



# Simulation-Based Optimization Frameworks for Dynamic Ride-Sharing

Negin Alisoltani Dehkordi

## ► To cite this version:

Negin Alisoltani Dehkordi. Simulation-Based Optimization Frameworks for Dynamic Ride-Sharing. Modeling and Simulation. Université de Lyon, 2020. English. NNT: . tel-03019291v1

**HAL Id: tel-03019291**

**<https://hal.science/tel-03019291v1>**

Submitted on 23 Nov 2020 (v1), last revised 24 Aug 2021 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre NNT: 2020LYSET012



## THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de  
l'École Nationale des Travaux Publics de l'État

École Doctorale N° 162  
MEGA (Mécanique, Energétique, Génie Civil et Acoustique)

Spécialité / discipline de doctorat : Génie Civil

---

# Méthodes d'optimisation basée sur la simulation pour le covoiturage dynamique

---

Soutenue publiquement le 2 octobre 2020 par :

**Negin ALISOLTANI DEHKORDI**

Devant le jury composé de :

Nikolas GEROLIMINIS	Full Prof. (Ecole Polytechnique Fédérale de Lausanne)	Rapporteur
Jorge A. LAVAL	Full Prof. (Georgia Institute of Technology)	Rapporteur
Salima HASSAS	Prof. HDR (Université Claude Bernard Lyon 1)	Présidente du jury
Amal EL FALLAH SEGHROUCHNI	Full Prof. (Sorbonne Université)	Examinateur
Ludovic LECLERCQ	Prof. HDR (Univ. Gustave Eiffel, ENTPE)	Directeur de thèse
Mahdi ZARGAYOUNA	Prof. HDR (Univ. Gustave Eiffel)	Directeur de thèse

Thèse préparée au  
GRETTIA (Genie des Réseaux de Transport Terrestres et Informatique Avancé), Paris, France  
&  
LICIT (Laboratoire Ingénierie Circulation Transports) , Lyon, France





A THESIS OF THE UNIVERSITY OF LYON

Prepared at  
École Nationale des Travaux Publics de l'État

Doctoral school No. 162  
MEGA (Mechanics, Energy, Civil Engineering and Acoustics)

To obtain the graduation of  
PhD in Civil Engineering

---

# Simulation-Based Optimization Frameworks for Dynamic Ride-Sharing

---

Defense date: October 2, 2020 by:

**Negin ALISOLTANI DEHKORDI**

In front of the following examination committee:

Nikolas GEROLIMINIS	Prof. (Ecole Polytechnique Fédérale de Lausanne)	Reviewer
Jorge A. LAVAL	Prof. (Georgia Institute of Technology)	Reviewer
Salima HASSAS	Full Prof (Université Claude Bernard Lyon1)	Committee chair
Amal EL FALLAH SEGHROUCHNI	Full Prof. (Sorbonne Université)	Examiner
Ludovic LECLERCQ	Research director. HDR (Univ. Gustave Eiffel, ENTPE)	Supervisor
Mahdi ZARGAYOUNA	Researcher. HDR (Univ. Gustave Eiffel)	Supervisor

Thèse préparée au  
GRETIA (Genie des Réseaux de Transport Terrestres et Informatique Avancée), Paris, France  
&  
LICIT (Laboratoire Ingénierie Circulation Transports) , Lyon, France





## Abstract

The main goal of this dissertation is to answer one of the critical questions about dynamic ride-sharing services: Can dynamic ride-sharing reduce congestion?

In this thesis, we propose a simulation-based optimization framework for dynamic ride-sharing. Then using this framework, we assess the dynamic ride-sharing impact on two different network scales to find the answer to this question. When assessing the dynamic ride-sharing problem, two important points should be considered. First, how the ride-sharing system serves the network demand and second, how the ride-sharing system is impacted by the network and in particular by congestion. Then we can assess the impact of such a service on the network. Most of the existing approaches focus on the first point, i.e., designing the demand matching while using basic assumptions for the second point, mainly constant travel times.

The proposed method in this thesis can outperform the existing methods in the literature. The optimization algorithm can provide high-quality solutions in a short time. Our solution approach is designed to be exact for small samples. Then, to be able to handle the large-scale problems, it is extended with several heuristics that keep the general design for the solution method but significantly reduce its computation time. In the simulation component, a "Plant Model" is applied based on the "Trip-based Macroscopic Fundamental Diagram (MFD)" to represent the traffic dynamics reality and a "Prediction Model" is applied based on the mean-speed to be used during the assignment process. We perform an extensive simulation study (based on real-world traffic patterns) to assess the influence of dynamic ride-sharing systems on traffic congestion. In the medium-scale (Lyon 6 + Villeurbanne), the results showed that ride-sharing could not significantly improve the traffic situation. High levels of market-share add additional travel distance and travel time to the trips and lead to more traffic in the network. In large cities, the results are entirely different from those in small and medium-sized cities. In large-scale (Lyon city in France) simulations, the proposed dynamic ride-sharing system can significantly improve traffic conditions, especially during peak hours. Increasing the market-share and the number of sharing can enhance this improvement. Therefore, the proposed dynamic ride-sharing system is a viable option, alleviating stress on existing public transport, to reduce the network traffic in populated and large-scale cities.



## Dedication

I dedicate this thesis to my mother, Nasrin,  
for her endless love, support and sacrifices.



## Acknowledgements

It is my pleasure to acknowledge the roles of several individuals who were instrumental in completing my Ph.D. research.

First of all, I would like to express my deep and sincere gratitude to my supervisors, Prof. Ludovic Leclercq, deputy director of the LICIT lab, and Prof. Mahdi Zargayouna, deputy director of the GRETTIA lab, for their constant support, patience, motivation, and immense knowledge. It was a great privilege and honor to work and study under their guidance. I am grateful for their confidence in my capabilities, which always inspired me to produce quality research.

Also, I would like to thank my committee members, Prof. Nikolas Geroliminis from EPFL, Prof. Jorge Laval from Georgia Institute of Technology, Prof. Salima Hassas from Université Claude Bernard Lyon 1, and Prof. Amal El Fallah from Sorbonne Université for their encouragement and insightful comments.

My special appreciation to all the members of the GRETTIA lab in Paris and the LICIT lab in Lyon, for being part of my research journey and becoming very good friends. Many thanks for having spent great times together. I have special thanks to Cécile Becarie and Jean Krug for their help with the data. I also thank Sonia Cenille (ENTPE), Anne-Christine Demanny (LICIT), Marie-Laure Poirer (GRETTIA), Francesca Labrousse (GRETTIA) and Mustapha Tendjaoui for their respective help in many and diverse administrative tasks.

I wish to express my sincere thanks to my parents Nasrin and Dariush, my sister Negar and my parents-in-law Faranak and Hassan, for their love, caring, encouragement, and support.

Last but not least, I would like to say a heartfelt thanks to my husband Mostafa, for his optimism and love, for listening and discussing ideas about this thesis on many occasions, for his guidance, especially at the beginning of my Ph.D. and for his support when I was writing this dissertation at home during the covid-19 lockdown.

This PhD has been funded by Univ. Gustave Eiffel. This study has also received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No 646592 – MAGnUM project).



# Contents

<b>Abstract</b>	<b>i</b>
<b>Dedication</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>I What is ride-sharing?</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Context . . . . .	3
1.2 Research background and motivation . . . . .	5
1.2.1 The performance of ride-sharing on the network demand . . . . .	6
1.2.2 The impact of the network on the ride-sharing system and vice versa	8
1.3 Research questions . . . . .	9
1.4 Methodology . . . . .	13
1.5 Major contributions . . . . .	14
1.6 Thesis outline . . . . .	15
1.7 Publication list . . . . .	17



<b>2</b>	<b>Literature Review</b>	<b>19</b>
2.1	Ride-sharing definition and related terms . . . . .	19
2.1.1	Dynamic ride-sharing . . . . .	21
2.1.2	Ride-sharing history and advantages . . . . .	23
2.2	Dynamic ride-sharing system settings . . . . .	25
2.3	Ride-sharing algorithms . . . . .	29
2.3.1	Objectives and constraints . . . . .	29
2.3.2	Solving methods . . . . .	33
2.3.3	Clustering methods . . . . .	35
2.4	Dynamic traffic conditions in ride-sharing . . . . .	36
<b>II</b>	<b>How to assess the performance of ride-sharing on the network demand?</b>	<b>39</b>
<b>3</b>	<b>Ride-Sharing Problem Formulation</b>	<b>41</b>
3.1	System components . . . . .	42
3.2	Fleet management component . . . . .	43
3.2.1	Notations and variables . . . . .	44
3.2.2	Problem formulation . . . . .	46
3.2.3	Mathematical model objective function . . . . .	47
3.2.4	Mathematical model constraints . . . . .	48
3.2.5	Model complexity . . . . .	50
<b>4</b>	<b>Solution Method for the Dynamic Ride-Sharing Problem</b>	<b>52</b>
4.1	Branch-and-Cut based algorithm . . . . .	53
4.1.1	Examples of the exact method . . . . .	56
4.2	Test cases . . . . .	57
4.2.1	Lyon 6 + villeurbanne . . . . .	58
4.2.2	Lyon . . . . .	59
4.3	Rolling horizon method and re-scheduling . . . . .	61

4.4	Clustering method . . . . .	64
4.4.1	Shareability function . . . . .	64
4.4.2	Clustering based on the similarity matrix . . . . .	66
4.4.3	Determining the proper clustering method . . . . .	69
4.5	Force the sharing method (FOSH method) . . . . .	71

### **III How to assess the impact of the network on the ride-sharing system and vice versa? 75**

## **5 Simulation Framework 77**

5.1	Dynamic simulation component . . . . .	78
5.1.1	Plant model . . . . .	79
5.1.2	Prediction model . . . . .	81
5.2	Depot management . . . . .	85
5.3	Optimization component performance . . . . .	86
5.3.1	Simulations' configuration . . . . .	86
5.3.2	Exact assignment algorithm over the full-time horizon . . . . .	88
5.3.3	Comparing the exact algorithm with the existing methods . . . . .	90
5.3.4	Heuristic 1: Rolling horizon and re-scheduling performance . . . . .	91
5.3.5	Heuristic 2: Clustering method performance . . . . .	93
5.3.6	Heuristic 3: Force the sharing method (FOSH method) performance . . . . .	94
5.3.7	Comparing the final algorithm with the existing methods . . . . .	97
5.4	Dynamic simulation component performance . . . . .	101

## **6 Transportation Analyses 103**

6.1	Analysis in medium-scale . . . . .	103
6.1.1	Market-share . . . . .	104
6.1.2	Number of sharing . . . . .	108
6.1.3	Time window . . . . .	112
6.1.4	Rolling horizon . . . . .	114

6.1.5	Depot size . . . . .	115
6.1.6	Operational settings . . . . .	116
6.2	Analysis in large-scale . . . . .	118
6.2.1	Simulations' configuration in large-scale . . . . .	118
6.2.2	Optimal size of clusters . . . . .	119
6.2.3	Market-share . . . . .	122
6.2.4	Number of sharing . . . . .	126
6.2.5	Vehicle capacity . . . . .	129
6.2.6	Parking effect . . . . .	131
<b>IV</b>	<b>What are the major outcomes of this thesis?</b>	<b>134</b>
<b>7</b>	<b>Conclusion</b>	<b>136</b>
7.1	Summary and discussions . . . . .	136
7.1.1	Dynamic ride-sharing optimization method . . . . .	137
7.1.2	Dynamic ride-sharing simulation framework . . . . .	139
7.1.3	Impact of dynamic ride-sharing on network congestion . . . . .	139
7.2	Conclusion . . . . .	143
7.3	Future work . . . . .	145
	<b>Bibliography</b>	<b>145</b>
	<b>Bibliography</b>	<b>146</b>
	<b>Appendices</b>	<b>166</b>
<b>A</b>		<b>166</b>

# List of Tables

2.1	Transportation demand and service characteristics . . . . .	28
3.1	Notation . . . . .	45
3.2	Solution methods comparison . . . . .	51
4.1	Example with 4 requests (configuration) . . . . .	56
4.2	Example with 4 requests (solutions) . . . . .	56
4.3	Clustering methods comparison . . . . .	70
5.1	Simulations configuration . . . . .	87
5.2	Simulation results for optimal assignment . . . . .	89
5.3	Solution methods comparison . . . . .	91
5.4	Simulation time for the exact method and heuristic 1 . . . . .	92
5.5	Simulation time for heuristic 1 and heuristic 2 . . . . .	93
5.6	Simulation time for heuristic 2 and heuristic 3 . . . . .	96
5.7	Solution methods comparison (computation time for market-share = 10%) . . . . .	97
5.8	Simulation results for heuristic methods (number of sharing = 1) . . . . .	98
5.9	Simulations results for optimal assignment . . . . .	102
6.1	Simulations results . . . . .	105
6.2	Simulation results for different time window lengths . . . . .	113
6.3	Simulation results for different rolling horizon scenarios . . . . .	114
6.4	Simulation results for different depot management scenarios . . . . .	115
6.5	Simulation results for operational settings . . . . .	118

6.6	Sum of Squares method . . . . .	120
6.7	Simulations results . . . . .	121
6.8	Simulation results for different vehicle's capacity . . . . .	130
6.9	Simulation results for different vehicle's capacity . . . . .	132
A.1	List of papers . . . . .	167
A.2	List of characteristics . . . . .	168

# List of Figures

1.1	Ride-sharing as an intermediate layer in transportation . . . . .	6
1.2	Research objectives and contributions associated with the thesis chapters .	12
1.3	System components . . . . .	13
2.1	Different categories of dynamic ride-sharing . . . . .	25
2.2	Proposed dynamic ride-sharing system . . . . .	27
3.1	System components (fleet management component) . . . . .	42
3.2	Trip time schedule . . . . .	44
4.1	The assignment algorithm function . . . . .	54
4.2	Feasible routes for the example . . . . .	57
4.3	Lyon 6e + Villeurbanne: Mapping data ©Google 2018 and the traffic network	58
4.4	Temporal demand pattern in Lyon6 + villeurbanne (Medium-scale) . . . .	59
4.5	Lyon city in France . . . . .	60
4.6	Network of Lyon . . . . .	60
4.7	Temporal demand pattern in Lyon (large-scale) . . . . .	61
4.8	Dynamic assignment . . . . .	62
4.9	Trip situations . . . . .	65
4.10	Final routes after multidimensional scaling and k-means clustering . . . . .	68
4.11	Final routes after hierarchical clustering . . . . .	69
4.12	The assigned routes for requests 1 to 5 with and without the FOSH method	72
4.13	Final assignment algorithm . . . . .	73

5.1	System components (dynamic simulation component) . . . . .	78
5.2	Speed evolution in Lyon 6 + Villeurbanne . . . . .	81
5.3	Speed evolution in the whole Lyon city . . . . .	81
5.4	Rolling horizon and simulation period . . . . .	82
5.5	Predicted speed and mean speed during the simulation horizon (before $\epsilon$ ) .	83
5.6	Predicted speed and mean speed (after $\epsilon$ ) in Lyon 6 + Villeurbanne . . . .	83
5.7	Predicted speed and mean speed during the simulation horizon (before $\epsilon$ ) .	84
5.8	Predicted speed and mean speed (after $\epsilon$ ) in Lyon . . . . .	84
5.9	Central depot in Lyon 6 + Villeurbanne . . . . .	85
5.10	Comparison between the exact method and heuristic 1 (market-share = 1%)	92
5.11	Comparing heuristic 1 and heuristic 2 when the market-share = 10% . . .	93
5.12	Comparing the heuristic 2 and the heuristic 3 when market-share = 40% .	95
5.13	Comparing the number of trips for heuristic 2 and heuristic 3 when market- share = 40% . . . . .	97
5.14	Objective function difference for the solution method steps (market-share = 10%) . . . . .	98
5.15	Comparing clustering methods' objective function (market-share=50%) . .	100
5.16	Comparing clustering methods' computation time (market-share=50%) . .	100
6.1	Traffic situation for the number of sharing 0 with different market-shares (medium-scale) . . . . .	106
6.2	Traffic situation for the number of sharing 1 with different market-shares (medium-scale) . . . . .	107
6.3	Total travel time for all the cars for the number of sharing 0 and 1 with different market-shares (medium-scale) . . . . .	108
6.4	Total travel distance for all the cars for the number of sharing 0 and 1 with different market-shares (medium-scale) . . . . .	108
6.5	Traffic situation for market-share 100% with different numbers of sharing (medium-scale) . . . . .	109
6.6	Average vehicle speed (medium-scale) . . . . .	110

6.7	Traffic situation at peak hours (medium-scale) . . . . .	110
6.8	Passengers' waiting time for different sharing scenarios when the market-share is 100% (medium-scale) . . . . .	111
6.9	Passengers' waiting time for different time window (market-share = 100%, nshare = 1) . . . . .	113
6.10	Comparing different scenarios for depot management . . . . .	116
6.11	Traffic situation for market-share 100% with operational settings . . . . .	117
6.12	Sum of squares method for finding the optimal size of clusters . . . . .	120
6.13	Traffic situation for the number of sharing 0 with different market-shares (large-scale) . . . . .	122
6.14	Traffic situation for the number of sharing 1 with different market-shares (large-scale) . . . . .	123
6.15	Traffic situation for the number of sharing 2 with different market-shares (large-scale) . . . . .	124
6.16	Total travel time for all the cars for the number of sharing 0, 1 and 2 with different market-shares (large-scale) . . . . .	125
6.17	Total travel distance for all the cars for the number of sharing 0, 1 and 2 with different market-shares (large-scale) . . . . .	125
6.18	Traffic situation for market-share 100% with different numbers of sharing (large-scale) . . . . .	126
6.19	Traffic situation at peak hour (large-scale) . . . . .	127
6.20	Average vehicle speed (large-scale) . . . . .	128
6.21	Traffic situation for different vehicle capacity (market-share = 100%) . . .	130
7.1	Thesis summary . . . . .	136
7.2	Medium-scale results . . . . .	141
7.3	Large-scale results . . . . .	142
A.1	Paper 1 and Paper 2 . . . . .	169
A.2	Paper 3 and Paper 4 . . . . .	170



A.3	Paper 5 and Paper 6 . . . . .	171
A.4	Paper 7 and Paper 8 . . . . .	172
A.5	Paper 9 and Paper 10 . . . . .	173
A.6	Paper 11 and Paper 12 . . . . .	174
A.7	Paper 13, Paper 14 and Paper 15 . . . . .	175

# Part I

What is ride-sharing?

This part explains the motivation and background of the thesis. We explain the origin and the concept of ride-sharing and the related terms and concepts. A discussion is developed on the different parts of a dynamic ride-sharing problem and the interaction of such a service with the transportation network.

# Chapter 1

## Introduction

### 1.1 Context

Today's cities face challenges in terms of congestion, lack of space, growing population, air quality, noise, health, and economic development. Citizens want to be mobile and move from a to b - within and between cities - easily, cheap, smart, and clean. Expanding infrastructure in the urban environment is almost never an option and not a sustainable long-term solution: It is not cost-effective, there is no space, and it leads to environmental issues. Meanwhile, innovation and competitiveness are crucial for the future of transportation.

Large-scale deployment of New Mobility Services (NMS) is part of the solution in dealing with these challenges. Mobility as a Service (MaaS), shared mobility concepts, and smart bicycle solutions can contribute to wealthy, healthy, clean, and accessible cities.

Shared mobility has gained popularity in recent years, especially after the mini-revolution in transportation with the launch of shared mobility services like Vélib in Paris ([132]), Zipcar in America ([34]), Autolib in France ([82]), Car2Go in Germany ([59]) and others. Shared-use mobility describes transportation services that are shared among users, including public transit, taxis and limos, bike-sharing, car-sharing (round-trip, one-way, and personal vehicle sharing), ride-sharing (carpooling, vanpooling), ride-sourcing/ride-splitting, scooter sharing, shuttle services, neighborhood jitneys, and commercial delivery vehicles providing flexible goods movement. These new services represent innovative re-

sponses to the demand for new options and offer an opportunity to provide more mobility choices. Each service has specific operational specifications, constraints and variants. Among these services, ride-sharing and taxi-sharing are getting more popular, and big companies like Uber and Lyft ([47, 77]) are becoming reputed in this background.

Ride-sharing refers to a mode of transportation in which individual travelers share a vehicle for a trip and split travel costs.

The large travel demand for personal car transportation, together with low occupancy, leads to traffic congestion that is an increasingly important issue in many urban areas with rapid population and economic growth [139].

The most recent data for the average number of passengers per car (including the driver) for the European countries is approximately 1.45 passengers per vehicle [4]. This low occupancies together with the large demand for automobile transportation leads to traffic congestion in many urban areas. The congestion has cost nearly 100 billion euro, or 1% of the EU's GDP, annually [3]. In addition, private cars are the dominant transportation mode consuming fuel and producing carbon dioxide emissions [78, 154]. An average European car can emit more than  $120 \text{ gCo}_2/\text{km}$  [167, 61]. Therefore, a successful ride-sharing program that increases the occupancy of vehicles may make a significant saving on driving costs of the roadway system.

Dynamic ride-sharing (real-time ride-sharing or ad-hoc ride-sharing) refers to a system which supports an automatic ride-matching process between participants at very short notice or even en-route [6].

The dynamic ride-sharing problem involves two sub-problems: 1) Satisfying user requests and managing a vehicle fleet - 2) Accurately predicting travel times to determine vehicle availability and pickup/drop off times.

The first sub-problem corresponds to a fleet management optimization problem with multiple objectives and has recently attracted much attention.

The second sub-problem has been given less attention in the studies on the fleet management problem, but is very important for real field operations. Network congestion can have significant impacts on the ride-sharing service. The optimization system of the

ride-sharing service uses estimates for the predicted travel time obtained from a "prediction model". When the rides are executed, a gap can exist between the estimation and the real traffic condition. The "plant" represents the real traffic condition and it may require dynamic adjustment of the initial assignment to fit with the conditions observed. When simulating a dynamic ride-sharing service, it is essential to accurately distinguish the prediction from the plant to provide a realistic service. In most research, the plant is assimilated to the prediction model [68, 119].

Dynamic ride-sharing service is often advertised as a significant way of alleviating congestion and more generally as being eco-friendly, but few results exist to support this claim and some claims to the contrary have also been expressed [40]. Ride-sharing definitely reduces the number of cars travelling, but it can also increase travel distance. Both phenomena must be considered to evaluate the actual impact of ride-sharing. This PhD tackles this question, considering two sub-problems of dynamic ride-sharing, by simulating and solving an optimal fleet management problem.

## 1.2 Research background and motivation

Ride-sharing originated as a general concept to carry more than two people in the car of one of the persons who share a portion or all of their time schedules and routes [62]. However, the concept of ride-sharing evolves nowadays as a real-time and ad-hoc arrangement of cars to combine passengers' travels.

Dynamic ride-sharing refers to a system that supports an automatic ride-matching process between participants on very short notice or even en-route. It is distinguished from traditional carpooling and is focused on single, non-recurring trips that do not require long-term commitments between people to travel together for a particular purpose. Since ride-shares are established on-demand, a ride-sharing system is similar to other on-demand forms of passenger transit such as taxis and dial-a-ride services like airport shuttles [62, 39]. This compelling transportation mode has been the subject of many studies during the last years. A lot of researches has emphasized different advantages of ride-sharing for participants (drivers and passengers) like saving travel costs, reducing travel time, mit-

igating traffic congestion, conserving fuel, and reducing air pollution [41, 57, 129, 95]. The main mentioned advantage of this system is reducing traffic congestion. However, there is still a lack of studies that investigate the performance of the ride-sharing system while considering the real functioning of the transportation network, including congestion periods.

As figure 1.1 shows, dynamic ride-sharing (as a new mobility service) acts as an intermediate layer. On the one hand, it serves the passenger requests, i.e., the mobility demand, and, on the other hand, it can impact the network capacity and can be affected by the network conditions.

