) A convex programming approach for ridesharing user equilibrium under fixed driver/rider demand, *Transportation Research Part B: Methodological*, 149, 33–51. https://doi.org/10.1016/j.trb.2021.04.007

Wang, X., Yang, H. and Zhu, D. (2018) Driver–rider cost-sharing strategies and equilibria in a ridesharing program, *Transportation Science*, 52(4), 868–881. https://doi.org/10.1287/trsc.2017.0801

Wang, X. (2013) *Optimizing ride matches for dynamic ride-sharing systems*. [Thesis/Report].

Wang, Y., Wang, S., Wang, J., Wei, J. and Wang, C. (2020) An empirical study of consumers intention to use ride-sharing services: Using an extended technology acceptance model, *Transportation*, 47, 397–415.

Wang, Z., Li, L., Tang, J., Xie, B. and Ge, J. (2022) Effect of ride sharing on air quality: Evidence from Shenzhen, China, *Journal of Applied Economics*, 25, 197–219. https://doi.org/10.1080/15140326.2021.2023421

Wardrop, J.G. (1952) Some theoretical aspects of road traffic research, *Proceedings of the Institution of Civil Engineers*, 1(3), 325–362. https://doi.org/10.1680/ipeds.1952.11259

Wei, B., Saberi, M., Zhang, F., Liu, W. and Waller, S.T. (2020) Modeling and managing ridesharing in a multi-modal network with an aggregate traffic representation: A doubly dynamical approach, *Transportation Research Part C: Emerging Technologies*, 117, 102670.

Winter, S. and Nittel, S. (2006) Ad hoc shared-ride trip planning by mobile geosensor networks, *International Journal of Geographical Information Science*, 20(8), 899–916.

Wong, K.I., Wong, S.C., Bell, M.G.H. and Yang, H. (2005) Modeling the bilateral micro-searching behavior for urban taxi services using the absorbing Markov chain approach, *Journal of Advanced Transportation*, 39(1), 81–104. https://doi.org/10.1002/atr.5670390107

Wong, K.I., Wong, S.C. and Yang, H. (2001) Modeling urban taxi services in congested road networks with elastic demand, *Transportation Research Part B: Methodological*, 35(9), 819–842. https://doi.org/10.1016/S0191-2615(00)00021-7

Wong, R.C.P., Szeto, W.Y. and Wong, S.C. (2014) A cell-based logit-opportunity taxi customer-search model, *Transportation Research Part C: Emerging Technologies*, 48, 84–96. https://doi.org/10.1016/j.trc.2014.08.010

Wong, R.C.P., Szeto, W.Y. and Wong, S.C. (2015a) Behavior of taxi customers in hailing vacant taxis: A nested logit model for policy analysis, *Journal of Advanced Transportation*, 49(8), 867–883. https://doi.org/10.1002/atr.1307

Wong, R.C.P., Szeto, W.Y. and Wong, S.C. (2015b) A two-stage approach to modeling vacant taxi movements, *Transportation Research Procedia*, 7, 254–275. https://doi.org/10.1016/j.trpro.2015.06.014

Wu, J.H., Florian, M. and Marcotte, P. (1994) Transit equilibrium assignment: A model and solution algorithms, *Transportation Science*, 28(3), 193–203. https://doi.org/10.1287/trsc.28.3.193

Xia, J., Curtin, K.M., Li, W. and Zhao, Y. (2015) A new model for a carpool matching service, *PLOS ONE*, 10(6), e0129257. https://doi.org/10.1371/journal.pone.0129257

Xing, X., Warden, T., Nicolai, T. and Herzog, O. (2009) Smize: A spontaneous ride-sharing system for individual urban transit, in *German Conference on Multiagent System Technologies*, 165–176.

Xu, H., Pang, J., Ordóñez, F. and Dessouky, M. (2015) Complementarity models for traffic equilibrium with ridesharing, *Transportation Research Part B: Methodological*, 81, 161–182.

Yan, L., Luo, X., Zhu, R., Santi, P., Wang, H., Wang, D., Zhang, S. and Ratti, C. (2020) Quantifying and analyzing traffic emission reductions from ridesharing: A case study of Shanghai, *Transportation Research Part D: Transport and Environment*, 89, 102629. https://doi.org/10.1016/j.trd.2020.102629

Yang, H. and Wong, S.C. (1998) A network model of urban taxi services, *Transportation Research Part B: Methodological*, 32(4), 235–246. https://doi.org/10.1016/S0191-2615(97)00042-8
Yen, J.Y. (1971) Finding the k shortest loopless paths in a network, *Management Science*, 17(11), 712–716.
Yen, J.Y. (1972) Finding the lengths of all shortest paths in N-node nonnegative-distance complete networks using ½N^3 additions and N^3 comparisons, *Journal of the ACM*, 19(3), 423–424.
Yu, B., Ma, Y., Xue, M., Tang, B., Wang, B., Yan, J. and Wei, Y. (2017) Environmental benefits from ridesharing: A case of Beijing, *Applied Energy*, 191, 141–152.
Yu, Y., Han, K. and Ochieng, W. (2020) Day-to-day dynamic traffic assignment with imperfect information, bounded rationality and information sharing, *Transportation Research Part C: Emerging Technologies*, 114, 59–83.
Zha, L., Yin, Y. and Xu, Z. (2018) Geometric matching and spatial pricing in ride-sourcing markets, *Transportation Research Part C: Emerging Technologies*, 92, 58–75. https://doi.org/10.1016/j.trc.2018.04.015
Zhang, C., Xie, J., Wu, F., Gao, X. and Chen, G. (2020) Pricing and allocation algorithm designs in dynamic ridesharing system, *Theoretical Computer Science*, 803, 94–104. https://doi.org/10.1016/j.tcs.2019.05.045
Zhang, D. and Nagurney, A. (1996) On the local and global stability of a travel route choice adjustment process, *Transportation Research Part B: Methodological*, 30(4), 245–262.
Zhang, H., Chen, J., Li, W., Song, X. and Shibasaki, R. (2020) Mobile phone GPS data in urban ride-sharing: An assessment method for emission reduction potential, *Applied Energy*, 269, 115038. https://doi.org/10.1016/j.apenergy.2020.115038
Zhang, J., Pourazarm, S., Cassandras, C.G. and Paschalidis, I. (2016) The price of anarchy in transportation networks by estimating user cost functions from actual traffic data, in *IEEE Conference on Decision and Control (CDC)*. https://doi.org/10.1109/CDC.2016.7798364
Zhang, X., Shi, J., Zhao, R., Deng, F., Li, Q., Zhang, J., Sun, C., Miao, G. and Song, S. (2018) Research and application of overall energy consumption optimization technology under low oil price, in *SPE Middle East Artificial Lift Conference and Exhibition*, Paper D022S015R001. https://doi.org/10.2118/192473-MS
Zheng, Z., Rasouli, S. and Timmermans, H. (2018) Modeling taxi driver anticipatory behavior, *Computers, Environment and Urban Systems*, 69, 133–141.
Zhong, L., Zhang, K., Nie, Y.M. and Xu, J. (2020) Dynamic carpool in morning commute: Role of high-occupancy-vehicle (HOV) and high-occupancy-

toll (HOT) lanes, *Transportation Research Part B: Methodological*, 135, 98–119.
Zhou, B., Xu, M., Meng, Q. and Huang, Z. (2017) A day-to-day route flow evolution process towards the mixed equilibria, *Transportation Research Part C: Emerging Technologies*, 82, 210–228.
Zhou, Z. and Roncoli, C. (2022) A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion, *Transportation Research Part C: Emerging Technologies*, 139, 103658.
Zou, M., Chen, X., Yu, H., Tong, Y., Huang, Z., Li, M. and Zou, H. (2013) Dynamic transportation planning and operations: Concept, framework and applications in China, *Procedia – Social and Behavioral Sciences*, 96, 2332–2343.

APPENDIX A

Variable and Equation look up tables

Appendix A1

Variable Look up table

Variable	Description
t	Time horizon (or time window)
d	Simulation day (or simulation iteration)
m	Mode of transport: 1 = Ridesharing (Driver) 2 = Ridesharing / Taxi-sharing (Passenger) 3 = Public Transport 4 = Private Transport
$f_{i,j,m}^{t,d}$	Number of trips from origin *i* to destination *j* at time *t* and day *d* with mode *m* (Mode flow)
$U_{i,j,m}^{t,d}$	Utility function (systematic utility or proposed mode cost function) of mode *m*
$\bar{U}_{i,j,m}^{t,(d)}$	Perceived mode cost of mode *m*
θ	Scale of the Gumbel distribution
ϕ	Scale of the Gumbel distribution at route choice level
λ	Weight of past days' experienced costs
t_0	Free-flow travel time per unit distance
V	Link flow (Volume)
C	Link capacity
x_a	Equilibrium flow in link *a*
α, β	Parameters for calibration (default 0.15, 4)
VKTS	Vehicle Kilometres Travelled Saved
C_k	Capacity of link *k*
$x_k^{t,d}$	Total link flow
$\hat{x}_k^{t,d}$	Existing link flow
$x'^{t,d}_k$	New vehicle flows on link *k*
$RTT_r^{t,d}$	Actual route travel time for route *r*
$LTT_k^{t,d}$	Actual link *k* travel time
$n_r^{t*,d}$	Number of horizons demand of route *r* persists

Variable	Description
t_{Length}	Length of the time horizon (minutes)
C_s	Travel cost of route section *s*
$\tilde{V}_s$	Users already occupying vehicle (crowding factor)
K_s	Vehicle capacity (practical, for crowding)
f_p	Ridesharing fare (passenger fare)
β_{VoT}	Value of Time coefficient
ρ	Surge pricing ratio
$\dot{f}_i$	Number of vacant taxis at node *i*
$X_{ijk1l1k2l2p}$	Decision variable: number of matched trips
R_{ij}	Rate of Return
E_j	Relative Passenger Demand
D_j^c	Intra-zonal circulation distance
D_{ij}^t	Cross-zonal travel distance
$\beta^E \beta^t \beta^C \beta^R$	Coefficients for Wong's probabilistic function

Appendix A2

Equation Look up table

Equation/Formulation/Algorithm		Equation Name / Chapter / Page
given minimize subject to	$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$ $f(\vec{x})$ $g_1(\vec{x}) \leq b_1$ $g_2(\vec{x}) \leq b_2$ $g_3(\vec{x}) \leq b_3$	Standard Optimisation Problem Chapter 3 / Page 60
Driver earliest departure Passenger earliest departure Driver latest arrival Passenger latest arrival	$T_d^0 - T \leq 0$ $T_p^0 - T_{(O_d,O_p)} - T \leq 0$ $T + T_{(O_d,O_p)} + T_{(O_p,D_p)} + T_{(D_p,D_d)} \leq T_d^1$ $T + T_{(O_d,O_p)} + T_{(O_p,D_p)} \leq T_p^1$	Time Constraints Chapter 3 / Page 63
Driver cost Passenger cost System cost	$\left[C_{(O_d,O_p)} + C_{(O_p,D_p)} + C_{(D_p,D_d)} - F_{d,p}^d\right] - C_{(O_d,D_d)} \leq 0$ $\left[F_{d,p}^p + C'_{(O_p,D_p)}\right] - C_{(O_p,D_p)} \leq 0$ $\left[C_{(O_d,O_p)} + C_{(O_p,D_p)} + C_{(D_p,D_d)}\right] - \left[C_{(O_d,D_d)} + C_{(O_p,D_p)}\right] \leq 0$	Cost Constraints Chapter 3 / Page 64
given Maximize	$x_{dp} \in \{0,1\} \quad \forall p \in P, d \in D$ $\sum_{d \in D} \sum_{p \in P} w_{dp} x_{dp}$	Binary Integer Programming Chapter 3 / Page 68

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
subject to $\sum_{p \in P} x_{dp} \leq 1 \quad \forall\, p \in P,$ $\sum_{d \in D} x_{dp} \leq 1 \quad \forall\, d \in D,$	
Driver cost Pattern #1 $\left[C_{(o_d,o_{p1})} + C_{(o_{p1},D_{p1})} + C_{(D_{p1},o_{p2})} + C_{(o_{p2},D_{p2})} + C_{(D_{p2},D_d)} - F^d_{d,p1,p2}\right] - C_{(o_d,D_d)} \leq 0$ Pattern #2 $\left[C_{(o_d,o_{p1})} + C_{(o_{p1},o_{p2})} + C_{(o_{p2},D_{p1})} + C_{(D_{p1},D_{p2})} + C_{(D_{p2},D_d)} - F^d_{d,p1,p2}\right] - C_{(o_d,D_d)} \leq 0$ Pattern #3 $\left[C_{(o_d,o_{p1})} + C_{(o_{p1},o_{p2})} + C_{(o_{p2},D_{p2})} + C_{(D_{p2},D_{p1})} + C_{(D_{p1},D_d)} - F^d_{d,p1,p2}\right] - C_{(o_d,D_d)} \leq 0$	Cost constraints (Driver – Multiple Passengers) Chapter 3 / Page 72
1st passenger cost Pattern #1 $\left[F^{p1}_{d,p1,p2} + C'_{(o_{p1},D_{p1})}\right] - C_{(o_{p1},D_{p1})} \leq 0$ Pattern #2 $\left[F^{p1}_{d,p1,p2} + C'_{(o_{p1},o_{p2})} + C'_{(o_{p2},D_{p1})}\right] - C_{(o_{p1},D_{p1})} \leq 0$ Pattern #3 $\left[F^{p1}_{d,p1,p2} + C'_{(o_{p1},o_{p2})} + C'_{(o_{p2},D_{p2})} + C'_{(D_{p2},D_{p1})}\right] - C_{(o_{p1},D_{p1})} \leq$	
2nd passenger cost Pattern #1 $\left[F^{p2}_{d,p1,p2} + C'_{(o_{p2},D_{p2})}\right] - C_{(o_{p2},D_{p2})} \leq 0$ Pattern #2 $\left[F^{p2}_{d,p1,p2} + C'_{(o_{p2},D_{p1})} + C'_{(D_{p1},D_{p2})}\right] - C_{(o_{p2},D_{p2})} \leq 0$ Pattern #3 $\left[F^{p2}_{d,p1,p2} + C'_{(o_{p2},D_{p2})}\right] - C_{(o_{p2},D_{p2})} \leq 0$	

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
Driver $C_{(N_t,O_{p1})} + C_{(O_{p1},D_{p1})} + C_{(D_{p1},O_{p2})} + C_{(O_{p2},D_{p2})} - F^{v}_{p1,p2,m} \leq 0$	Cost constraint for Taxi Driver Chapter 3 / Page 76
$R_{ij} = \frac{-C^t_{ij} - C^c_j + \bar{P}_j}{T^t_{ij} + T^c_j + \bar{T}^t_j}$	Wong's Rate of Return Chapter 3 / Page 80
$p(j\|i) = \begin{cases} 0 & ,if\ j \notin H_i \cup \{i\}; \\ \frac{\exp(\beta^E E_j + \beta^t D^t_{ij} + \beta^c D^c_j + \beta^R R_{ij})}{\sum_{n\epsilon H_i \cup \{i\}} \exp(\beta^E E_n + \beta^t D^t_{in} + \beta^c D^c_n + \beta^R R_{in})} & ,if\ j \in H_i \cup \{i\}, \end{cases}$	Wong's Probabilistic Function Chapter 3 / Page 80
$W_s = \frac{0.5}{\sum_{l\varepsilon L_s} f_l}$	Expected Waiting Time Chapter 4 / Page 89
$.T_s = E(W_s) + \sum_{i\epsilon S} t_i H_i(S)$	Expected Travel Time (Routes) Chapter 4 / Page 92
$t_S = \frac{\sum_{j\in S} f_j t_j}{\sum_{i\in S} f_i}$	Expected Travel Time (Multiple Frequencies) Chapter 4 / Page 94
$C_s = \bar{t}_s + \frac{\alpha}{f_l} + \varphi\left(\frac{\tilde{V}_s}{K_s}\right)$ $w^i_l = \frac{\alpha}{f_l} + \varphi\left(\frac{\tilde{V}_s}{K_s}\right)$	De Cea's Cost Function Chapter 4 / Page 96
$t = t_0\, exp\left(\frac{V}{C}\right)$	Smock's Volume-Delay Function Chapter 5 / Page 109

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
$T(V) = \begin{cases} d/S_0 & V < F_1 \\ d/S(V) = \dfrac{d}{S_0 + SS_{01}F_1 - SS_{01}V} & F_1 \le V \le F_2 \\ d/S_1 + (V/F_2 - 1)/8 & V > F_2 \end{cases}$ $SS_{01} = \dfrac{S_0 - S_1}{F_1 - F_2}$	UK DoT Time-Flow Function Chapter 5 / Page 110
$t = t_0\left[1 + \alpha\left(\dfrac{V}{C}\right)^{\beta}\right]$	US BPR Volume-Delay Function Chapter 5 / Page 110
$t = t_o\left\{1 + 0.25r_f\left[z + \sqrt{z^2 + 8\,J_D x/r_f}\right]\right\}$	Akçelik's Volume-Delay Function Chapter 5 / Page 111
Minimize Z $= \sum_a \int_0^{x_a} t_a(x_a)dx,$ Subject to, $\sum_k f_k^{rs} = q_{rs}: \forall r, s$ $x_a = \sum_r \sum_s \sum_k \delta_{a,k}^{rs} f_k^{rs}: \forall a$ $f_k^{rs} \ge 0: \forall k, r, s$ $x_a \ge 0: a \in A$	Beckmann's UE Formulation Chapter 5 / Page 115
Minimize Z $= \sum_a x_a t_a(x_a)dx,$ Subject to, $\sum_k f_k^{rs} = q_{rs}: \forall r, s$ $x_a = \sum_r \sum_s \sum_k \delta_{a,k}^{rs} f_k^{rs}: \forall a$ $f_k^{rs} \ge 0: \forall k, r, s$ $x_a \ge 0: a \in A$	Sheffi's SO Formulation Chapter 5 / Page 116
$v_a^n = v_a^{n-1} + F_a$	Accumulate Flows Chapter 5 / Page 118
$v_a^n = (1 - \emptyset_n)v_a^{n-1} + \emptyset_n F_a$ $= v_a^{n-1} + \emptyset_n(F_a - v_a^{n-1})$ $with\ 0 \le \emptyset_n \le 1$	MSA Flow Calculation Chapter 5 / Page 120

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
$\emptyset_n = \left\{1, \frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{n}, \frac{1}{n}, , \dots, \frac{1}{n}, \dots\right\}$	Zhang's Sequence Chapter 5 / Page 121
$u_a^n = v_a^{n-1} + \emptyset_{n,}(v_a^n - v_a^{n-1})$	Frank-Wolfe Linear Function Chapter 5 / Page 122
Writing the objective function in terms of u_a^n: Minimize $Z \quad = \sum_a \int_0^{x_a} t_a(u_a^n)dx$, Find $\emptyset_{n,}$ by taking $\frac{dz}{d\emptyset} = 0$ Calculates u_a^n with $\emptyset_{n,}$, then $v_a^{n+1} = u_a^n$. If $\lvert v_a^{n+1} - v_a^n \rvert \leq \varepsilon \ \forall a$ then STOP, else set $n = n + 1$and go to 2nd step.	Frank-Wolfe Algorithm Chapter 5 / Page 123
$U_{ir} = V(-c_r) + \varepsilon_{ir}$	Utility function (stochastic) Chapter 5 / Page 124
$p_{i,k} = \Pr(U_{ir}(-c_r) \geq U_{ik}(-c_r), \forall k \neq r)$	Probabilistic Function Chapter 5 / Page 125
$w_{ij} \begin{cases} W_i e^{(-c_{ij}\theta)} & C_{o,i} < C_{o,j} \\ 0 & C_{o,i} < C_{o,j} \end{cases}$, $W_j = \sum_n w_{nj}$	Dial's Forward pass Chapter 5/126
Initialize all links and nodes flow to zero: $v_i, v_{ij} = 0 \ \forall i \ \forall j$ Set the destination node as node i , and equal to the OD demand: $v_i = V_{od}$ For all links (n, i) that going to node i, set: $v_{ni} = v_i \frac{w_{ni}}{W_i}$	Dial's Backward pass Chapter 5/127
$v_r = V_{ij} \frac{e^{-\theta t_r}}{\sum_{k \in R(i.j)} e^{-\theta t_k}}, \qquad \forall r \in R(i.j)$	MNL Function

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
	Chapter 5 / Page 128
$CF_r^{ij} = \beta ln \sum_{k \epsilon R(i.j)} \left(\frac{L_{rk}^{ij}}{\sqrt{L_r^{ij} \cdot L_k^{ij}}} \right)^{\gamma}, \qquad \forall r \in R(i.j)$	Commonality Function Chapter 5 / Page 128
$v_r = V_{ij} \frac{e^{-\left(t_r - CF_r^{ij}\right)}}{\sum_{k \in R(i.j)} e^{-\left(t_k - CF_k^{ij}\right)}}, \qquad \forall r \in R(i.j)$	C-Logit Function Chapter 5 / Page 128
Minimize Z_{MNL} $= \sum_{a \epsilon A} \int_0^{x_a = \sum_{ij \epsilon OD} \sum_{k \in R(i.j)} v_k^{ij} \delta_{ka}^{ij}} t_a(w) dw + \frac{1}{\theta} \sum_{ij \epsilon OD} \sum_{k \in R(i.j)} v_k^{ij} \ln v_k^{ij}$, Subject to $\sum_{k \in R(i.j)} v_k^{ij} = V_{ij}, \quad \forall ij \epsilon OD,$ $v_k^{ij} \geq 0, \quad \forall k \in R(i.j), \forall ij \epsilon OD$	SUE Optimisation Problem Chapter 5 / Page 129
$P_n(i) = \frac{exp(U_{in})}{\sum_{j=1}^{k} exp(U_{jn})},$	MNL Function Chapter 6 / Page 142
$P_{in} = P_{mn} P_{i\|m,n}, \qquad IV_{mn} = \ln \left(\sum_{j \epsilon m} \exp \left(\frac{V_{jn}}{\lambda_m} \right) \right)$	Nested Logit Function Chapter 6 / Page 142
$\begin{pmatrix} f_{i,j,1}^{t,d} \\ f_{i,j,2}^{t,d} \\ f_{i,j,3}^{t,d} \\ f_{i,j,4}^{t,d} \end{pmatrix} = \begin{pmatrix} \exp\left(-\theta U_{i,j,1}^{t,d}\right) \\ \exp\left(-\theta U_{i,j,2}^{t,d}\right) \\ \exp\left(-\theta U_{i,j,3}^{t,d}\right) \\ \exp\left(-\theta U_{i,j,4}^{t,d}\right) \end{pmatrix} \bullet \frac{f_{i,j}^{t,d}}{\sum_m \exp\left(-\theta U_{i,j,m}^{t,d}\right)}$	Revised MNL Function Chapter 6 / Page 147
Ridesharing as a driver: $U_{i,j,1}^{t,d} = \alpha + \beta_f D_{i\,j\,t\,d}^{D} + \beta_{VOT} TT_{i\,j\,t\,d}^{D} - f_{P(1)} - f_{P(2)} + f_{fee}$ Ridesharing/Taxi-sharing as a passenger: $U_{i,j,2}^{t,d} = \beta_{VOT}\left(WT_{i\,j\,t\,d}^{P} + TT_{i\,j\,t\,d}^{P}\right) + f_p + f_{fee}$	Proposed Mode Cost Functions Chapter 6 / Page 148

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
Public Transport: $U_{i,j,3}^{t,d} = \beta_{VOT}\left(WK_i + WT_b^B + TT_{i\,j\,t\,d}^B\right) + f_b + C_{i\,j\,t\,d\,b}$ Self-driving (Private Transport): $U_{i,j,4}^{t,d} == \alpha + \beta_f D_{i\,j\,t\,d}^S + \beta_{VOT} TT_{i\,j\,t\,d}^s$	
$sd^{t,d} = \begin{Bmatrix} f_{1,1,1}^{t,d} & \cdots & f_{i,1,1}^{t,d} \\ \cdots & \cdots & \cdots \\ f_{1,j,1}^{t,d} & \cdots & f_{i,j,1}^{t,d} \end{Bmatrix}, sp1^{t,d} = \begin{Bmatrix} f_{1,1,2}^{t,d} & \cdots & f_{i,1,2}^{t,d} \\ \cdots & \cdots & \cdots \\ f_{1,j,2}^{t,d} & \cdots & f_{i,j,2}^{t,d} \end{Bmatrix}, sp2^{t,d} = \begin{Bmatrix} f_{1,1,2}^{t,d} & \cdots & f_{i,1,2}^{t,d} \\ \cdots & \cdots & \cdots \\ f_{1,j,2}^{t,d} & \cdots & f_{i,j,2}^{t,d} \end{Bmatrix}$	Matching Possibility Chapter 6 / Page 152
given $x_{dp} \in R$ minimize $f(x_1, x_2, \ldots x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$ subject to $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$ $a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2$ $\cdots$ $a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m$ $x_n \geq 0 \quad \forall\, n \in N$	Linear Programming Chapter 6 / Page 152
Objective function $\max \sum_{i,j}\sum_{k,l}\sum_{p} X_i$ Subject to $\sum_{(i,j)\in D} X_{ijk_1l_1k_2\,l_2p}$ $\leq F_{(k,l)}\ \forall(k,l)\ \in$ $\sum_{(k,l)\in P} X_{ijk_1l_1k_2\,l_2p} \leq F_{(i,j)}\ \forall(i,j)\ \in D$ $X_{ijk_1l_1k_2\,l_2p}\ \geq 0\ \forall(i,j) \in D\ ,\ \forall(k,l) \in P$	Proposed Optimisation Problem (Ridesharing) Chapter 6 / Page 154
Objective function $\max \sum_{i,j}\sum_{k,l}\sum_{p} X_{ik_1l_1k_2\,l_2p} VKTS_{ik_1l_1k_2\,l_2p}$ Subject to $\sum_{(i,j)\in D} X_{ik_1l_1k_2\,l_2p} \leq F_{(k,l)}\ \forall(k,l)\ \in P$	Proposed Optimisation Problem (Taxi-sharing) Chapter 6 / Page 155

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
$\sum_{(k,l)\in P} X_{ik_1l_1k_2\,l_2p} \le \dot{f}_i \;\forall i \in T$ $X_{ik_1l_1k_2\,l_2p} \ge 0 \;\forall i \in T,\; \forall (k,l) \in P$	
$R_{ij}^{t,d} = \frac{C_{(N_t,O_{p1})} + C_{(O_{p1},D_{p1})} + C_{(D_{p1},O_{p2})} + C_{(O_{p2},D_{p2})} - F_{p1,p2,m}^{v}}{TT_{i\,j\,t\,d}^{D}}$	Proposed Rate of Return Function Chapter 6 / Page 158
$P_q(j\|i) = \frac{\exp\left(\beta^E E_j^{t,d} + \beta^t TT_{i\,j\,t\,d}^{D} + \beta^R R_{ij}^{t,d}\right)}{\sum_{n\epsilon H_i\cup\{i\}}\left(\beta^E E_n^{t,d} + \beta^t TT_{i\,n\,t\,d}^{D} + \beta^R R_{in}^{t,d}\right)} \qquad j \in H_i \cup \{i\}$	ATT Revenue Probabilistic Function Chapter 6 / Page 159
$T_s = E(W_s) + \sum_{r\epsilon S} rtt_r H_r(S)$	Proposed Expected Travel Time Chapter 6 / Page 166
$f_{i,j,4,r}^{t,d} = f_{i,j,4}^{t,d} \frac{e^{-\emptyset RTT'^{t,d}_r}}{\sum_{k\in R(i.j)} e^{-\emptyset RTT'^{t,d}_k}}, \qquad \forall r \in R(i.j)$ $RTT'^{t,d}_r = \sum_l \delta_{l,r}^{t,d} LTT'^{t,d}_l \qquad \forall l \in r$	Stochastic Traffic Assignment Chapter 6 / Page 169
$x'^{t,d}_k = \sum_i \sum_j \sum_m \sum_r f_{i,j,m,r}^{t,d} \delta_{k,i.j.m.r}^{t,d}$	Link Flow Conservation Chapter 6 / Page 172
$x_k^{t,d} = x'^{t,d}_k + \hat{x}_k^{t,d}$ $LTT'^{t,d}_k = LTT_k^o \left[1 + 0.15\left(\frac{\hat{x}_k^{t,d}}{C_k}\right)^4\right]$ $LTT_k^{t,d} = LTT_k^o \left[1 + 0.15\left(\frac{x_k^{t,d}}{C_k}\right)^4\right]$	Link Travel Time Calculation Chapter 6 / Page 173
$RTT_r^{t,d} = \sum_k \delta_{k,r}^{t,d} LTT_k^{t,d} \qquad \forall k \in r, \forall r \in R(i.j)$	Actual Route Travel Time

Equation/Formulation/Algorithm	Equation Name / Chapter / Page
	Chapter 6 / Page 174
$n_r^{t^*,d} = \left\lceil \frac{RTT_r^{t^*,d}}{t_{Length}} \right\rceil \qquad \forall r \in R(i.j)$	Number of Horizons Chapter 6 / Page 175
$\hat{f}_{i,j,m,r}^{t+1,d} = \begin{cases} f_{i,j,m,r}^{t^*,d} & \text{, if } t+1 \leq (t^* + n_r^{t^*,d} - 1) \\ 0 & \text{, if } t+1 > (t^* + n_r^{t^*,d} - 1) \end{cases}$ $\hat{x}_k^{t+1,d} = \sum_i \sum_j \sum_m \sum_r \hat{f}_{i,j,m,r}^{t+1,d} \delta_{k,i.j.m.r}^{t+1,d}$	Flow Propagation Model Chapter 6 / Page 176
$C_b = f_b \cdot c_b$ $\omega_{i\ j\ t\ d\ b} = \frac{\max\{\ddot{x}_k^{t,d}\}}{C_b}$ $C_{i\ j\ t\ d\ b} = \beta_{PT} exp^{\omega_{i\ j\ t\ d\ b}}$	Crowdedness Factor Chapter 6 / Page 179
$U_{i,j,m}^{t,(d)} = \frac{1}{\sum_{i=1}^{k} \lambda^{i-1}} \left(U_{i,j,m}^{t,d-1} + \lambda U_{i,j,m}^{t,d-2} + \lambda^2 U_{i,j,m}^{t,d-3} \ldots \lambda^{k-1} U_{i,j,m}^{t,d-k} \right)$	Perceived Mode Cost Chapter 6 / Page 180
Objective function: $\max \sum_{i,j} \sum_{k,l} \sum_{p} X_{ijk_1l_1} VKTS_{ijk_1l_1}$ Subject to: $\sum_{(i,j)\in D} X_{ijk_1l_1} \leq F_{(k,l)}\ \forall (k,l)\ \in P$ $\sum_{(k,l)\in P} X_{ijk_1l_1} \leq F_{(i,j)}\ \forall (i,j)\ \in D$ $X_{ijk_1l_1}\ \geq 0\ \forall (i,j) \in D\ ,\ \forall (k,l) \in P$	Optimisation Problem Chapter 8 / Page 232
Objective function: $\max \sum_{i,j} \sum_{k,l} \sum_{p} X_{ijk_1l_1k_2\ l_2p} F_{d,p1,p2}^{d}$ Subject to: $\sum_{(i,j)\in D} X_{ijk_1l_1k_2\ l_2p} \leq F_{(k,l)}\ \forall (k,l)\ \in P$ $\sum_{(k,l)\in P} X_{ijk_1l_1k_2\ l_2p} \leq F_{(i,j)}\ \forall (i,j)\ \in D$ $X_{ijk_1l_1k_2\ l_2p}\ \geq 0\ \forall (i,j) \in D\ ,\ \forall (k,l) \in P$	Profit Maximisation Problem Chapter 8 / Page 236

Equation/Formulation/Algorithm	**Equation Name / Chapter / Page**
$Adjusted\ fare = Original\ fare \times \rho \frac{n(Pax)}{n(Taxi)}$	Surge Pricing Chapter 8 / Page 262

APPENDIX B

Simulation code descriptions

Appendix B1

Input Data

1. Network & demand data

The network data that used in this thesis can be obtained from:

Transportation Networks for Research Core Team. *Transportation Networks for Research*.
https://github.com/bstabler/TransportationNetworks.
Accessed June, 03, 2023.

The network and demand data must have the same formatted as the aforementioned reference. In order to be compatible with the proposed simulation framework.

2. Path data

In this simulation the path data is obtained from Yen's algorithm. MATLAB function "kShortestPath" can be used to generate K shortest path that compatible with the proposed simulation framework. Following table provide a detailed explanation of the Path data format.

Name	Size	Description
DestNode	A x 1 double	Destination node
NumOD	integer	Total OD number
OD_Set	A x 3 double	-the first column is origin node of that OD. -the second column is destination node of that OD -the third column is number of available paths.
ODpath_set	A x 1 cell	path index for that OD
ODpathEnd	A x 1 double	starting index of path
ODpathStart	A x 1 double	ending path ending
OriNode	A x 1 double	Origin node
pathList	B x C double	path list (node order)

Where A = number of available OD, B = number of paths/routes and C= the maximum length of path

3. Bus (or public transport) data

Following table describe formatted of bus (or public transport) data as an input to the simulation framework/

Name	Size	Description
BusDemand	A integer	Bus frequency
BusNode	A x B double	Bus route by Node
BusRoute	A x C double	Bus route by Link

Where A = number of bus lines OD, B = the maximum length of route (by Node) and C= the maximum length of route (by Link).

4. **Taxi data**

Following table describe formatted of taxi data as an input to the simulation framework.

Name	**Size**	**Description**
TaxiDemand	A x T double	Number of vacant taxis on each node

Where A = number of nodes. and T is the total time horizon.

5. **Constants**

Most constant variable are located at **Constants.m**. please refer to data dictionary and source codes for further detail.

Appendix B2

Definition of functions

The entire simulation framework employed throughout this thesis has been developed in the MATLAB programming language. I hereby affirm that all the code contained within the explained functions in this section is original and was exclusively written by the dissertation author.

import_network.m: This is our main simulation framework that manages all input reception. All computation functions exist within this domain. This function controls a simulation loop representing time and the evolution of day-to-day traffic.

GenODArrayVNV.m: This function is used when the possession status of a traveller's vehicle is undetermined. It randomly assigns the proportion of vehicle possession among travellers, thereby separating travellers into two classes - those with vehicle possession (V) and those without (NV).

Constants.m: This function holds all the constant variables required for the simulation framework.

Checkbus.m: This function checks if there are any traveller Origin-Destination (OD) pairs that contain NV travellers and lack access to public transport. If any such pairs exist, they are transferred to V travellers.

Calculate_trip3.m: Utilizing V and NV traveller demand, this function represents our mode choice model, calculating the modal split for each transportation mode (For ridesharing).

Calculate_trip4.m: Utilizing V and NV traveller demand, this function represents our mode choice model, calculating the modal split for each transportation mode (For taxi-sharing).

Matching4C.m: This function contains a matching algorithm based on linear programming optimization. It identifies unmatched passengers and drivers before redirecting them to other available modes of transportation (For ridesharing).

Matching5C.m: This function contains a matching algorithm based on linear programming optimization. It identifies unmatched passengers and drivers before redirecting them to other available modes of transportation (For taxi-sharing).

Routesummary.**m**: This function summarizes the routes taken by shared mobility vehicles (For ridesharing).

Routesummary2.**m**: This function summarizes the routes taken by shared mobility vehicles (For taxi-sharing).

VacantTaxi.m: This function identifies vacant taxi models within the network.

routechoice2.m: After all routes for each mode of transportation are determined, this function calculates the flow numbers for each route. It then loads these demands into the network under existing traffic conditions before calculating actual travel times. It concludes with a within-day traffic model that propagates route flow over time.

day2day.m: This function hosts a day-to-day traffic variation model complemented with a learning operator function.

test_cluster.m: This function calculates the distances between each shared mobility OD based on similarity measures and executes hierarchical clustering based on the maximum number of ODs per cluster.

Prematching.m: Relying on the cluster index generated by test_cluster.m, this function separates OD demands into clusters and prepares all necessary data for the matching algorithm.

Appendix B3

Data dictionary

Following table shows all the parameters used in the simulation framework and with explanation.

Parameter	**Descriptions**
a	for loop parameters
Aeq	Optimisation related parameter
aflow1	temporary flow for solo drivers
aflow2	temporary flow for solo drivers
aflow22	temporary flow for unmatched users and change to solo drivers
aflow3	temporary flow for shared mobility passengers
aflow4	temporary flow for ridesharing drivers
aflowBus	temporary flow for bus passengers
aflowBus2	temporary flow for bus unmatched users and change to bus passengers
alpha	constant variable for d2d model
ans	MATLAB answer for specific task
Beq	Optimisation related parameter
BusAccess	Bus accessibility
BusCapaciy	Bus capacity
BusCost	Cost for using bus for each OD

BusDemand	Bus frequency
BusFare	Bus frae
BusNode	Bus route in node
BusPeak	Maximum flow that OD facing while travelling in the bus
BusRoute	Bus route in link
BusRouteCost	Average bus travel cost by link
BusRouteFlow	Number of bus users by link
BusTT	Average bus travel time by link
BusVehTime	Total bus vehicle time
charge1	ridesharing fare for passenger 1
charge2	ridesharing fare for passenger 2
chargecost	ridesharing fare factor
check1	checking function
check2	checking function
Cluster_ODArrayV	OD in array for may for a specific cluster
ClusterDistances	Similarity measures
ClusterIndex	Cluster index
ClusterNum	Number of clusters
ClusterSize	Size of each cluster
Cnetwork	Network related parameters
Con	Optimisation related parameter
Con1	Optimisation related parameter
Con2	Optimisation related parameter
ConEQ	Optimisation related parameter
ConEQ1	Optimisation related parameter

ConEQ2	Optimisation related parameter
current_iteration	current iteration
d	simulation day d/D
D	total simulation day
damping	damping variable for learning operator
Dataset1	OD demand data in array form
DestNode	Destination Node
Distances	shortest path travel cost
DriverVehTime	Driver in vehicle time
extra	extra cost for ridesharing
extra_dist	extra distance for ridesharing
f	objective function
f2	objective function
f3	objective function
Find_Cluster	parameter used for finding cluster
flag	flag parameter
flow1	flow for solo drivers
flow2	flow for shared mobility passengers
flow22	flow unmatched users and change to solo drivers
flow2_Cluster	flow for passengers and change to solo drivers per cluster
flow3	flow for ridesharing driver
flow3_Cluster	flow for unmatched user and
flow4	flow for solo driver with unmatched travellers
flowBus	flow for buses

flowBus2	flow for unmatched users and change to buses
fuelcost	fuel cost
G	network related data
GLength	network related data
i	origin node
idx	removing parameter
idx2	removing parameter
InVehTime	In vehicle time
j	destination node
lb	lower bound
Lia	optimisation output
Link	Link Data
LinkCost	Link travel cost
LinkCost0	Link trave cost before loaded
LinkFlow	Link flow
linkID	Link ID
LinkLength	Link length
Locb	optimisation output
MatchedRate	matching rate
MatchedRouteCost	Matched demand route cost
MatchedRouteCost1-4	part of matched demand route cost
MatchedRouteFlow	Matched demand
MatcedRouteHorizon	Number of horizons that the demand last
MatchingLoop	Matching loop
MatchVehRoute	Matched vehicle route

MaxClusterMem	Maximum member of each cluster
maxroutelength	Maximum route length
ModeCost	Mode Cost
n	numbering
network	network data
networkArray	network data in Array form
NLength	Link Length
num_ATT	number of attractiveness
num_UnMatched	number of unmatched travellers
NumOD	Number of solo drivers OD
OD	OD matrix
OD_set	OD in form of Array
ODArray	OD in form of Array
ODArrayNV	OD in form of Array for NV travellers
ODArrayNVC	OD in form of Array for NC travellers and each cluster
ODArrayV	OD in form of Array for V travellers
ODArrayVC	OD in form of Array for V travellers and each cluster
ODCostNV	Average OD cost for NV travellers
ODCostV	Average OD cost for V travellers
ODMax	Maximum OD
ODNV	OD matric for NV travellers
ODpath_set	A set of OD routes
ODpathEnd	Start Node
ODpathStart	End Node
ODPax	Passenger OD matric