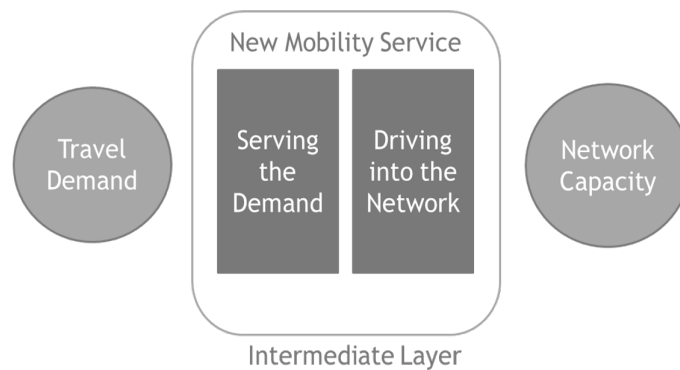


Figure 1.1: Ride-sharing as an intermediate layer in transportation

Thus, to assess the ride-sharing<sup>1</sup> system, and to investigate the impacts of service on traffic, we need to assess both problems. First, we have to find the best way to serve the network demand. Second, we have to assess the impact of the network condition on the ride-sharing system performance and finally, the impact of the ride-sharing system on the network traffic.

### 1.2.1 The performance of ride-sharing on the network demand

The first issue is matching the passengers and the fleet vehicles by solving a fleet management problem. Effective and efficient optimization technology that matches drivers and riders in real-time is one of the necessary components for a successful dynamic ride-share

---

<sup>1</sup>For the sake of simplicity, we will use the word ride-sharing instead of dynamic ride-sharing in this chapter.

system [6]. This problem has recently attracted much attention. Most approaches attempt to find the near-optimal solution to the matching problem in ride-sharing systems by considering specific constraints like vehicle capacity and the time window, to minimize the total additional distance [135, 143, 53] and maximize the match between vehicles and passengers [162, 118, 68]. They usually rank the possible, feasible matches for passengers and cars close to each other, based on the objective function and then choose the best match for the requests. When modeling the fleet management problem, it is important to consider the essential objectives and constraints of both passengers and providers. Much research work on ride-sharing has tried to minimize the total distance for cars to accommodate the trips requested [135, 53, 143]. This leads to minimizing the provider's costs. Travel time is an important feature for the participants [133].

Another important part of the first problem is to maximize the matching between cars and passengers. This is not only beneficial for the provider but also for the users, as it reduces the number of requests that cannot be served [6]. A large number of studies have tried to maximize the number of matches between participants [118, 68, 162]. The method we propose for fleet management matches all the requests with vehicles, based on travel time prediction. These matches between cars and riders are feasible only if they satisfy the participants' constraints. One of the most important considerations in dynamic ride-sharing is the time constraint. Many of the systems proposed in the literature let users choose their earliest and latest pickup times [113, 7]. Besides the constraints on travel time, the number of passengers that are sharing a ride at the same time is crucial as it can impact the passengers satisfaction. This parameter can also affect traffic.

In ride-sharing, the participant's willingness to share their ride is critical. Therefore, it is important to consider the rider's objectives too. In the dynamic ride-sharing method proposed by [7], the objective was to minimize the total vehicle-miles driven by all participants. They showed that this objective is aligned with societal objectives for reducing emissions and traffic congestion. Another important objective for the passengers is the time that they have to wait for the ride. As [162] showed, if no match is found before a specified time, the passenger is likely to leave the system. Previous research usually



focused on one of these objectives at a time, but it is important to take them into account simultaneously. We try to use an objective function that combines all these criteria. Matching users to trips is very challenging in real-time since it must happen very quickly. In a great deal of research, the optimal assignment is formulated as an integer linear programming problem and then different approaches are taken to optimize the problem. A further discussion on how we formulate this challenging problem are given in Chapter 3. The primary target of this research is not the computation time but to obtain solutions that are close to the global optimum. In Chapter 4, first, we propose an algorithm to find the optimal solution. However, as we are targeting problems with large instances, we face with the curse of dimensionality. Our solution approach is designed to be exact for small samples. It is then extended with several heuristics that keep the general design for the solution method but significantly reduce its computation time.

### **1.2.2 The impact of the network on the ride-sharing system and vice versa**

The second problem is related to the interactions between the ride-sharing system and the network capacity. This problem has been given less attention in the literature, but it is very important for real field operations. Network congestion can have significant impacts on the ride-sharing service. The optimization system of the ride-sharing service uses estimates for the predicted travel time obtained from a "prediction model". When the rides are executed, a gap can exist between the estimation and the real traffic condition. The "plant model" represents the real traffic condition, and it may require dynamic adjustment of the initial assignment to fit with the conditions observed. When simulating a dynamic ride-sharing service, it is essential to accurately distinguish the prediction model from the plant to provide a realistic service.

In most research, the plant and the prediction model are the same [68, 119]. There is no benchmark considering traffic conditions, but a few studies have considered the impact of traffic conditions on ride-sharing. [68] proposed an approach where the pick-up and drop off locations for passengers are selected from a fixed set. They considered

a randomly chosen overhead of 10-20 percent to reflect different traffic conditions when computing the end time for a driver. Even with this consideration, they used only the prediction model and assumed that the travel times during the assignment process stayed the same during the execution of the vehicle schedules. In some research, only the plant model is considered. A simulator was used to assess the dynamic ride-sharing but they did not optimize vehicle allocation as in [113], [119] and [89]. Other works used only static travel times in the optimization process [79, 80]. In Chapter 6, we define the plant model in addition to the prediction model to assess the impact of traffic conditions on the performance of the dynamic ride-sharing system for large-scale problems and see how the dynamic ride-sharing system can impact traffic congestion.

### 1.3 Research questions

A large body of literature has been dedicated to assess different parts of ride-sharing systems. From this literature review, it appears that (1) there is a lack of studies on efficient fleet management methods that can provide fast and qualified solutions in large-scale [130] and (2) there is very few studies to appropriately address the impact of traffic conditions on ride-sharing system performance.

In this thesis we aim to assess the performance of the dynamic ride-sharing system integrated with the current transportation network. The main research question is to assess the reduction in network congestion that can be obtained with ride-sharing. To answer this question, we have to consider the performance of ride-sharing on serving the passenger requests as well as the interactions between the network capacity and the ride-sharing system performance. Nested under general umbrella, the thesis investigates several research questions that are associated with six chapters (figure 1.2).

- *How to assess the performance of ride-sharing on the network demand?*

Three chapters of this thesis are dedicated to answering this problem. We solve this problem in a hierarchical manner by answering the following questions:

- *How to model a fleet management problem, for a ride-sharing system that con-*

*siders both participants main objectives and constraints, into the mathematical problem?*

In Chapter 2 we do a complete review of similar problems to find the gaps in previous researches on the fleet management problem. Then, considering these gaps, Chapter 3 describes the constraints and objective functions of the fleet management problem and finally presents the integer linear mathematical problem of the proposed ride-sharing system.

- *How to find high quality and fast solution for the fleet management problem in large-scale?*

First, in Chapter 4, we propose an innovative algorithm to find the optimal solution for the mathematical problem presented in Chapter 3. Our solution approach is designed to be exact for small samples. Then, the method is extended with several heuristics that keep the general design of the solution method but significantly reduce its computation time to make it scalable for large-scale problems.

- ***How to assess the impact of the network on the ride-sharing system and vice versa?***

The proposed algorithms in the first 4 chapters can answer the first question about serving the network demand by a ride-sharing system in large-scale networks. Then to assess the interactions of the network and the ride-sharing service, we face the following questions:

- *How to consider the real traffic dynamics?*

The first section of Chapter 5 presents a "plant model" which is based on the trip-based Macroscopic Fundamental Diagram and can represent the real traffic situation.

- *How to compute the vehicles travel time and speed during the assignment process?* A "prediction model" is presented in Chapter 5 that can be used during the assignment process.

- *How to manage the stop locations for the ride-sharing fleet vehicles in the network?*

A "Depot management" method is proposed in the second section of Chapter 5 to answer this question.

- *Can ride-sharing reduce traffic congestion?*

The last question is the main research question of this thesis. In chapter 5, first, we show the performance of the proposed algorithm, and then in chapter 6 we assess the performance of the proposed ride-sharing system in terms of congestion and do a sensitivity analysis on the different service parameters.

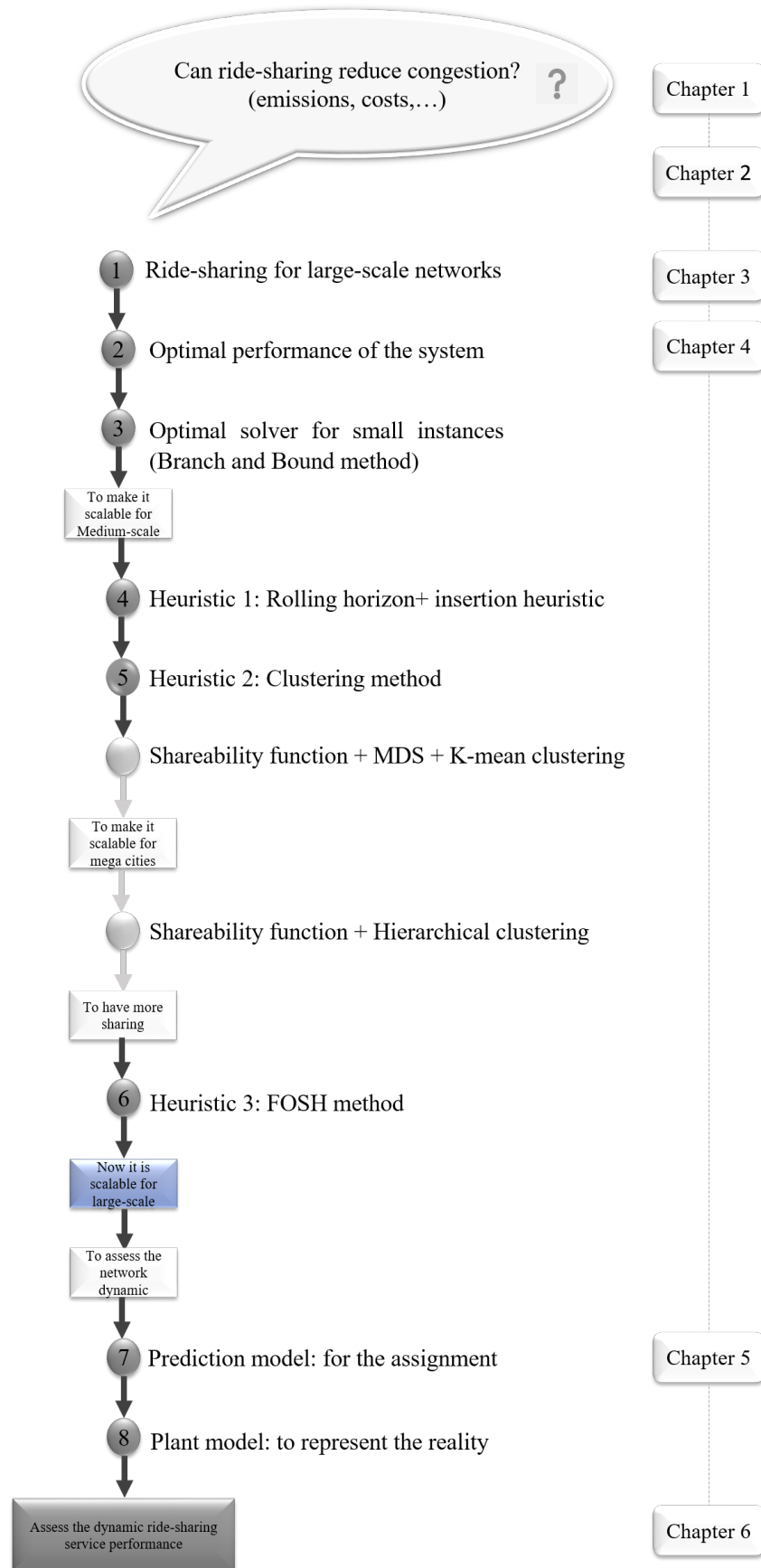


Figure 1.2: Research objectives and contributions associated with the thesis chapters

## 1.4 Methodology

The main research goal is to assess the performance of a ride-sharing system integrated with the network. We have designed a global framework to solve the problem. Our system has two main parts. The fleet management part deals with the matching process between riders and assigning the optimized match of riders to the vehicles. To handle this part, first, we use operation research methods to solve the fleet management problem and then we exert heuristic and data science methods to solve the problem in large scale. Then in the simulation part, we see how the optimal car schedule is realized considering the complete dynamic traffic conditions. The main components of the system are shown in figure 1.3.

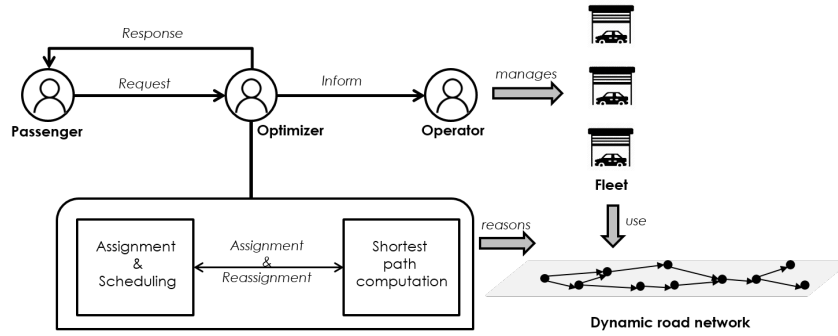


Figure 1.3: System components

The main characteristics of the ride-sharing problem we investigate are:

- Door-to-door dynamic ride-sharing (the passenger obtains service at the exact defined origin and destination).
- Passengers define the earliest pickup time and the latest arrival time. The passenger must be picked up, transported and dropped off at the destination inside this time window.
- All requests over the next prediction horizon (usually 20 minutes) are considered known at the beginning of the horizon.
- Each passenger defines the maximum number of persons they are ready to share a

trip with. We call it "number of sharers". The service has to guarantee that the number of sharers constraint is always satisfied for all the cars.

- Service time is added to each trip to reflect the time to stop and get in and out of the car.
- The service is provided by a limited number of vehicles that are initially all in the central depot. Local depots are uniformly distributed over the network to represent locations where cars can wait for further assignments. When an idle waiting car is needed to serve a passenger, it comes from the nearest non-empty depot. Note that the central depot can always generate new cars if necessary. When a car ends a trip without any further short-term assignment, it goes to the nearest depot and waits there.

## 1.5 Major contributions

In this section, we resume the major contributions of this dissertation. The contributions can be classified according to the designed system main parts (In figure 1.3):

- Fleet management:
  1. This thesis presents an innovative algorithm based on branch-and-bound method to find the exact optimal solution for the fleet management problem. This algorithm is able to provide all the feasible routes for the fleet vehicles. Then the operator can choose the best route regarding to the weights of objective functions.
  2. A rolling horizon approach is defined for the dynamic ride-sharing system to deal with the planning uncertainty. Also, An insertion heuristic method is developed to do the re-scheduling for the vehicles that are moving in the system at the beginning of each horizon.
  3. A clustering method is developed to cluster the trips that have more potential to be shared to solve large-scale problems. We have defined a "Shareability

Index" based on 3 different situations of sharing the trips or serving them in sequence. Then the clustering is done based on this index.

4. A heuristic approach called "Force the Sharing (FOSH) method" is presented to favor sharing opportunities at the cost of slight increases in the cost function, decreasing passenger waiting time.
- Dynamic simulation platform:
    1. A "Plant Model" is applied based on "Trip-based Macroscopic Fundamental Diagram (MFD)" to represent the traffic dynamics reality.
    2. A "Prediction Model" is applied based on the mean-speed to be used during the assignment process.
    3. A depot management system is developed to locate the empty waiting cars in the stop locations.

## 1.6 Thesis outline

This thesis includes four parts:

- Part 1: What is ride-sharing?

This part contains Chapter 1 (the current chapter) and Chapter 2. Chapter 1 explains the motivation and background of the thesis, the research questions, and the contributions to answer the questions. Chapter 2 provides a comprehensive review of the ride-sharing, the current algorithms designed for ride-sharing, the simulation models that are used for ride-sharing in the literature, and the consideration of dynamic traffic conditions in new mobility services and ride-sharing.

- Part 2: How to assess the performance of ride-sharing on the network demand?

Chapter 3 and 4 are dedicated to answer the question of part 2. In Chapter 3, the mathematical formulation of the ride-sharing problem is presented. Chapter 4 introduces an optimal solution method for the mathematical problem of ride-sharing. To show the quality of the proposed algorithm, we have compared the



algorithm with the existing methods in this chapter. To make the optimal solution scalable for mega cities and large-scale networks, Chapter 4, then presents a rolling horizon method and re-scheduling and a heuristic method based on clustering. The clustering method is compared with the current clustering methods in the literature. A new method called "FOSH method" is presented in this chapter to favor sharing opportunities and speed up the computations.

- Part 3: How to assess the impact of the network on the ride-sharing system and vice versa?

In Chapter 5 we answer this question. We have implemented two models to deal with traffic dynamics. A depot management system is developed to locate the waiting vehicles. In this chapter we show the performance of the optimal assignment algorithm and the heuristic methods. In chapter 6 we assess the impact of ride-sharing on traffic conditions. Also an extensive analysis is done on the problem parameters.

- Part 4: What are the major outcomes of this thesis?

In the last chapter, Chapter 7, we summarize the thesis and clarify the major conclusion and outcomes of this thesis.

## 1.7 Publication list

Most ideas and figures have been presented previously in the following publications:

### Journal papers

- **Alisoltani, N.**, Zargayouna, M. & Leclercq, L. (2020). A Sequential Clustering Method for the Taxi Dispatching Problem Considering Traffic Dynamics, *IEEE Intelligent Transportation System Magazine*, vol. 12, no. 4, pp. 169-181, 2020 [14].
- **Alisoltani, N.**, Leclercq, L. & Zargayouna, M. (2019). Can dynamic ride-sharing reduce traffic congestion in medium-scale cities? *Transportation Research Part B: Methodological*, (under review).

### Book chapters

- **Alisoltani, N.**, Zargayouna, M. & Leclercq, L. (2019). A multi-agent system for real-time ride-sharing in congested networks, *Agents and Multi-agent Systems: Technologies and Applications 2019*, pages 333–343, Springer [12]
- **Alisoltani, N.**, Zargayouna, M. & Leclercq, L. (2020). Real-Time Autonomous Taxi Service: An Agent Based Simulation, *Agents and Multi-agent Systems: Technologies and Applications 2020*, pages 199–207, Springer [13]

### Peer-reviewed conference proceeding

- **Alisoltani, N.**, Leclercq, L. & Zargayouna, M. (2019). Optimal fleet management for real-time ride-sharing service considering network congestion, In *Transportation Research Board 98th Annual Meeting*, Washington DC, USA [10]
- **Alisoltani, N.**, Zargayouna, M. & Leclercq, L. (2020). Real-Time Autonomous Taxi Service: An Agent Based Simulation, *KES AMSTA 2020, Agents and Multi-agent Systems: Technologies and Applications*, Split, Croatia
- **Alisoltani, N.**, Zargayouna, M. & Leclercq, L. (2019). A multi-agent system for real-time ride-sharing in congested networks, *KES AMSTA 2019, Agents and Multi-agent Systems: Technologies and Applications 2019*, St.Julians, Malta

## International conference presentations

- **Alisoltani, N.**, Leclercq, L. & Zargayouna, M. (2019). Real-time ride-sharing systems performance considering network congestion, *the 8th Symposium of the European Association for Research in Transportation (hEART 2019)*, Budapest, Hungary.[9]
- **Alisoltani, N.**, Leclercq, L. & Zargayouna, M. (2019). Network performance under different levels of ride-sharing: a simulation study, *the 10th Triennial Symposium on Transportation Analysis (TRISTAN X)*, Hamilton Island, Australia. [8]
- **Alisoltani, N.**, Zargayouna, M. & Leclercq, L. (2019). Data-Oriented Approach for the Dial-A-Ride Problem, *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, UAE [11]

## Working paper

- **Alisoltani, N.**, Leclercq, L. & Zargayouna, M. (2020). The impact of dynamic ride-sharing on the network congestion in large-scale.

# Chapter 2

## Literature Review

Ride-sharing was the subject of many studies undertaken in the past. In the previous chapter, we mentioned that to assess the ride-sharing problem's performance, we have to assess the interactions of the ride-sharing system and the transportation network. To evaluate this problem, we face two sub-problems: First, how to serve the network demand and second, how to assess the interactions with the traffic dynamics. Previous studies on this field mainly focus on the first sub-problem, and the second sub-problem has been given less attention in the literature, while it is very important to consider both sub-problems, especially if an operational deployment is envisioned.

Chapter 2 does a review of the previous research on this domain, considering these two sub-problems.

Section 2.1 reviews the ride-sharing definition and similar concepts. Section 2.2 presents different system settings for ride-sharing. Section 2.3 discusses different algorithms and methods being used in the literature to match the rides and riders in ride-sharing. Finally, in section 2.4, we present a literature review about considering the dynamic traffic condition in the ride-sharing performance.

### 2.1 Ride-sharing definition and related terms

The growing pressure on urban transportation systems needs innovative solutions that can increase its efficiency. In recent years, intelligent transportation systems have reshaped

traditional transportation supply with the rapid introduction of new mobility services. Among these services, ride-sharing is becoming popular [166]. In this section, we review the definition of ride-sharing and similar concepts to it, the advantages of ride-sharing, and a brief history of ride-sharing and its existing applications.

There have been several different ways to define the term ‘ride-sharing’ over the last decades. In 1989, the State of Virginia in the united states has described one of the first definitions as below:

“Ride-sharing arrangement means the transportation of persons in a motor vehicle when such transportation is incidental to the principal purpose of the driver, which is to reach a destination and not to transport a person for profit.” [2] After that, many scientific publications described and analysed the early schemes of ride-sharing, such as [147, 76, 57, 36, 95]

In 2010, Agatz et al. ([5]) define the ride-sharing as a system that aims to bring together travelers with similar itineraries and time schedules. They underline the point that to be widely adopted, ride-sharing must be easy, safe, flexible, efficient, and economical. Also, it must be able to compete with one of the main private cars’ preferences, which is the immediate access to door-to-door transportation.

The definition of ride-sharing has changed over time, depending on the specific setup and location [19]. Chan and Shaheen in a survey on the ride-sharing in North America, define the ride-sharing as: “The grouping of travelers into common trips by car or van.” In 2013 Furuhashi et al. made this definition more complete and defined the ride-sharing as: “A mode of transportation in which individual travelers share a vehicle for a trip and split travel costs such as gas, toll, and parking fees with others that have similar itineraries and schedule.” [62]

Cohen and Kietzmann define a different concept of ride-sharing in [44]. They represent a financial model of ride-sharing as “Drivers earn extra money while intermediaries earn up to 20% of each transaction”.

Recently, Wang et al. present one of the latest definitions: “Ride-sharing is an emerging transport mode that harnesses both private cars and taxis to combine two (groups) of

travelers into the same vehicle if all or part of the two groups' travels is overlapped in space and time" [175]. It consists of sharing a car using an e-hailing application to save costs and resources.

The important point is to distinguish ride-sharing from similar transportation modes such as dial-a-ride, car-sharing, carpooling, ride-sourcing, etc.

In a recent study, the authors in [130] review the definitions and characteristics of different mobility services. The dial-a-ride (DARP) provides shared trips between any origin and destination in response to advanced passenger requests within a specific area [128, 83]. The main difference between these services and ride-sharing services is that most shared services aim to minimize the response time to passenger requests, whereas dial-a-ride systems aim to minimize vehicle operating cost by reducing the number of vehicles used to serve given passenger demands [92].