ODPax2	Unmatched passenger OD matric
ODTaxi	Taxi OD matrices
ODTaxi2	Unmatched Taxi OD matric
OriNode	Origin Nodes
PairArray	Matching possibility table
PairArrayCost	Matching cost table
PairArrayCostTemp	Temporary Matching cost
PairArrayTemp	Temporary Matching Table
PairTable	Matching Table
path1-5	a part of matched route
pathList	mated route
Pax1VehTime	1st passenger in veh time
Pax2VehTIme	2nd passenger in veh time
PCostNV	Perceived cost for NV travellers
PCostV	Perceived cost for V travellers
PCU	PCU unit
prod	optimisation output
ProdLoop	optimisation related parameters
PRouteCOst	Perceived route cost
record	for loop parameter
RouteDemand	Demand for each Route flow
RouteLink	Route travelled link
RouteNode	Route travelled now
row	for loop parameter
rowselected	selecting parameter
run1-5	for loop parameter
SavedRouteCost	Saved route cost

SavedRouteFlow	Saved route flow
Select	selecting parameter
ShareCost	Sharing cost
Sol	optimisation output
sumcost	summation of network cost
t	time horizon
T	total time horizon
t1-t6	parameter for repositioning
TakePaxCost	Cost for picking passenger
Taxi_Cluster	Taxi cluster
TaxiATT	Attractiveness factor for taxis
TaxiDemand	Number of vacant taxis for each node
TaxiMatchedRate	Taxi matching rate
TLength	Length of horizon
total_iterations	total number of iterations
type	matching pattern
UnMatchedCost	Unmatched traveller cost
UnMatchedRouteCost	Unmatched route cost
UnMatchedRouteFlow	Unmatched route flow
UnMatchedRouteHorizon	Duration of unmatched route last in the network
UnmatechedVehTime	Unmatched travellers in vehicle time
vehcost	vehicle cost constant
vot	value of time
VoTDriver	value of time for Driver
VoTPax	value of time for Passengers
VoTTakePax	value of time for taking a passenger

VotWait	value of time for waiting
VTaxi	Vacant Taxi
VTaxiRiute	Vacant Taxi route
VTaxiRouteCost	Vacant Taxi route cost
VTaxiRouteFlow	Vacant Taxi route flow
VTaxiRouteHorizon	Duration of vacant taxi flow last in the network
VTaxiRouteLink	Vacant Taxi travelled links
VTaxiRouteNode	Vacant Taxi travelled nodes
WaitTime	Waiting time
WalkCost	Waiting cost
weight	weight factor

APPENDIX C

Source code: Simulation Framework

(Ridesharing system)

import_network.m

```
tic;

output1=zeros(10,15);



% Clearing everything
clc;
clearvars -except maxcount charge output1;
% @@@@@@@  DAY INPUT @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

%  Day input
D=15;
%  damping variable (for day to day dynamic)
damping=0.3;

% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% @@@@@@@ TIME INPUT @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

%  Input Initial Value
%  Total Time Horizon
T=1;
t=1;
% Length of Time Horizon in a unit of minutes
TLength=60;



% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% Constant value for Stochastic traffic assignment
alpha=0.1;

% Import Network
network=readtable('SiouxFalls_net.txt');
Cnetwork=readtable('SiouxFalls_node.txt');
i=network.InitNode;
j=network.TermNode;
weight=network.FreeFlowTime;
NLength=network.Length;

% Create Graph with Directed Edges (Traveltime = weight)
G=digraph(i,j,weight);
GLength=digraph(i,j,NLength);



% % Plot Network
% x=Cnetwork.X;
% y=Cnetwork.Y;
% p = plot(G,'XData',x,'YData',y,'EdgeLabel',G.Edges.Weight);
```

```
% Creat Adjacency matrix (Total cost of shortest route between node
pairs)
% Distances = FreeFlowTravelTime   ; LinkLength = Actual distance
Distances=zeros(height(G.Nodes),height(G.Nodes),T);
LinkLength=zeros(height(GLength.Nodes),height(GLength.Nodes));


% Only for the first horizon, other horizon will perform at
% clculate_trips programme
for run1=1:height(G.Nodes)
    for run2=1:height(G.Nodes)
        [a,Distances(run1,run2,t)]=shortestpath(G,run1,run2);
        [a1,LinkLength(run1,run2)]=shortestpath(GLength,run1,run2);

    end
end


% Load all avaliable path (obtain by Frank wolve algorithm)
load("pathData.mat");

% Load public transport data
load("BusData2.mat");
BusCapacity=30;
WalkCost=3;
BusFare=5;
BusRouteCost=zeros(size(BusRoute));
BusRouteFlow=zeros(size(BusRoute));
Delta=3;

% Load Taxi Ridesharing data
% load("TaxiData2.mat");



% Creat tables of route proporties for using later
SavedRouteCost=zeros(size(pathList,1),T+1,D+1);
SavedRouteflow=SavedRouteCost;
% Creat a table of link propoties for using later
Link=table2array(network(:,2:11));
LinkFlow=zeros(length(Link),T+1);
LinkCost=zeros(length(Link),T+1);



% Not sure if needs to remove
% flow1=ones(height(G.Nodes),height(G.Nodes),T);
% flow2=ones(height(G.Nodes),height(G.Nodes),T);
% flow3=ones(height(G.Nodes),height(G.Nodes),T);

ShareCost=0.5*Distances(:,:,1);

%  Create a table of report (Cost on each mode)
```

```
ModeCost=zeros(5,T);



% Initial value of Link cost and link flow where ther first horizon is
zero
% flow and free flow speed at all link
LinkCost(:,1)=Link(:,5);
LinkFlow(:,1)=0;


% Load constants value
Constants;
OD=zeros(height(G.Nodes),height(G.Nodes),T);
ODNV=OD;
ODArray=[OriNode DestNode];

% load('ODData30.mat');
%
GenODArrayVNV;

% PathSelection;

% Check if OD can acesss bus
CheckBus;


% Generate flow specified by a number
for row=1:size(ODArrayV,1)
    for a=1:T
%       here is where you pick flow on each OD
%         ODArray(row,3)=randi([0 10])*100;

%         Constant value of flow per hour
%           ODArray(row,3)=6000;
%          change it into total number of vehicle per horizon, this later
%          will be converted back into flow when calculateing the route
%          cost

% OD and ODNV in matrix form

      OD(ODArrayV(row,1),ODArrayV(row,2),a)=ODArrayV(row,3)*TLength/60;

ODNV(ODArrayNV(row,1),ODArrayNV(row,2),a)=ODArrayNV(row,3)*TLength/60;


    end

end

% OD=OD;
% ODNV=ODNV;


ODCostV=zeros(size(ODArrayV,1),4,T,D);
```

```
ODCostNV=zeros(size(ODArrayNV,1),2,T,D);
MatchedRate=zeros(size(ODArrayNV,1),7,T,D);
BusCost=zeros(size(ODArrayNV,1),1);
PCostV=ODCostV;
PCostNV=ODCostNV;
PRouteCost=SavedRouteCost;

% % WaitBar
h = waitbar(0, 'Processing...'); % creates waitbar

total_iterations = T * D; % total iterations
current_iteration = 0; % keeps track of current iteration


% % EndWaitBar


for d=1:D
LinkFlow=zeros(length(Link),T+1);
LinkCost=zeros(length(Link),T+1);
LinkCost0=zeros(length(Link),T+1);
flow1=zeros(height(G.Nodes),height(G.Nodes),T);
flow2=zeros(height(G.Nodes),height(G.Nodes),T);
flow3=zeros(height(G.Nodes),height(G.Nodes),T);
flow4=zeros(height(G.Nodes),height(G.Nodes),T);
flow22=zeros(height(G.Nodes),height(G.Nodes),T);
flowBus=zeros(height(G.Nodes),height(G.Nodes),T);
flowBus2=zeros(height(G.Nodes),height(G.Nodes),T);


% Temp
aflow1=zeros(height(G.Nodes),height(G.Nodes),T);
aflow2=zeros(height(G.Nodes),height(G.Nodes),T);
aflow3=zeros(height(G.Nodes),height(G.Nodes),T);
aflow4=zeros(height(G.Nodes),height(G.Nodes),T);
aflow22=zeros(height(G.Nodes),height(G.Nodes),T);
aflowBus=zeros(height(G.Nodes),height(G.Nodes),T);
aflowBus2=zeros(height(G.Nodes),height(G.Nodes),T);


weight=network.FreeFlowTime;
tic;


for t=1:T


% Inner Loop for Waitbar
% ... your computations ...
current_iteration = current_iteration + 1;

% update waitbar
waitbar(current_iteration / total_iterations, h, sprintf('Processing...
(%.2f%%)', (current_iteration / total_iterations) * 100));
% End Inner Loop for Waitbar
```

```
% Initial value of Link cost and link flow where ther first horizon is
zero
    % flow and free flow speed at all link
LinkCost(:,1)=Link(:,5);

LinkFlow(:,1)=0;


Calculate_trips3;

% Clustering
test_cluster;
pre_mataching;

%
%     With Ridesharing system
    Matching4C;


% %     Without Ridesharing system
% PairArray=[];
% flow4(:,:,t)=flow1(:,:,t)+flow2(:,:,t)+flow3(:,:,t);




    if isempty(PairArray)==0
    routesummary;
%     VacantTaxi;
    routechoice;
    size(PairArray)

    continue
    end

    if isempty(PairArray)==1
%     PairArray=zeros(1,16);
    RouteNode=zeros(1,maxroutelength);
    RouteDemand=zeros(1,7);
    RouteLink=zeros(1,maxroutelength);

    routechoice2;



    continue
    end

    h = waitbar(0, 'Processing...'); % creates waitbar

total_iterations = T * D; % total iterations
current_iteration = 0; % keeps track of current iteration



end
```

```
if t==T+1
    clear sum;
%     FlowDist(simulation,4)=sum(sum(LinkCost0));
%     FlowDist(simulation,5)=mean(ModeCost(1,:));
%     FlowDist(simulation,6)=mean(ModeCost(2,:));
%     FlowDist(simulation,7)=mean(ModeCost(3,:));
%     FlowDist(simulation,8)=mean(ModeCost(4,:));
    disp("End of Simulation");



   day2day;

    disp(sum(sum(LinkCost0)));

    output3(d-1)=sum(sum(LinkCost0));



end
toc;
end



toc;
```

GenODArrayVNV.m

```
load('dataset.mat');

ODMax=70;
rowselected=randperm(528,ODMax);

%
ODArrayV=Dataset1(:,1:6);
ODArrayV(:,7)=0;
ODArrayNV=ODArrayV(:,1:3);
ODArrayNV(:,4:5)=0;
```

Constants.m

```
% Input Variables
% ########################################################


% Constants related to travel cost

% Vehicle Fixed cost
VehCost=1;

% Value of Time
VoTDriver=1;
VoTPax=0;
VoTWait=0;
VoTTakePax=0;
UnMatchedCost=0.5;
% Join cost (Fee of Ridesharing program)
JoinCost=0;

% Cost of taking a Pax
TakePaxCost=0;

% ShareUtil(i,j,t)= flow2(i,j,t)/flow3(i,j,t);
ShareUtil=ones(height(G.Nodes),height(G.Nodes),T);
WaitTime=zeros(height(G.Nodes),T);



networkArray=table2array(network(:,2:11));

maxroutelength=100;

vot=(40/60);
% fuelcost=1+(maxcount/10);
fuelcost=1;
vehcost=4;
chargecost=0.2;
% chargecost=1;
```

Checkbus.m

```
BusAccess=zeros(size(ODArrayNV,1),1);



% Check if ORArray have a bus acess or not
for run1=1:size(ODArrayNV,1)
    for run2=1:size(BusNode,1)

        check1=ODArrayNV(run1,1:2);
        check2=BusNode(run2,:);
        [Lia,Locb]=ismember(check1,check2);
```

```
        if Lia(1)~=0 && Lia(2)~=0 && Locb(1)<Locb(2)
           BusAccess(run1,1)=run2;
        end

    end
end

ODArrayNV=horzcat(ODArrayNV,BusAccess);
ODArrayV=horzcat(ODArrayV,BusAccess);

% Adjust value for OD that no bus Access
for run1=1:size(ODArrayNV,1)
        if ODArrayNV(run1,6)==0
            ODArrayNV(run1,4)=100;
            ODArrayNV(run1,5)=0;
            ODArrayV(run1,4)=ODArrayV(run1,4)+(ODArrayV(run1,7)/3);
            ODArrayV(run1,5)=ODArrayV(run1,5)+(ODArrayV(run1,7)/3);
            ODArrayV(run1,6)=ODArrayV(run1,6)+(ODArrayV(run1,7)/3);
            ODArrayV(run1,7)=0;

        end
end
```

test_cluster.m

```
% Max cluster member
MaxClusterMem=20;

ClusterDistances=zeros(size(ODArrayV,1),size(ODArrayV,1));
CDistance=[];

%  Define similarity matrix (Based on O and D shorthest distances)
for run1=1:size(ODArrayV,1)
    for run2=1:size(ODArrayV,1)


%         Saved distances similarity measures

CDistance(1)=Distances(ODArrayV(run1,1),ODArrayV(run2,1))+Distances(ODArr
ayV(run2,1),ODArrayV(run2,2))+Distances(ODArrayV(run2,2),ODArrayV(run1,2)
);

CDistance(2)=Distances(ODArrayV(run2,1),ODArrayV(run1,1))+Distances(ODArr
ayV(run1,1),ODArrayV(run1,2))+Distances(ODArrayV(run1,2),ODArrayV(run2,2)
);

CDistance(3)=Distances(ODArrayV(run1,1),ODArrayV(run2,1))+Distances(ODArr
ayV(run2,1),ODArrayV(run1,2))+Distances(ODArrayV(run1,2),ODArrayV(run2,2)
);

CDistance(4)=Distances(ODArrayV(run2,1),ODArrayV(run1,1))+Distances(ODArr
ayV(run1,1),ODArrayV(run2,2))+Distances(ODArrayV(run2,2),ODArrayV(run1,2)
);
```

```
%         Mor possitive the better

ClusterDistances(run1,run2)=Distances(ODArrayV(run1,1),ODArrayV(run1,2))+
Distances(ODArrayV(run2,1),ODArrayV(run2,2))-min(CDistance);
%
% Change to more positive the worse because this is use for measure
distance

ClusterDistances(run1,run2)=-1*ClusterDistances(run1,run2);
% ClusterDistances(run1,run2)=exp(-0.01.*ClusterDistances(run1,run2));



%          PD distances similarity measures
%
ClusterDistances(run1,run2)=Distances(ODArrayV(run1,1),ODArrayV(run2,1))+
Distances(ODArrayV(run1,2),ODArrayV(run2,2));



    end
end



for run1=1:size(ODArrayV,1)
ClusterIndex=cluster(linkage(ClusterDistances,'average'),'maxclust',run1)
;

ClusterSize=groupcounts(ClusterIndex);


if ClusterSize<MaxClusterMem+1
    break
end

end
```

pre_matching.m

```
Cluster_ODArrayV=[];
flow3_Cluster=[];
flow2_Cluster=[];
```

```
flow3_Cluster=zeros(size(flow3,1),size(flow3,1),size(ClusterSize,1));
flow2_Cluster=flow3_Cluster;

for ClusterNum=1:ClusterSize

Find_Cluster=find(ClusterIndex(:,1)==ClusterNum);

Cluster_ODArrayV=ODArrayV(Find_Cluster,1:2);







    for record=1:size(Cluster_ODArrayV,1)


flow3_Cluster(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2),Clust
erNum)=flow3(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2));

flow2_Cluster(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2),Clust
erNum)=flow2(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2));
    end

ClusterNum;
end
```

Matching4C.m

```
% MAtching algorithm with clustering for RIDESHARING SYSTEM.


ProdLoop=zeros(size(ClusterSize,1),1);
PairArrayTemp=[];
PairArrayCostTemp=[];

for MatchingLoop=1:size(ClusterSize,1)

    ODArrayVC=[];
    ODArrayNVC=[];
    ODArrayVC=ODArrayV(find(ClusterIndex==MatchingLoop),:);
    ODArrayNVC=ODArrayNV(find(ClusterIndex==MatchingLoop),:);


% Define Current OD matrices for both Driver abd Pax
ODDriver=flow3_Cluster(:,:,MatchingLoop);
ODPax=flow2_Cluster(:,:,MatchingLoop);
% ODTaxi=TaxiDemand(:,t);
```

```
% Initial Value
record=0;
PairArray=[];
PairArrayCost=[];
PairTable=[];
Con=[];
Con1=[];
Con2=[];
ConEQ=[];
ConEQ1=[];
ConEQ2=[];
idx=[];
Sol=[];

tic;

PairArray=zeros(1,15);

% Creat Pair Array for 2Pax 1 Driver (Both Paxs may or maynot have the
% same OD) Note that types refer to moving pattern refer to
for run3=1:size(ODArrayVC,1)
    for run2=1:size(ODArrayVC,1)
        for run1=1:size(ODArrayVC,1)
            for type=1:6
            record=record+1;
            PairArray(record,1)=ODArrayVC(run1,1);
            PairArray(record,2)=ODArrayVC(run1,2);
            PairArray(record,3)=ODArrayVC(run2,1);
            PairArray(record,4)=ODArrayVC(run2,2);
            PairArray(record,5)=ODArrayVC(run3,1);
            PairArray(record,6)=ODArrayVC(run3,2);


PairArray(record,7)=Distances(PairArray(record,1),PairArray(record,2),t);

PairArray(record,8)=Distances(PairArray(record,3),PairArray(record,4),t);

PairArray(record,9)=Distances(PairArray(record,5),PairArray(record,6),t);

            if type==1

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,5),t)
+Distances(PairArray(record,5),PairArray(record,6),t)+Distances(PairArray
(record,6),PairArray(record,3),t)+Distances(PairArray(record,3),PairArray
(record,4),t)+Distances(PairArray(record,4),PairArray(record,2),t);
            end
            if type==2

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,4),t)+Distances(PairArray
(record,4),PairArray(record,5),t)+Distances(PairArray(record,5),PairArray
(record,6),t)+Distances(PairArray(record,6),PairArray(record,2),t);
            end
            if type==3

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,5),t)
+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairArray
```

```
(record,3),PairArray(record,4),t)+Distances(PairArray(record,4),PairArray
(record,6),t)+Distances(PairArray(record,6),PairArray(record,2),t);
            end
            if type==4

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairArray
(record,5),PairArray(record,6),t)+Distances(PairArray(record,6),PairArray
(record,4),t)+Distances(PairArray(record,4),PairArray(record,2),t);
            end
            if type==5

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,5),t)
+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairArray
(record,3),PairArray(record,6),t)+Distances(PairArray(record,6),PairArray
(record,4),t)+Distances(PairArray(record,4),PairArray(record,2),t);
            end
            if type==6

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairArray
(record,5),PairArray(record,4),t)+Distances(PairArray(record,4),PairArray
(record,6),t)+Distances(PairArray(record,6),PairArray(record,2),t);
            end



            PairArray(record,11)=PairArray(record,10)-
PairArray(record,9)- PairArray(record,8)- PairArray(record,7);

PairArray(record,12)=PairArray(record,10)*(PairArray(record,7)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,13)=PairArray(record,10)*(PairArray(record,8)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,14)=PairArray(record,10)*(PairArray(record,9)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));
            PairArray(record,15)=type;
            end
        end
    end
end

% Create Pair Array for 1Pax 1 Driver matching
type=0;
for run2=1:size(ODArrayVC,1)
        for run1=1:size(ODArrayVC,1)
            record=record+1;
            PairArray(record,1)=ODArrayVC(run1,1);
            PairArray(record,2)=ODArrayVC(run1,2);
            PairArray(record,3)=ODArrayVC(run2,1);
            PairArray(record,4)=ODArrayVC(run2,2);
            PairArray(record,5)=0;
            PairArray(record,6)=0;

PairArray(record,7)=Distances(PairArray(record,1),PairArray(record,2),t);

PairArray(record,8)=Distances(PairArray(record,3),PairArray(record,4),t);
```

```
            PairArray(record,9)=0;

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,4),t)+Distances(PairArray
(record,4),PairArray(record,2),t);
            PairArray(record,11)=PairArray(record,10)-
PairArray(record,9)- PairArray(record,8)- PairArray(record,7);

PairArray(record,12)=PairArray(record,10)*(PairArray(record,7)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,13)=PairArray(record,10)*(PairArray(record,8)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,14)=PairArray(record,10)*(PairArray(record,9)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));
            PairArray(record,15)=type;
        end
end



% define cost function for ridesharing

PairArrayCost=PairArray;

for record=1:size(PairArrayCost,1)







%     Calculate ridesharing distance

PairArrayCost(record,7)=LinkLength(PairArray(record,1),PairArray(record,2
));

PairArrayCost(record,8)=LinkLength(PairArray(record,3),PairArray(record,4
));

            if PairArrayCost(record,15)~=0

PairArrayCost(record,9)=LinkLength(PairArray(record,5),PairArray(record,6
));
            end

            if PairArrayCost(record,15)==1

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,5))+LinkLength(PairArrayCost(record,5),PairArrayCost(record,6))+L
inkLength(PairArrayCost(record,6),PairArrayCost(record,3))+LinkLength(Pai
rArrayCost(record,3),PairArrayCost(record,4))+LinkLength(PairArrayCost(re
cord,4),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==2
```

```
PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,4))+L
inkLength(PairArrayCost(record,4),PairArrayCost(record,5))+LinkLength(Pai
rArrayCost(record,5),PairArrayCost(record,6))+LinkLength(PairArrayCost(re
cord,6),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==3

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,5))+LinkLength(PairArrayCost(record,5),PairArrayCost(record,3))+L
inkLength(PairArrayCost(record,3),PairArrayCost(record,4))+LinkLength(Pai
rArrayCost(record,4),PairArrayCost(record,6))+LinkLength(PairArrayCost(re
cord,6),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==4

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,5))+L
inkLength(PairArrayCost(record,5),PairArrayCost(record,6))+LinkLength(Pai
rArrayCost(record,6),PairArrayCost(record,4))+LinkLength(PairArrayCost(re
cord,4),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==5

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,5))+LinkLength(PairArrayCost(record,5),PairArrayCost(record,3))+L
inkLength(PairArrayCost(record,3),PairArrayCost(record,6))+LinkLength(Pai
rArrayCost(record,6),PairArrayCost(record,4))+LinkLength(PairArrayCost(re
cord,4),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==6

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,5))+L
inkLength(PairArrayCost(record,5),PairArrayCost(record,4))+LinkLength(Pai
rArrayCost(record,4),PairArrayCost(record,6))+LinkLength(PairArrayCost(re
cord,6),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==0

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,4))+L
inkLength(PairArrayCost(record,4),PairArrayCost(record,2));
            end
%  Charges to pax based on non-sharing trip
    charge1=chargecost*PairArrayCost(record,8);
    charge2=chargecost*PairArrayCost(record,9);

%         charge1=chargecost;
%     charge2=chargecost;



%   Actual Vehicle distance saved (VKT ot Miles)
    PairArrayCost(record,11)=PairArrayCost(record,10)-
PairArrayCost(record,9)- PairArrayCost(record,8)-
PairArrayCost(record,7);
```

```
%   Cost before Ridesharing

PairArrayCost(record,7)=vot*PairArray(record,7)+fuelcost*PairArrayCost(re
cord,7)+vehcost;

PairArrayCost(record,8)=vot*PairArray(record,8)+fuelcost*PairArrayCost(re
cord,8)+vehcost;

    if PairArrayCost(record,15)~=0

PairArrayCost(record,9)=vot*PairArray(record,9)+fuelcost*PairArrayCost(re
cord,9)+vehcost;
    end


%   Cost after ridesharing

PairArrayCost(record,12)=vot*PairArray(record,10)+fuelcost*PairArrayCost(
record,10)+vehcost-charge1-charge2;

%     Waiting time for all cases
    if PairArray(record,15)==1

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,6),t)+Distances(PairA
rray(record,6),PairArray(record,3),t)+Distances(PairArray(record,3),PairA
rray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,8))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,9))+charge2;
    end
    if PairArray(record,15)==2

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,4),t)+Distances(PairA
rray(record,4),PairArray(record,5),t)+Distances(PairArray(record,5),PairA
rray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,8))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,9))+charge2;
    end
    if PairArray(record,15)==3

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,4),t)+Distances(PairArray(record,4),PairA
rray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,8))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,9))+charge2;
    end
```

```
    if PairArray(record,15)==4

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,6),t)+Distances(PairArray(record,6),PairA
rray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,8))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,9))+charge2;
    end
    if PairArray(record,15)==5

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,6),t)+Distances(PairArray(record,6),PairA
rray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,8))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,9))+charge2;
    end
    if PairArray(record,15)==6

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,4),t)+Distances(PairArray(record,4),PairA
rray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,8))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,9))+charge2;
    end
    if PairArray(record,15)==0

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,4),t);
        PairArrayCost(record,14)=0;
        PairArrayCost(record,8)=vot*(PairArrayCost(record,8))+charge1;
    end

    PairArrayCost(record,17)=charge1;
    PairArrayCost(record,18)=charge2;




end
```

```
% Remove Pair which participants pay more when use ridesharing
idx=find(PairArrayCost(:,12) > PairArrayCost(:,7) | PairArrayCost(:,13) >
PairArrayCost(:,8)| PairArrayCost(:,14) > PairArrayCost(:,9) );
PairArray(idx,:)=[];
PairArrayCost(idx,:)=[];t




% idx=find(PairArray(:,15)~=0);
% PairArray(idx,:)=[];



% % only 1 to 1
%
% idx=find(PairArray(:,15) == 6 | PairArray(:,15) == 5 | PairArray(:,3)
== PairArray(:,5 )) ;
% PairArray(idx,:)=[];
%

%Store VAlue for all clusters

PairArrayTemp=vertcat(PairArrayTemp,PairArray);
PairArrayCostTemp=vertcat(PairArrayCostTemp,PairArrayCost);




end


% Return all Pair Array for all matching cluster
PairArray=[];
PairArrayCost=[];
PairArray=PairArrayTemp;
PairArrayCost=PairArrayCostTemp;
%
PairArray=PairArray(1,:);

% Define Current OD matrices for both Driver abd Pax
ODDriver=sum(flow3_Cluster,3);
ODPax=sum(flow2_Cluster,3);




% Listing remainging and reposition (If some node is not matching)
t1=unique(PairArray(:,1));
```

```
t2=unique(PairArray(:,2));
t3=unique(PairArray(:,3));
t4=unique(PairArray(:,4));
t5=unique(PairArray(:,5));
t6=unique(PairArray(:,6));


%  Remove result from case 0
idx=find(t5(:)==0);
t5(idx)=[];
idx=find(t6(:)==0);
t6(idx)=[];


% Con 1 All zeros (Matching less than Total driver)
Con1=zeros(length(t1)*length(t2),size(PairArray,1));
row=1;
for run1=1:length(t1)
    for run2=1:length(t2)
        for run3=1:size(PairArray,1)

           if PairArray(run3,1)==t1(run1) && PairArray(run3,2)==t2(run2)
               Con1(row,run3)=1;
           end

        end
        row=row+1;
    end

end

% Con 2 All zeros (Mathicng less than total Pax)
Con2=zeros(length(t3)*length(t4),size(PairArray,1));
row=1;

for run1=1:length(t3)
    for run2=1:length(t4)




        row=row+1;
    end

end

row=1;
for run1=1:length(t5)
    for run2=1:length(t6)
        for run3=1:size(PairArray,1)
           if PairArray(run3,5)==t5(run1) && PairArray(run3,6)==t6(run2)
               Con2(row,run3)=Con2(row,run3)+1;

           end
        end
        for run3=1:size(PairArray,1)
```

```
            if PairArray(run3,3)==t3(run1) && PairArray(run3,4)==t4(run2)
                Con2(row,run3)=Con2(row,run3)+1;

            end
        end

        row=row+1;
    end

end

% Creat ConEQ 1=DriverOD 2=PaxOD
i=1;
for run1=1:length(t1)
    for run2=1:length(t2)
        ConEQ1(1,i)=ODDriver(t1(run1),t2(run2));
        i=i+1;
    end
end
i=1;
for run3=1:length(t3)
    for run4=1:length(t4)
        ConEQ2(1,i)=ODPax(t3(run3),t4(run4));
        i=i+1;
    end
end

%  Begin Optimisation proceduce
 Con=[Con1;Con2];
 ConEQ=[ConEQ1 ConEQ2];
 Aeq=[];
 Beq=[];

% Objective Function
% f2=double(PairArray(:,11).');
% f=-1*ones(1,size(PairArray,1));
f2=-1*double((PairArray(:,8)+PairArray(:,9)).');
f=-1*ones(1,size(PairArray,1));


%  Non negativity
lb = zeros(size(f));


toc;
%  Start Optimisation
tic;


[Sol,prod,flag]=linprog(f2,Con,ConEQ,Aeq,Beq,lb);

toc;



% [Sol,prod,flag]=linprog(f2,Con,ConEQ,Aeq,Beq,lb);
```

```
% Merge result into PairArray # matched = PairArray(:,10)
PairArray=[PairArray Sol];




%  Remove Unmatched Pairs, Now PairArray has only matched
idx=find(PairArray(:,16)<=0);
PairArray(idx,:)=[];
PairArrayCost(idx,:)=[];



% Copy ODs 2=Unmatched
ODDriver2=ODDriver;
ODPax2=ODPax;

% Calculate Unmatched and change into OD

for run1=1:size(PairArray,1)
    if PairArray(run1,15)~=0

ODDriver2(PairArray(run1,1),PairArray(run1,2))=ODDriver2(PairArray(run1,1
),PairArray(run1,2))-PairArray(run1,16);

ODPax2(PairArray(run1,3),PairArray(run1,4))=ODPax2(PairArray(run1,3),Pair
Array(run1,4))-PairArray(run1,16);

ODPax2(PairArray(run1,5),PairArray(run1,6))=ODPax2(PairArray(run1,5),Pair
Array(run1,6))-PairArray(run1,16);

    end
    if PairArray(run1,15)==0

ODDriver2(PairArray(run1,1),PairArray(run1,2))=ODDriver2(PairArray(run1,1
),PairArray(run1,2))-PairArray(run1,16);

ODPax2(PairArray(run1,3),PairArray(run1,4))=ODPax2(PairArray(run1,3),Pair
Array(run1,4))-PairArray(run1,16);
    end



end




sum(ProdLoop)

% Calculate Matching rate
for run1=1:size(ODArrayV,1)
% Matched rate of that particular OD
```

```
MatchedRate(run1,1,t,d)=1-
abs(ODPax2(ODArrayV(run1,1),ODArrayV(run1,2))/ODPax(ODArrayV(run1,1),ODAr
rayV(run1,2)));
if ODPax(ODArrayV(run1,1),ODArrayV(run1,2))==0
    MatchedRate(run1,1,t,d)=0;
end
MatchedRate(run1,2,t,d)=ODArrayV(run1,3)*ODArrayV(run1,5)./100.*TLength./
60;
MatchedRate(run1,3,t,d)=ODArrayNV(run1,3)*ODArrayNV(run1,4)./100.*TLength
./60;
MatchedRate(run1,4,t,d)=MatchedRate(run1,1,t,d)*MatchedRate(run1,2,t,d);
MatchedRate(run1,5,t,d)=MatchedRate(run1,1,t,d)*MatchedRate(run1,3,t,d);
MatchedRate(run1,6,t,d)=MatchedRate(run1,2,t,d)-MatchedRate(run1,4,t,d);
MatchedRate(run1,7,t,d)=MatchedRate(run1,3,t,d)-MatchedRate(run1,5,t,d);
end



% All Conventional Vehicle



for run1=1:size(ODArrayV,1)

flow4(ODArrayV(run1,1),ODArrayV(run1,2),t)=flow4(ODArrayV(run1,1),ODArray
V(run1,2),t)+MatchedRate(run1,6,t,d);

end
flow4(:,:,t)=flow4(:,:,t)+flow1(:,:,t);

for run1=1:size(ODArrayNV,1)
%
flowBus(ODArrayNV(run1,1),ODArrayV(run1,2),t)=flowBus(ODArrayNV(run1,1),O
DArrayNV(run1,2),t)+MatchedRate(run1,7,t,d);


flowBus(ODArrayNV(run1,1),ODArrayV(run1,2),t)=flowBus(ODArrayNV(run1,1),O
DArrayNV(run1,2),t)+MatchedRate(run1,6,t,d)+MatchedRate(run1,7,t,d);
end
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % %
```

routesummary.m

```
% For ridesharing

Select=[1,2,3,4,5,6,10,15,16];
MatchVehRoute=PairArray(:,Select);

for row=1:size(PairArray,1)
```

```
    RouteDemand(row,1)=PairArray(row,16)*(60/TLength);
    if PairArray(row,15)==1
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,5);
        RouteDemand(row,4)=PairArray(row,6);
        RouteDemand(row,5)=PairArray(row,3);
        RouteDemand(row,6)=PairArray(row,4);
        RouteDemand(row,7)=PairArray(row,2);
    end
    if PairArray(row,15)==2
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,3);
        RouteDemand(row,4)=PairArray(row,4);
        RouteDemand(row,5)=PairArray(row,5);
        RouteDemand(row,6)=PairArray(row,6);
        RouteDemand(row,7)=PairArray(row,2);
    end
    if PairArray(row,15)==3
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,5);
        RouteDemand(row,4)=PairArray(row,3);
        RouteDemand(row,5)=PairArray(row,4);
        RouteDemand(row,6)=PairArray(row,6);
        RouteDemand(row,7)=PairArray(row,2);
    end
    if PairArray(row,15)==4
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,3);
        RouteDemand(row,4)=PairArray(row,5);
        RouteDemand(row,5)=PairArray(row,6);
        RouteDemand(row,6)=PairArray(row,4);
        RouteDemand(row,7)=PairArray(row,2);
    end
    if PairArray(row,15)==5
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,5);
        RouteDemand(row,4)=PairArray(row,3);
        RouteDemand(row,5)=PairArray(row,6);
        RouteDemand(row,6)=PairArray(row,4);
        RouteDemand(row,7)=PairArray(row,2);
    end
    if PairArray(row,15)==6
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,3);
        RouteDemand(row,4)=PairArray(row,5);
        RouteDemand(row,5)=PairArray(row,4);
        RouteDemand(row,6)=PairArray(row,6);
        RouteDemand(row,7)=PairArray(row,2);
    end
    if PairArray(row,15)==0
        RouteDemand(row,2)=PairArray(row,1);
        RouteDemand(row,3)=PairArray(row,3);
        RouteDemand(row,4)=PairArray(row,4);
        RouteDemand(row,5)=PairArray(row,2);
        RouteDemand(row,6)=0;
        RouteDemand(row,7)=0;
    end
```

```
end




% Create RouteNodeFull for matched vehicle
RouteNode=zeros(length(RouteDemand),maxroutelength);
for record=1:size(RouteDemand,1)

        if RouteDemand(record,6)~=0

path1=shortestpath(G,RouteDemand(record,2),RouteDemand(record,3));

path2=shortestpath(G,RouteDemand(record,3),RouteDemand(record,4));

path3=shortestpath(G,RouteDemand(record,4),RouteDemand(record,5));

path4=shortestpath(G,RouteDemand(record,5),RouteDemand(record,6));

path5=shortestpath(G,RouteDemand(record,6),RouteDemand(record,7));
        c=horzcat(path1,path2,path3,path4,path5);
        b=zeros(1,maxroutelength-length(c));
        c=horzcat(c,b);
        end

        if RouteDemand(record,6)==0

path1=shortestpath(G,RouteDemand(record,2),RouteDemand(record,3));

path2=shortestpath(G,RouteDemand(record,3),RouteDemand(record,4));

path3=shortestpath(G,RouteDemand(record,4),RouteDemand(record,5));
        c=horzcat(path1,path2,path3);
        b=zeros(1,maxroutelength-length(c));
        c=horzcat(c,b);
        end

    RouteNode(record,:)=c;
end
% Remove Unused Variable
c=[];
b=[];

% Create Route Link full for matched vehicle
RouteLink=zeros(length(RouteDemand),maxroutelength);

for record=1:size(RouteDemand,1)
    for run1=1:maxroutelength-1
       if RouteNode(record,run1+1)==0
       break
       end

       if RouteNode(record,run1)==RouteNode(record,run1+1)
```

```
        continue
        end

        linkID=find(Link(:,1)==RouteNode(record,run1) &
Link(:,2)==RouteNode(record,run1+1));
        c=[c linkID];
    end
    b=zeros(1,maxroutelength-length(c));
    c=horzcat(c,b);
    RouteLink(record,:)=c;
    c=[];


end
```