Car-sharing consists of a group of individuals who share a fleet of cars with other members [94]. In British English, the words car-sharing and lift-share are used anonymously to ride-sharing [41]. Generally, the term carpooling is interchanged with ride-sharing, but researchers use this term to show a transportation mode where an individual gets access to a pool of cars for hourly or daily use for the commute to and from a shared location like university or workplace [41, 5, 74, 158]. Ride-sourcing is another mode that performs like traditional taxi cabs that serve one request at a time without sharing. The difference is that ride-sourcing uses smartphone technology and dynamic matching algorithms [145, 91].

In private-car ride-sharing, the rider is both a driver and a passenger. Taxi-sharing is a type of ride-sharing where the driver is just a professional taxi driver. Currently ride-sharing is undergoing an expansion from the traditional private car ride-sharing to taxi-sharing [125, 84], and in the near future to autonomous vehicle taxi-sharing [101].

### 2.1.1 Dynamic ride-sharing

Nowadays, transportation, like many other aspects of daily life is being transformed by the information technology (IT) revolution [69]. The spread of mobile devices and the

development of Global Positioning System (GPS) make it possible for all the transport operators to adapt in real-time the transportation supply to travel demand. These new technologies have made considerable changes in the transportation modes as well as taxis [160, 15]. These options can make the possibility to have access to the vehicles position at any time and perform the matching process of ride-sharing in real-time. These possibilities has led to the development and progress of a new type of ride-sharing which is called dynamic ride-sharing.

Dynamic ride-sharing, which is also known as real-time ride-sharing, real-time peer to peer ride-sharing, *ad hoc* ride-sharing, instant ride-sharing, and dynamic carpooling [7, 66] is a transportation mode that provides rides for single, one-way trips. In dynamic ride-sharing, the sharing is arranged on a per-trip basis rather than for trips made regularly [39]. Dynamic ride-sharing systems need to support arbitrary locations and travel times to match users [158, 49, 64]

In 1994, Schweiger *et al.* defined the dynamic ride-sharing for the first time: "a mode of transportation that is ready when you are. This service is multipurpose and can be arranged either in real-time or close to it. Participants pre-qualify and are put into a database. Upon receipt of a trip enquiry, the database is searched for others who are traveling in the same direction at the same time. Participants can not only use this database to arrange for carpools to and from work but also to a shopping center, medical facility or any other trip generator" [155]. Two years later, the authors in [100] presented the next definition: "a one-time rideshare match obtained for a oneway trip either the same day or the evening before."

In 1997, Dailey *et al.* [49] defined the dynamic ride-sharing as:

"two or more people sharing a single trip, without regard to previous arrangements or history among the individuals involved. In comparison to traditional ride-matching services, which focus on commuters traveling to and from the same origins and destinations on fixed schedules, a dynamic ride-sharing system must be able to match random trip requests at any time. Thus, the system must be able to match potential carpoolers quickly to respond to same-day trip requests, as well as the more traditional commute trips".

'dynamicridesharing.org' defines dynamic ride-sharing as "A system that facilitates the ability of drivers and passengers to make one-time ride matches close to their departure time, with sufficient convenience and flexibility to be used on a daily basis" [97].

Another recent definition is presented by [20]: "A single or recurring rideshare trip with no fixed schedule, organized on a one-time basis, with matching of participants occurring as little as a few minutes before departure or as far in advance as the evening before a trip is scheduled to take place."

Agatz *et al.* [7] describes the dynamic ride-sharing as "an automated system that facilitates drivers and riders to share one-time trips close to their desired departure times." They mention that the system is dynamic. It serves independent and non-recurring trips (not like the traditional carpooling or van-pooling). The trips are prearranged (not like casual ride-sharing, hitch-hiking and hailing a taxi) and the matching process is automated.

### 2.1.2 Ride-sharing history and advantages

Chan and Shaheen [41] categorize the ride-sharing evolution into 5 phases. These five phases begin from car-sharing clubs to today's technology-enabled ride-matching systems. For the first time, during the World War II, the U.S. government introduced the first organized ride-sharing to save resources. [62, 41]. The second time in history was in the 1970s during the oil crises when ride-sharing gained attention[141]. During this phase vanpooling, casual carpooling, park-and-ride facilities, and innovative ride-matching programs used as strategies to facilitate ride-sharing.

In the 1980s and 1990s, in the third phase of ride-sharing evolution, transportation management starts focusing on solving congestion and air pollution issues. At this time, the dynamic ride-sharing becomes more considerable because of the advances in internet-based ride-matching programs and the facilitation of using telephones everywhere [75, 109]. However, due to low oil prices and strong economic growth, participation in ride-sharing decreased at the beginning of the 1980s. From 1999 to 2004 ride-sharing has gained momentum with the use of technology-enabled devices and most ride-sharing systems try



to develop this mode among travelers who had the most reliable trip schedules [67, 41]. Nowadays, modern information technology and the flexibility they have provided has made ride-sharing more attractive than ever before. In many countries around the world, different companies are providing ride-sharing services [99, 90].

Another considerable advancement in ride-sharing is the appearance of autonomous vehicles (AV). The development of AVs can make a significant improvement in the ride-sharing systems and, consequently, the transportation system. One of the ride-sharing obstacles is the drivers' limitations regarding distance, route and time window, removing these limitations is a valuable point to make AVs appropriate for ride-sharing, besides reducing emissions, making streets safer, saving time, saving space and reducing congestion [101]. Nowadays, a significant part of studies on this field focuses on the assessment of autonomous vehicles sharing [55, 108, 157, 72, 33].

The sharing community has a lot of social, environmental and economic advantages [73, 99]. Andreasson [21] mentions that besides the objections like longer waiting times, loss of privacy and risk of assaults or harassment by other passengers, ride-sharing can reduce the fleet size, increase the system capacity and reduce the requirement for short headways. Specifically, different studies have focused on different advantages of ride-sharing. We mention some of the recent studies on this objective:

Lokhandwala and Cai in 2018 [117] propose an agent-based simulation model to study the impacts of dynamic ride sharing. They show that ride-sharing combining autonomous driving with autonomous vehicles in New York city, can potentially decrease the fleet size by up to 59%, increase the occupancy rate from 1.2 to 3, decrease total travel distance up to 55%, and reduce carbon emissions by 725 metric tonnes per day. A detailed analysis about the environmental benefits of ride-sharing in terms of reduced emissions and vehicle kilometres travelled is demonstrated by [40]. Another study in Paris shows that ride-sharing can play a key role in reducing CO<sub>2</sub> emissions from road transport [176]. Research in Beijing shows that if implemented for the entire taxi fleet, shared taxis can save 28.3 million gallons of gasoline and reduce 2392 tons CO emissions annually [37]. Similar researches are done in different mega-cities around the world to show the environ-

mental benefits of ride-sharing [164, 115, 88]. Besides reducing CO2 emissions and fuel consumption, it also saves the users' money and time [168].

## 2.2 Dynamic ride-sharing system settings

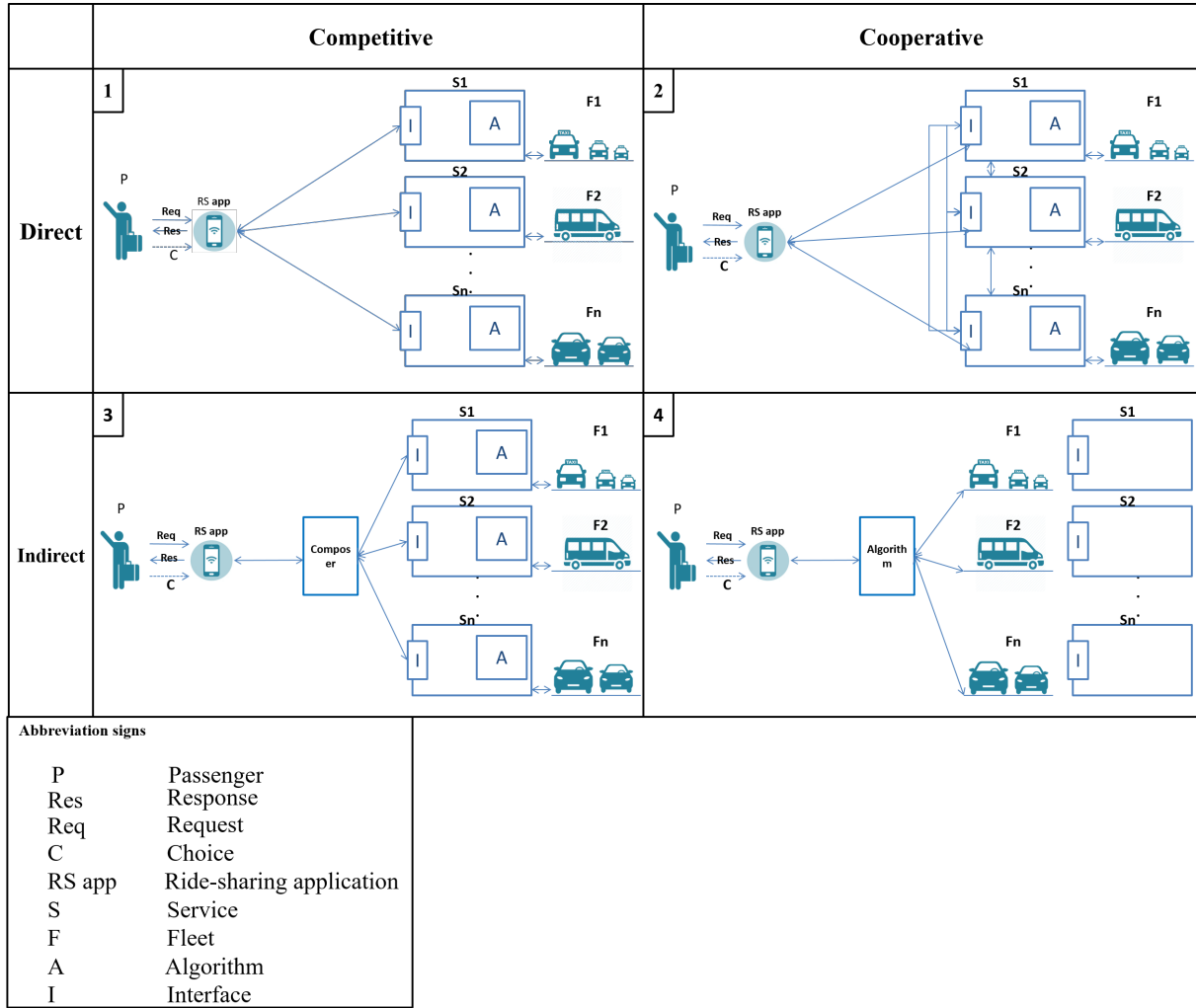


Figure 2.1: Different categories of dynamic ride-sharing

Considering the history and definitions of ride-sharing, we define the dynamic ride-sharing in this thesis as below:

The main parts of this dynamic ride-sharing system are the passenger, ride provider, and the matching algorithm. The passenger seeks a ride to pick her/him up at the origin point and drop her/him off at the desired destination within a time interval. The ride provider has a fleet of vehicles (taxi, van, autonomous car, etc.) that is ready to serve

the passengers' requests. The matching algorithm receives the requests and the fleet information and tries to find the best matches in short notice.

We classify this system into four categories, considering the interactions between different fleet providers and the performance of the matching algorithm. Fleet providers can work competitively or cooperatively. In the case of cooperative interaction, they share the fleet information so the passenger may get the ride easier, and the benefits of ride-sharing will be more achievable. In the competitive scenario, each provider tries to gain a higher profit. The matching algorithm communicates with the providers directly or indirectly, (figure 2.1). This classification can be expanded for other new mobility services as well.

Nowadays, the common dynamic ride-sharing systems are placed in the first category. The research proposed by this dissertation considers this category. Figure 2.2 shows the system functioning. Passengers send their requests for the ride via an application and define their trip specifications. On the other side, the application has access to the fleet information and can track the fleet vehicles at any time. Then the algorithm starts to assign vehicles to the passengers' requests and send the response to the passengers in real-time.

The matching algorithm is a key element in the efficiency of such a system. It is possible to provide the best real-time match between vehicle and passenger in terms of time and location, using efficient algorithms. A huge number of studies like [98, 43, 29, 153] focused on finding such an algorithm. In section 2.3, we review the different ride-sharing algorithms in the literature.

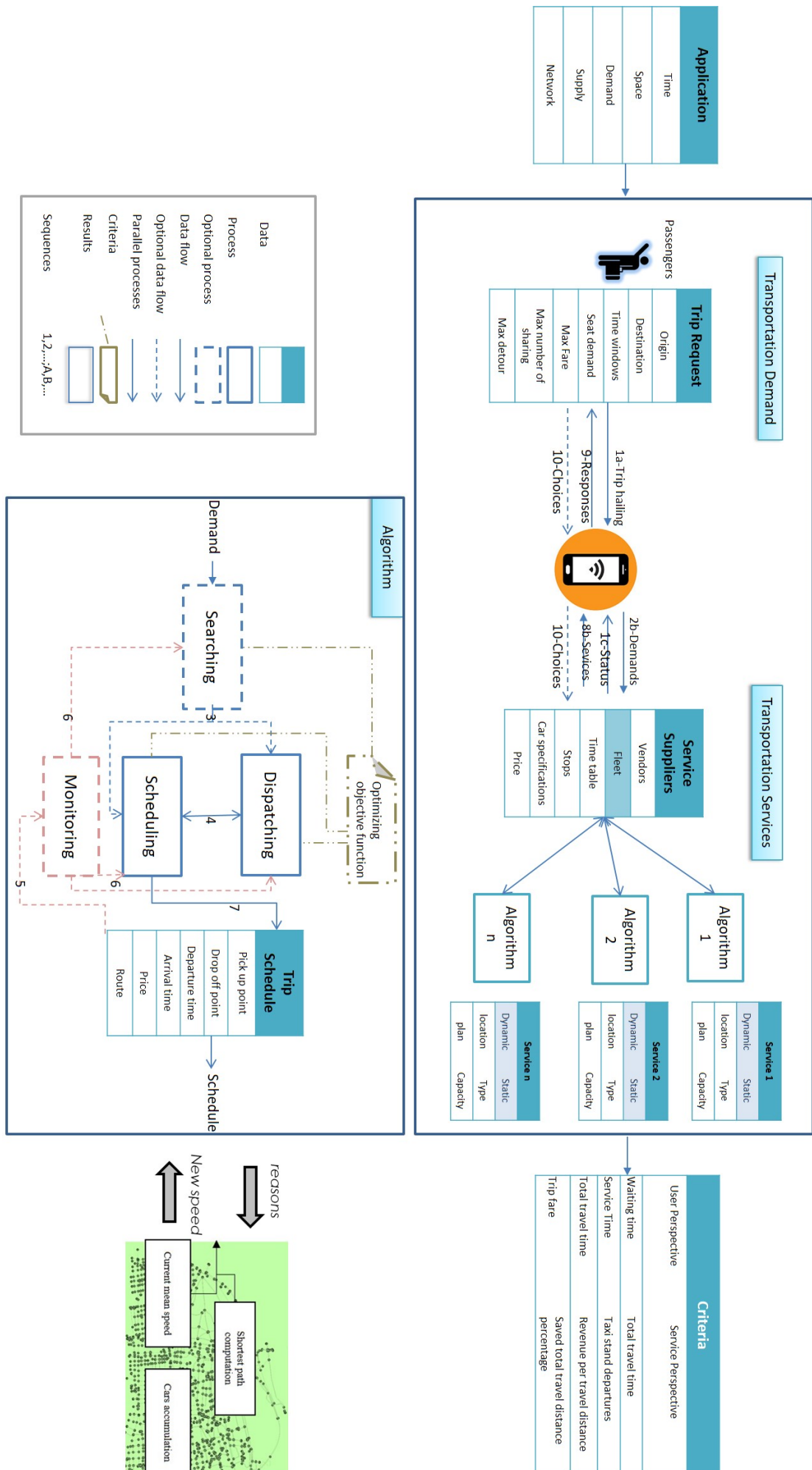


Figure 2.2: Proposed dynamic ride-sharing system

Transportation demand	Trip Planning	Time Dependent
		Real time demand
		Trip hailing
		Authentication System
		Trip Motivation
		Special passenger
		Origin
		Destination
		Seat Demand
	Trip	Max waiting time
		Max Travel time
		Max Detour
		Max Fare
		Time window
		Desired pick up time
		Desired arrival time
		Max delay
		Max number of sharing
		Distance from origin to pick up
		Departure time
		Waiting time
		Number of Sharing
		Detour
		Arrival time
		Time for picking up and dropping off
		Fare
Transportation Service	Vendors	
	Fixed Fleet	
	Time table	
	Fixed stops (pick up or drop off locations)	
	Waiting locations	
	Car Specifications (homogeneous/heterogeneous)	
	Supply type (pro/not pro)	
	Car capacity	
	Pricing scheme	
Algorithm	Objective function	
	Dispatching	
	Searching	
	Scheduling	
	Monitoring	
	Method	
	Shortest path	
	Empty vehicle behavior	
	Accept/reject behavior of riders	
Traffic		
Application	Network	Characteristics
	Scale	Time
		Space
		Demand
		Supply
Criteria	Passengers' objectives	Waiting time
		Service Time
		Total travel time/distance
		Trip fare
		Rejected Requests
		Max delay
	Providers' objectives	Total travel time/distance
		Number of served requests
		Number of required vehicles
		Taxi stand departures
		Revenue per travel distance
		Average number of persons per vehicle
		Vehicle emissions
Saved total travel distance percentage		

Table 2.1: Transportation demand and service characteristics

The proposed system in figure 2.2 has three main parts: Transportation demand, transportation service, and algorithm. Different studies on ride-sharing consider different con-

figurations for this system. Usually, they use data from a real network to assess the performance of the ride-sharing system, and they consider different criteria based on the study's goals. We extract all the related parameters and variables for these parts. Then we can have a comprehensive review of different methods. Table 2.1 shows these items for each box of figure 2.2. We have evaluated 15 most cited or related research proposals on ride-sharing using this table (See Appendix A).

In dynamic ride-sharing, the system works to serve the network demand, and at the same time, it is affected by the network. The traffic situation can impact the shared vehicle travel times. On the other side, the system vehicles can have an impact on the congestion (Figure 2.2). So, we can divide the dynamic ride-sharing problem into two sub-problems. First, how to serve the network demand and manage a fleet of vehicles. Second, how to assess the impact of network and accurately predict the travel times to determine vehicle availability at pickup/drop off times. In the next sections, we provide state of the art on the ride-sharing algorithms and dynamic traffic conditions in ride-sharing.

## 2.3 Ride-sharing algorithms

One of the most challenging facets of dynamic ride-sharing problems is the automated matching process. Matching users to trips is very challenging in real-time since it must happen very quickly.

The matching problem in ride-sharing corresponds to a fleet management optimization problem with multiple objectives and constraints and has recently attracted much attention. In section 2.3.1, we first review the different objectives and constraints that have been considered for this problem in previous studies and then in section 2.3.2 we review the methods and algorithms to solve the problem.

### 2.3.1 Objectives and constraints

When modeling the fleet management problem, it is important to consider the essential objectives and constraints of both passengers and providers. In table 2.1, we have

listed the constraints for both passengers and providers. Furthermore, the criteria for the problem show the different objectives that can be considered in the fleet management problem.

**Objective functions** Different researches focus on different objectives in the problem related to passengers [156, 120], drivers [116, 48], providers [140, 121, 86] or the network [30].

Different objectives of the ride-sharing problem used in the literature review are listed in table 2.1 as criteria. We categorize the ride-sharing problem objectives into providers' objectives and passengers' objectives. Most approaches attempt to find the near-optimal solution to the matching problem in ride-sharing systems by considering specific constraints, to minimize the total travel distance or time [135, 143, 53] and maximize the match between vehicles and passengers [162, 118, 68]. This leads to minimizing the provider's costs.

Maximizing the number of serviced requests [80, 28, 161, 152] and minimizing the number of required vehicles [96, 107], saved total travel distance percentage [135] or vehicle emissions [22] are other objectives used in the state of the art of this problem that can yield a better performance from the system (provider) perspective.

In ride-sharing, the participant's willingness to share their ride is critical. Therefore, it is important to consider the rider's objectives too. In the dynamic ride-sharing method proposed by [7], the objective was to minimize the total vehicle-miles driven by all participants. They showed that this objective is aligned with societal objectives for reducing emissions and traffic congestion.

Travel time is an important feature for the passengers [133]. Another important objective for the passengers is the time that they have to wait for the ride. Usually, the main motivation of passengers is to reduce travel costs. However, [162] showed that, if no match is found before a specified time, the passenger is likely to leave the system and refuse to use shared mobility systems. Minimizing total passenger ride or waiting time might yield better performance from the passenger perspective but not from a system-wide perspective. Reducing the passenger waiting time can make shared services comparable

with the other traditional taxi services. Different methods in the literature try to minimize the passengers' waiting time as an objective function like in [87, 96, 126].

Much of the research on dynamic ride-sharing services is focused on optimizing a single providers' objective. In these studies usually they consider some constraints for the problem to keep the passengers' objectives in an acceptable level [128, 38]. There are also studies that consider multiple-objective systems combining passengers' and providers' objectives. In [53], total travel distance, taxi stand departures (number of exits from all taxi stands) and revenue per travel distance (revenue per km earned by all vehicles) are used to demonstrate vehicle owners' performance in terms of operation mode and costs for all trips. Percentage of served requests, waiting time, travel time and trip fare are considered to demonstrate the passengers' preferences. [96] combined six different objectives: Minimizing routing cost, excess ride time, passenger waiting time, route durations, early arrival times at pickup and delivery nodes, and number of unserved requests.

When assessing a dynamic ride-sharing problem it is important to consider both passengers' and providers' essential objectives. In this thesis, we aggregate the different passengers' and providers' objectives into a weighted-sum objective with different measures. We define a weight for each of the combined objectives (Chapter 3).

**Constraints** A set of constraints and features should be considered for both transportation demand and transportation service in the ride-sharing problem. Many studies on ride-sharing formulate the optimal assignment problem using different vehicle routing problem (VRP) formulations. They use different constraints and features for various types of problems. Assignment constraints, synchronization constraints, time window constraints and capacity constraints are examples of different formulation constraints.

Assignment constraints are the very first constraints of a ride-sharing problem. A passenger should be transported from the pickup point to the drop off point, and the pickup point should be visited before the destination point. Therefore, the assignment constraints are strict and should be respected in solving a ride-sharing problem. This process has modeled with different methods in the literature. The authors in [163] define an intermediate location called meeting point for passenger picked up or dropped off. They show



that this meeting point can lead to shorter detours. The authors in [68] present an approach to ride-sharing where the pick up/drop off locations for passengers are selected from a fixed set. They present a scheme that chooses optimally fixed locations of Pick up Points (PuPs) and aim to maximize the car occupancy rates and preserve user privacy and safety. In [133] the drop off point is a common destination like university or company. Other researches focus on serving the passengers door-to-door to make more comfort for the sharing participants. In [135, 110, 165] passengers can define the pick-up and drop off locations. In [53, 56, 98], the pick-up point is the current location of the passenger, and passengers only need to define the destination.

With synchronization constraints, the system ensures that each passenger is served exactly once by one or more vehicles [58, 54].

One of the most important considerations in dynamic ride-sharing is the time constraint. For the ride-sharing, it is important to define a time window, especially for the cases that both drivers and riders have a time schedule that should be matched. Much research on ride-sharing, consider the time window in their problem constraints. They define the time window via different strategies. In [7], the passenger defines the earliest departure time and time flexibility that specifies the difference between the earliest departure time and the latest arrival time. Then the latest arrival time can be computed. In [133, 173, 80, 113, 7], the passengers define the earliest pick up time and latest arrival time. So, the passenger should be picked up at the origin point not earlier than the defined pick up time, and he/she should be dropped off at the destination point before the latest arrival time.