routechoice.m

```
UnMatchedRouteCost0=zeros(size(pathList,1),1);
UnMatchedRouteCost=zeros(size(pathList,1),1);
UnMatchedRouteFlow=zeros(size(pathList,1),1);
MatchedRouteCost=zeros(size(PairArray,1),1);
MatchedRouteFlow=zeros(size(PairArray,1),1);




% Start Stochastic Trafic assignment with nested logit

% Calculate route travel time for unmatched trip.

for record=1:size(pathList,1)
    for run1=1:size(pathList,2)
        if pathList(record,run1+1)==0
        break
        end

        extra=LinkCost(pathList(record,run1),t);
        UnMatchedRouteCost0(record)=UnMatchedRouteCost0(record)+extra;
    end
end

% Calculate route flow for unmatched trip

sumcost=zeros(size(ODArray,1),1);
for record=1:size(ODArray,1)
```

```
record0=intersect(find(OriNode(:)==ODArray(record,1)),find(DestNode(:)==O
DArray(record,2)));

    for run1=ODpathStart(record0):ODpathEnd(record0)
        sumcost(record)=sumcost(record)+exp(-
alpha*UnMatchedRouteCost0(run1));
    end

    for run1=ODpathStart(record0):ODpathEnd(record0)

UnMatchedRouteFlow(run1)=flow4(ODArray(record,1),ODArray(record,2),t)*(ex
p(-alpha*UnMatchedRouteCost0(run1))/sumcost(record));
%         Change veh unit per hirzon into flow

        UnMatchedRouteFlow(run1)=(UnMatchedRouteFlow(run1)/TLength)*60;
    end
end




% Calculate link flow (at the same horizon) (unmatched+matched trip) +
add
% flow to all horizon from current horizon
run2=t;
for record=1:size(LinkFlow,1)
%      Matched trip
    for run1=1:size(RouteLink,1)
      if ismember(record,RouteLink(run1,:))==1
      LinkFlow(record,run2)=LinkFlow(record,run2)+RouteDemand(run1,1);
      end
          end
%  Unmatched trip
    for run1=1:size(pathList,1)
      if ismember(record,pathList(run1,:))==1

LinkFlow(record,run2)=LinkFlow(record,run2)+UnMatchedRouteFlow(run1);
      end
    end


end




% Add bus traffic
    for run1=1:size(BusRoute,1)
        for run3=1:size(BusRoute,2)

%             Only link not ~=0
              if BusRoute(run1,run3)~=0

LinkFlow(BusRoute(run1,run3),run2)=LinkFlow(BusRoute(run1,run3),run2)+(4.
5*BusDemand(run1));
```

```
            end
        end
    end


PCU(t,d)=(sum(RouteDemand(:,1))+sum(UnMatchedRouteFlow)+4.5*sum(BusDemand
))*(TLength/60);



%  Calculate Actual Link Travel time (LinkCost) based on BRP fomulation


for record=1:size(LinkFlow,1)

LinkCost(record,t)=networkArray(record,5)*(1+(networkArray(record,6)*(Lin
kFlow(record,t)/networkArray(record,3))^networkArray(record,7)));
end



% Calculate Actual Route Travel time that suffer

for record=1:size(MatchedRouteCost,1)
%      matched
    for run1=1:size(LinkFlow,1)
    if ismember(run1,RouteLink(record,:))==1
    MatchedRouteCost(record)=MatchedRouteCost(record)+LinkCost(run1,t);
    end
    end
end

for record=1:size(UnMatchedRouteCost,1)
%      Unmatched
    for run1=1:size(LinkFlow,1)
    if ismember(run1,pathList(record,:))==1

UnMatchedRouteCost(record)=UnMatchedRouteCost(record)+LinkCost(run1,t);
    end
    end
end

bussummary;



MatchedRouteHorizon=zeros(size(MatchedRouteCost,1),2);
UnMatchedRouteHorizon=zeros(size(UnMatchedRouteCost,1),2);


% Calculate how long the flow remain in the network
% Matched
for record=1:size(MatchedRouteCost,1)
    MatchedRouteHorizon(record,1)=t;
```

```
MatchedRouteHorizon(record,2)=t+floor(MatchedRouteCost(record)/TLength)+1
;
    if MatchedRouteHorizon(record,2)>T+1
        MatchedRouteHorizon(record,2)=T+1;
    end
end
% UnMatched
for record=1:size(UnMatchedRouteCost,1)
    UnMatchedRouteHorizon(record,1)=t;

UnMatchedRouteHorizon(record,2)=t+floor(UnMatchedRouteCost(record)/TLengt
h)+1;
    if UnMatchedRouteHorizon(record,2)>T+1
        UnMatchedRouteHorizon(record,2)=T+1;
    end
end



% Calculate link flow (at the same horizon) (unmatched+matched trip) +
add
% flow to all horizon from current horizon

for record=1:size(LinkFlow,1)
%      Matched trip
    for run1=1:size(RouteLink,1)
      if ismember(record,RouteLink(run1,:))==1
      for run2=MatchedRouteHorizon(run1,1)+1:MatchedRouteHorizon(run1,2)
      LinkFlow(record,run2)=LinkFlow(record,run2)+RouteDemand(run1,1);
      end
      end
      end
%  Unmatched trip
    for run1=1:size(pathList,1)
      if ismember(record,pathList(run1,:))==1
      for
run2=UnMatchedRouteHorizon(run1,1)+1:UnMatchedRouteHorizon(run1,2)

LinkFlow(record,run2)=LinkFlow(record,run2)+UnMatchedRouteFlow(run1);
      end
      end
    end


end




% Calculate Cost on Each mode



ModeCost(1,t)=sum((UnMatchedRouteFlow/60*TLength).*UnMatchedRouteCost)/su
m(UnMatchedRouteFlow/60*TLength);
```

```
%  Total of pax1

pax1weight=PairArray(:,3)>0;
pax2weight=PairArray(:,5)>0;
%  Total of pax2

% sum cost 2 pax


ModeCost(2,t)=sum(PairArray(:,13)./(PairArray(:,13)+PairArray(:,14)+PairA
rray(:,15)).*(pax1weight.*PairArray(:,16).*MatchedRouteCost))/sum(pax1wei
ght.*PairArray(:,16));

ModeCost(3,t)=sum(PairArray(:,14)./(PairArray(:,13)+PairArray(:,14)+PairA
rray(:,15)).*(pax2weight.*PairArray(:,16).*MatchedRouteCost))/sum(pax2wei
ght.*PairArray(:,16));

ModeCost(4,t)=sum((PairArray(:,12)./(PairArray(:,13)+PairArray(:,14)+Pai
rArray(:,15)).*(PairArray(:,16).*MatchedRouteCost))/sum(PairArray(:,16));




% OD Cost
sumcost=zeros(size(ODArray,1),1);

for record=1:size(ODArray,1)
    idx2=find(OD_set(:,1)==ODArray(record,1) &
OD_set(:,2)==ODArray(record,2));
%     Need to fix something hear to get n exact cost of conventional
%     vehicle
%     if flow4(ODArray(record,1),ODArray(record,2),t)~=0
%
%
%
%
ODCost(record,1,t,d)=mean(UnMatchedRouteCost(ODpathStart(idx2):ODpathEnd(
idx2)));
%
%
%
%
%     end


  for run1=ODpathStart(idx2):ODpathEnd(idx2)

sumcost(record)=sumcost(record)+(UnMatchedRouteCost(run1)*UnMatchedRouteF
low(run1));
    end


ODCost(record,1,t,d)=sumcost(record)/sum(UnMatchedRouteFlow(ODpathStart(i
dx2):ODpathEnd(idx2)));

end
```

```
% For Driver (Included thoose unmatched drivers)
PairArray1=PairArray(:,1:2);
PairArray1=horzcat(PairArray1,PairArray(:,12).*MatchedRouteCost./PairArra
y(:,10));
PairArray1=horzcat(PairArray1,PairArray(:,16));

for record=1:size(ODArray,1)
    idx2=find(PairArray1(:,1)==ODArray(record,1) &
PairArray1(:,2)==ODArray(record,2));
    num_UnMatched=ODDriver2(ODArray(record,1),ODArray(record,2));

ODCost(record,3,t,d)=(sum(PairArray1(idx2,3).*PairArray1(idx2,4))+(UnMatc
hedCost+ODCost(record,1,t,d))*num_UnMatched)/(sum(PairArray1(idx2,4))+num
_UnMatched);
end

% For Passenger (Included thoose unmatched passengers)
PairArray2=vertcat(PairArray(:,3:4),PairArray(:,5:6));
PairArray2=horzcat(PairArray2,vertcat(PairArray(:,13).*MatchedRouteCost./
PairArray(:,10),PairArray(:,14).*MatchedRouteCost./PairArray(:,10)));
PairArray2=horzcat(PairArray2,vertcat(PairArray(:,16),PairArray(:,16)));

for record=1:size(ODArray,1)
    idx2=find(PairArray2(:,1)==ODArray(record,1) &
PairArray2(:,2)==ODArray(record,2));
    num_UnMatched=ODPax2(ODArray(record,1),ODArray(record,2));

ODCost(record,2,t,d)=(sum(PairArray2(idx2,3).*PairArray2(idx2,4))+(UnMatc
hedCost+ODCost(record,1,t,d))*num_UnMatched)/(sum(PairArray2(idx2,4))+num
_UnMatched);
end

for record=1:size(ODArray,1)
    idx2=find(isnan(ODCost(record,:,t,d)));
    ODCost(record,idx2,t,d)=max(ODCost(record,:,t,d));
end

idx2=[];
PairArray1=[];
PairArray2=[];
sumcost=[];


% Move Time Horizon
t=t+1;
RouteDemand=[];
RouteLink=[];
RouteNode=[];

% Update LinkCost on the new horizon
if t<=T+1
    for record=1:size(LinkFlow,1)

LinkCost(record,t)=networkArray(record,5)*(1+(networkArray(record,6)*(Lin
kFlow(record,t)/networkArray(record,3))^networkArray(record,7)));
    end
```

```
    LinkCost0(:,t)=LinkCost(:,t);

    G.Edges.Weight=LinkCost(:,t);
end


outputLC(:,t,d)=LinkCost0(:,t);
outputLF(:,t,d)=LinkFlow(:,t);
```

day2day.m

```
t=1;
d=d+1;

a4=d-1;
n=0;


if a4>2
PCostV(:,:,:,d)=power(damping,n).*ODCostV(:,:,:,a4)+power(damping,n+1).*O
DCostV(:,:,:,a4-1)+power(damping,n+2).*ODCostV(:,:,:,a4-
2)/(power(damping,n)+power(damping,n+1)+power(damping,n+2));
PCostNV(:,:,:,d)=power(damping,n).*ODCostNV(:,:,:,a4)+power(damping,n+1).
*ODCostNV(:,:,:,a4-1)+power(damping,n+2).*ODCostNV(:,:,:,a4-
2)/(power(damping,n)+power(damping,n+1)+power(damping,n+2));
PRouteCost(:,:,d)=power(damping,n).*SavedRouteCost(:,:,a4)+power(damping,
n+1).*SavedRouteCost(:,:,a4-1)+power(damping,n+2).*SavedRouteCost(:,:,a4-
2)/(power(damping,n)+power(damping,n+1)+power(damping,n+2));
end

if a4==2
PCostV(:,:,:,d)=power(damping,n).*ODCostV(:,:,:,a4)+power(damping,n+1).*O
DCostV(:,:,:,a4-1)/(power(damping,n)+power(damping,n+1));
PCostNV(:,:,:,d)=power(damping,n).*ODCostNV(:,:,:,a4)+power(damping,n+1).
*ODCostNV(:,:,:,a4-1)/(power(damping,n)+power(damping,n+1));
PRouteCost(:,:,d)=power(damping,n).*SavedRouteCost(:,:,a4)+power(damping,
n+1).*SavedRouteCost(:,:,a4-1)/(power(damping,n)+power(damping,n+1));
end


if a4<2
PCostV(:,:,:,d)=ODCostV(:,:,:,a4);
PCostNV(:,:,:,d)=ODCostNV(:,:,:,a4);
PRouteCost(:,:,d)=SavedRouteCost(:,:,a4);

end
```

APPENDIX D

Source code: Simulation Framework

(Taxi-sharing system)

import_network.m

```
tic;


% Clearing everything
clc;
clearvars -except maxcount charge output1;
% @@@@@@@  DAY INPUT @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

%  Day input
D=15;
%  damping variable (for day to day dynamic)
damping=0.3;

% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% @@@@@@@ TIME INPUT @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

%  Input Initial Value
%  Total Time Horizon
T=1;
t=1;
% Length of Time Horizon in a unit of minutes
TLength=60;



% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% Constant value for Stochastic traffic assignment
alpha=0.1;

% Import Network
network=readtable('SiouxFalls_net.txt');
Cnetwork=readtable('SiouxFalls_node.txt');
i=network.InitNode;
j=network.TermNode;
weight=network.FreeFlowTime;
NLength=network.Length;

% Create Graph with Directed Edges (Traveltime = weight)
G=digraph(i,j,weight);
GLength=digraph(i,j,NLength);



% % Plot Network
% x=Cnetwork.X;
% y=Cnetwork.Y;
% p = plot(G,'XData',x,'YData',y,'EdgeLabel',G.Edges.Weight);


% Creat Adjacency matrix (Total cost of shortest route between node
pairs)
```

```
% Distances = FreeFlowTravelTime   ; LinkLength = Actual distance
Distances=zeros(height(G.Nodes),height(G.Nodes),T);
LinkLength=zeros(height(GLength.Nodes),height(GLength.Nodes));


% Only for the first horizon, other horizon will perform at
% clculate_trips programme
for run1=1:height(G.Nodes)
    for run2=1:height(G.Nodes)
        [a,Distances(run1,run2,t)]=shortestpath(G,run1,run2);
        [a1,LinkLength(run1,run2)]=shortestpath(GLength,run1,run2);

    end
end


% Load all avaliable path (obtain by Frank wolve algorithm and Dr.KE HAN
load("pathData.mat");

% Load public transport data
load("BusData.mat");
BusCapacity=30;
WalkCost=3;
BusFare=5;
BusRouteCost=zeros(size(BusRoute));
BusRouteFlow=zeros(size(BusRoute));
Delta=3;

% Load Taxi Ridesharing data
load("TaxiData3.mat");

TaxiDemand=TaxiDemand*0.1;

% Creat tables of route proporties for using later
SavedRouteCost=zeros(size(pathList,1),T+1,D+1);
SavedRouteflow=SavedRouteCost;
% Creat a table of link propoties for using later
Link=table2array(network(:,2:11));
LinkFlow=zeros(length(Link),T+1);
LinkCost=zeros(length(Link),T+1);

ShareCost=0.5*Distances(:,:,1);

%  Create a table of report (Cost on each mode)

ModeCost=zeros(5,T);



% Initial value of Link cost and link flow where ther first horizon is
zero
% flow and free flow speed at all link
LinkCost(:,1)=Link(:,5);
LinkFlow(:,1)=0;
```

```
% Load constants value
Constants;
OD=zeros(height(G.Nodes),height(G.Nodes),T);
ODNV=OD;
ODArray=[OriNode DestNode];

% load('ODData30.mat');
%
GenODArrayVNV;

% PathSelection;

% Check if OD can acesss bus
CheckBus;


% Generate flow specified by a number
for row=1:size(ODArrayV,1)
    for a=1:T

%       here is where you pick flow on each OD
%         ODArray(row,3)=randi([0 10])*100;

%         Constant value of flow per hour
%           ODArray(row,3)=6000;
%          change it into total number of vehicle per horizon, this later
%          will be converted back into flow when calculateing the route
%          cost

% OD and ODNV in matrix form

      OD(ODArrayV(row,1),ODArrayV(row,2),a)=ODArrayV(row,3)*TLength/60;

ODNV(ODArrayNV(row,1),ODArrayNV(row,2),a)=ODArrayNV(row,3)*TLength/60;


    end

end

% OD=OD;
% ODNV=ODNV;


ODCostV=zeros(size(ODArrayV,1),4,T,D);
ODCostNV=zeros(size(ODArrayNV,1),2,T,D);
MatchedRate=zeros(size(ODArrayNV,1),7,T,D);
BusCost=zeros(size(ODArrayNV,1),1);
PCostV=ODCostV;
PCostNV=ODCostNV;
PRouteCost=SavedRouteCost;

% % WaitBar
h = waitbar(0, 'Processing...'); % creates waitbar
```

```
total_iterations = T * D; % total iterations
current_iteration = 0; % keeps track of current iteration
% % EndWaitBar


for d=1:D


LinkFlow=zeros(length(Link),T+1);
LinkCost=zeros(length(Link),T+1);
LinkCost0=zeros(length(Link),T+1);
flow1=zeros(height(G.Nodes),height(G.Nodes),T);
flow2=zeros(height(G.Nodes),height(G.Nodes),T);
flow3=zeros(height(G.Nodes),height(G.Nodes),T);
flow4=zeros(height(G.Nodes),height(G.Nodes),T);
flow22=zeros(height(G.Nodes),height(G.Nodes),T);
flowBus=zeros(height(G.Nodes),height(G.Nodes),T);
flowBus2=zeros(height(G.Nodes),height(G.Nodes),T);


% Temp
aflow1=zeros(height(G.Nodes),height(G.Nodes),T);
aflow2=zeros(height(G.Nodes),height(G.Nodes),T);
aflow3=zeros(height(G.Nodes),height(G.Nodes),T);
aflow4=zeros(height(G.Nodes),height(G.Nodes),T);
aflow22=zeros(height(G.Nodes),height(G.Nodes),T);
aflowBus=zeros(height(G.Nodes),height(G.Nodes),T);
aflowBus2=zeros(height(G.Nodes),height(G.Nodes),T);


weight=network.FreeFlowTime;
tic;


for t=1:T




% Initial value of Link cost and link flow where ther first horizon is
zero
    % flow and free flow speed at all link
LinkCost(:,1)=Link(:,5);

LinkFlow(:,1)=0;


Calculate_trips4;

% Clustering
test_cluster;
pre_mataching;
```

```
%     With Ridesharing system
    Matching5C;


%     Without Ridesharing system
% PairArray=[];
% flow4(:,:,t)=flow1(:,:,t)+flow2(:,:,t)+flow3(:,:,t);




    if isempty(PairArray)==0
    routesummary;
    VacantTaxi;
    routechoice2;
    size(PairArray)

    continue
    end

    if isempty(PairArray)==1
    PairArray=zeros(1,16);
    RouteNode=zeros(1,maxroutelength);
    RouteDemand=zeros(1,7);
    RouteLink=zeros(1,maxroutelength);

    routechoice2;


    continue
    end



end

if t==T+1
    clear sum;
     disp("End of Simulation");
   Untitled555;
    disp(sum(sum(LinkCost0)));

    output3(d-1)=sum(sum(LinkCost0));

end
toc;
end

toc;
```

GenODArrayVNV.m

```
load('dataset.mat');

ODMax=70;
rowselected=randperm(528,ODMax);

%
ODArrayV=Dataset1(:,1:6);
ODArrayV(:,7)=0;
ODArrayNV=ODArrayV(:,1:3);
ODArrayNV(:,4:5)=0;
```

Constants.m

```
% Input Variables
% ########################################################


% Constants related to travel cost

% Vehicle Fixed cost
VehCost=1;

% Value of Time
VoTDriver=1;
VoTPax=0;
VoTWait=0;
VoTTakePax=0;
UnMatchedCost=0.5;
% Join cost (Fee of Ridesharing program)
JoinCost=0;

% Cost of taking a Pax
TakePaxCost=0;

% ShareUtil(i,j,t)= flow2(i,j,t)/flow3(i,j,t);
```

```
ShareUtil=ones(height(G.Nodes),height(G.Nodes),T);
WaitTime=zeros(height(G.Nodes),T);



networkArray=table2array(network(:,2:11));

maxroutelength=100;

vot=(40/60);
% fuelcost=1+(maxcount/10);
fuelcost=1;
vehcost=4;
chargecost=0.2;
% chargecost=1;
```

Checkbus.m

```
BusAccess=zeros(size(ODArrayNV,1),1);



% Check if ORArray have a bus acess or not
for run1=1:size(ODArrayNV,1)
    for run2=1:size(BusNode,1)

        check1=ODArrayNV(run1,1:2);
        check2=BusNode(run2,:);
        [Lia,Locb]=ismember(check1,check2);

        if Lia(1)~=0 && Lia(2)~=0 && Locb(1)<Locb(2)
           BusAccess(run1,1)=run2;
        end

    end
end

ODArrayNV=horzcat(ODArrayNV,BusAccess);
ODArrayV=horzcat(ODArrayV,BusAccess);

% Adjust value for OD that no bus Access
for run1=1:size(ODArrayNV,1)
        if ODArrayNV(run1,6)==0
            ODArrayNV(run1,4)=100;
            ODArrayNV(run1,5)=0;
            ODArrayV(run1,4)=ODArrayV(run1,4)+(ODArrayV(run1,7)/3);
            ODArrayV(run1,5)=ODArrayV(run1,5)+(ODArrayV(run1,7)/3);
            ODArrayV(run1,6)=ODArrayV(run1,6)+(ODArrayV(run1,7)/3);
            ODArrayV(run1,7)=0;

        end
end
```

test_cluster.m

```
% Max cluster member
MaxClusterMem=20;

ClusterDistances=zeros(size(ODArrayV,1),size(TaxiDemand,1));
CDistance=[];

%  Define similarity matrix (Based on O and D shorthest distances)
for run1=1:size(ODArrayV,1)
    for run2=1:size(TaxiDemand,1)


%  Saved distances similarity measures

CDistance(1)=Distances(run2,ODArrayV(run2,1))+Distances(ODArrayV(run2,1),
ODArrayV(run2,2))+Distances(ODArrayV(run2,2),ODArrayV(run1,2));

CDistance(2)=Distances(run2,ODArrayV(run1,1))+Distances(ODArrayV(run1,1),
ODArrayV(run1,2))+Distances(ODArrayV(run1,2),ODArrayV(run2,2));

CDistance(3)=Distances(run2,ODArrayV(run2,1))+Distances(ODArrayV(run2,1),
ODArrayV(run1,2))+Distances(ODArrayV(run1,2),ODArrayV(run2,2));

CDistance(4)=Distances(run2,ODArrayV(run1,1))+Distances(ODArrayV(run1,1),
ODArrayV(run2,2))+Distances(ODArrayV(run2,2),ODArrayV(run1,2));

%         Mor positive the better

ClusterDistances(run1,run2)=Distances(ODArrayV(run1,1),ODArrayV(run1,2))+
Distances(ODArrayV(run2,1),ODArrayV(run2,2))-min(CDistance);
%
% Change to more positive the worse because this is use for measure
distance

ClusterDistances(run1,run2)=-1*ClusterDistances(run1,run2);
% ClusterDistances(run1,run2)=exp(-0.01.*ClusterDistances(run1,run2));



%   PD distances similarity measures
%
ClusterDistances(run1,run2)=Distances(ODArrayV(run1,1),ODArrayV(run2,1))+
Distances(ODArrayV(run1,2),ODArrayV(run2,2));



    end
end
```

```
for run1=1:size(ODArrayV,1)
ClusterIndex=cluster(linkage(ClusterDistances,'average'),'maxclust',run1)
;

ClusterSize=groupcounts(ClusterIndex);


if ClusterSize<MaxClusterMem+1
    break
end

end
```

Prematching.m

```
Cluster_ODArrayV=[];
flow3_Cluster=[];
flow2_Cluster=[];

flow3_Cluster=zeros(size(flow3,1),size(flow3,1),size(ClusterSize,1));
flow2_Cluster=flow3_Cluster;
Taxi_Cluster=zeros(size(TaxiDemand,1),T,size(ClusterSize,1));

for ClusterNum=1:ClusterSize

Find_Cluster=find(ClusterIndex(:,1)==ClusterNum);

Cluster_ODArrayV=ODArrayV(Find_Cluster,1:2);







    for record=1:size(Cluster_ODArrayV,1)


flow3_Cluster(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2),Clust
erNum)=flow3(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2));

flow2_Cluster(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2),Clust
erNum)=flow2(Cluster_ODArrayV(record,1),Cluster_ODArrayV(record,2));
    end

ClusterNum;
end
```