In [87], the travelers have implicit rather than explicit time constraints, and only passenger waiting time is considered in the objective function. Also, the capacity constraint for each vehicle is two. It means that each car can serve only two traveler requests at the same time. Some studies try to use the vehicle's maximum capacity [68, 53]. Besides limiting the maximum number of passengers, some studies on van-pooling systems put a constraint for the minimum number of passengers to form a van-pool for a shared trip [93].

Besides the constraints on vehicles' capacity, this thesis introduces the desired maximum number of ride-sharing for all passengers. Thus, in a given vehicle, the number of on-

board passengers cannot exceed the lowest number of passengers willing to share for all the on-board passengers. This parameter can also affect traffic.

### 2.3.2 Solving methods

Most approaches attempt to find the near-optimal solution to the matching problem in ride-sharing systems by considering specific constraints. They usually rank the possible, feasible matches for passengers and cars close to each other, based on the objective function and then choose the best match for the requests [162, 118, 68, 135, 143, 53].

The authors in [87] assigned the passenger to a vehicle only if the car is 20% closer to the passenger than any idle shared car.

The ride-sharing assignment is a pickup and delivery problem with time windows (PDPTW) [122]. [27, 138] did reviews of dynamic pickup and delivery problems, bringing up some interesting but unsolved questions, such as optimal waiting strategies, modifications of the objective function on a rolling basis, to name a few. It is shown that this problem is NP-hard. Even simplified variants of the problem with a single-driver single-rider setting, single pickup, and drop off or a single-objective function are still NP-hard [71]. In a great deal of research, the optimal assignment is formulated as an integer linear programming problem, and then different approaches are taken to optimize the problem [178]. There is a vast literature on solution methods and algorithms for these problems. However, there is still room for improvement in these methods.

Due to the complexity of the problem, the exact solution methods are introduced to solve very small instances of the problem. The most frequently cited literature on PDPTW is [46], where they present a mixed linear integer programming formulation of PDPTW and a branch and cut solution for it. [148] later introduces an enhanced branch-and-cut-and-pricing solution to further improve the solution. These exact methods are usually used to solve static problems with deterministic data [46, 24, 65]. In the PDPTW, increasing the number of vehicles and passengers increases the dimension of solution space and so the computational time. The method proposed by [122] takes almost two hours to compute a case with 50 passengers and 15 vehicles.

To handle larger problems, the assignment problem is solved with different heuristic methods in the literature. [35] propose an exact Branch-and-Cut algorithm for Dial-a-Ride problem (DARP) that can outperform the state-of-the-art solver CPLEX. The exact method is followed by a lean heuristic algorithm based on Large Neighborhood Search (LNS) for larger instances of the problem. In another study on pick up and delivery problem for ride-sharing by [173], the exact method is proposed to solve small instances of the problem. Then they present a Tabu search heuristic for the pick and delivery problems for ride-sharing. They use a ratio (the objective value of the optimal solution divided by the output objective value of the heuristic method) to compare the heuristic method with optimal situation.

Dynamic ride-sharing problem addresses short-term matching or even en-route matching [7, 118, 51]. This fact makes the assignment problem more complex.

In some studies, researchers try to narrow the feasible solution space to make the computations faster and be able to assign the vehicles to the requests that are coming at each time to the system. For example, [114] present a method to tighten travelers time windows and eliminate unnecessary variables and constraints to narrow the solution space. [85] proposed a branch-and-bound algorithm for solving real-time ride-sharing problem. They introduced a kinetic tree algorithm to schedule dynamic requests and adjust the routes on-the-fly. [114] proposed a branch-and-cut algorithm to solve a realistic DARP with multiple trips and request types and a heterogeneous fleet of vehicles.

Some researches have implemented meta-heuristic methods to solve the assignment problem [16, 18]. [81] used a genetic algorithm to find a sub-optimal solution for the ride-matching problem, and then an insertion heuristic took care of the newly received requests by modifying the solution of a genetic algorithm when possible. [92] proposed hybrid-simulated annealing (HSA) method to dynamically assign passenger requests to shared taxis. Masmoudi et al. (2017) propose an improved ALNS-based method, Hybrid Bees Algorithm with Simulated Annealing (BA-SA), and with Deterministic Annealing (BA-DA) to solve the Heterogeneous Dial-a-Ride problem (HDARP).

[7] introduced a rolling horizon approach that can provide high-quality solutions for dy-

dynamic ride-sharing systems where trip requests continuously enter the system.

Recently [130] has presented a survey of models and algorithms for optimizing shared mobility, and they have shown that one of the most critical problems in the solution for these systems is computation time and the quality of the results.

In this dissertation, we propose heuristic methods for the mathematical problem that gives a high-quality optimal assignment in a few minutes. The algorithm searches for the matches between riders only in the area feasible for each passenger. Then, the match between two participants is possible only if their feasible area has an intersection.

### 2.3.3 Clustering methods

In the large-scale problems, the number of received requests at every time is a huge number. It has been indicated in the literature that the patterns of demands and the patterns of supplies are spatially-temporally dependent [172].

A lot of researches on this domain uses different clustering methods to consider these dependencies. They use methods like dividing the time into several time slots or dividing the space into several clusters, road segments, or cells [70, 50, 142, 177].

Qiang in [144] propose an algorithm to use the dataset of taxi get-off points to achieve the clustering of taxis on urban roads and compare their method with the classical clustering methods. However, the taxi clustering data in their study are conducted in a static environment.

In [42] all pickup points are partitioned into several clusters and the vehicle dispatch and ride-sharing problem is solved in each cluster. Bard and Jarrah in [26] show that for large-scale problems, an appropriate solution is clustering the demand nodes and downsizing the network. Some researches try to limit the feasible region with clustering methods to speed up the computation. They usually divide the demand nodes in the network into geographically dense clusters [136, 150].

One of the recent researches on the clustering of the trips is done by [151]. They introduce the notion of a shareability network to quantify the spatial and temporal compatibility of individual trips in a dynamic environment. In their method, two trips are shareable

if they would incur a delay of no more than five minutes. Then, [169] modify the idea to model the sharing of vehicles instead of rides and address the minimum fleet problem in on-demand urban mobility. In these clustering methods, the trips are clustered just based on the situation of the origin points. But, in ride-sharing, other combinations of trips should be considered. So, in this research, we propose the concept of "sequential index" and "Shareability index" for the same purpose to assess the possibility of serving two trips with the same car in sequence or sharing the trips. Our proposal employs a method that reduces the number of required vehicles.

## 2.4 Dynamic traffic conditions in ride-sharing

One of the main mentioned advantages of ride-sharing is reducing congestion. Most of the studies on this domain assess the impact of ride-sharing on the congestion, and with statistical analysis, they show this impact [111, 112, 52].

The critical point which has not been considered is that dynamic traffic conditions in the network can have significant impacts on the ride-sharing service as well.

The optimization system of the ride-sharing service uses estimates for the predicted travel time obtained from a so called "prediction model". When the rides are executed, a gap usually exists between the estimation and the real traffic condition. The so called "plant model" represents the real traffic condition and it may require dynamic adjustment of the initial assignment to fit with the conditions observed. When simulating a dynamic ride-sharing service, it is essential to accurately distinguish the prediction and the plant models to provide a realistic service.

In most research, the plant model and the prediction model are the same [179, 68, 119]. There is no benchmark considering traffic conditions, but a few studies have considered the impact of traffic conditions on ride-sharing [134, 174]. For instance, the authors in [68] proposed an approach where the pick-up and drop off locations for passengers are selected from a fixed set. They considered a randomly chosen overhead of 10-20 percent to reflect different traffic conditions when computing the end time for a driver. Even with this consideration, the authors used only the prediction model and assumed that the

travel times during the assignment process stayed the same during the execution of the vehicle schedules. Other works used only static travel times in the optimization process [80].

In some research, only the plant model is considered. For instance, in [113], [119] and [89], the authors used a simulator to assess the dynamic ride-sharing but they did not optimize vehicle allocation. In [55], the authors described advances in existing shared autonomous vehicle simulations, by enabling dynamic ride-sharing. They proposed a method that fits for passengers with similar origins, destinations and departure times. In our method, each passenger defines their origin, destination and desired time window.

In [25], the authors develop a general economic equilibrium model at the macroscopic level to describe the equilibrium state of a transportation system composed of solo drivers and the e-hailing service providers. Experimental results show that when there is little in the symmetry in the network demand, the travelled distances increase significantly with the service usage due to the increase in deadhead miles. However, as the symmetry increases the impact on deadhead miles significantly reduces with increased service usage. These results are coherent with our findings, where the mobility services significantly increase the total travel distance, and sharing is a means of combating this trend without eliminating it.

In our method, we define the plant model in addition to the prediction model to assess the impact of traffic conditions on the performance of the dynamic ride-sharing system for large-scale problems and see how the dynamic ride-sharing system can impact traffic congestion. In Chapter 5 we present a complete framework for the ride-sharing simulation to fill the previous researches gaps.

## Conclusion

In this chapter, we reviewed the researches on dynamic ride-sharing systems. From this literature review, we can conclude that:

- There is a lack of studies on efficient fleet management methods that can provide

fast and qualified solutions in large-scale.

- There are very few studies to address the impact of traffic appropriately on ride-sharing system performance.

To figure out these points, in this thesis, we try to answer two main questions:

- How to assess the performance of ride-sharing on the network demand?
- How to assess the impact of the network on the ride-sharing system and vice versa?

In the following chapters, we propose simulation-based optimization frameworks for dynamic ride-sharing to answer these questions.

## Part II

How to assess the performance of  
ride-sharing on the network demand?



The dynamic ride-sharing problem involves two sub-problem. The first sub-problem is the fleet management problem and matching the rides to the passengers. This part formulates the ride-sharing problem into a mathematical model in Chapter 3. In this dissertation, we aim to approach the global optimal solution, whereas previous studies mostly focused on optimizing local sub-problems. The search for the global solution may be computationally expensive, but it permits answering the question of the maximum gain we can expect from ride-sharing in the transportation system. This is a crucial point that would worth being elaborated deeper. To approximate the global solution, in Chapter 4, we resort to an algorithm to solve the mathematical problem. Our solution approach is designed to be exact for small samples. Then, to be able to handle the large-scale problems, it is extended with several heuristics that keep the general design for the solution method but significantly reduce its computation time. The main strategy is to cluster the requests depending on a shareability index to create smaller samples that are faster to solve.

# Chapter 3

## Ride-Sharing Problem Formulation

### Introduction

In this chapter we present the components of the system that we have designed for dynamic ride-sharing. The system has two main components.

1) The fleet management component that solves the assignment problem based on the predicted travel times. The fleet management component solves a mathematical problem to assign the vehicles to the trip requests. The mathematical formulation of this problem is presented in this chapter. Then we explain how we solve this mathematical problem in the next two chapters.

2) The simulation component that represents the actual evaluation of the system. The performance of this component is explained in chapter 5.

In this chapter, section 3.2 represents the fleet management component. We express the ride-sharing problem into a mathematical problem. The purpose of the proposed mathematical formulation is to present a complete framework for the fleet management problem. We rigorously formulate the problem considering the essential objectives and constraints for the passengers and providers and we define the number of sharing constraints to limit the number of shared trips. This constraint can help us to analyze the transportation system under different sharing situations.

### 3.1 System components

The global functioning of the system is shown in figure 3.1. In this chapter we present the fleet management component. At the beginning of a new rolling horizon, the fleet management components solve the optimal assignment problem based on predicted travel times. Then, the simulation component implements this assignment over the next horizon to determine the actual evaluation of the system and the effective pickup/drop off and travel times for all the vehicles. The effective value of the objective function (objective function realized) can be computed at the end of each horizon and be compared to the optimal objective value (estimated objective function) derived from solving the assignment problem with predicted travel times.

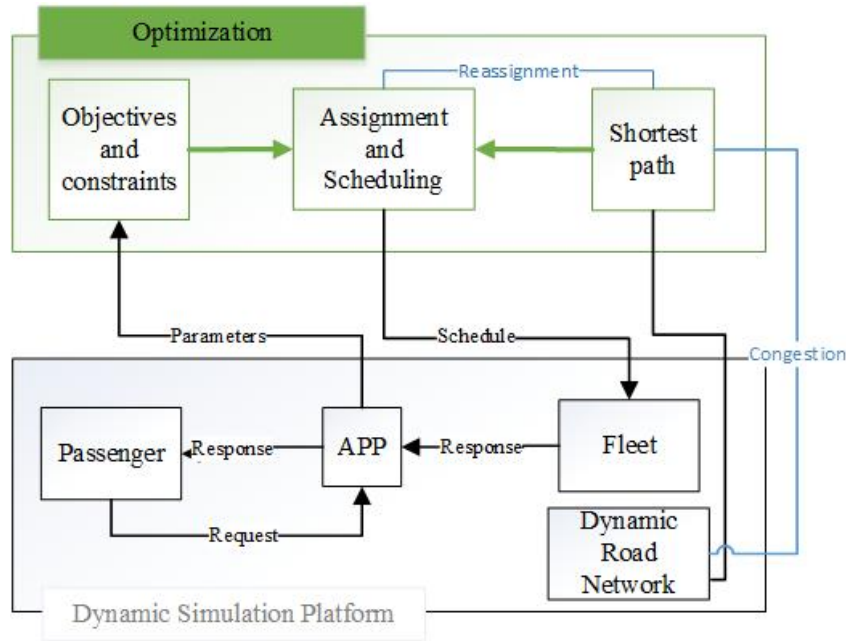


Figure 3.1: System components (fleet management component)

The main characteristics of the ride-sharing problem we investigate are:

1. Demand characteristics:

- Door-to-door dynamic ride-sharing (the passenger obtains a service at the exact defined origin and destination).
- Passengers determine origin and destination.

- Passengers define the earliest pickup time and the latest arrival time. The passenger must be picked up, transported and dropped off at the destination inside this time window (time schedule for each trip is shown in figure 3.2).
- Passengers determine the number of requested seats that can then be more than one.
- All requests over the next prediction horizon (usually 20 minutes) are considered known at the beginning of the horizon.
- Each passenger defines the maximum number of persons they are ready to share a trip with. We call it "number of sharing". The service has to guarantee that the number of sharing constraint is always satisfied for all the cars.
- A request has an origin, a destination, a time window, a number of seats and a number of sharing.

## 2. Service characteristics:

- There is a fleet of service vehicles to serve the requests dynamically.
- Service time is added to each trip to reflect the time to stop and get in and out of the car.
- The service is provided by a limited number of vehicles that are initially all in the central depot. Local depots are uniformly distributed over the network to represent locations where cars can wait for further assignments. When an idle waiting car is needed to serve a passenger, it comes from the nearest non-empty depot. Note that the central depot can always generate new cars if necessary. So there is no limit on the fleet size. When a car ends a trip without any further short-term assignment, it goes to the nearest depot and waits there.

## 3.2 Fleet management component

Fleet management component works to assign the vehicles to the trip requests. In this component, an algorithm computes the shortest path and then solves a constrained multi-

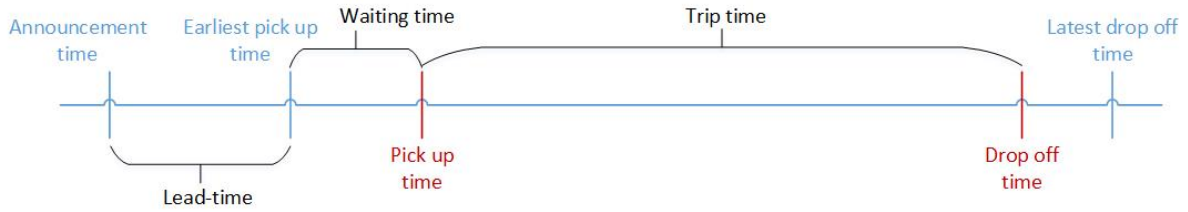


Figure 3.2: Trip time schedule

objective optimization problem to minimize the total travel time and distance for vehicles and the total travel time and total waiting time for passengers.

The constraint functions are capacity constraints, time windows constraints, assignment constraints, and the number of sharing constraints.

A trade-off between the objectives will be found to be sent as the schedule to the fleet.

Then the assigned cars leave their location or take a detour to serve the new demands.

The mathematical description of the fleet management component is provided in this section. First, we explain the objective functions, the constraints, and the inputs and outputs of the model, and finally, we present the mathematical model.

Note that the predicted travel times used in the fleet management correspond to the trip length divided by the mean speed over the full network at the end of the previous simulation period.

### 3.2.1 Notations and variables

At each period (rolling horizon width), the system receives  $n$  passenger requests. Every request has a pickup point  $i$  and a drop-off point  $i + n$ . Thus, there will be  $2n$  nodes ( $\{1, \dots, 2n\}$ ) plus nodes 0 (corresponding to the depot from where the vehicle comes) and  $2n + 1$  (corresponding to the depot where the vehicle goes after finishing all assignments). If vehicle  $k$  is assigned to passenger  $i$ , the decision variable  $y_i^k$  equals 1 and if not it equals 0. If vehicle  $k$  takes the passenger from point  $i$  to  $j$ , the decision variable  $x_{i,j}^k$  equals 1 and if not it equals 0. Table 3.1 gives the list of the notations that we use for the mathematical description of the problem.

The predicted direct travel time between every two points at each time is  $DTT_{ij}^t$ . Other variables in the model are computed based on  $DTT_{ij}^t$ .

Table 3.1: Notation

$m$	Fleet size
$n$	Number of passengers
$M$	Set of vehicles, $M = \{1, \dots, m\}$
$N$	Set of passengers, $N = \{1, \dots, n\}$
$P$	Set of pick up points, $P = \{1, \dots, n\}$
$D$	Set of drop off points, $D = \{n + 1, \dots, 2n\}$
$A$	Set of all pick up and drop off points, $A = P \cup D$
$O$	Set of all the stop points, $O = A \cup \{0, 2n + 1\}$
$TH$	Time horizon
$org_i$	Origin point of passenger $i \in N$
$des_i$	Destination point of passenger $i \in N$
$n_i^{share}$	Maximum number of sharers for passenger $i \in N$
$d_i$	Number of seats demanded for passenger $i$
$Cap$	Vehicle capacity
$c_i^k$	Capacity of vehicle $k \in M$ at point $i$
$ST_i$	Service time for passenger $i$
$EP_i$	Earliest pick up time for passenger $i$
$LP_i$	Latest pick up time for passenger $i$
$ED_i$	Earliest drop off time for passenger $i$
$LD_i$	Latest drop off time for passenger $i$
$P_i^k$	The time when the vehicle $k$ arrives at point $i$
$TT_i$	Total travel time for passenger $i$
$WT_i$	Waiting time for passenger $i$
$WT'_i$	Waiting time for passenger $i$ when served individually
$T_k$	Travel time for vehicle $k$
$TD_k$	Travel distance for vehicle $k$
$N_{trips}$	Number of trips
$\alpha$	Weight of waiting time
$\beta$	Weight of passenger total travel time
$\gamma$	Weight of vehicle total travel time
$\delta$	Weight of vehicle total travel distance
$DTT_{i,j}^t$	Direct travel time from point $i$ to $j$ at time $t$ , $(i, j \in A, t \in TH)$
$DTD_{i,j}$	Direct travel distance from point $i$ to $j$ , $(i, j \in A)$
$x_{i,j}^k$	Decision variable equal to 1 if vehicle $k$ takes the passenger from point $i$ to $j$ and 0, otherwise
$y_i^k$	Decision variable equal to 1 if vehicle $k$ is assigned to passenger $i$ and 0, otherwise

### 3.2.2 Problem formulation

We express the fleet management problem as the following integer linear program:

$$\min \sum_{i \in P} (\alpha.WT_i + \beta.TT_i) + \sum_{k \in M} (\gamma.T_k + \delta.TD_k) \quad (3.1)$$

subject to:

$$d_i \leq \sum_{k \in M} c_i^k \cdot y_i^k, \forall i \in P \quad (3.2)$$

$$\sum_{j \in O} x_{i+n,j}^k \cdot y_i^k (c_{i+n}^k + d_i - c_j^k) = 0, \forall i \in P, \forall k \in M \quad (3.3)$$

$$\sum_{j \in A} x_{i,j}^k \cdot y_i^k (c_i^k - d_i - c_j^k) = 0, \forall i \in P, \forall k \in M \quad (3.4)$$

$$EP_i - P_i^k - ST_i \leq 0, \forall i \in P, \forall k \in M \quad (3.5)$$

$$P_i^k + ST_i - LD_i + DTT_{i,i+n}^t \leq 0, \forall i \in P, \forall k \in M \quad (3.6)$$

$$EP_i + DTT_{i,i+n}^t - D_{i+n}^k - ST_i \leq 0, \forall i \in P, \forall k \in M \quad (3.7)$$

$$D_{i+n}^k + ST_i - LD_i \leq 0, \forall i \in P, \forall k \in M \quad (3.8)$$

$$x_{i,j}^k (P_i^k + DTT_{i,j}^t + ST_i - P_j^k) \leq 0, \forall i, j \in A \quad (3.9)$$

$$\sum_{k=1}^m y_i^k = 1, \forall i \in P \quad (3.10)$$

$$\sum_{j \in A} x_{i,j}^k - \sum_{j \in A} x_{j,i+n}^k = 0, \forall i \in P, \forall k \in M \quad (3.11)$$

$$\sum_{i \in P \cup \{0\}} x_{i,j}^k - \sum_{i \in D \cup \{2n+1\}} x_{j,i}^k = 0, \forall j \in O, \forall k \in M \quad (3.12)$$

$$c_j^k - M2 \cdot (1 - x_{0,j}^k) - Cap \leq 0, \forall j \in P, \forall k \in M \quad (3.13)$$

$$c_j^k + M2 \cdot (1 - x_{0,j}^k) - Cap \geq 0, \forall j \in P, \forall k \in M \quad (3.14)$$

$$\sum_{j \in P} y_j^k \cdot d_j - \sum_{j \in D} y_j^k \cdot d_j - n_i^{share} \leq M1 \cdot (1 - y_i^k), \forall k \in M, \forall i \in N \quad (3.15)$$

$$m \leq M3 \quad (3.16)$$

$$x_{i,j}^k \in \{0, 1\}, \forall i, j \in O \quad (3.17)$$

$$y_i^k \in \{0, 1\}, \forall i \in N, k \in M \quad (3.18)$$

The objective function and constraints of the mathematical model are described in detail in the next sections.

### 3.2.3 Mathematical model objective function

In this research, we aim to optimize both providers' and passengers' objective functions. According to the state-of-the-art, the most important operation objective for the service provider is to minimize the total travel time and the total travel distance of vehicles. Also, the passengers need to get to the destination on time and have the minimum waiting time. So we model the problem to minimize the passenger waiting time and travel time and vehicles' total travel time and distance.

The total travel time for each vehicle is the summation of the direct travel time for all the trips that are served over the time horizon by this vehicle.

$$T_k = \sum_{i \in O} \sum_{j \in O} DTT_{i,j}^t . x_{i,j}^k \quad \forall k \in M, t \in TH \quad (3.19)$$

Also, the total travel distance for a vehicle is the summation of the direct travel distance for all the trips that are served over the time horizon by this vehicle:

$$TD_k = \sum_{i \in O} \sum_{j \in O} DTD_{i,j} . x_{i,j}^k \quad \forall k \in M \quad (3.20)$$

Waiting time is the time that the passenger must wait before being picked up. When the passenger defines the earliest pickup time, the vehicle cannot serve them before this time or after the latest pick up time. If the car arrives at point  $i$  before  $EP_i$ , it must wait to pick up the passenger at their desired time. So, the waiting time, in this case, is zero for the passenger. But if the car arrives after  $EP_i$ , the waiting time is the difference between the pickup time and the lower bound of the pickup time window:

$$WT_i = P_i^k - EP_i \quad \forall i \in P \quad (3.21)$$



The exact passenger pickup time is the time that the vehicle arrives at the passenger's location. This time can be computed as follows:

$$P_i^k = \sum_{g=0}^{i-1} \sum_{j=n+1}^{i+n-1} \sum_{k=1}^m DTT_{i,j}^t \cdot x_{gj}^k \cdot y_i^k + DTT_{i+n-1,i}^t + ST_i \quad \forall i \in P, t \in TH \quad (3.22)$$

The total travel time for the passenger is defined as the sum of the service time and the predicted travel time.

$$TT_i = P_{i+n}^k - P_i^k + ST \quad \forall i \in P \quad (3.23)$$

We aggregate these four objectives into a weighted-sum objective with different measures [96].  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\delta$  are the weights for the passenger waiting time and travel time and the vehicle travel time and travel distance respectively. The objective weights are determined according to a prior definition of their relative importance.

Finally, the objective function of the mathematical problem would be as in Equation 3.1.

### 3.2.4 Mathematical model constraints

In the formulation of this problem, the main constraints are capacity constraints, time constraints and assignment constraints. Also we consider limitations on the number of sharing and we make sure that there is a sufficient number of vehicles in the fleet.

**Capacity constraints** The capacity of vehicle  $k$  when it arrives at point  $i$  is the summation of the number of passengers that are picked up at point  $i$  by this car minus the number of passengers that are dropped off from the vehicle at this point.

$$c_i^k = Cap - \sum_{g=1}^{i-1} \sum_{j=1}^{i+n-1} \sum_{k=1}^m d_g \cdot x_{g,j}^k \cdot y_i^k + \sum_{g=1}^i \sum_{j=1}^{i+n-1} \sum_{k=1}^m d_g \cdot x_{g,j}^k \cdot y_i^k \quad \forall i \in P \quad (3.24)$$

We define constraint 3.2 to 3.4 on capacity. The first ensures that at each pickup point, the demand does not exceed the vehicle's capacity at that point. The second is to be assured that all the passengers who are picked up at the origin, will be dropped off at the

corresponding destination. The third constraint is to ensure that the passengers who are picked up at point  $i$  stay in the vehicle up to their destination.

**Time window constraints** The time window for passenger pickup and drop off can be derived from the departure time (earliest pickup time) and arrival time (latest arrival time) defined by the passenger. The difference between the earliest pickup time and the earliest drop off time is the minimum time needed to go from the passenger's origin to their destination (direct free flow travel time):

- pickup time window:  $(EP_i, LD_i - DTT_{i,i+n}^t)$
- drop off time window:  $(EP_i + DTT_{i,i+n}^t, LD_i)$

Constraint 3.5 to 3.9 are on time. The time windows for the pickup and drop off are ensured with constraint 3.5 to 3.8. The drop off point must be visited after the pickup point and sufficient time must be guaranteed for service time and the travel time between the origin and destination. This constraint is imposed by inequality 3.9.

**Assignment constraints** Constraint 3.10 to 3.12 are related to the assignment. Constraint 3.10 ensures that just one vehicle is assigned to passenger  $i$ . Constraint 3.11 guarantees that the same vehicle is handling a passenger pickup and drop off. Constraint 3.12 is the flow constraint, to be sure that the vehicle that enters a service node will also exit from it.

Constraint 3.13 and 3.14 work together to guarantee that when a vehicle exits the depot, it has no passenger on board.

**Number of sharing constraint** We define the number of sharing constraints to limit the number of shared trips. For example, when the number of sharing is 0, the system performance is like a taxi service that serves only one passenger on each trip. When we increase the number of sharing to 1, we allow the system to serve two passengers simultaneously. With the number of sharing 2, the system is allowed to share the passenger's trip with two other passengers. This parameter can affect the traffic by changing the number of required trips to serve the network demand.

Constraint 3.15 ensures that the number of passengers in a car is lower than or equal to the number of sharing that the passenger has defined

**Fleet size constraint** There should be enough vehicles in the network to serve all the requests. Constraint 3.16 ensures that there is a sufficient number of vehicles in the fleet.

**Decision variables** The possible values of the variables  $x_{i,j}^k$  and  $y_i^k$  are given by 3.17 and 3.18.

### 3.2.5 Model complexity

If  $n$  is the number of ride requests and  $m$  is the number of vehicles, the number of decision variables in the model is as follows:

- Binary variables  $x_{i,j}^k : m \cdot (2n + 1) \cdot (2n + 1) = 4m \cdot n^2 + 4 \cdot m \cdot n + m$
- Binary variables  $y_i^k : m \cdot n$

So the total number of decision variables ( $NDV$ ) is:

$$NDV = 4mn^2 + 5mn + m$$

It can be seen that even very small instances of this problem give a number of variables and constraints to compute that encompasses the acceptable dimension of a linear programming problem. For example, with four requests when there are 4 service vehicles in the network, the number of decision variables for the problem is 350. This number increases to 1804 for ten requests and 4 vehicles. With a small increase in the number of vehicles and passengers, we will have a large increase in the number of variables. When we have 1000 requests, the number of decision variables increases to more than 4 million. Table 3.2, shows the computation time to solve small instances of the problem with CPLEX solver. For each instance the value of  $M3$  is set equal to the number of requests to be sure that there are sufficient number of vehicles to serve all the requests. For 4 requests, the computation time is 0.4 seconds but it exponentially increases by increasing the number of requests and for 7 requests the computation time is almost 5 minutes.

Table 3.2: Solution methods comparison

Number of requests	Objective function	Computation time (s) (normalized value)
4	0.54	0.40
5	0.65	5.30
6	0.74	43.50
7	1.14	287.60
8	1.24	2945.90

## Conclusion

In this chapter we presented the mathematical formulation of the proposed ride-sharing system. The purpose of the proposed formulation is to present a complete framework for the fleet management problem. We rigorously formulate the problem considering the essential objectives and constraints for the passengers and providers and we define the number of sharing constraints to limit the number of shared trips.

The complexity of the optimization problem grows exponentially by small increases in the number of requests and vehicles. Also, softening the constraints, for example, increasing the number of sharing will increase complexity.

One of the most important problems in the solution approaches for shared mobility systems is computation time and quality of the results [130].

In the next chapters, we present a solution method with multiple steps that starts from finding the exact solution for small instances. Furthermore, we introduce extensions that speed up the solution method and can address bigger networks, even large-scale networks, while assessing the difference in quality at each step. We show that the proposed heuristics can keep the quality of solutions at an acceptable level (near-optimal solution) while significantly decreasing the computation time. Thus, we design our solution method based on the classical branch and bound algorithm [149] but with specific properties to cope with a fleet management problem.

# Chapter 4

## Solution Method for the Dynamic Ride-Sharing Problem

### Introduction

In this thesis, we aim to approach the global optimal solution, whereas previous studies mostly focused on optimizing local sub-problems. The search for the global solution may be computationally expensive, but it permits answering the question of the maximum gain we can expect from ride-sharing in the transportation system.

To approximate the global solution, we resort to an algorithm to solve an integer linear program. However, as we are targeting problems with large instances, we still face the curse of dimensionality. Our solution approach is designed to be exact for small samples. It is then extended with several heuristics that keep the general design for the solution method but significantly reduce its computation time.

The main strategy is to cluster the requests depending on a shareability index to create smaller samples that are faster to solve. This method narrows the exploration of the space to feasible and promising states only. As the number of assigned passengers increases for a car, the intersection of feasible areas becomes smaller, and the algorithm can compute the assignment of running cars. In this chapter, section 4.1 presents the optimal algorithm to solve the assignment problem, in section 4.2, we present our test cases, and we present

the heuristic method step by step in sections 4.3, 4.4 and 4.5.

## 4.1 Branch-and-Cut based algorithm

In chapter 3, we formulated the ride-sharing problem into an Integer Linear Programming (ILP) model, and we showed that the problem is NP-hard. Due to the NP-hardness of the problem, the optimal solution can be computed only for small instances of the problem. The strategies of branch-and-bound, branch-and-cut, branch-and-price and branch-and-cut-and-price are among the most used in solving ILP [31].

The branch-and-bound methods are based on partitioning a complete problem into sub-problems and reducing the solution space by divide-and-conquer strategies [60]. Then the branch-and-cut methods solve the problem using a combination of the branch-and-bound algorithm and the cutting plane method [127]. The idea of branch-and-cut methods is to solve a sequence of linear programming relaxations of the integer programming problem. The cutting plane methods are used to improve the relaxation of the problem, and then branch-and-bound methods are used with divide-and-conquer strategies to finally solve the problem [45]. In this section we present a branch-and-cut based algorithm to exactly solve the mathematical problem in previous chapter.

The algorithm builds a tree of routes and tries to add the feasible points to the best branch of the tree at each step. It checks the feasibility of the points regarding the model constraints. In the beginning, it starts from the closest non-empty depot to the origins and adds the origin points to the branches of the tree (figure 4.1(a)). The algorithm can add a destination point if and only if its related origin point has been added to the route before (figure 4.1(b)). Also, it can add a new origin point if the capacity constraints and the number of sharing constraints are satisfied (figure 4.1(c)). The time window constraints must be checked when adding new stop points to the routes (figure 4.1(d)). When the algorithm finds a feasible point for a route, it creates a new route by adding this possible point and puts the newly created route in the set of paths.

Finally, the best route is the route that has the minimum objective function.

Algorithm 1 shows the optimization algorithm. Each part of the algorithm corresponds

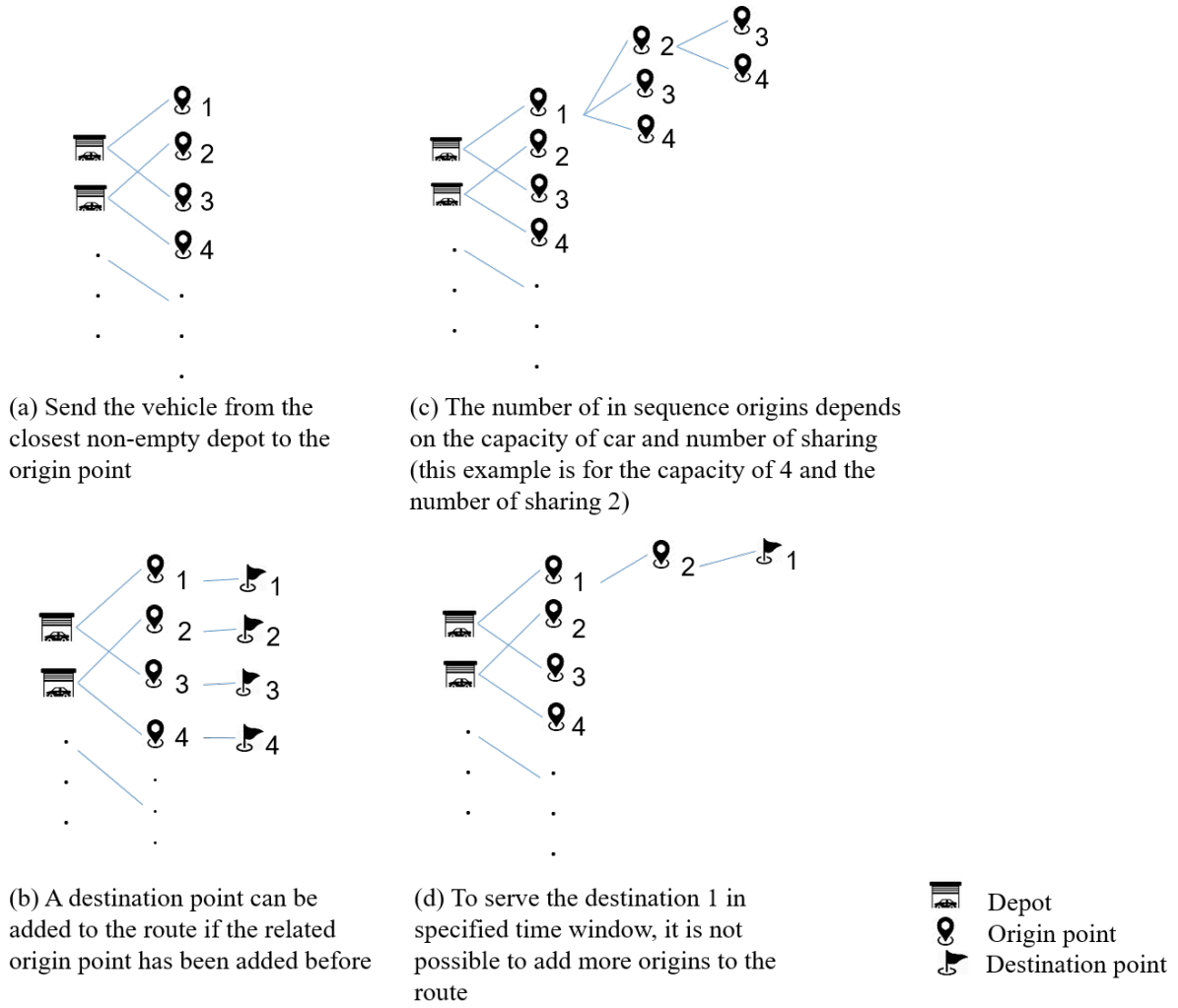


Figure 4.1: The assignment algorithm function

to one or multiple equations of our mathematical model (the number of the corresponding equations is shown at each level of the algorithm).

The initial route set  $S$  contains the origin points of the requests that are not already being assigned to a car. The algorithm builds the primary branches from the depots nearest the origins. Then, it finds the best branch with respect to the objective function (equation 3.1) among these primary branches. The next branches are created as extensions of this primary branch. To add points to the primary branch, equation 3.11 of the model must be satisfied. Then the algorithm finds a set of points that can be added to the first branch. The feasibility of adding points to the branch is checked by other equations in the model. Then, the algorithm creates new branches by adding the feasible points. At each iteration, the optimal branch in terms of the objective function is selected to be the

base branch.

---

**Algorithm 1:** Assign requests to the depot vehicles
 

---

**input:** New requests: direct travel times ( $DTT_{i,j}^t$ ), direct travel distances ( $DTD_{i,j}$ ) set of vehicles ( $M$ ), set of points ( $P, D, A$ ), time windows ( $EP_i, LD_i$ ), number of seats demanded ( $d_i$ ), number of sharing ( $n_i^{share}$ ), maximum detours ( $SQ$ ), vehicle capacity ( $Cap$ ), weights of objective function ( $\alpha, \beta, \gamma, \delta$ )

**output:** Vehicle schedules

**while** *Not all the points in  $A$  are assigned* **do**

- Create the new car  $m \in M$ ;
- Create initial routes set  $S$  from remaining origins in origin set  $P$ ;
- while**  $S$  is not empty **do**
  - Find the optimized route  $s \in S$  (in terms of objective function);
  - Find the set of points  $SP$  that can be added to  $s$ ;
  - for**  $sp \in SP$  **do**
    - if**  $sp$  is feasible for time constraints on  $s$  **then**
      - Compute new vehicle capacity ;
      - if**  $sp$  is feasible for capacity, number of sharing, detour constraints on  $s$  **then**
        - Create new route  $ns$  by adding the point  $sp$  to the route  $s$ ;
        - Add route  $ns$  to the routes set  $S$ ;
  - if** All  $sp \in SP$  are non-feasible in route  $s$  **then**
    - if** number of route origins = number of route destinations **then**
      - Put route  $s$  in the results set  $Result$ ;
    - else**
      - Remove route  $s$  from routes set  $S$ ;
- Find the optimized route  $optimal-route \in Result$ ;
- Assign the  $optimal-route$  to the car  $m$ ;
- Remove pickup points on  $optimal-route$  from  $P$ ;
- Remove  $m$  from  $M$ ;
- Add  $m$  to en-route vehicles set  $eM$ ;

---

At the end, when it is no longer possible to add more points to a branch, if the points on the branch satisfy equations 3.10 and 3.12 of the model, the branch can be added to the results set. Finally, the optimal branch is selected from the set of results.

When the schedule is received, the algorithm puts the associated vehicle in the en-route vehicles set, and as the car finishes the assigned schedule, it goes back to the nearest depot. At each iteration of the algorithm, a large number of branches are added to the route set. The critical point that makes our method efficient is that we remove the branches that are not feasible with respect to three kinds of constraint (on time, capacity and number



of sharing).

This algorithm is exact and its complexity explodes with the number of branches. This is, for example, the case when we increase the number of sharing or requests. In this case, we need to introduce heuristics that reduce the exploration of the feasible solution.

#### 4.1.1 Examples of the exact method

In this section, we present an examples with four requests to show the performance of the algorithm. Table 4.1 shows the requests. Each request has a demanded seat number, a number of sharing, and the earliest pick-up time and latest arrival time.

Table 4.1: Example with 4 requests (configuration)

Request	Travel disance	demand	nshare	EPT	LDT
1	19	1	3	8:00	8:45
2	11	2	2	8:00	8:25
3	24	2	1	8:15	9:00
4	18	2	3	8:30	9:20

The algorithm starts to create branches of routes to serve the requests. It sends a car from the closest stop location to pick up the passengers at the origin point, and then it continues by adding the feasible points to the branches. First, the algorithm finds feasible branches. Figure 4.2 shows the final feasible solutions for the problem. The algorithm can find four different solutions for the problem.

Table 4.2: Example with 4 requests (solutions)

Solution	Total waiting time (s)	Total travel distance (m)	Number of cars
A	28	3030	4
B	51	2085	2
C	53	1785	2
D	29	2265	3

Table 4.2 shows the total passengers' waiting time, total vehicles' travel distance, and the number of vehicles for these solutions. Solution *A* is when a car serves each passenger separately without sharing. In this solution, the waiting time is minimum, and the passengers wait for 28 seconds to be picked up. Solution *B* serves passengers 1 and 4 with

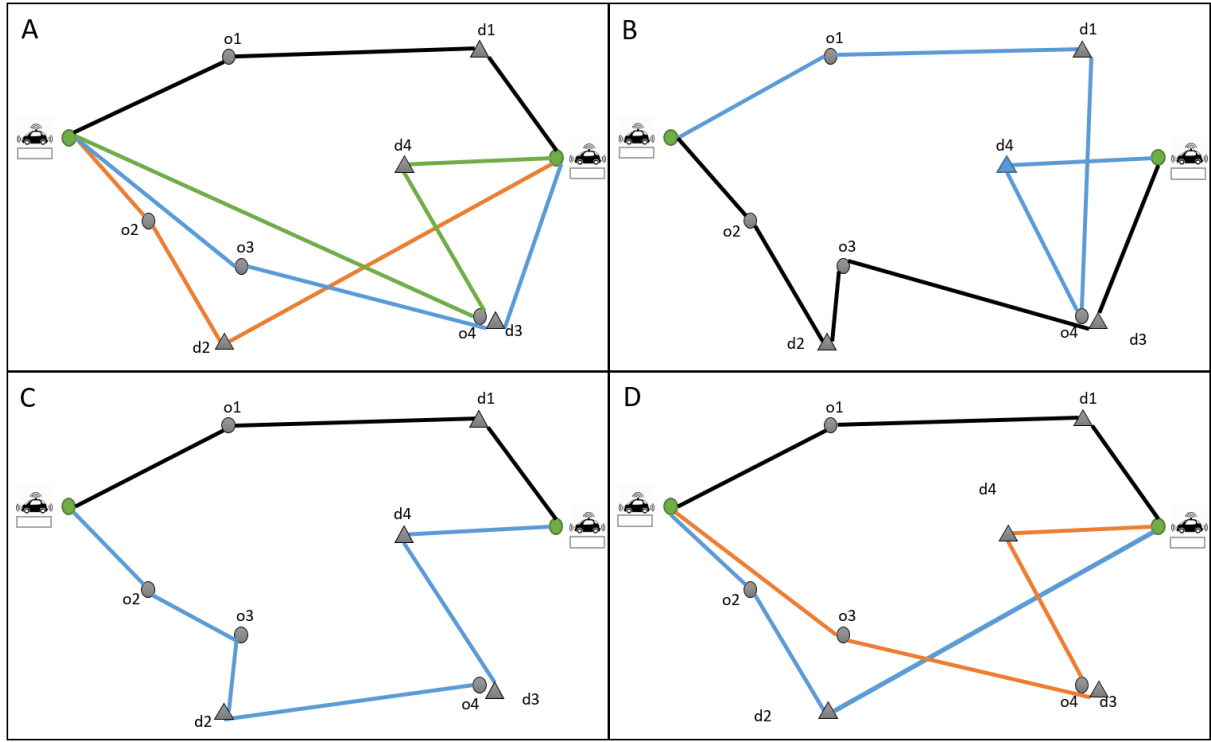


Figure 4.2: Feasible routes for the example

one car and passengers 2 and 3 with another car. It should be mentioned that in this solution, the trips are not shared. The first car serves passenger 1, and after dropping off this passenger, it goes to pick up passenger 4 at his destination. This solution increases the waiting time to 51 seconds, but it can reduce the travel distance from 3,030 meters to 2,085. Solution *C* serves the four requests with two cars like solution *B*. However, it shares the trip for passengers 2 and 3, and it can make more significant progress in reducing the travel distance to 1,785 meters. The waiting time with this solution is 53 seconds. Solution *D* increases the waiting time just 1 second compared to solution *A*. It serves requests 3 and 4 with a car in sequence (without sharing) and requests 1 and 2 with two cars separately. If the weights for all the objective functions are equal, after normalizing the objective functions, the optimal solution will be solution *C*.

## 4.2 Test cases

The main goal of this dissertation is to assess the impact of dynamic ride-sharing on network traffic. In mega-cities and large-scale networks, the performance of ride-sharing in

terms of reducing congestion can be different from its impact on small-scale and medium-scale networks. To the best of our knowledge, the stream of research on ride-sharing services focuses on situations in large cities like New York, Beijing, and San Francisco. However, these services are increasingly popular in large cities and small and medium-sized cities.

In this thesis, the goal is to assess dynamic ride-sharing systems' performance in reducing congestion in both medium and large-scale cities. So, we implement the method on two networks. First, to assess the service in small and medium scales, we apply our method to a realistic O-D trip matrix for Lyon's northern half in France. Then, to assess the impact of ride-sharing on large-scale networks, we apply the method to the whole Lyon city network in France.

#### 4.2.1 Lyon 6 + villeurbanne

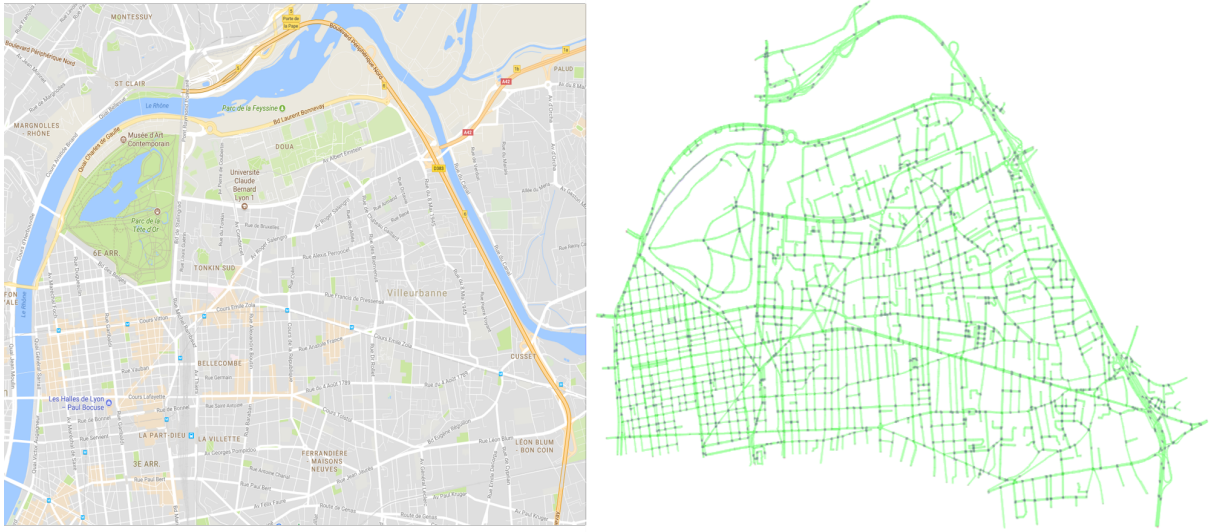


Figure 4.3: Lyon 6e + Villeurbanne: Mapping data ©Google 2018 and the traffic network

This network has a scale of  $25 \text{ km}^2$  and has 1,883 nodes and 3,383 links. The area is shown in figure 4.3. The origins set contains 94 points, the destinations set includes 227 points and the local depots (stop locations) set contains 237 points on the network.