Calculate_trip4.m

```
% Update new shotest route cost
for run1=1:height(G.Nodes)
    for run2=1:height(G.Nodes)
        [a,Distances(run1,run2,t)]=shortestpath(G,run1,run2);
    end
end

% FIXED MODE SPLIT
% Calculate flow on each mode
% for run1=1:height(G.Nodes)
%     for run2 =1:height(G.Nodes)
%         flow1(run1,run2,t)=OD(run1,run2,t)*FlowDist(simulation,1)/100;
%         flow2(run1,run2,t)=OD(run1,run2,t)*FlowDist(simulation,2)/100;
% %
flow3(run1,run2,t)=OD(run1,run2,t)*exp(cost3(run1,run2,t+1))/(exp(cost1(r
un1,run2,t+1))+exp(cost2(run1,run2,t+1))+exp(cost3(run1,run2,t+1)));
%         flow3(run1,run2,t)=OD(run1,run2,t)-flow1(run1,run2,t)-
flow2(run1,run2,t);
%         flow3(run1,run2,t)=OD(run1,run2,t)*FlowDist(simulation,3)/100;
%      end
% end


% FIXED FLOW


% Calculate flow on each mode

% For traveller who have vehicle
for record=1:size(ODArrayV,1)
flow1(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),ODAr
rayV(record,2),t)*ODArrayV(record,4)/100;
flow2(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),ODAr
rayV(record,2),t)*ODArrayV(record,5)/100;
flow3(ODArrayV(record,1),ODArrayV(record,2),t)=0;
flowBus(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),OD
ArrayV(record,2),t)*ODArrayV(record,7)/100;


% Temp
if ODArrayV(record,8)~=0
aflow1(ODArrayV(record,1),ODArrayV(record,2),t)=flow1(ODArrayV(record,1),
ODArrayV(record,2),t);
aflow2(ODArrayV(record,1),ODArrayV(record,2),t)=flow2(ODArrayV(record,1),
ODArrayV(record,2),t);
aflow3(ODArrayV(record,1),ODArrayV(record,2),t)=0;
aflowBus(ODArrayV(record,1),ODArrayV(record,2),t)=flowBus(ODArrayV(record
,1),ODArrayV(record,2),t);

end

end
```

```
if d>1
for record=1:size(ODArrayV,1)

    if ODArrayV(record,8)==0

flow1(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),ODAr
rayV(record,2),t).*exp(-alpha*PCostV(record,1,t,d))./sum(exp(-
alpha*PCostV(record,1:2,t,d)));

flow2(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),ODAr
rayV(record,2),t).*exp(-alpha*PCostV(record,2,t,d))./sum(exp(-
alpha*PCostV(record,1:2,t,d)));
    flow3(ODArrayV(record,1),ODArrayV(record,2),t)=0;
    flowBus(ODArrayV(record,1),ODArrayV(record,2),t)=0;
    end




    if ODArrayV(record,8)~=0

flow1(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),ODAr
rayV(record,2),t).*exp(-alpha*PCostV(record,1,t,d))./sum(exp(-
alpha*PCostV(record,[1,2,4],t,d)));

flow2(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),ODAr
rayV(record,2),t).*exp(-alpha*PCostV(record,2,t,d))./sum(exp(-
alpha*PCostV(record,[1,2,4],t,d)));
    flow3(ODArrayV(record,1),ODArrayV(record,2),t)=0;

flowBus(ODArrayV(record,1),ODArrayV(record,2),t)=OD(ODArrayV(record,1),OD
ArrayV(record,2),t).*exp(-alpha*PCostV(record,4,t,d))./sum(exp(-
alpha*PCostV(record,[1,2,4],t,d)));

%     Temp

aflow1(ODArrayV(record,1),ODArrayV(record,2),t)=flow1(ODArrayV(record,1),
ODArrayV(record,2),t);

aflow2(ODArrayV(record,1),ODArrayV(record,2),t)=flow2(ODArrayV(record,1),
ODArrayV(record,2),t);

aflow3(ODArrayV(record,1),ODArrayV(record,2),t)=flow3(ODArrayV(record,1),
ODArrayV(record,2),t);

aflowBus(ODArrayV(record,1),ODArrayV(record,2),t)=flowBus(ODArrayV(record
,1),ODArrayV(record,2),t);
    end

end
end
```

```
% For traveller who does not have vehicle
for record=1:size(ODArrayNV,1)
flow22(ODArrayNV(record,1),ODArrayNV(record,2),t)=ODNV(ODArrayNV(record,1
),ODArrayNV(record,2),t)*ODArrayNV(record,4)/100;
flowBus2(ODArrayNV(record,1),ODArrayNV(record,2),t)=ODNV(ODArrayNV(record
,1),ODArrayNV(record,2),t)*ODArrayNV(record,5)/100;


% Temp
if ODArrayV(record,8)~=0

aflowBus2(ODArrayNV(record,1),ODArrayNV(record,2),t)=flow22(ODArrayNV(rec
ord,1),ODArrayNV(record,2),t);

aflowBus2(ODArrayNV(record,1),ODArrayNV(record,2),t)=flowBus2(ODArrayNV(r
ecord,1),ODArrayNV(record,2),t);
end



end

if d>1
for record=1:size(ODArrayNV,1)




    if ODArrayNV(record,6)==0

flow22(ODArrayNV(record,1),ODArrayNV(record,2),t)=ODNV(ODArrayNV(record,1
),ODArrayNV(record,2),t);
    flowBus2(ODArrayNV(record,1),ODArrayNV(record,2),t)=0;
    end

     if ODArrayNV(record,6)~=0

flow22(ODArrayNV(record,1),ODArrayNV(record,2),t)=ODNV(ODArrayNV(record,1
),ODArrayNV(record,2),t).*exp(-alpha*PCostNV(record,1,t,d))./sum(exp(-
alpha*PCostNV(record,:,t,d)));

flowBus2(ODArrayNV(record,1),ODArrayNV(record,2),t)=ODNV(ODArrayNV(record
,1),ODArrayNV(record,2),t).*exp(-alpha*PCostNV(record,2,t,d))./sum(exp(-
alpha*PCostNV(record,:,t,d)));
%     Temp

aflow22(ODArrayNV(record,1),ODArrayNV(record,2),t)=flow22(ODArrayNV(recor
d,1),ODArrayNV(record,2),t);

aflowBus2(ODArrayNV(record,1),ODArrayNV(record,2),t)=flowBus2(ODArrayNV(r
ecord,1),ODArrayNV(record,2),t);
    end




    end
end
```

```
flow1(isnan(flow1)) = 0;
flow2(isnan(flow2)) = 0;
flow3(isnan(flow3)) = 0;
flowBus(isnan(flowBus)) = 0;
flow2(isnan(flow2)) = 0;
flow22(isnan(flow22)) = 0;
flowBus2(isnan(flowBus2)) = 0;



output4(:,:,1,t,d)=flow1(:,:,t);
output4(:,:,2,t,d)=flow2(:,:,t);
output4(:,:,3,t,d)=flow3(:,:,t);
output4(:,:,4,t,d)=flowBus(:,:,t);
output4(:,:,5,t,d)=flow22(:,:,t);
output4(:,:,6,t,d)=flowBus2(:,:,t);

% Temp
output6(:,:,1,t,d)=aflow1(:,:,t);
output6(:,:,2,t,d)=aflow2(:,:,t);
output6(:,:,3,t,d)=aflow3(:,:,t);
output6(:,:,4,t,d)=aflowBus(:,:,t);
output6(:,:,5,t,d)=aflow22(:,:,t);
output6(:,:,6,t,d)=aflowBus2(:,:,t);


flow2(:,:,t)=flow2(:,:,t)+flow22(:,:,t);
flowBus(:,:,t)=flowBus(:,:,t)+flowBus2(:,:,t);

for n=2:D


end
```

Matching5C.m

```
% % create random od matrices
% ODDriver=10*randi(10,[height(G.Nodes),height(G.Nodes)]);
% ODPax=10*randi(10,[height(G.Nodes),height(G.Nodes)]);
% for run1=1:height(G.Nodes)
%     ODDriver(run1,run1)=0;
%     ODPax(run1,run1)=0;
% end

ProdLoop=zeros(size(ClusterSize,1),1);
PairArrayTemp=[];
PairArrayCostTemp=[];
```

```
% % Define Current OD matrices for both Driver abd Pax
% ODDriver=flow3(:,:,t);

% All ridesharing drivers go to passengers

for MatchingLoop=1:size(ClusterSize,1)

disp([num2str(MatchingLoop) ' \ ' num2str(size(ClusterSize, 1))]);


     ODArrayVC=[];
    ODArrayNVC=[];
    ODArrayVC=ODArrayV(find(ClusterIndex==MatchingLoop),:);
    ODArrayNVC=ODArrayNV(find(ClusterIndex==MatchingLoop),:);


ODPax=[];
ODTaxi=[];
ODPax=flow2_Cluster(:,:,MatchingLoop)+flow3_Cluster(:,:,MatchingLoop);
ODTaxi=TaxiDemand(:,t);

% Initial Value
record=0;
PairArray=[];
PairArrayCost=[];
PairTable=[];
Con=[];
Con1=[];
Con2=[];
ConEQ=[];
ConEQ1=[];
ConEQ2=[];
idx=[];
Sol=[];


PairArray=zeros(1,15);


% % % % % % From now is the new matching with taxi

% Creat Pair Array for 2Pax 1 Driver (Both Paxs may or maynot have the
% same OD) Note that types refer to moving pattern refer to
for run3=1:size(ODArrayVC,1)
    for run2=1:size(ODArrayVC,1)
        for run1=1:size(ODTaxi,1)
            for type=1:6
            record=record+1;
            PairArray(record,1)=run1;
            PairArray(record,2)=0;
            PairArray(record,3)=ODArrayVC(run2,1);
            PairArray(record,4)=ODArrayVC(run2,2);
            PairArray(record,5)=ODArrayVC(run3,1);
            PairArray(record,6)=ODArrayVC(run3,2);
```

```
            if type==1
               PairArray(record,2)=PairArray(record,4);
            end
            if type==2
               PairArray(record,2)=PairArray(record,6);
            end
            if type==3
               PairArray(record,2)=PairArray(record,6);
            end
            if type==4
               PairArray(record,2)=PairArray(record,4);
            end
            if type==5
               PairArray(record,2)=PairArray(record,4);
            end
            if type==6
               PairArray(record,2)=PairArray(record,6);
            end



PairArray(record,7)=Distances(PairArray(record,1),PairArray(record,2),t);

PairArray(record,8)=Distances(PairArray(record,3),PairArray(record,4),t);

PairArray(record,9)=Distances(PairArray(record,5),PairArray(record,6),t);

            if type==1

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,5),t)
+Distances(PairArray(record,5),PairArray(record,6),t)+Distances(PairArray
(record,6),PairArray(record,3),t)+Distances(PairArray(record,3),PairArray
(record,4),t)+Distances(PairArray(record,4),PairArray(record,2),t);
            end
            if type==2

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,4),t)+Distances(PairArray
(record,4),PairArray(record,5),t)+Distances(PairArray(record,5),PairArray
(record,6),t)+Distances(PairArray(record,6),PairArray(record,2),t);
            end
            if type==3

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,5),t)
+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairArray
(record,3),PairArray(record,4),t)+Distances(PairArray(record,4),PairArray
(record,6),t)+Distances(PairArray(record,6),PairArray(record,2),t);
            end
            if type==4

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairArray
(record,5),PairArray(record,6),t)+Distances(PairArray(record,6),PairArray
(record,4),t)+Distances(PairArray(record,4),PairArray(record,2),t);
            end
            if type==5

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,5),t)
```

```
+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairArray
(record,3),PairArray(record,6),t)+Distances(PairArray(record,6),PairArray
(record,4),t)+Distances(PairArray(record,4),PairArray(record,2),t);
            end
            if type==6

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairArray
(record,5),PairArray(record,4),t)+Distances(PairArray(record,4),PairArray
(record,6),t)+Distances(PairArray(record,6),PairArray(record,2),t);
            end



            PairArray(record,11)=PairArray(record,10)-
PairArray(record,9)- PairArray(record,8)- PairArray(record,7);

PairArray(record,12)=PairArray(record,10)*(PairArray(record,7)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,13)=PairArray(record,10)*(PairArray(record,8)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,14)=PairArray(record,10)*(PairArray(record,9)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));
            PairArray(record,15)=type;
            end
        end
    end
end

%

% Create Pair Array for 1Pax 1 Driver matching
type=0;
for run2=1:size(ODArrayVC,1)
        for run1=1:size(ODTaxi,1)
            record=record+1;
            PairArray(record,1)=run1;

            PairArray(record,3)=ODArrayVC(run2,1);
            PairArray(record,4)=ODArrayVC(run2,2);
            PairArray(record,5)=0;
            PairArray(record,6)=0;
            PairArray(record,2)=PairArray(record,4);



PairArray(record,7)=Distances(PairArray(record,1),PairArray(record,2),t);

PairArray(record,8)=Distances(PairArray(record,3),PairArray(record,4),t);
            PairArray(record,9)=0;

PairArray(record,10)=Distances(PairArray(record,1),PairArray(record,3),t)
+Distances(PairArray(record,3),PairArray(record,4),t)+Distances(PairArray
(record,4),PairArray(record,2),t);
            PairArray(record,11)=PairArray(record,10)-
PairArray(record,9)- PairArray(record,8)- PairArray(record,7);
```

```
PairArray(record,12)=PairArray(record,10)*(PairArray(record,7)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,13)=PairArray(record,10)*(PairArray(record,8)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));

PairArray(record,14)=PairArray(record,10)*(PairArray(record,9)/(PairArray
(record,9)+ PairArray(record,8)+ PairArray(record,7)));
            PairArray(record,15)=type;
        end
end




% define cost function for ridesharing

PairArrayCost=PairArray;

for record=1:size(PairArrayCost,1)

 %     Calculate ridesharing distance

PairArrayCost(record,7)=LinkLength(PairArray(record,1),PairArray(record,2
));

PairArrayCost(record,8)=LinkLength(PairArray(record,3),PairArray(record,4
));

            if PairArrayCost(record,15)~=0

PairArrayCost(record,9)=LinkLength(PairArray(record,5),PairArray(record,6
));
            end

            if PairArrayCost(record,15)==1

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,5))+LinkLength(PairArrayCost(record,5),PairArrayCost(record,6))+L
inkLength(PairArrayCost(record,6),PairArrayCost(record,3))+LinkLength(Pai
rArrayCost(record,3),PairArrayCost(record,4))+LinkLength(PairArrayCost(re
cord,4),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==2

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,4))+L
inkLength(PairArrayCost(record,4),PairArrayCost(record,5))+LinkLength(Pai
rArrayCost(record,5),PairArrayCost(record,6))+LinkLength(PairArrayCost(re
cord,6),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==3

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,5))+LinkLength(PairArrayCost(record,5),PairArrayCost(record,3))+L
inkLength(PairArrayCost(record,3),PairArrayCost(record,4))+LinkLength(Pai
```

```
rArrayCost(record,4),PairArrayCost(record,6))+LinkLength(PairArrayCost(re
cord,6),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==4

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,5))+L
inkLength(PairArrayCost(record,5),PairArrayCost(record,6))+LinkLength(Pai
rArrayCost(record,6),PairArrayCost(record,4))+LinkLength(PairArrayCost(re
cord,4),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==5

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,5))+LinkLength(PairArrayCost(record,5),PairArrayCost(record,3))+L
inkLength(PairArrayCost(record,3),PairArrayCost(record,6))+LinkLength(Pai
rArrayCost(record,6),PairArrayCost(record,4))+LinkLength(PairArrayCost(re
cord,4),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==6

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,5))+L
inkLength(PairArrayCost(record,5),PairArrayCost(record,4))+LinkLength(Pai
rArrayCost(record,4),PairArrayCost(record,6))+LinkLength(PairArrayCost(re
cord,6),PairArrayCost(record,2));
            end
            if PairArrayCost(record,15)==0

PairArrayCost(record,10)=LinkLength(PairArrayCost(record,1),PairArrayCost
(record,3))+LinkLength(PairArrayCost(record,3),PairArrayCost(record,4))+L
inkLength(PairArrayCost(record,4),PairArrayCost(record,2));
            end

%  Charges to pax based on non-sharing trip
    charge1=chargecost*PairArrayCost(record,8);
    charge2=chargecost*PairArrayCost(record,9);



%     %  Charges to pax based on surgepricing
%
charge1=chargecost*PairArrayCost(record,8)*(ODPax(PairArray(record,3),Pai
rArray(record,4))./200);
%     if PairArrayCost(record,15)>0
%
charge2=chargecost*PairArrayCost(record,9)*(ODPax(PairArray(record,5),Pai
rArray(record,6))./200);
%     end
%     if PairArrayCost(record,15)==0
%     charge2=0;
%     end



%         charge1=chargecost;
%     charge2=chargecost;

% % % % OBJECTIVE FUNCTION
%   Actual Vehicle distance saved (VKT ot Miles)
```

```
    PairArrayCost(record,11)=PairArrayCost(record,10)-
PairArrayCost(record,9)- PairArrayCost(record,8)-
PairArrayCost(record,7);

%   Cost before Ridesharing

PairArrayCost(record,7)=vot*PairArray(record,7)+fuelcost*PairArrayCost(re
cord,7)+vehcost;

PairArrayCost(record,8)=vot*PairArray(record,8)+fuelcost*PairArrayCost(re
cord,8)+vehcost;

    if PairArrayCost(record,15)~=0

PairArrayCost(record,9)=vot*PairArray(record,9)+fuelcost*PairArrayCost(re
cord,9)+vehcost;
    end



%   Cost after ridesharing

PairArrayCost(record,12)=vot*PairArray(record,10)+fuelcost*PairArrayCost(
record,10)+vehcost-charge1-charge2;

%     Waiting time for all cases
%  fixed 17_NOV_19
    if PairArray(record,15)==1

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,6),t)+Distances(PairA
rray(record,6),PairArray(record,3),t)+Distances(PairArray(record,3),PairA
rray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,14))+charge2;
    end
    if PairArray(record,15)==2

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,4),t)+Distances(PairA
rray(record,4),PairArray(record,5),t)+Distances(PairArray(record,5),PairA
rray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,14))+charge2;
    end
    if PairArray(record,15)==3

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
```

```
rray(record,3),PairArray(record,4),t)+Distances(PairArray(record,4),PairA
rray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,14))+charge2;
    end
    if PairArray(record,15)==4

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,6),t)+Distances(PairArray(record,6),PairA
rray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,14))+charge2;
    end
    if PairArray(record,15)==5

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,6),t)+Distances(PairArray(record,6),PairA
rray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,5
),t)+Distances(PairArray(record,5),PairArray(record,3),t)+Distances(PairA
rray(record,3),PairArray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,14))+charge2;
    end
    if PairArray(record,15)==6

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,4),t);

PairArrayCost(record,14)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,5),t)+Distances(PairA
rray(record,5),PairArray(record,4),t)+Distances(PairArray(record,4),PairA
rray(record,6),t);
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
        PairArrayCost(record,14)=vot*(PairArrayCost(record,14))+charge2;
    end
    if PairArray(record,15)==0

PairArrayCost(record,13)=Distances(PairArray(record,1),PairArray(record,3
),t)+Distances(PairArray(record,3),PairArray(record,4),t);
        PairArrayCost(record,14)=0;
        PairArrayCost(record,13)=vot*(PairArrayCost(record,13))+charge1;
    end

    PairArrayCost(record,17)=charge1;
    PairArrayCost(record,18)=charge2;
```

```
end



% Remove Pair which participants pay more when use ridesharing
idx=find(PairArrayCost(:,13) >= PairArrayCost(:,8) | PairArrayCost(:,14)
>= PairArrayCost(:,9));
PairArray(idx,:)=[];
PairArrayCost(idx,:)=[];


% idx=find(PairArray(:,15)~=0);
% PairArray(idx,:)=[];




% % only 1 to 1
%
STARRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
% idx=find(PairArray(:,15) == 6 | PairArray(:,15) == 5 | PairArray(:,3)
== PairArray(:,5 )) ;
% PairArray(idx,:)=[];
%
STARRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR

%Store VAlue for all clusters

PairArrayTemp=vertcat(PairArrayTemp,PairArray);
PairArrayCostTemp=vertcat(PairArrayCostTemp,PairArrayCost);

end



% Return all Pair Array for all matching cluster
PairArray=[];
PairArrayCost=[];
PairArray=PairArrayTemp;
PairArrayCost=PairArrayCostTemp;
%
% PairArray=PairArray(1,:);

% Define Current OD matrices for both Driver abd Pax

ODPax=sum(flow2_Cluster,3)+sum(flow3_Cluster,3);



% Listing remainging and reposition (If some node is not matching)
t1=unique(PairArray(:,1));
t2=unique(PairArray(:,2));
t3=unique(PairArray(:,3));
t4=unique(PairArray(:,4));
```

```
t5=unique(PairArray(:,5));
t6=unique(PairArray(:,6));


%  Remove result from case 0
idx=find(t5(:)==0);
t5(idx)=[];
idx=find(t6(:)==0);
t6(idx)=[];


% Con 1 All zeros (Matching less than Total driver)
Con1=zeros(length(t1),size(PairArray,1));
row=1;
for run1=1:length(t1)

        for run3=1:size(PairArray,1)
%             Let cookie tum nai gunnnn
           if PairArray(run3,1)==run1
               Con1(row,run3)=1;
           end
% TEMP
%            if PairArray(run3,1)==t1(run1) &&
PairArray(run3,2)==t2(run2)
%                Con1(row,run3)=1;
%            end
% TEMP

        end
        row=row+1;



end

% Con 2 All zeros (Mathicng less than total Pax)
Con2=zeros(length(t3)*length(t4),size(PairArray,1));
row=1;

for run1=1:length(t3)
    for run2=1:length(t4)

        row=row+1;
    end
end

row=1;
for run1=1:length(t5)
    for run2=1:length(t6)
        for run3=1:size(PairArray,1)
           if PairArray(run3,5)==t5(run1) && PairArray(run3,6)==t6(run2)
               Con2(row,run3)=Con2(row,run3)+1;

           end
        end
        for run3=1:size(PairArray,1)
           if PairArray(run3,3)==t3(run1) && PairArray(run3,4)==t4(run2)
```

```
                Con2(row,run3)=Con2(row,run3)+1;

            end
        end




        row=row+1;
    end

end

% Creat ConEQ 1=DriverOD 2=PaxOD
i=1;
for run1=1:length(t1)

        ConEQ1(1,i)=TaxiDemand(run1,t);
%         ConEQ1(1,i)=ODDriver(t1(run1),t2(run2));
        i=i+1;

end
i=1;
for run3=1:length(t3)
    for run4=1:length(t4)
        ConEQ2(1,i)=ODPax(t3(run3),t4(run4));
        i=i+1;
    end
end

%  Begin Optimisation proceduce
 Con=[Con1;Con2];
 ConEQ=[ConEQ1 ConEQ2];
 Aeq=[];
 Beq=[];

% Objective Function
f3=-1*double(PairArrayCost(:,17).'+PairArrayCost(:,18).');
f2=double(PairArray(:,11).');
f=-2*ones(1,size(PairArray,1));

%  Non negativity
lb = zeros(size(f));

%  Start Optimisation

[Sol,prod,flag]=linprog(f,Con,ConEQ,Aeq,Beq,lb);

% [Sol,prod,flag]=linprog(f2,Con,ConEQ,Aeq,Beq,lb);

% Merge result into PairArray # matched = PairArray(:,10)
PairArray=[PairArray Sol];

%  Remove Unmatched Pairs, Now PairArray has only matched
idx=find(PairArray(:,16)<=0);
```

```
PairArray(idx,:)=[];
PairArrayCost(idx,:)=[];

output11(t,d)=sum(PairArrayCost(:,17)+PairArrayCost(:,18));
output12(t,d)=2*sum(PairArray(:,16));



% Copy ODs 2=Unmatched


ODTaxi2=zeros(size(TaxiDemand));
ODTaxi2(:,t)=TaxiDemand(:,t);
ODPax2=ODPax;


% Calculate Unmatched and change into OD

for run1=1:size(PairArray,1)
    if PairArray(run1,15)~=0

    ODTaxi2(PairArray(run1,1),t)=ODTaxi2(PairArray(run1,1),t)-
PairArray(run1,16);

ODPax2(PairArray(run1,3),PairArray(run1,4))=ODPax2(PairArray(run1,3),Pair
Array(run1,4))-PairArray(run1,16);

ODPax2(PairArray(run1,5),PairArray(run1,6))=ODPax2(PairArray(run1,5),Pair
Array(run1,6))-PairArray(run1,16);

    end
    if PairArray(run1,15)==0

ODDriver2(PairArray(run1,1),PairArray(run1,2))=ODDriver2(PairArray(run1,1
),PairArray(run1,2))-PairArray(run1,16);
    ODTaxi2(PairArray(run1,1),t)=ODTaxi2(PairArray(run1,1))-
PairArray(run1,16);

ODPax2(PairArray(run1,3),PairArray(run1,4))=ODPax2(PairArray(run1,3),Pair
Array(run1,4))-PairArray(run1,16);
    end



end

% if t<T
% ODTaxi2(:,t)=TaxiDemand(:,t+1);
% end
%



% Calculate Matching rate
for run1=1:size(ODArrayV,1)
```

```
% Matched rate of that particular OD
MatchedRate(run1,1,t,d)=1-
abs(ODPax2(ODArrayV(run1,1),ODArrayV(run1,2))/ODPax(ODArrayV(run1,1),ODAr
rayV(run1,2)));
if ODPax(ODArrayV(run1,1),ODArrayV(run1,2))==0
    MatchedRate(run1,1,t,d)=0;
end
MatchedRate(run1,2,t,d)=ODArrayV(run1,3)*ODArrayV(run1,5)./100.*TLength./
60;
MatchedRate(run1,3,t,d)=ODArrayNV(run1,3)*ODArrayNV(run1,4)./100.*TLength
./60;
MatchedRate(run1,4,t,d)=MatchedRate(run1,1,t,d)*MatchedRate(run1,2,t,d);
MatchedRate(run1,5,t,d)=MatchedRate(run1,1,t,d)*MatchedRate(run1,3,t,d);
MatchedRate(run1,6,t,d)=MatchedRate(run1,2,t,d)-MatchedRate(run1,4,t,d);
MatchedRate(run1,7,t,d)=MatchedRate(run1,3,t,d)-MatchedRate(run1,5,t,d);
end
for run1=1:size(TaxiDemand,1)
TaxiMatchedRate(run1,t,d)=1-abs(ODTaxi2(run1,t)/TaxiDemand(run1,t));
end




% All Conventional Vehicle


for run1=1:size(ODArrayV,1)

flow4(ODArrayV(run1,1),ODArrayV(run1,2),t)=flow4(ODArrayV(run1,1),ODArray
V(run1,2),t)+MatchedRate(run1,6,t,d);

end
flow4(:,:,t)=flow4(:,:,t)+flow1(:,:,t);

for run1=1:size(ODArrayNV,1)
%
flowBus(ODArrayNV(run1,1),ODArrayV(run1,2),t)=flowBus(ODArrayNV(run1,1),O
DArrayNV(run1,2),t)+MatchedRate(run1,7,t,d);


flowBus(ODArrayNV(run1,1),ODArrayV(run1,2),t)=flowBus(ODArrayNV(run1,1),O
DArrayNV(run1,2),t)+MatchedRate(run1,6,t,d)+MatchedRate(run1,7,t,d);
end



if isempty(PairArray)==0
output2(t,d)=prod;
end
t
d
```