The network is loaded with travelers of all ODs with a given departure time to represent the morning peak hour (4 hours from 6:30 AM to 10:30 AM), based on the study of [102]. The number of trips during this period is 62,450. Some trips start from or end outside

the network. Only trips wholly inside the network (11,235 trips) can be assigned to the service depending on the market-share. Market-share is the percentage of the trips that will be served with the service vehicles. This corresponds to 11,235 trips and defines the maximal dimension of our optimization problem when the market-share is equal to 100%. Figure 4.4 shows the temporal pattern of the demand for the private and service vehicles. We have 51,215 personal trips in the network. Besides that we have 11,235 demand for the service cars in the system. Based on the market-share, we select uniformly a part of this demand to be served with the service vehicles. Then we serve the rest of the trips with personal cars.

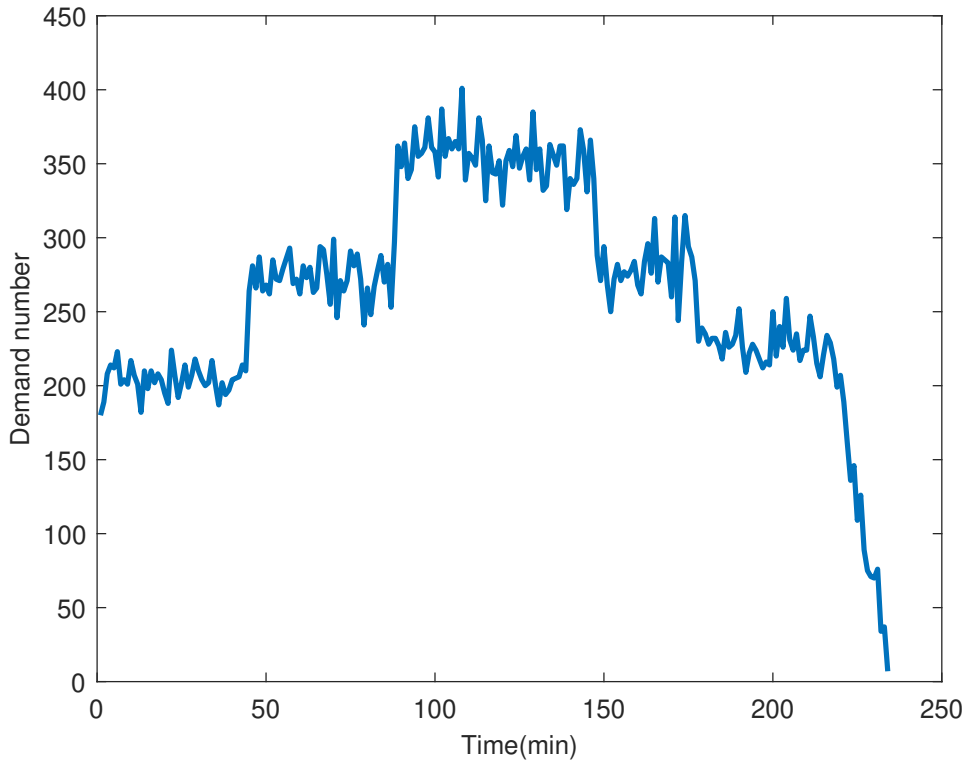


Figure 4.4: Temporal demand pattern in Lyon6 + villeurbanne (Medium-scale)

### 4.2.2 Lyon

Lyon is the second-largest urban area of France with an area of  $80 \text{ km}^2$ . The city is located in the south of Paris and is close to other mega-cities in France and Switzerland (Marseille, Grenoble, and Geneva). Figure 4.5 shows the network of the city and its geographical position.



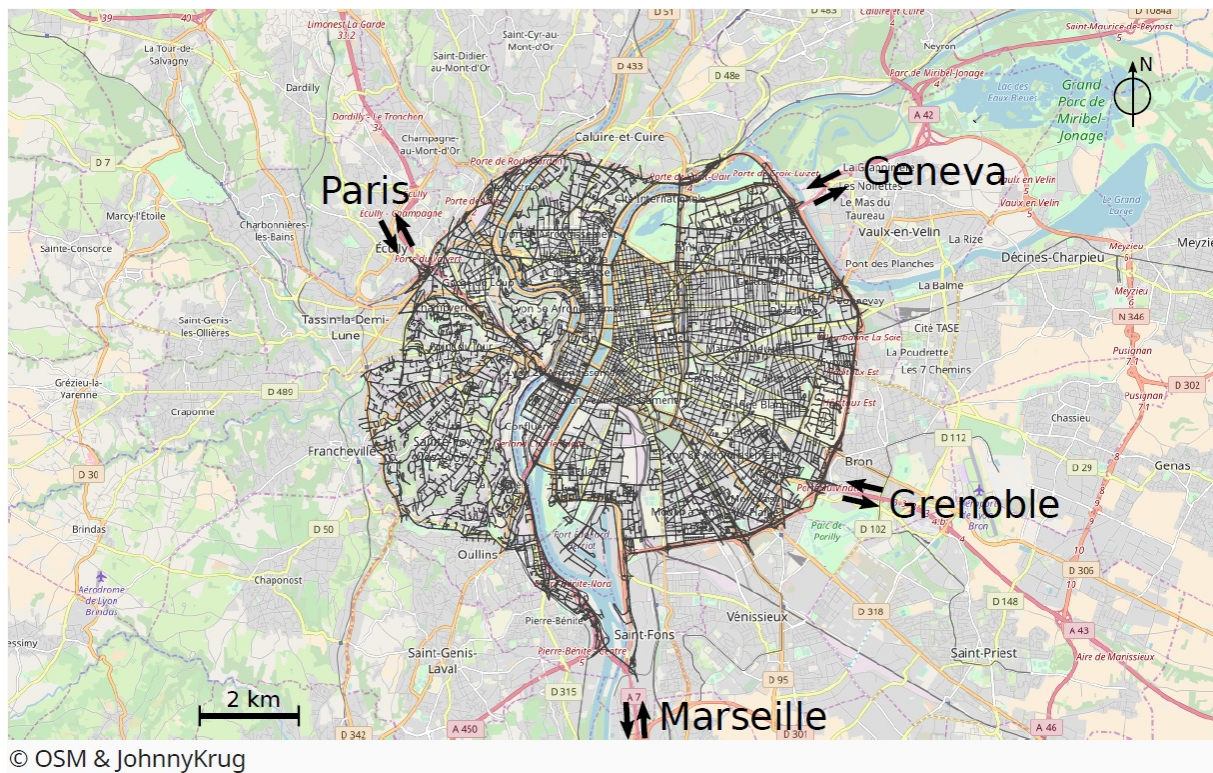


Figure 4.5: Lyon city in France



Figure 4.6: Network of Lyon

The origins/destination set contains 11,314 points and the local depots (stop locations) set contains 2,272 points including 9 central depots on the network. Figure 4.6 shows the links and nodes on the network. The network is loaded with travelers of all ODs with a given departure time to represent the morning peak hour from 6 AM to 10 AM. The number of trips during this period is 484,690. We have 279,382 personal trips in the network and 205,308 demand for the service cars in the system.

Figure 4.7 shows the temporal pattern of the demand in Lyon network.

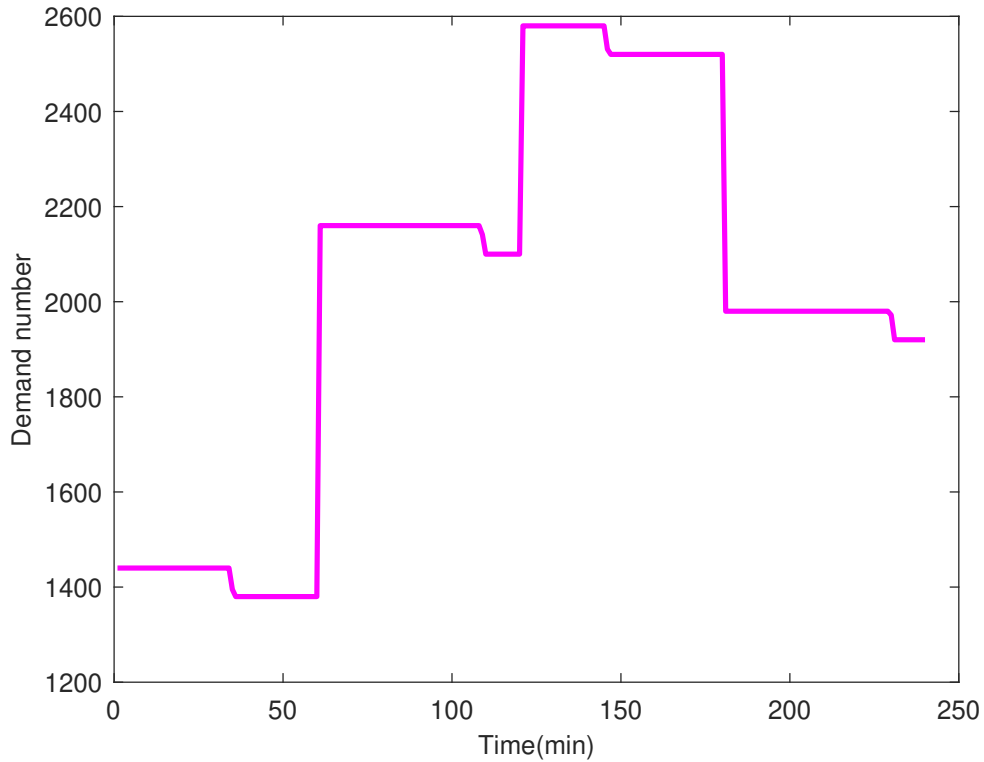


Figure 4.7: Temporal demand pattern in Lyon (large-scale)

### 4.3 Rolling horizon method and re-scheduling

Considering all the requests over the full-time horizon can provide the global optimum solution. However, this greatly increases the number of variables and is not reasonable in practice. To reduce the number of variables, but also to bring the expression of the problem more in line with common practice, we now implement a rolling horizon, generally about 20 minutes. The requests are assumed known only over the next rolling horizon.

**Algorithm 2:** Assign requests to the en-route vehicles

---

**input:** New requests  
**output:** Vehicles re-schedules

**for**  $origin\ p \in P$  **do**

**for**  $c\text{-schedule}, \text{ the schedule of car } m \in eM$  **do**

**if** *Detour is possible from any of the remaining origins on  $c\text{-schedule}$*  **then**

Build the *re-schedule* by adding the  $p$  after *origin*;

**if**  $p$  is feasible for time window, capacity and number of sharing constraints on  $c\text{-schedule}$  **then**

**if**  $d$  the destination of  $p$  is feasible for time window on  $c\text{-schedule}$  **then**

Create new schedule  $n\text{-schedule}$  by adding  $p$  and  $d$  to  $c\text{-schedule}$ ;

Put  $n\text{-schedule}$  to the *Result* set;

Find the optimized route  $optimal\text{-schedule} \in Result$ ;

Re-assign the  $optimal\text{-schedule}$  to the car  $m$ ;

Remove  $p$  from  $P$ ;

---

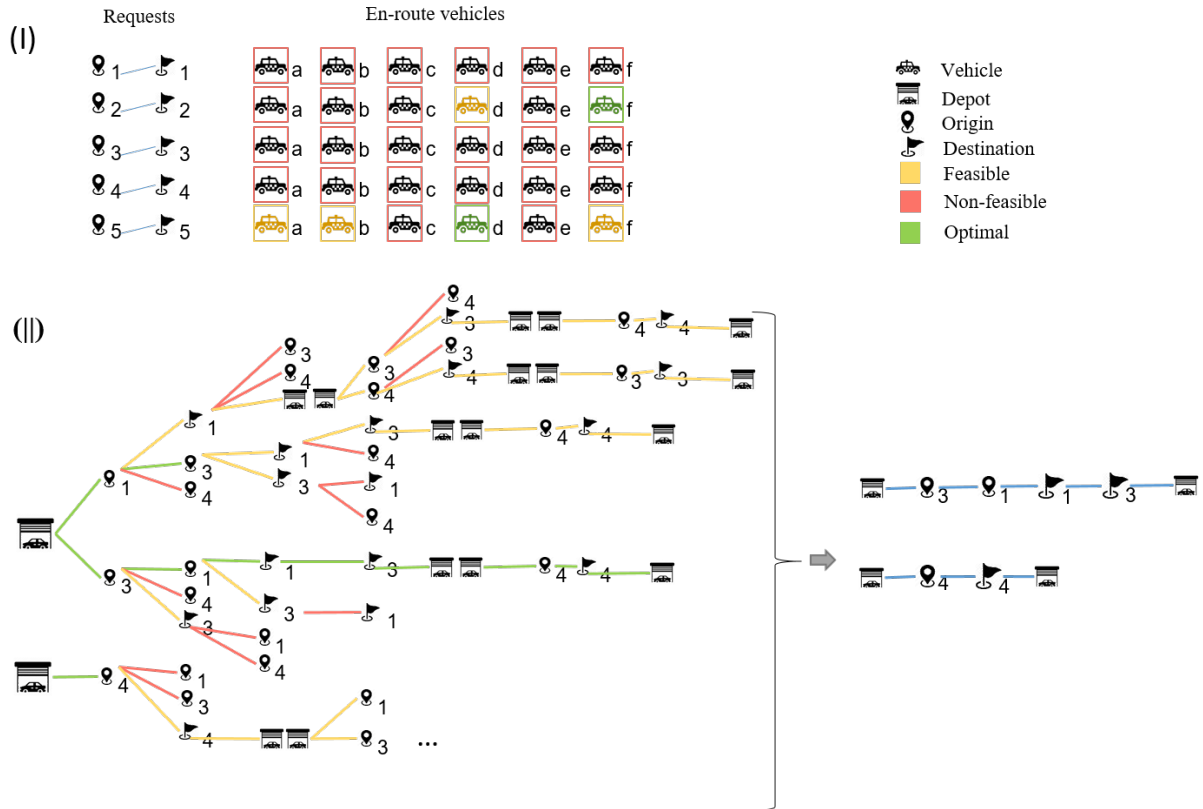


Figure 4.8: Dynamic assignment

The corollary is that we have to introduce a new process to handle travelling cars that have not yet reached their maximal occupancy because of the car or the passenger constraints. We therefore introduce a specific algorithm to assign the new requests in priority to en-

route vehicles. The remaining requests are handled by the first algorithm presented in the previous section.

In dynamic ride-sharing setting, new trip request announcements arrive continuously at arbitrary times. Therefore, the optimizer must adapt vehicle schedules multiple times during the day. This implies that future requests are unknown during each assignment time step. To handle the new requests and optimize the problem dynamically, a rolling-horizon approach is implemented.

In this study, the assignment procedure rolls over a specific horizon for the announced requests of a particular optimization step. Every  $\frac{TH}{2}$  minute, we optimize the requests of the next  $TH$  minutes. It is clear that some of the requests are re-optimized every  $\frac{TH}{2}$ .

If a trip has been assigned to a vehicle which has left the depot, the algorithm does not assign it again, but if a trip is in the schedule of a waiting vehicle, the algorithm puts it in the set of optimizing trips in the particular horizon and re-optimizes it.

In dynamic assignment, when a car drops off a passenger, it detects the potential to be assigned to a new request. We add a second part to the algorithm to re-assign the potential cars. In this part, receiving a new request at each optimization time step, the algorithm works to put the new request in the en-route vehicle schedule if possible.

The second part of the algorithm to assign requests to the en-route vehicles is shown in algorithm 2. First, the algorithm checks the possibility of adding the origin point of the request to the vehicle schedule. It must check the capacity of the car, the number of sharing for all the on-board passengers after adding the new origin as well as the time window for all the stop points. If the vehicle route remains feasible after adding the new origin, the algorithm checks the possibility of adding the related destination point. In this step, it must check the time window for all the points after adding the new origin and destination points. Then, the algorithm puts all the feasible vehicles for the new request in the *Result* set. Finally, it chooses the vehicle that has a minimum increase in the objective function after adding the new request and sends the re-scheduled route to the car.

Figure 4.8 shows an example of the dynamic ride-sharing algorithm when the number of



sharing is one. In the first part, when the optimizer receives the requests, algorithm 2 searches to put new trips in the en-route vehicle schedule and then in the second part, algorithm 1 starts to work and assign the optimal routes to the vehicles waiting in depots. For example, in the figure 4.8, the optimizer has received five new requests. For request number 2, the re-scheduling algorithm finds vehicles  $d$  and  $f$  which can serve this request, satisfying all the constraints and, finally, vehicle  $f$  has a smaller increase in objective function after adding request 2 to its schedule. Likewise, for request 5 and vehicle  $d$ . Then for the remaining not-assigned trips, the main algorithm finds the best routes and assigns them to the vehicles in the nearest depots. The optimal solution serves requests 3 and 1 in the same car and request number 4 individually.

## 4.4 Clustering method

The proposed algorithm creates branches of origin and destination points as mentioned before. The algorithm must try all the different arrangements of stop points to find the optimal assignment. Thus, the computation time increases exponentially as the number of requests increases.

Restricting the exploration of the feasible area to the branches that are more likely to create the optimal assignment can narrow the search of feasible solutions. To overcome this limitation, we define a clustering method to make clusters of the requests which are more likely to be shared. Then the algorithm is executed within each cluster independently.

### 4.4.1 Shareability function

To perform the clustering on the requests received by the system over each rolling horizon, we define the "Shareability Index" ( $SF_{i,j}$ ) between request  $i$  and request  $j$  ( $\forall i, j \in N$ ). We compute  $SF_{i,j}$  for each pair of trips, and the function value is the difference between the travel time when the two trips are shared and the travel time to serve each trip individually.

Three situations exist for every two trips (figure 4.9).

In figure 4.9(a), two trips can be shared, and the first passenger drop off is before that of the second passenger. So, the travel time for the first passenger is the summation of their waiting time, the travel time between the first origin and the second origin and the travel time between the second origin and the first destination. Also, the travel time for the second passenger is the summation of their waiting time, the travel time between the second origin and the first destination and the travel time between the first destination and the second destination.

In situation (b), the vehicle serves the second passenger while the first passenger is on board. Thus, the travel time for the second passenger is the same as when served individually and the travel time for the first passenger is the travel time of all the links from the first stop point to the last one.

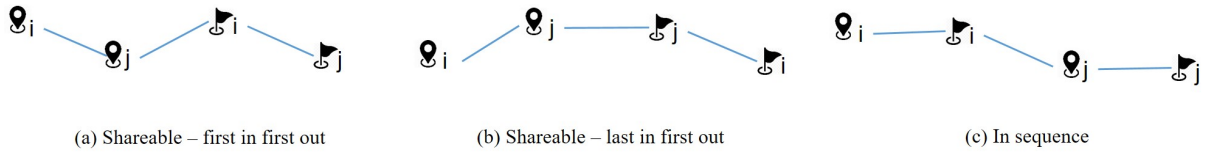


Figure 4.9: Trip situations

There is a third situation when the trips are not shared, but the vehicle can serve two passengers sequentially. This situation must be considered in the shareability index in order to put these trips in the same group while solving the optimization problem. The travel time for the passengers is the same as when served individually. But in this situation, the vehicle travel time can decrease if the travel time between the first destination and the second origin is less than the summation of the travel time between the first origin and the closest depot and the travel time between the start depot and the second origin. This means that the  $SF$  here is the difference between the passenger waiting times when the trips are in sequence and the passenger waiting times when the trips are individual.

The following equations show how we compute  $SF_{i,j}$  for each pair of trips, considering

the three situations ( $i, j$  show the origin and the destination of the trip ( $r_n$ )):

$$\begin{aligned}
 (a) : TT_i + TT_j &= WT_i + DTT_{org_i, org_j} + WT_j + DTT_{org_j, des_i} + DTT_{des_i, des_j} \quad \forall i, j \in N \\
 SF_{i,j}^a &= TT_i + TT_j - (DTT_{org_i, des_i} + DTT_{org_j, des_j} + WT_i' + WT_j')
 \end{aligned} \tag{4.1}$$

$$\begin{aligned}
 (b) : TT_i + TT_j &= WT_i + DTT_{org_i, org_j} + WT_j + DTT_{org_j, des_j} + DTT_{des_j, des_i} \quad \forall i, j \in N \\
 SF_{i,j}^b &= TT_i + TT_j - (DTT_{org_i, des_i} + DTT_{org_j, des_j} + WT_i' + WT_j')
 \end{aligned} \tag{4.2}$$

$$\begin{aligned}
 (c) : TT_i + TT_j &= WT_i + DTT_{org_i, des_i} + WT_j + DTT_{org_j, des_j} \quad \forall i, j \in N \\
 SF_{i,j}^c &= TT_i + TT_j - (DTT_{org_i, des_i} + DTT_{org_j, des_j} + WT_i' + WT_j') \\
 SF_{i,j}^c &= WT_i + WT_j - (WT_i' + WT_j')
 \end{aligned} \tag{4.3}$$

Finally, the  $SF$  value for each pair of passengers is the minimum value among three different situations. It means that the algorithm chooses the condition that the additional travel time is minimum for sharing each pair of trips.

$$SF_{i,j} = \text{minimum}\{SF_{i,j}^a, SF_{i,j}^b, SF_{i,j}^c\} \tag{4.4}$$

#### 4.4.2 Clustering based on the similarity matrix

After computing the shareability function, we have the function value for each pair of requests that creates the shareability matrix. The shareability matrix is a kind of similarity matrix for the requests received that can be used in the clustering process.

After computing the shareability function for all the pairs of requests, we do the clustering

using the computed similarity matrix. When we make clusters based on the  $SF$ , we put the trip requests that are more potential to be shared in the same cluster (the trips that have lower  $SF$ ).

There are two main categories for clustering based on the similarity matrix:

1- Partitioned clustering algorithms cluster the data into  $k$  number of clusters. One of the usual algorithms for partitioned clustering is k-means clustering. K-means clustering is simple, fast, and flexible and is a proper method for our application.

2- Hierarchical clustering methods in which the clusters are arranged in a tree-like structure. Hierarchical clustering can be divided into Agglomerative hierarchical clustering (AHC) and divisive clustering [159, 170].

In [144], the authors have compared the hierarchical clustering and k-means clustering for urban taxi carpooling in a static environment. They show that compactness and separation are almost the same for hierarchical and k-means clustering for large cluster sizes. In dynamic large-scale problems, the results would be different. Also, the computation time becomes critical. Thus, to assess more, we implement both clustering methods on the shareability matrix to cluster the requests. In the next chapter, we assess the different clustering methods' performance, considering the quality and computation time.

### **Multidimensional scaling and k-means clustering**

K-Means method is a partitional clustering approach for decomposing the problem into independent subsets, following the principle of finding clusters of data based on their similarity. In the clustering procedure, the preferred number of clusters should be specified in the algorithm before execution. Each observation (data points) will randomly be assigned to a cluster, and find a set of centroids for each group of observations. Afterward, iterations will be done for optimizing the clusters.

k-means clustering takes place based on the distance between points. So we use the multidimensional scaling method ([171]) to convert the similarity matrix into a distance matrix which makes it possible to apply the appropriate clustering method based on the study in [137] (we have implemented the method using the mathtoolbox in C++).

After extracting the distance matrix, we use the modified k-means clustering method to create the same size clusters for the data received at every assignment time step. The modified k-means algorithm can be used to obtain clusters in preferred sizes [63]. Accordingly, we can find the best trade-off between cluster size and computation time, considering the objective function value. We favor a uniform distribution of requests among clusters to decrease computation times and facilitate parallel computations of each sub-problem.

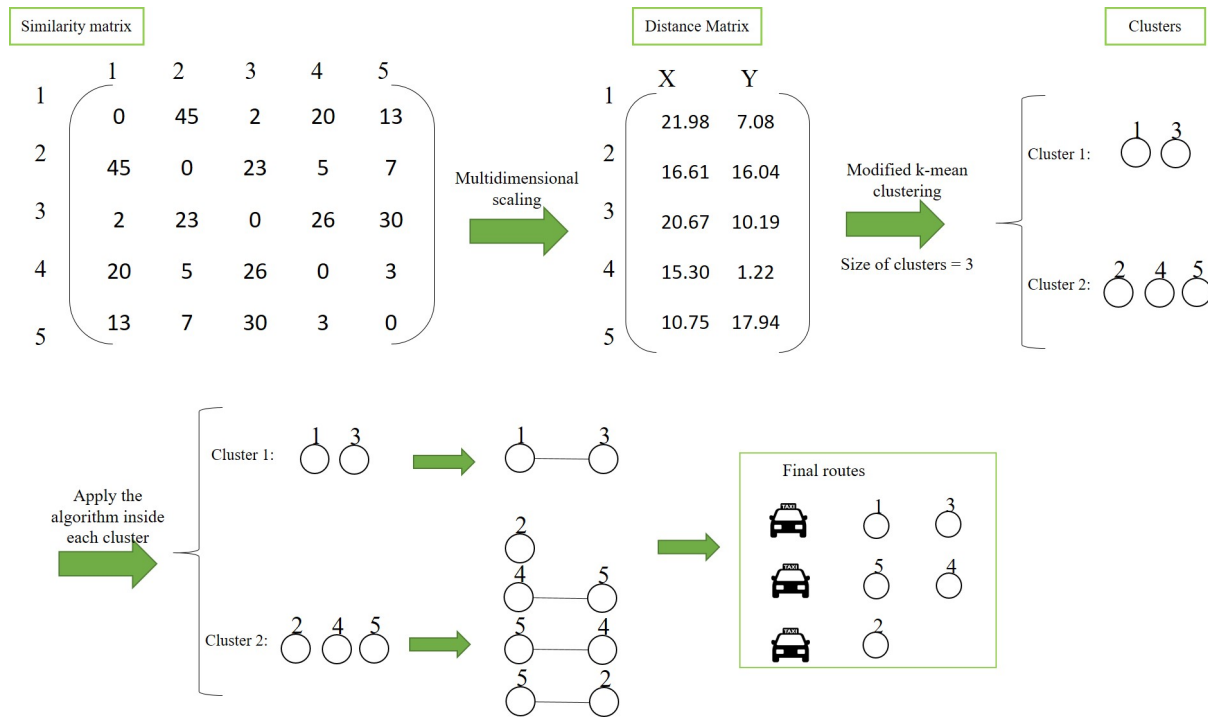


Figure 4.10: Final routes after multidimensional scaling and k-means clustering

Figure 4.10 shows a small example with five requests for the k-means clustering approach. The similarity matrix is computed and then converted to the distance matrix. K-means clustering puts requests 1 and 3 in one cluster and requests 2, 4, and 5 in another cluster. Then the assignment algorithm can be solved in each cluster.

### Hierarchical clustering

Hierarchical clustering offers a flexible and no-parametric approach and is an algorithm that builds hierarchy of clusters [131].

We use the agglomerative hierarchical method, which starts with taking singleton clusters

(that contain only one request per cluster) at the bottom level and continue merging two clusters at a time to build a bottom-up hierarchy of the clusters [146].

Figure 4.11 shows an example to show the hierarchical clustering method. It works directly with the similarity matrix. As the example is small, the results are the same as the k-means clustering method.

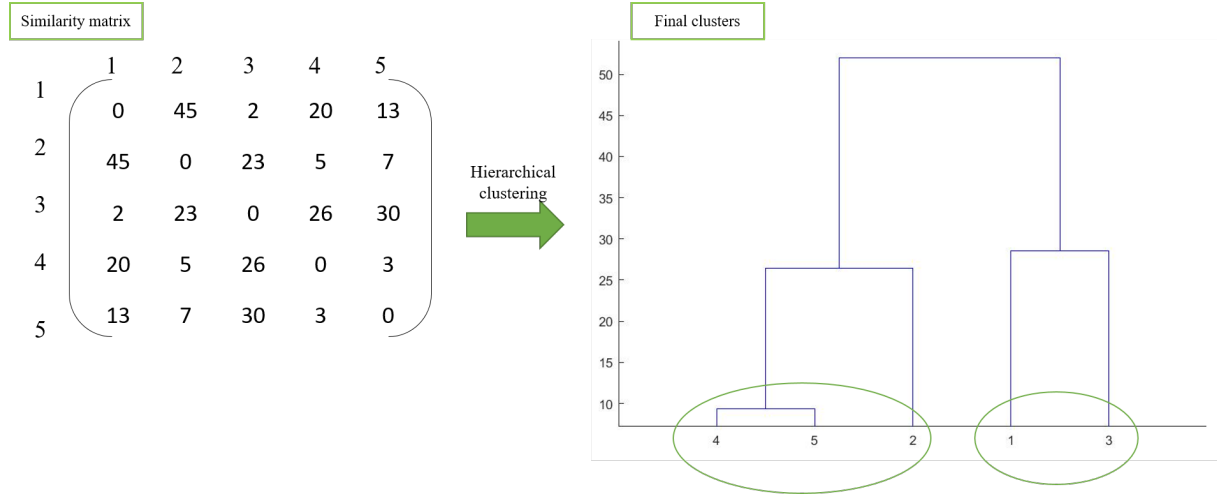


Figure 4.11: Final routes after hierarchical clustering

### 4.4.3 Determining the proper clustering method

In previous sections, we showed that we can use the k-means clustering method and the hierarchical method to cluster the requests based on the presented similarity matrix. In our method, the quality of the clustering method and computation time is very important. The time complexity of K-means is linear, while that of hierarchical clustering is quadratic. On the other side, k-means clustering requires prior knowledge of  $k$  and also multidimensional scaling to convert the similarity matrix into the distance matrix. We can stop at whatever number of clusters we find appropriate in hierarchical clustering by interpreting the dendrogram. As we use the agglomerative hierarchical method, we can have larger clusters faster with the hierarchical method.

To choose the best clustering method, we have compared both methods considering the quality of objective function and the computation time for different sizes of problems.

Table 4.3 shows the objective function and computation time with k-means clustering and hierarchical clustering method for different sizes of problems.

Table 4.3: Clustering methods comparison

Method	Number of requests	Objective function (normalized value)	Computation time (s)
Exact	112	29.99	12966.00
	1092	1924.00	288000.00
K-means clustering	112	30.96	100.10
	1092	1994.18	1134.00
	4482	7761.03	5340.00
	11160	20773.70	20981.10
Hierarchical clustering	112	30.96	99.03
	1092	1994.59	1131.10
	4482	7791.90	5187.50
	11160	20905.87	19950.00

In both the clustering methods, we try to have clusters with 50 requests. To have a baseline for the comparisons, we have computed the optimal objective function for 112 and 1,092 requests.

K-means clustering and hierarchical clustering methods increase the objective function by 3.22% for 112 requests and 3.64% and 3.66% for 1,092 requests comparing with the optimal solution. Both the clustering methods can decrease the computation time from 216 minutes to less than 2 minutes. It shows that both clustering methods are very effective in terms of reducing the computation time while keeping the quality of the solution acceptable. Then by increasing the number of requests, the computation time for both methods exponentially increases (the major part of the k-means method computation time is dedicated to the multidimensional scaling method, which exponentially increases by increasing the size of the problem).

K-means can give smaller objective function while hierarchical computation can result in a lower time. For 11,160 requests, the objective function is 0.64% lower for the k-means method, while the computation time is 5% more.

In our Lyon6 + villeurbanne test case, the maximum number of requests is 11,235. So we can use k-means clustering for this test case to have better solutions. However, for the Lyon network, we have more than 200,000 service requests. So the hierarchical method is

a better solution for this network as it works directly with the similarity matrix, and it can provide high-quality solutions.

## 4.5 Force the sharing method (FOSH method)

The optimizer aims to minimize the objective function, which combines both passengers' and operators' objectives. Therefore, the algorithm may choose a branch which has less sharing, compared with other feasible branches in the tree built by the algorithm. We propose the third heuristic method to force the algorithm to favor the longest possible route, which is in favor of more sharing.

When we increase the number of passengers assigned to a vehicle, the passengers' waiting time and travel time increase, so we reduce the length of the trip time window to keep the passenger's objectives acceptable.

The algorithm finds the longest possible routes, and then it chooses the path with the minimum objective. Algorithm 3 shows the modification in algorithm 1 to force the sharing. Thus, the command in line 10 of the algorithm is modified, as shown below:

---

### Algorithm 3: FOSH method

---

```

max-route-size = 0;
for feasible solution route  $\in$  Result do
    | if number of stops on route > max-route-size then
    | | max-route-size = number of stops on route;
for feasible solution route  $\in$  Result do
    | if number of stops on route = max-route-size then
    | | Put the route in re-Result;
Find the optimized route optimal-route  $\in$  Result;

```

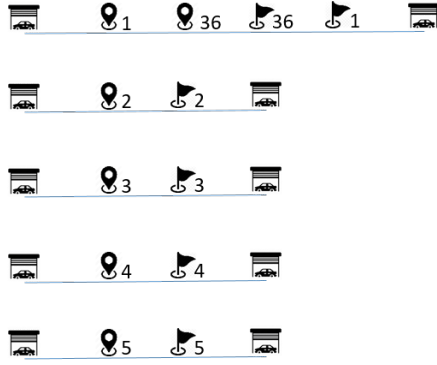
---

When the algorithm assigns more trips to a vehicle, the number of all the trips decreases. Thus, with the FOSH method, we expect to use fewer trips and consequently fewer service vehicles.

Figure 4.12 shows the routes for requests 1 to 5 with and without the FOSH method when the number of sharing is 1. Without the FOSH method, five vehicles can serve just six passengers, but with the FOSH method, these five vehicles can serve ten passengers.



Without FOSH



With FOSH

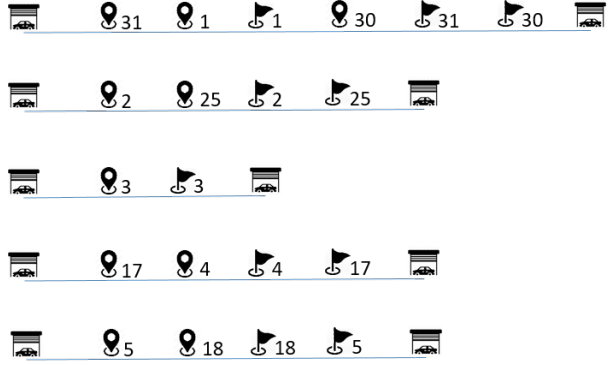


Figure 4.12: The assigned routes for requests 1 to 5 with and without the FOSH method

With the FOSH method, the algorithm can build fewer branches at each step. Because the number of trips on a single route is higher, it needs fewer branches to assign all the requests to the vehicles. Therefore, this method can also decrease computation time.

The final heuristic algorithm to solve the fleet management problem is presented in figure 4.13. In the next chapter, we first show the quality of different steps of the solution method (three heuristic methods). Then we do our analysis with the final algorithm.

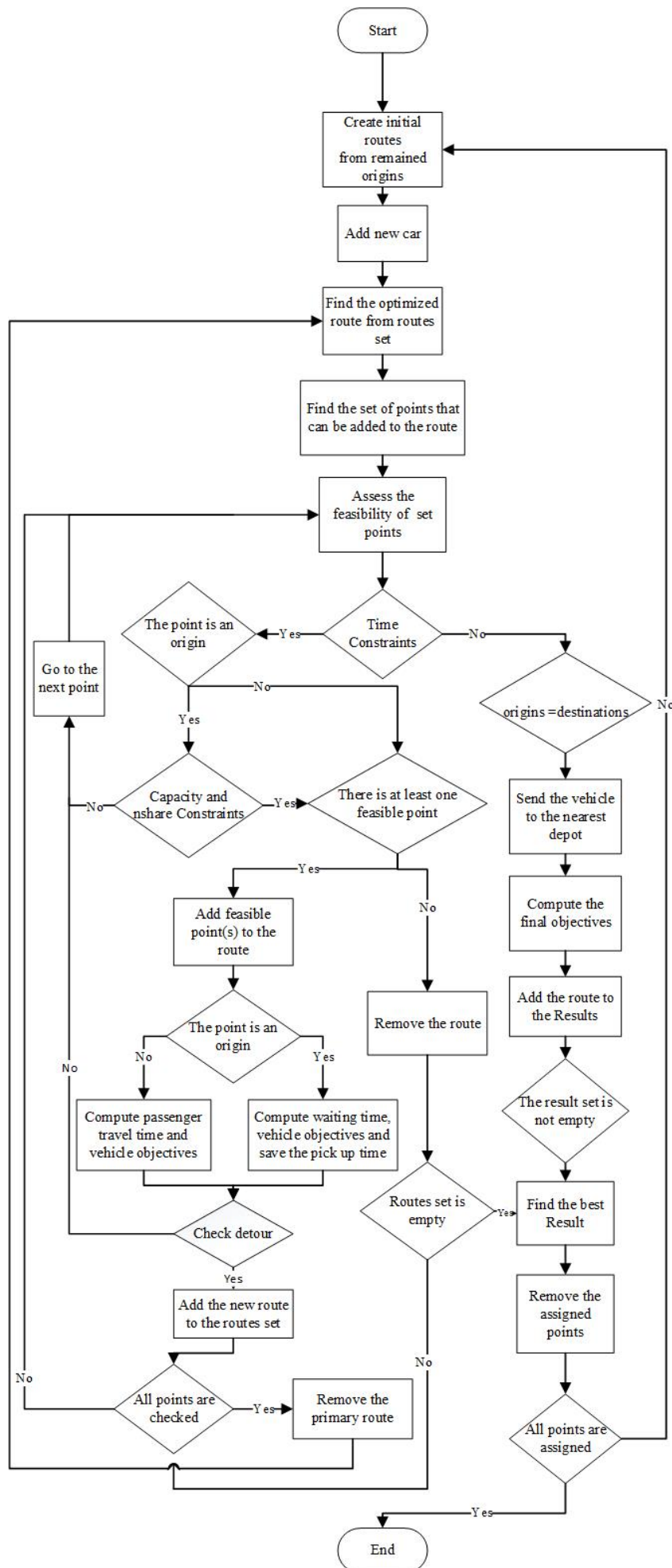


Figure 4.13: Final assignment algorithm

## Conclusion

In this chapter, we presented solution approaches to solve the dynamic ride-sharing problem. The approach is designed to be exact for small samples. It is then extended with several heuristics that keep the general design for the solution method but significantly reduce its computation time.

In the first heuristic method, we implement a rolling horizon approach with re-scheduling. The exact algorithm is solved every optimization time step, and at first, tries to put the requests in the moving cars' schedules. In the second heuristic method, the main strategy is to cluster the requests depending on a shareability index to create smaller samples that are faster to solve. This method narrows the exploration of the space to feasible and promising states only. We try both partitioned (k-means clustering) and hierarchical (agglomerative hierarchical clustering) methods. In the last heuristic step, we have presented the Force the Sharing (FOSH) method to favor the sharing. The final algorithm can solve a dynamic ride-sharing problem with high quality in a short time comparing with other methods in the literature.

In the next chapter, we show the performance of different heuristic steps.

## Part III

How to assess the impact of the  
network on the ride-sharing system and  
vice versa?

In the previous part, we presented the answer to the first sub-problem of dynamic ride-sharing systems. In this part, we investigate the second sub-problem: How to assess the impact of the network on the ride-sharing system and vice versa? To answer this question, in chapter 5, we present a "plant model," which is based on the trip-based Macroscopic Fundamental Diagram and can represent the real traffic situation. Then we present the "prediction model" based on the mean speed that can be used during the assignment process. Then, we assess the performance of the presented heuristic methods step by step to show the solution method's quality. In chapter 6, we analyze the impact of dynamic ride-sharing on network congestion in medium-scale and large-scale networks.

# Chapter 5

## Simulation Framework

### Introduction

In the previous chapters, to assess ride-sharing performance on the network demand, we presented the formulation and solving method of the dynamic ride-sharing problem. In this chapter, we investigate the interactions between transportation networks and dynamic ride-sharing systems. This problem has received less attention in the literature but is very important from the operational point of view. In fact, network congestion can have significant impacts on the ride-sharing service.

The ride-sharing service's optimization system uses estimates for the predicted travel time obtained from a so-called "prediction model". When the rides are executed, a gap usually exists between the estimation and the real traffic condition. The so-called "plant model" represents the real traffic condition, and it may require dynamic adjustment of the initial assignment to fit with the conditions observed. When simulating a dynamic ride-sharing service, it is essential to accurately distinguish the prediction and the plant models to provide a realistic assessment of the system functioning. In this chapter, we define the prediction model and the plant model for dynamic ride-sharing.

We use real data from the Lyon network in our simulations. The prediction model is based on the last observed travel times, while the plant model considered is a trip-based Macroscopic Fundamental Diagram (MFD) model able to reproduce the evolution over time of mean traffic conditions for a full road network using the MFD as a global behavioral

curve [104, 124, 123]. The macroscopic fundamental diagram (MFD) provides an overall overview of the network states [17].

We also explain the depot management method in our study.

Finally, the performance of the solution method is evaluated with our simulation-based optimization framework. In this chapter, in section 5.1, we present the dynamic simulation component, in section 5.2 we explain the depot management method, and in sections 5.3 and 5.4 we evaluate the performance of optimization and simulation components.

## 5.1 Dynamic simulation component

The dynamic ride-sharing system has two main components: 1- the fleet management component and 2- the dynamic simulation component. In chapter 3, we evaluated the fleet management component. In this chapter, we present our dynamic simulation component's performance, which is shown in green color in figure 5.1.

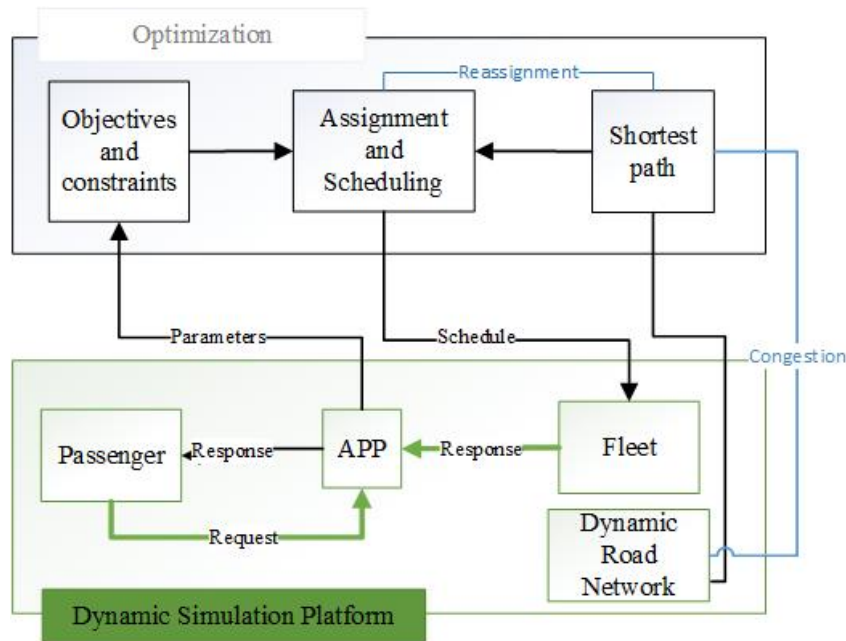


Figure 5.1: System components (dynamic simulation component)

### 5.1.1 Plant model

The plant model considers all the vehicles in the transportation network and not only the ride-sharing vehicles. That means that we also have a baseline of personal trips composed of: (i) trips that come from or go outside the area studied and that cannot be shared; (ii) trips that are wholly inside the region but for which rides are not requested and personal cars are chosen.

The market-share defines among the several trip categories how many users will request a ride or choose their personal car. Simulating all the trips whatever their mode of transportation mode (personal car or service car) guarantees that we can properly track congestion during peak hours.

In this research, the trip-based MFD is used to consider individual trips while keeping a very simple description of traffic dynamics [104, 124, 123, 106]. The general principle of this approach is to derive the inflow and outflow curves, noting that the travel distance  $L_i$  by a car  $i$  entering at time  $t - T(t)$  when  $n(t)$  is the number of en-route vehicles at time  $t$  and the mean speed of travelers is  $V(n(t))$  at every time  $t$ , must satisfy the following equation:

$$L_i = \int_{t-T(t)}^t V(n(s))ds \quad (5.1)$$

The function  $V(n(t))$  is the speed macroscopic fundamental diagram and can be derived from common observations for a transportation network [105]. For more details on the functioning of trip-based MFD, readers can refer to [106, 123].

Service cars can have two situations: They are waiting in depots for new passengers, or they are servicing the assigned passengers. In addition to the shared cars that are circulating to serve the passengers, personal cars are making trips that are not shared in the network.

The accumulation at each time  $t$  is the summation of the number of circulating service vehicles and the number of personal vehicles in the system. Therefore, at each time  $t$ , the



mean speed of travelers can be computed. Both shared trips and non-shared or personal trips can pass the length of a time period, based on the current mean speed at that time. The general assumption here is that personal and service vehicles have the same behaviors when driving and have the same marginal impact on congestion.

At each time step, the simulator computes the current speed of the cars considering the current traffic situation (the number of en-route vehicles). Then, the vehicle can cover a distance based on the current speed at every time step. So, the situation of cars is updated every time step, with the speed computed in the time and the remaining travel distance to cover. The time step that we use in our plant model is 1 second. So, the state of en-route cars is updated every second in the simulations.

Equation 5.2 shows the trip-based MFD to compute the speed in Lyon 6 + Villeurbanne and equation 5.3 shows the function for whole Lyon network where  $speed_t$  shows the speed of vehicles at time  $t$  and  $acc_t$  shows the accumulation of cars at time  $t$

$$speed_t = \begin{cases} md - (mc \times acc_t), & \text{if } acc_t < nj, acc_t > nc \\ mf - (mb \times acc_t)(ma \times acc_t^2), & \text{if } nj \leq acc_t \leq nc \end{cases} \quad (5.2)$$

$$nj = 1850, nc = 4400$$

$$ma = 1.369e - 6, mb = 8.835e - 3, mc = 1.575e - 3, md = 13.28, mf = 21.38$$

$$speed_t = \begin{cases} \frac{2.mp}{nc}, & \text{if } acc_t = 0 \\ \frac{(mp/nc^2)(acc_t)(2.nc - acc_t)}{acc_t}, & \text{if } acc_t \leq nc \\ \frac{(mp/(nj - nc^2))(nj - acc_t)(nj + acc_t - 2.nc)}{acc_t}, & \text{if } acc_t > nc \end{cases} \quad (5.3)$$

$$nj = 1.3393e + 05, nc = 5.6518e + 04, mp = 2.8887e + 05$$

The time and evolution of the mean speed over the full network in the network of Lyon 6 + Villeurbanne and the network of Lyon is shown in figure 5.2 and 5.3.