Routesummary2.m

```
UnMatchedRouteCost0=zeros(size(pathList,1),1);
UnMatchedRouteCost=zeros(size(pathList,1),1);
```

```
UnMatchedRouteFlow=zeros(size(pathList,1),1);
UnMatchedRouteDistance=zeros(size(pathList,1),1);
MatchedRouteCost=zeros(size(PairArray,1),1);
MatchedRouteFlow=zeros(size(PairArray,1),1);
VTaxiRouteCost=zeros(size(VTaxiRoute,1),1);
VTaxiRouteFlow=zeros(size(VTaxiRoute,1),1);

% Total travel time since waiting till drop off 1=pax1 2=pax2
MatchedRouteCost1=zeros(size(PairArray,1),1);
MatchedRouteCost2=zeros(size(PairArray,1),1);
MatchedRouteCost3=zeros(size(PairArray,1),1);
MatchedRouteCost4=zeros(size(PairArray,1),1);




% Start Stochastic Trafic assignment with nested logit



% Calculate route travel time and distance for unmatched trip.

for record=1:size(pathList,1)
    for run1=1:size(pathList,2)
       if pathList(record,run1)==0
       break
       end
       extra_dist=networkArray(pathList(record,run1),4);
       extra=LinkCost(pathList(record,run1),t);
       UnMatchedRouteCost0(record)=UnMatchedRouteCost0(record)+extra;

UnMatchedRouteDistance(record)=UnMatchedRouteDistance(record)+extra_dist;
    end
end


if d==1
PRouteCost(:,t,d)=UnMatchedRouteCost0;
end

% Calculate route flow for unmatched trip

sumcost=zeros(size(ODArrayV,1),1);
for record=1:size(ODArrayV,1)


record0=intersect(find(OriNode(:)==ODArrayV(record,1)),find(DestNode(:)==
ODArrayV(record,2)));

    for run1=ODpathStart(record0):ODpathEnd(record0)
        sumcost(record)=sumcost(record)+exp(-alpha*PRouteCost(run1,t,d));
    end

    for run1=ODpathStart(record0):ODpathEnd(record0)
```

```
UnMatchedRouteFlow(run1)=flow4(ODArrayV(record,1),ODArrayV(record,2),t)*(
exp(-alpha*PRouteCost(run1,t,d))/sumcost(record));
%         Change veh unit per hirzon into flow

        UnMatchedRouteFlow(run1)=(UnMatchedRouteFlow(run1)/TLength)*60;
    end
end

SavedRouteFlow(:,t,d)=UnMatchedRouteFlow;



% Calculate link flow (at the same horizon) (unmatched+matched trip) +
add
% flow to all horizon from current horizon
run2=t;
for record=1:size(LinkFlow,1)
%      Matched trip
    for run1=1:size(RouteLink,1)
      if ismember(record,RouteLink(run1,:))==1
      LinkFlow(record,run2)=LinkFlow(record,run2)+RouteDemand(run1,1);
      end
    end
%  Unmatched trip
    for run1=1:size(pathList,1)
      if ismember(record,pathList(run1,:))==1

LinkFlow(record,run2)=LinkFlow(record,run2)+UnMatchedRouteFlow(run1);
      end
    end

%  Vacant Taxi
    for run1=1:size(VTaxiRoute,1)
      if ismember(record,VTaxiRouteLink(run1,:))==1
      LinkFlow(record,run2)=LinkFlow(record,run2)+VTaxiRoute(run1,1);
      end
    end




end


% Add bus traffic
    for run1=1:size(BusRoute,1)
        for run3=1:size(BusRoute,2)

%             Only link not ~=0
              if BusRoute(run1,run3)~=0

LinkFlow(BusRoute(run1,run3),run2)=LinkFlow(BusRoute(run1,run3),run2)+(4.
5*BusDemand(run1));
              end
        end
    end
```

```
PCU(t,d)=(sum(RouteDemand(:,1))+sum(UnMatchedRouteFlow)+4.5*sum(BusDemand
))*(TLength/60);




%  Calculate Actual Link Travel time (LinkCost) based on BRP fomulation


for record=1:size(LinkFlow,1)

LinkCost(record,t)=networkArray(record,5)*(1+(networkArray(record,6)*(Lin
kFlow(record,t)/networkArray(record,3))^networkArray(record,7)));
end



% Calculate Actual Route Travel time that suffer

for record=1:size(MatchedRouteCost,1)
%      matched : Taxi
    for run1=1:size(LinkFlow,1)
    if ismember(run1,RouteLink(record,:))==1
    MatchedRouteCost(record)=MatchedRouteCost(record)+LinkCost(run1,t);
    end
    end
end

for record=1:size(UnMatchedRouteCost,1)
%      Unmatched
    for run1=1:size(LinkFlow,1)
    if ismember(run1,pathList(record,:))==1

UnMatchedRouteCost(record)=UnMatchedRouteCost(record)+LinkCost(run1,t);
    end
    end
end

for record=1:size(VTaxiRouteCost,1)
%      Vacant Taxi
    for run1=1:size(LinkFlow,1)
    if ismember(run1,VTaxiRouteLink(record,:))==1
    VTaxiRouteCost(record)=VTaxiRouteCost(record)+LinkCost(run1,t);
    end
    end
end



UnmatchedVehTime=sum(UnMatchedRouteCost.*UnMatchedRouteFlow)/4;

% Calculate total travel time+waitingtime passenger 1 and 2
routesummary1;
```

```
routesummary2;

routesummary3;
routesummary4;

% Calculate essentail bus cost
bussummary;

% Saved unmatched route cost
SavedRouteCost(:,t,d)=UnMatchedRouteCost;




MatchedRouteHorizon=zeros(size(MatchedRouteCost,1),2);
UnMatchedRouteHorizon=zeros(size(UnMatchedRouteCost,1),2);
VTaxiRouteHorizon=zeros(size(VTaxiRouteCost,1),2);

% Calculate how long the flow remain in the network
% Matched
for record=1:size(MatchedRouteCost,1)
    MatchedRouteHorizon(record,1)=t;

MatchedRouteHorizon(record,2)=t+floor(MatchedRouteCost(record)/TLength)+1
;
    if MatchedRouteHorizon(record,2)>T+1
        MatchedRouteHorizon(record,2)=T+1;
    end
end
% UnMatched
for record=1:size(UnMatchedRouteCost,1)
    UnMatchedRouteHorizon(record,1)=t;

UnMatchedRouteHorizon(record,2)=t+floor(UnMatchedRouteCost(record)/TLengt
h)+1;
    if UnMatchedRouteHorizon(record,2)>T+1
        UnMatchedRouteHorizon(record,2)=T+1;
    end
end
% Vacant Taxi
for record=1:size(VTaxiRouteCost,1)
    VTaxiRouteHorizon(record,1)=t;

VTaxiRouteHorizon(record,2)=t+floor(VTaxiRouteCost(record)/TLength)+1;
    if VTaxiRouteHorizon(record,2)>T+1
        VTaxiRouteHorizon(record,2)=T+1;
    end
end




 % Calculate Matched Taxi location
for record=1:size(MatchedRouteCost,1)
    if MatchedRouteHorizon(record,2)<=T
% %   Sent to desttination
```

```
TaxiDemand(PairArray(record,2),MatchedRouteHorizon(record,2))=TaxiDemand(
PairArray(record,2),MatchedRouteHorizon(record,2))+PairArray(record,16);
    end
end
% Calculate Vacant Taxi location

for record=1:size(VTaxiRouteCost,1)
    if VTaxiRouteHorizon(record,2)<=T
% %   Sent to desttination

TaxiDemand(VTaxiRoute(record,3),VTaxiRouteHorizon(record,2))=TaxiDemand(V
TaxiRoute(record,3),VTaxiRouteHorizon(record,2))+VTaxiRoute(record,1);
    end
end



% Calculate link flow (at the same horizon) (unmatched+matched trip) +
add
% flow to all horizon from current horizon

for record=1:size(LinkFlow,1)
%      Matched trip
    for run1=1:size(RouteLink,1)
      if ismember(record,RouteLink(run1,:))==1
      for run2=MatchedRouteHorizon(run1,1)+1:MatchedRouteHorizon(run1,2)
      LinkFlow(record,run2)=LinkFlow(record,run2)+RouteDemand(run1,1);
      end
      end
      end
%  Unmatched trip
    for run1=1:size(pathList,1)
      if ismember(record,pathList(run1,:))==1
      for
run2=UnMatchedRouteHorizon(run1,1)+1:UnMatchedRouteHorizon(run1,2)

LinkFlow(record,run2)=LinkFlow(record,run2)+UnMatchedRouteFlow(run1);
      end
      end
    end

%  Vacant taxi trip

        for run1=1:size(VTaxiRouteLink,1)
      if ismember(record,VTaxiRouteLink(run1,:))==1
      for run2=VTaxiRouteHorizon(run1,1)+1:VTaxiRouteHorizon(run1,2)
      LinkFlow(record,run2)=LinkFlow(record,run2)+VTaxiRoute(run1,1);
      end
      end
    end



end
```

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % %
% OD Cost for those who have VEHICLE
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % %

% OD Cost for SOLO DRIVER
sumcost=zeros(size(ODArrayV,1),1);

for record=1:size(ODArrayV,1)
    idx2=find(OD_set(:,1)==ODArrayV(record,1) &
OD_set(:,2)==ODArrayV(record,2));



  for run1=ODpathStart(idx2):ODpathEnd(idx2)

sumcost(record)=sumcost(record)+((vot*UnMatchedRouteCost(run1))+(fuelcost
*UnMatchedRouteDistance(run1)))*UnMatchedRouteFlow(run1);

    end


ODCostV(record,1,t,d)=vehcost+(sumcost(record)/sum(UnMatchedRouteFlow(ODp
athStart(idx2):ODpathEnd(idx2))));

end



% For BUS
for record=1:size(ODArrayV,1)
    ODCostV(record,4,t,d)=BusCost(record,1);
end

% For Driver (Included thoose unmatched drivers)
PairArray1=PairArray(:,1:2);

PairArray1=horzcat(PairArray1,(fuelcost.*PairArrayCost(:,10))+(vot.*Match
edRouteCost)+vehcost-PairArrayCost(:,17)-PairArrayCost(:,18));
PairArray1=horzcat(PairArray1,PairArray(:,16));

for record=1:size(ODArrayV,1)

% % % % % % % %     Replace it to taxi as it does not has cost
$$$$$$$$$$$$$$$$$


%     idx2=find(PairArray1(:,1)==ODArrayV(record,1) &
PairArray1(:,2)==ODArrayV(record,2));
%     num_UnMatched=ODDriver2(ODArrayV(record,1),ODArrayV(record,2));
%
ODCostV(record,3,t,d)=(sum(PairArray1(idx2,3).*PairArray1(idx2,4))+(UnMat
chedCost+ODCostV(record,1,t,d))*num_UnMatched)/(sum(PairArray1(idx2,4))+n
um_UnMatched);
```

```
% % % % % % %     Replace it to taxi as it does not has cost
    ODCostV(record,3,t,d)=0;
end

% For Passenger (Included thoose unmatched passengers)
PairArray2=vertcat(PairArray(:,3:4),PairArray(:,5:6));
PairArray2=horzcat(PairArray2,vertcat(vot.*(MatchedRouteCost1)+PairArrayC
ost(:,17),vot.*(MatchedRouteCost2)+PairArrayCost(:,18)));
PairArray2=horzcat(PairArray2,vertcat(PairArray(:,16),PairArray(:,16)));

for record=1:size(ODArrayV,1)
    idx2=find(PairArray2(:,1)==ODArrayV(record,1) &
PairArray2(:,2)==ODArrayV(record,2));
%     num_UnMatched=ODPax2(ODArrayV(record,1),ODArrayV(record,2));
    num_UnMatched=MatchedRate(record,6,t,d);
%
ODCostV(record,2,t,d)=(sum(PairArray2(idx2,3).*PairArray2(idx2,4))+(UnMat
chedCost+ODCostV(record,1,t,d))*num_UnMatched)/(sum(PairArray2(idx2,4))+n
um_UnMatched);

ODCostV(record,2,t,d)=(sum(PairArray2(idx2,3).*PairArray2(idx2,4))+(UnMat
chedCost+ODCostV(record,1,t,d))*num_UnMatched)/(sum(PairArray2(idx2,4))+n
um_UnMatched);
end

for record=1:size(ODArrayV,1)
    idx2=find(isnan(ODCostV(record,:,t,d)));
    ODCostV(record,idx2,t,d)=max(ODCostV(record,:,t,d));
end




% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % %
% OD Cost for those who DOES NOT HAVE VEHICLE
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % %



% For Bus

for record=1:size(ODArrayNV,1)
    ODCostNV(record,2,t,d)=BusCost(record,1);
end




% For Unmatched Passenger + Bus

for record=1:size(ODArrayNV,1)
    idx2=find(PairArray2(:,1)==ODArrayNV(record,1) &
PairArray2(:,2)==ODArrayNV(record,2));
%     num_UnMatched=ODPax2(ODArrayV(record,1),ODArrayV(record,2));
```

```
    num_UnMatched=MatchedRate(record,7,t,d);
%
ODCostV(record,2,t,d)=(sum(PairArray2(idx2,3).*PairArray2(idx2,4))+(UnMat
chedCost+ODCostV(record,1,t,d))*num_UnMatched)/(sum(PairArray2(idx2,4))+n
um_UnMatched);

ODCostNV(record,1,t,d)=(sum(PairArray2(idx2,3).*PairArray2(idx2,4))+(UnMa
tchedCost+ODCostNV(record,2,t,d))*num_UnMatched)/(sum(PairArray2(idx2,4))
+num_UnMatched);
end


for record=1:size(ODArrayNV,1)
    idx2=find(isnan(ODCostNV(record,:,t,d)));
    ODCostNV(record,idx2,t,d)=max(ODCostNV(record,:,t,d));
end

DriverVehTime=sum(PairArray(:,16).*MatchedRouteCost);

InVehTime(t,d)=UnmatchedVehTime+Pax1VehTime+Pax2VehTime+DriverVehTime+Bus
VehTime;

outputLC(:,t,d)=LinkCost0(:,t);
outputLF(:,t,d)=LinkFlow(:,t);



idx2=[];
PairArray1=[];
PairArray2=[];
sumcost=[];


% Move Time Horizon
t=t+1;
RouteDemand=[];
RouteDemand1=[];
RouteDemand2=[];
RouteDemand3=[];
RouteDemand4=[];

RouteLink=[];
RouteLink1=[];
RouteLink2=[];
RouteLink3=[];
RouteLink4=[];

RouteNode=[];
RouteNode1=[];
RouteNode2=[];
RouteNode3=[];
RouteNode4=[];

% Update LinkCost on the new horizon
if t<=T+1
    for record=1:size(LinkFlow,1)
```

```
LinkCost(record,t)=networkArray(record,5)*(1+(networkArray(record,6)*(Lin
kFlow(record,t)/networkArray(record,3))^networkArray(record,7)));
    end

    LinkCost0(:,t)=LinkCost(:,t);

    G.Edges.Weight=LinkCost(:,t);
end
```

VacantTaxi.m

```
VTaxi=zeros(size(ODTaxi2,1),3);
VTaxi(:,1)=ODTaxi2(:,t);



num_ATT=3;
for run1=1:size(Distances,1)
TaxiATT(run1,:)=sum(ODPax)./Distances(run1,:,t);

% % % % % %  Taxi travel to the closet 3 path
% % % % % TaxiATT(run1,:)=1./Distances(run1,:,t);

end

% Counter for creating new vacan taxi matrix
run4=1;
record=1;
% Creating taxi route matric based on attractioness nober of maximum =
% num_ATT
for run1=1:size(VTaxi,t)
   if VTaxi(run1)>0
       [run2,run3]=maxk(TaxiATT(1,:),num_ATT);



    for run5=1:num_ATT
% Number of taxi on each route
    VTaxiRoute(record,1)=VTaxi(run1)/num_ATT;

        %     Taxi origin node
    VTaxiRoute(record,2)=run1;
%     Taxi des node 1
    VTaxiRoute(record,3)=run3(run5);



    path1=shortestpath(G,run1,run3(run5));
    b=zeros(1,maxroutelength-length(path1));
    VTaxiRouteNode(record,:)=horzcat(path1,b);



    record=record+1;

    end
```

```
   end


end

% Crate taxi route link
VTaxiRouteLink=zeros(size(VTaxiRouteNode,1),maxroutelength);
c=[];
for record=1:size(VTaxiRouteNode,1)
    for run1=1:maxroutelength-1
       if VTaxiRouteNode(record,run1+1)==0
       break
       end

       if VTaxiRouteNode(record,run1)==VTaxiRouteNode(record,run1+1)
       continue
       end

       linkID=find(Link(:,1)==VTaxiRouteNode(record,run1) &
Link(:,2)==VTaxiRouteNode(record,run1+1));
       c=[c linkID];
    end
    b=zeros(1,maxroutelength-length(c));
    c=horzcat(c,b);
    VTaxiRouteLink(record,:)=c;
    c=[];


end
```

day2day.m

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % %
t=1;
d=d+1;

a4=d-1;
n=0;


if a4>2
```

```
PCostV(:,:,:,d)=power(damping,n).*ODCostV(:,:,:,a4)+power(damping,n+1).*O
DCostV(:,:,:,a4-1)+power(damping,n+2).*ODCostV(:,:,:,a4-
2)/(power(damping,n)+power(damping,n+1)+power(damping,n+2));
PCostNV(:,:,:,d)=power(damping,n).*ODCostNV(:,:,:,a4)+power(damping,n+1).
*ODCostNV(:,:,:,a4-1)+power(damping,n+2).*ODCostNV(:,:,:,a4-
2)/(power(damping,n)+power(damping,n+1)+power(damping,n+2));
PRouteCost(:,:,d)=power(damping,n).*SavedRouteCost(:,:,a4)+power(damping,
n+1).*SavedRouteCost(:,:,a4-1)+power(damping,n+2).*SavedRouteCost(:,:,a4-
2)/(power(damping,n)+power(damping,n+1)+power(damping,n+2));
end

if a4==2
PCostV(:,:,:,d)=power(damping,n).*ODCostV(:,:,:,a4)+power(damping,n+1).*O
DCostV(:,:,:,a4-1)/(power(damping,n)+power(damping,n+1));
PCostNV(:,:,:,d)=power(damping,n).*ODCostNV(:,:,:,a4)+power(damping,n+1).
*ODCostNV(:,:,:,a4-1)/(power(damping,n)+power(damping,n+1));
PRouteCost(:,:,d)=power(damping,n).*SavedRouteCost(:,:,a4)+power(damping,
n+1).*SavedRouteCost(:,:,a4-1)/(power(damping,n)+power(damping,n+1));
end


if a4<2
PCostV(:,:,:,d)=ODCostV(:,:,:,a4);
PCostNV(:,:,:,d)=ODCostNV(:,:,:,a4);
PRouteCost(:,:,d)=SavedRouteCost(:,:,a4);

end
```

APPENDIX E

Detailed Numerical Examples for Private Transport Route Choice Models

Appendix E1

The numerical results of incremental assignment algorithm

Iteration	Algorithm step	Route 1	Route 2	Route 3
1	Update	$t_1 = 7$	$t_2 = 10$	$t_3 = 16$
	Network loading	$F_1 = 40$	$F_2 = 0$	$F_3 = 0$
	Summation	$v_1^1 = 40$	$v_2^1 = 0$	$v_3^1 = 0$
2	Update	$t_1 = 10.32$	$t_2 = 10$	$t_3 = 16$
	Network loading	$F_1 = 0$	$F_2 = 30$	$F_3 = 0$
	Summation	$v_1^2 = 40$	$v_2^2 = 30$	$v_3^2 = 0$
3	Update	$t_1 = 10.32$	$t_2 = 11.5$	$t_3 = 16$
	Incremental loading	$F_1 = 20$	$F_2 = 0$	$F_3 = 0$
	Summation	$v_1^3 = 60$	$v_2^3 = 30$	$v_3^3 = 0$
4	Update	$t_1 = 23.8$	$t_2 = 11.5$	$t_3 = 16$
	Network loading	$F_1 = 0$	$F_2 = 10$	$F_3 = 0$
	Summation	$v_1^3 = 60$	$v_2^3 = 40$	$v_3^3 = 0$

Appendix E2

The numerical results of Method of Successive Averages algorithm

Iteration	$\emptyset_n$	**Algorithm step**	**Route 1**	**Route 2**	**Route 3**
1	1	Update	$t_1 = 7$	$t_2 = 10$	$t_3 = 16$
		Summation	$v_1^1 = 100$	$v_2^1 = 0$	$v_3^1 = 0$
2	1/2	Update	$t_1 = 136.63$	$t_2 = 10$	$t_3 = 16$
		Summation	$v_1^2 = 50$	$v_2^2 = 50$	$v_3^2 = 0$
3	1/3	Update	$t_1 = 15.10$	$t_2 = 21.57$	$t_3 = 16$
		Summation	$v_1^3 = 66.67$	$v_2^3 = 33.33$	$v_3^3 = 0$
4	1/4	Update	$t_1 = 32.50$	$t_2 = 12.28$	$t_3 = 16$
		Summation	$v_1^4 = 50$	$v_2^4 = 50$	$v_3^4 = 0$
5	1/5	Update	$t_1 = 15.10$	$t_2 = 21.57$	$t_3 = 16$
		Summation	$v_1^5 = 60$	$v_2^5 = 40$	$v_3^5 = 0$
6	1/6	Update	$t_1 = 23.8$	$t_2 = 14.74$	$t_3 = 16$
		Summation	$v_1^6 = 50$	$v_2^6 = 50$	$v_3^6 = 0$
7	1/7	Update	$t_1 = 15.10$	$t_2 = 21.57$	$t_3 = 16$
		Summation	$v_1^7 = 57.14$	$v_2^7 = 42.86$	$v_3^7 = 0$
…	…	…	…	…	…
16	1/16	Update	$t_1 = 17.48$	$t_2 = 14.74$	$t_3 = 16.02$
		Summation	$v_1^{16} = 50$	$v_2^{16} = 43.75$	$v_3^{16} = 6.25$

Appendix E3

The numerical results of Frank Wolfe algorithm

Initialize

Let $v_a^0 = 0\ a\forall$, therefore $t_1 = 7$, $t_2 = 10$ and $t_3 = 16$

Perform AON assignment $v_1^1 = 100$, $v_2^1 = 0$ and $v_3^1 = 0$

Updated travel costs: $t_1 = 136.63$, $t_2 = 10$ and $t_3 = 16$

Perform 2nd AON assignment $v_1^2 = 0, v_2^2 = 100$ and $v_3^2 = 0$

1st iteration

$$u_1^2 = 100 + \emptyset_2(0 - 100)$$

$$u_2^2 = 0 + \emptyset_2(100 - 0)$$

$$u_3^2 = 0 + \emptyset_2(0 - 0)$$

$$Min\ Z = \int_0^{u_1^2} 7\left(1 + 0.15\left(\frac{v_1}{30}\right)^4\right) dv_1 + \int_0^{u_2^2} 10\left(1 + 0.15\left(\frac{v_2}{30}\right)^4\right) dv_2 + \int_0^{u_3^2} 16\left(1 + 0.15\left(\frac{v_2}{30}\right)^4\right) dv_3$$

$\frac{dz}{d\emptyset} = \frac{d}{d\emptyset}(2.59 \times 10^{-7}(100 - 100\emptyset_2)^5 + 7(100 - 100\emptyset_2) + 3.70 \times 10^{-7}(100\emptyset_2)^5 + 10(100\emptyset_2) + 5.92 \times 10^{-7}(0)^5 + 16(0) = 0$

then $\emptyset_2 = 0.45823$

$$u_1^2 = 100 + 0.45823(0 - 100) = 54.177$$

$$u_2^2 = 0 + 0.45823(100 - 0) = 45.823$$

$$u_3^2 = 0 + \emptyset_2(0 - 0) = 0$$

then Updated travel costs: $t_1 = 18.16$, $t_2 = 18.16$ and $t_3 = 16$

2nd iteration

Perform AON assignment $v_1^3 = 0, v_2^3 = 0$ and $v_3^3 = 100$

$$u_1^3 = 54.177 + \emptyset_3(0 - 54.177)$$

$$u_2^3 = 45.823 + \emptyset_3(0 - 45.823)$$

$$u_3^3 = 0 + \emptyset_3(100 - 0)$$

then $\emptyset_3 = 0.060245$

$u_1^3 = 50.9131, u_2^3 = 43.0625$ and $u_3^3 = 6.0245$

then Updated travel costs: $t_1 = 15.71$, $t_2 = 16.37$ and $t_3 = 16.00$

3rd iteration

Perform AON assignment $v_1^4 = 100, v_2^4 = 0$ and $v_3^4 = 100$

$$u_1^4 = 50.9131 + \emptyset_4(100 - 50.9131)$$

$$u_2^4 = 43.0624 + \emptyset_4(0 - 43.0624)$$

$$u_3^4 = 6.0245 + \emptyset_4(0 - 6.0245)$$

then $\emptyset_4 = 0.0109948$

$u_1^4 = 51.4528, u_2^4 = 42.5889$ and $u_2^4 = 5.9583$

then Updated travel costs: $t_1 = 18.49$, $t_2 = 18.00$ and $t_3 = 16.00$

4th iteration

Perform AON assignment $v_1^5 = 0, v_2^5 = 0$ and $v_3^5 = 100$

$u_1^5 = 51.4528 + \emptyset_5(0 - 51.4528)$

$u_2^5 = 42.5889 + \emptyset_5(0 - 42.5889)$

$u_3^5 = 5.9583 + \emptyset_5(100 - 6.0245)$

Then $\emptyset_5 = 0.002485$

$u_1^5 = 51.3249, u_2^5 = 42.4831$ and $u_2^5 = 6.1918$

then Updated travel costs: $t_1 = 15.99$, $t_2 = 16.03$ and $t_3 = 16.00$

Appendix E4

The numerical results of the modified Method of Successive Averages algorithm

Iteration	$\emptyset_n$	**Algorithm step**	**Route 1**	**Route 2**	**Route 3**
1	1	Update	$t_1 = 7$	$t_2 = 10$	$t_3 = 16$
		Summation	$v_1^1 = 81.02$	$v_2^1 = 18.08$	$v_3^1 = 0.90$
2	1/2	Update	$t_1 = 379.40$	$t_2 = 11.31$	$t_3 = 16.00$
		Summation	$v_1^2 = 40.51$	$v_2^2 = 54.65$	$v_3^2 = 4.84$
3	1/3	Update	$t_1 = 30.28$	$t_2 = 120.11$	$t_3 = 16.01$
		Summation	$v_1^3 = 27.03$	$v_2^3 = 36.43$	$v_3^3 = 36.53$
4	1/4	Update	$t_1 = 11.62$	$t_2 = 31.75$	$t_3 = 51.19$
		Summation	$v_1^4 = 45.27$	$v_2^4 = 27.32$	$v_3^4 = 27.40$
5	1/5	Update	$t_1 = 43.31$	$t_2 = 16.88$	$t_3 = 27.13$
		Summation	$v_1^5 = 36.22$	$v_2^5 = 41.74$	$v_3^5 = 22.04$
6	1/6	Update	$t_1 = 21.88$	$t_2 = 47.48$	$t_3 = 20.66$
		Summation	$v_1^6 = 36.06$	$v_2^6 = 34.78$	$v_3^6 = 29.15$
7	1/7	Update	$t_1 = 21.62$	$t_2 = 28.08$	$t_3 = 30.26$
		Summation	$v_1^7 = 44.48$	$v_2^7 = 30.35$	$v_3^7 = 25.17$
…	…	…	…	…	…
29	1/32	Update	$t_1 = 26.29$	$t_2 = 26.55$	$t_3 = 26.99$
		Summation	$v_1^{16} = 38.65$	$v_2^{16} = 34.03$	$v_3^{16} = 27.31$

APPENDIX F

Single Driver – Single Passenger Result

(Ridesharing system)

The matching algorithm starting of single driver - single passenger matching scheme.

Driver		1st Passenger		2nd Passenger		NonRidesharing Cost			Ridesharing Cost			Total	Saved	Mathching	#
Origin	Destination	Origin	Destination	Origin	Destination	Driver	1st Pax	2nd Pax	Driver	1st Pax	2nd Pax	Cost	Cost	Pattern	Matched
19	20	19	20	0	0	4	4	0	2	2	0	4	-4	0	**120**
19	16	19	16	0	0	4	4	0	2	2	0	4	-4	0	**130**
19	17	19	17	0	0	2	2	0	1	1	0	2	-2	0	**510**
20	17	20	17	0	0	6	6	0	3	3	0	6	-6	0	**680**
20	18	20	18	0	0	4	4	0	2	2	0	4	-4	0	**96**
17	16	17	16	0	0	2	2	0	1	1	0	2	-2	0	**1400**
20	7	20	7	0	0	6	6	0	3	3	0	6	-6	0	**280**
20	16	20	16	0	0	7	7	0	3.5	3.5	0	7	-7	0	**640**
19	18	19	18	0	0	7	7	0	3.5	3.5	0	7	-7	0	**90**
19	7	19	7	0	0	9	9	0	4.5	4.5	0	9	-9	0	**40**
16	17	16	17	0	0	2	2	0	1	1	0	2	-2	0	**1120**
18	17	18	17	0	0	5	5	0	2.5	2.5	0	5	-5	0	**60**
17	20	17	20	0	0	6	6	0	3	3	0	6	-6	0	**170**
7	19	7	19	0	0	9	9	0	4.5	4.5	0	9	-9	0	**40**
16	18	16	18	0	0	3	3	0	1.5	1.5	0	3	-3	0	**50**
7	18	7	18	0	0	2	2	0	1	1	0	2	-2	0	**20**
18	20	18	20	0	0	4	4	0	2	2	0	4	-4	0	**160**
16	19	16	19	0	0	4	4	0	2	2	0	4	-4	0	**260**
16	20	16	20	0	0	7	7	0	3.5	3.5	0	7	-7	0	**480**
16	7	16	7	0	0	5	5	0	2.5	2.5	0	5	-5	0	**280**
17	19	17	19	0	0	2	2	0	1	1	0	2	-2	0	**510**
18	16	18	16	0	0	3	3	0	1.5	1.5	0	3	-3	0	**350**
17	18	17	18	0	0	5	5	0	2.5	2.5	0	5	-5	0	**300**
18	7	18	7	0	0	2	2	0	1	1	0	2	-2	0	**32**
18	19	18	19	0	0	7	7	0	3.5	3.5	0	7	-7	0	**150**
7	20	7	20	0	0	6	6	0	3	3	0	6	-6	0	**150**
7	16	7	16	0	0	5	5	0	2.5	2.5	0	5	-5	0	**140**
7	17	7	17	0	0	7	7	0	3.5	3.5	0	7	-7	0	**100**
20	8	20	8	0	0	9	9	0	4.5	4.5	0	9	-9	0	**216**
19	6	19	6	0	0	11	11	0	5.5	5.5	0	11	-11	0	**40**
19	2	19	2	0	0	16	16	0	8	8	0	16	-16	0	**60**
20	2	20	2	0	0	16	16	0	8	8	0	16	-16	0	**60**
7	6	7	6	0	0	5	5	0	2.5	2.5	0	5	-5	0	**288**
16	6	16	6	0	0	7	7	0	3.5	3.5	0	7	-7	0	**180**
17	6	17	6	0	0	9	9	0	4.5	4.5	0	9	-9	0	**50**
16	2	16	2	0	0	12	12	0	6	6	0	12	-12	0	**80**
16	8	16	8	0	0	5	5	0	2.5	2.5	0	5	-5	0	**660**
7	8	7	8	0	0	3	3	0	1.5	1.5	0	3	-3	0	**160**
18	6	18	6	0	0	7	7	0	3.5	3.5	0	7	-7	0	**32**
11	15	11	15	0	0	9	9	0	4.5	4.5	0	9	-9	0	**280**
10	23	10	23	0	0	13	13	0	6.5	6.5	0	13	-13	0	**540**
9	23	9	23	0	0	16	16	0	8	8	0	16	-16	0	**200**
9	21	9	21	0	0	14	14	0	7	7	0	14	-14	0	**48**
10	24	10	24	0	0	14	14	0	7	7	0	14	-14	0	**80**
9	22	9	22	0	0	12	12	0	6	6	0	12	-12	0	**56**
9	24	9	24	0	0	17	17	0	8.5	8.5	0	17	-17	0	**100**
15	11	15	11	0	0	9	9	0	4.5	4.5	0	9	-9	0	**980**
21	11	21	11	0	0	13	13	0	6.5	6.5	0	13	-13	0	**80**
22	11	22	11	0	0	12	12	0	6	6	0	12	-12	0	**440**
24	9	24	9	0	0	17	17	0	8.5	8.5	0	17	-17	0	**100**
24	10	24	10	0	0	14	14	0	7	7	0	14	-14	0	**320**
23	9	23	9	0	0	16	16	0	8	8	0	16	-16	0	**150**
22	10	22	10	0	0	9	9	0	4.5	4.5	0	9	-9	0	**1040**
22	9	22	9	0	0	12	12	0	6	6	0	12	-12	0	**210**
16	10	16	10	0	0	4	4	0	2	2	0	4	-4	0	**440**
18	10	18	10	0	0	7	7	0	3.5	3.5	0	7	-7	0	**70**
7	10	7	10	0	0	9	9	0	4.5	4.5	0	9	-9	0	**950**
17	9	17	9	0	0	9	9	0	4.5	4.5	0	9	-9	0	**90**
17	10	17	10	0	0	6	6	0	3	3	0	6	-6	0	**1170**
7	9	7	9	0	0	12	12	0	6	6	0	12	-12	0	**120**
16	9	16	9	0	0	7	7	0	3.5	3.5	0	7	-7	0	**840**
10	17	10	17	0	0	6	6	0	3	3	0	6	-6	0	**1950**
9	16	9	16	0	0	7	7	0	3.5	3.5	0	7	-7	0	**140**
9	17	9	17	0	0	9	9	0	4.5	4.5	0	9	-9	0	**720**
10	18	10	18	0	0	7	7	0	3.5	3.5	0	7	-7	0	**70**
9	18	9	18	0	0	10	10	0	5	5	0	10	-10	0	**40**
6	12	6	12	0	0	14	14	0	7	7	0	14	-14	0	**20**
8	12	8	12	0	0	16	16	0	8	8	0	16	-16	0	**60**
														sum	**21488**