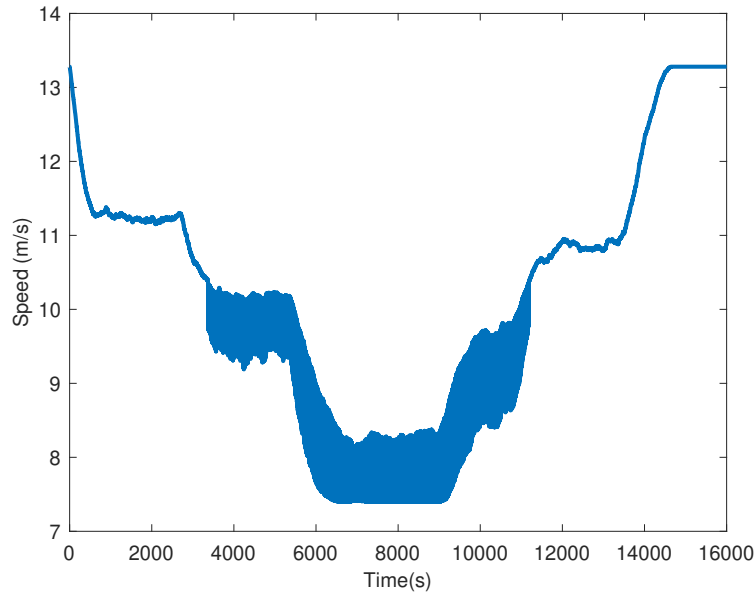


Figure 5.2: Speed evolution in Lyon 6 + Villeurbanne

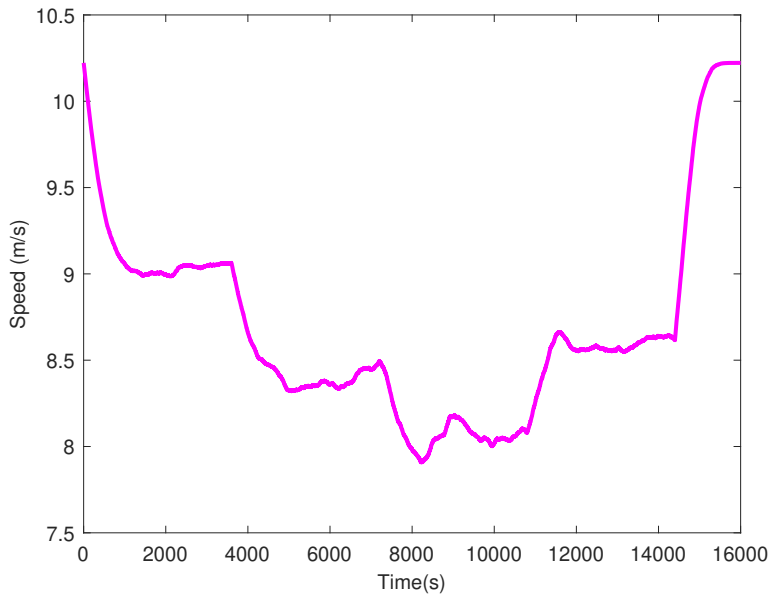


Figure 5.3: Speed evolution in the whole Lyon city

### 5.1.2 Prediction model

To carry out travel time prediction for the optimization part, in the prediction model, the traffic situation is predicted for the next assignment time horizon (every 10 minutes), and we assign the passengers to the cars based on this prediction. Thus, at each time step, the direct travel time ( $DTT_{i,j}^t$ ) from each point  $i$  to  $j$  is computed based on the current

mean speed and the shortest path between two points for the next 10 minutes. Then the optimization algorithm assigns all the requests for the next 10 minutes to the en-route cars or empty waiting cars.

In the rolling horizon method (Heuristic 1), the assignment procedure rolls over a specific horizon for the requests announced of a particular optimization step. In the current research, the rolling horizon is 20 minutes<sup>1</sup>, and the optimization time step is 10 minutes. So, the requests of the next 20 minutes that have not yet been assigned are optimized every 10 minutes. Some of the requests are re-optimized every 10 minutes. If a trip has been assigned to a vehicle which has left the depot, the algorithm does not assign it again, but if a trip is in the schedule of a waiting vehicle, the algorithm places it in the set of optimized trips in the particular horizon and re-optimizes it.

Note that we stop the simulations halfway on the rolling horizon ( $\frac{TH}{2}$ ) and solve a new fleet allocation problem over a new full rolling horizon ( $TH$ ). This prevents the system from being myopic to the new demand that may arrive just after the end of a simulation period, see figure 5.4.

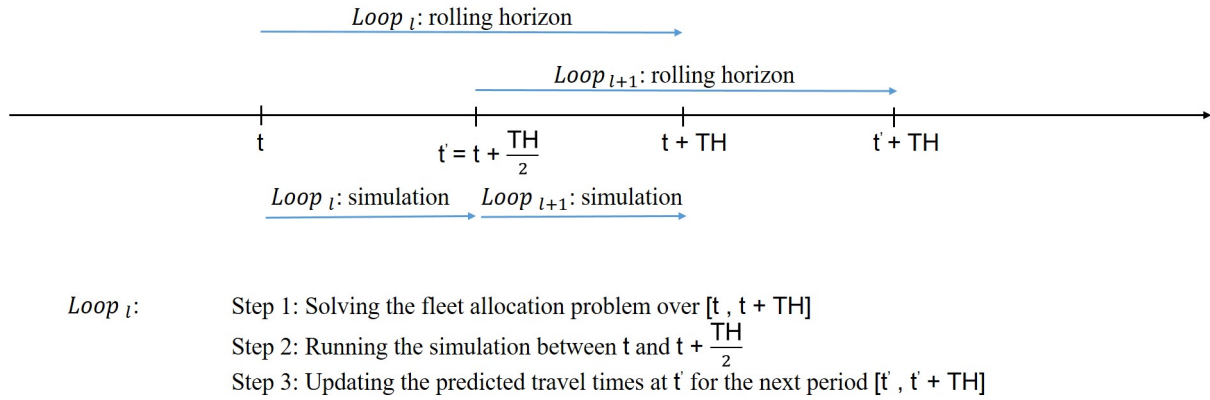


Figure 5.4: Rolling horizon and simulation period

We use the trips-based MFD as the dynamic simulator and the predicted speed at the beginning of each horizon as the prediction model, which can be calibrated, to do the

<sup>1</sup>The main goal of this study is to assess the performance of dynamic ride-sharing in terms of reducing congestion. To this purpose, it is more beneficial to have a longer optimization period to assess the system when we can have more sharing opportunities and be close to the optimal situation. Thus, if such a system that knows all the requests over a long horizon and is close to the optimal situation can not significantly decrease congestion in the network, we can say with quite high confidence that a real-time ride-sharing that matches the requests to the vehicles in real-time can not perform better

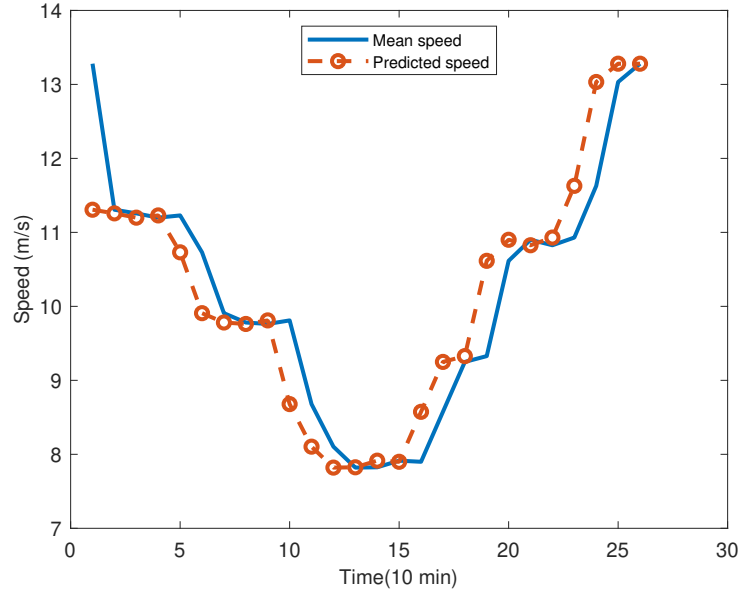


Figure 5.5: Predicted speed and mean speed during the simulation horizon (before  $\epsilon$ )

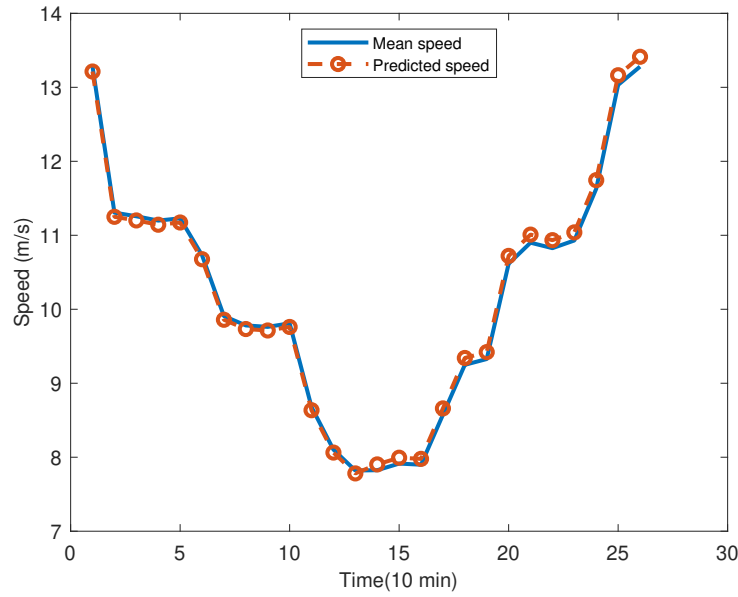


Figure 5.6: Predicted speed and mean speed (after  $\epsilon$ ) in Lyon 6 + Villeurbanne

optimization. To predict the mean speed over the next time horizon  $[t + TH]$ , we use the current mean speed at time  $t$  weighted by a proportional function, which is different for the network loading and unloading phases. This function accounts for the speed decrease that usually happens during the network loading (an increase of congestion) and the speed decrease during the unloading. Figure 5.5 (Lyon 6 + Villeurbanne) and figure 5.7 (Lyon) show the mean speed evaluation for a simulation with market-share = 100%. The results

of the prediction with the current speed only show a lag between the prediction and actual speed. The introduction of the correlation factor provides accurate predictions in figure 5.6 and figure 5.8. The proportional function has been set to  $\epsilon_l = 0.995$  (during the loading) and  $\epsilon_u = 1.010$  (during the unloading) for Lyon 6 + Villeurbanne and  $\epsilon_l = 0.993$  and  $\epsilon_u = 1.020$  for Lyon.

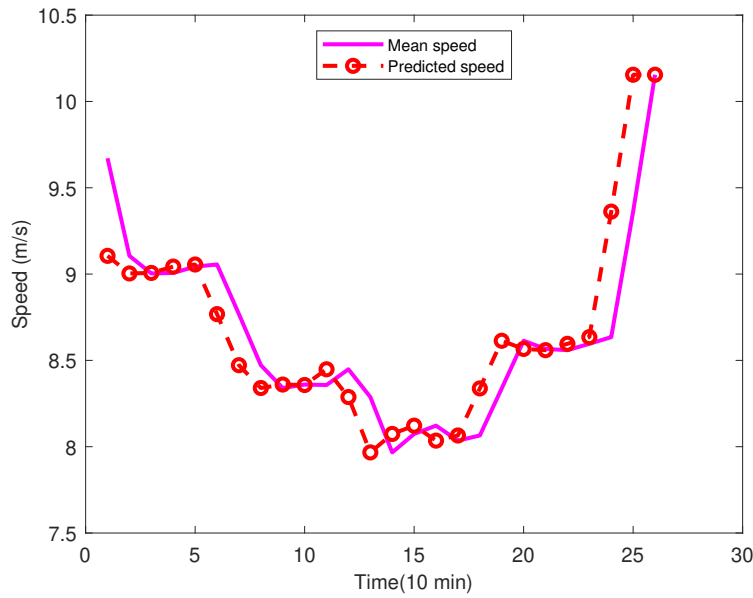


Figure 5.7: Predicted speed and mean speed during the simulation horizon (before  $\epsilon$ )

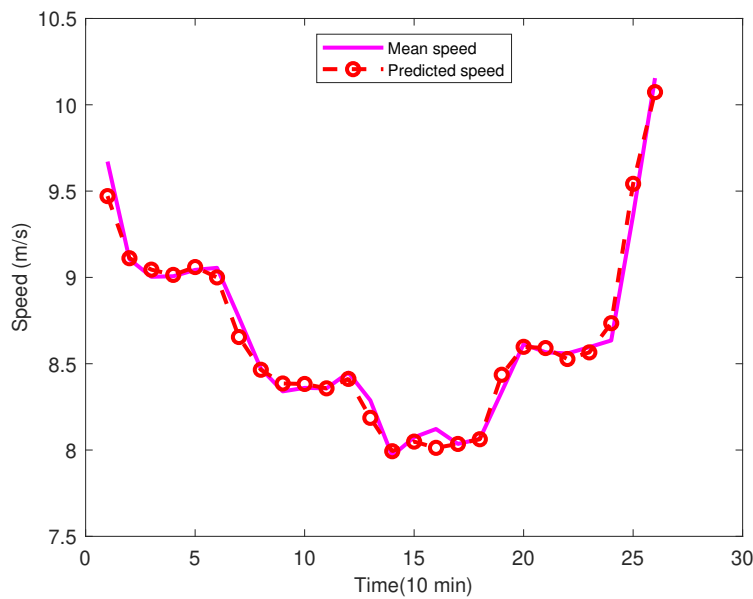


Figure 5.8: Predicted speed and mean speed (after  $\epsilon$ ) in Lyon

## 5.2 Depot management

The service provider has a fleet of vehicles in a ride-sharing system to serve the service requests. Participating service vehicles start up from a number of known locations or depots and after serving the assigned requests, they stop at this allowed locations to wait for the next passengers.

In our research we define two kinds of depots: local and the central depots. The central depot in the network can feed all the local depots. Thus, there is no limitation on the fleet size. On the one hand, distributing vehicles over the depots will decrease the waiting time for passengers. However, on the other hand, in the peak hour, if many vehicles are circulating in the network, the congestion will increase, and it leads to more travel time for vehicles and passengers. We analyze the number of vehicles in depots over the network to decide about the best distribution for the vehicles.



Figure 5.9: Central depot in Lyon 6 + Villeurbanne

To locate the cars at the beginning of the simulations, we use the historical data for the network demand to estimate the demand distribution over the network. Then we specify the number of cars at each location based on the demand for the depot. So if the demand is high, we consider more cars on the depot, and if the demand is low, we put fewer

vehicles at that location.

In Lyon 6 + Villeurbanne, we have defined 237 stop locations (all the allowed stop points in this network) for the service vehicles (local depots) and 1 central depot that can feed these local depots. Considering the demand distribution, at the beginning of the simulations, we feed 14 depots with two vehicles, 114 depots with one vehicle, and put 14 empty depots. The central depot can feed each depot when the demand is high. Also, the system sends back the vehicles to the central depot when there exceed the required number of vehicles for this depot.

Figure 5.9 shows the location of the central depot and some of the local depots in the network of Lyon 6 + Villeurbanne.

In the network of Lyon city, we have defined nine central depots that are uniformly located in the network. The number of allowed stop locations is 2,272 points on the network. As the primary purpose of the thesis is to assess the impact of ride-sharing on reducing congestion, we ensure that the system can always have close service vehicles to any request, and the central depots can always feed the local depots.

## 5.3 Optimization component performance

In this section, we assess the performance of different steps of the solving method by comparing them two by two.

### 5.3.1 Simulations' configuration

Table 5.1 shows the primary configuration of the simulations. The computations are carried out on a desktop with two Intel Xeon core E5-2620 processors, 64 GB RAM and the Windows 10 operating system running C++ Visual Studio 2013.

One of the goals of this thesis is to figure out the performance of a dynamic ride-sharing system under the optimal situation. Here we assess the different optimization steps using the data from the network of Lyon 6 + Villeurbanne. In this case study, we fully monitor traffic dynamics as we assess both service and personal trips in the network. In the

Table 5.1: Simulations configuration

Parameters	Values
Rolling horizon	20 min
Optimization time step	10 min
Simulation time step	1 s
Fixed time window length	6 min
Number of sharing	0, 1, 2
Market-share	1 to 100%
Car capacity	4
Number of cars in local depots	142
Number of cars in central depot	1000
Service time for each passenger	1 min

rolling horizon approach, to find the near-optimal matching for the service vehicles in the dynamic traffic conditions, we choose a longer time step comparing with other dynamic methods to guarantee that we can find near-optimal solutions for the matching problem. Hence, to solve the problem dynamically, we apply the method every 10 minutes over the requests received in the next 20 minutes, considering we have a perfect knowledge of all requests over such a time horizon. We update the system situation every 1 second.

In chapter 3, we explained that the pick-up time window is the time interval between the earliest pick-up time and the latest pick up time. Similarly, we can define the drop off time window. In our simulations, This time interval is computed based on the trip length. It has a fixed length, which is set to be 6 minutes (considering the average trip length). Then we add 1 minute for each 1-kilometer trip. Equation 5.4 shows how we compute the time window's interval ( $TWI$ ) for each request where  $Fixed_{TW}$  is the fixed-length (6 minutes in the primary configurations) and  $DTD_{i,i+n}^t$  is the trip length (direct travel distance from passenger  $i$  origin to destination). For example, for a trip of 7 kilometers, the waiting time is  $6 + 7 = 13$  minutes. The passenger  $i$  defines the earliest pick-up time ( $EP_i$ ) when he/she sends the trip request to the system. Then the pick-up and drop off time window can be defined as below ( $DTT_{i,i+n}^t$  is the direct travel time from passenger  $i$ 's



origin to destination at time  $t$ ).

$$TWI = Fixed_{TW} + 1 \times DTD_{i,i+n}^t \quad (5.4)$$

- pickup time window:  $(EP_i, EP_i + TWI)$
- drop off time window:  $(EP_i + DTT_{i,i+n}^t, EP_i + DTT_{i,i+n}^t + TWI)$

The capacity of vehicles in the primary configurations is 4 and we do the simulations with the number of sharing 0 (like traditional taxi services without sharing), number of sharing 1 and 2. The market-share can be from 1 percent to 100 percent of the trips.

We consider 1 minute service time for passenger pick up and drop off.

The result tables show the number of trips, the number of shared vehicles, the total travel time and distance for the vehicles, the number of requests, the total travel time and waiting time for all the passengers and the total travel time of personal vehicles for different market shares and numbers of sharing.

We assume that personal trips start at the origin precisely at the departure time without any waiting time and they end the trip at the destination. The travel time for the travelers of personal vehicles is equal to the vehicle travel time.

### 5.3.2 Exact assignment algorithm over the full-time horizon

Our first test case focuses on the exact global solution. It can be obtained only if the number of requests is low and no heuristics, including the rolling horizon, are running. The results are provided for a market-share of 4%, i.e. a total of 430 trips with the number of sharing 0, 1, 2 and 3. See Table 5.2.  $N_{trips}$  is the number of trips to serve all the requests,  $m$  is the number of vehicles used,  $\sum_{k \in M} T_k$  and  $\sum_{k \in M} TD_k$  are total travel time and distance for service vehicles,  $n$  is the number of passengers and  $\sum_{i \in N} TT_i$  and  $\sum_{i \in N} WT_i$  are total travel time and waiting time for passengers.

When the number of sharing is 0, each service vehicle serves only one passenger at a time, and the system does not share any ride. Then, with the number of sharing 1, the system

Table 5.2: Simulation results for optimal assignment

Configuration	Shared vehicles				Passengers			Simulation time(h)
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	$n$	$\sum_{i \in N} TT_i(h)$	$\sum_{i \in N} WT_i(h)$	
MS : 4 %								
nshare = 0	429	32	46.36	1509.3	430	39.01	2.50	0.33
nshare = 1	419	32	46.09	1499.7	430	39.18	2.69	35.81
nshare = 2	417	32	46.08	1494.9	430	39.36	2.70	224.50
nshare = 3	416	32	46.08	1494.3	430	39.60	3.07	505.11

is allowed to share a passenger ride with only one other passenger at the same time, as in e.g. [87], and in continuation with the number of sharing 2, three passengers can be in the same vehicle at the same time. With the number of sharing 3, the system uses all the vehicle capacity to serve the passengers.

It is clear that sharing can significantly improve the objective function. This means that with our proposed sharing method, the system can serve the requests with fewer vehicles and a better objective function.

In the general case, sharing can decrease the number of vehicles needed and maybe improve the objective function (this is our current investigation). However, with so few trips here, the impact of sharing is very limited. What is interesting is to assess the increase of the feasible solutions (branches in the tree) with the increase of the number of sharing and the impacts on computation times.

When the number of sharing is 0, each vehicle serves just one passenger at a time. The algorithm also considers the trips that can be in sequence. When the travel time from the first destination to the second origin is shorter than the travel time between the first destination and the closest depot to this point, the algorithm puts these two trips in sequence. So, for example, in the first step when the algorithm is building the tree with branches, the number of branches does not go further than the number of requests. The results show that it takes just 0.33 hour to simulate our ride-sharing system in the morning peak hour. Then, when we increase the number of sharing to 1, we reduce the constraint on the number of passengers in the vehicle at the same time. Thus, the exploration space is expanded, and the algorithm can extract more branches at each step. The simulation time when we have an optimal system with the number of sharing 1 is 35.81 hours. When

the number of sharing is 2, the algorithm can add any permutation of the other two trips to the first trips. Thus, the number of branches increases exponentially and, as can be seen the computation time for the number of sharing 2 is much longer than the computation time for the number of sharing 0 and 1. It takes almost 224 hours to simulate the system function with the number of sharing 2.

When the number of sharing is 3, the total travel distance is reduced by 600 meters while the total waiting time is increased by 22 minutes. Also, the computation time is 505 hours for the number of sharing 3.

### 5.3.3 Comparing the exact algorithm with the existing methods

As it is shown in section 3.2.5, increasing the number of requests will exponentially increase the computation time of the algorithm. We have developed the algorithm in terms of computations speed with paralleling the third main loop in the algorithm (for  $sp \in SP$ ). In the algorithm, the feasible points are added to the feasible branches to extend the tree and find all the possible routes. By paralleling the third main loop, the algorithm can add the points to the branches in parallel simultaneously (24 points at each time). This can enable us to solve larger problems with the exact algorithm.

To show the quality of our exact solution method, we have compared the computation time with a CPLEX solver for the same problem. Table 5.3 shows the computation time for different number of requests. As the problem is NP-hard, it is very expensive in terms of time to compute the exact solution with CPLEX for more than 8 requests. Only for 4 requests, the computation time for our presented algorithm is 26 times better than CPLEX.

[35] propose an exact Branch-and-Cut algorithm for similar problem that can outperform the state-of-the-art solver CPLEX. The computation time of their proposed algorithm with 40 requests is 578 seconds. Our presented algorithm can solve the problem with 40 requests in 230 seconds. Then they propose a lean heuristic algorithm based on Large Neighborhood Search (LNS) to find near-optimal solutions. In another study on pick up and delivery problem for ride-sharing by [173], the exact solution for the ride-sharing

Table 5.3: Solution methods comparison

Method	Number of requests	Objective function (normalized value)	Computation time (s)
CPLEX	4	0.54	0.40
	5	0.65	5.30
	6	0.74	43.50
	7	1.14	287.60
	8	1.24	2945.90
Exact solution Method	4	0.54	0.02
	5	0.65	0.05
	6	0.74	0.08
	7	1.14	0.19
	8	1.24	0.25

problem takes less than 8 seconds for 9 requests. Our method can find the exact solution for 9 requests in less than a second.

So our solution method outperforms existing solvers when determining the exact solution for small instances. It allows us to compute the exact solution for medium-size instances, e.g., market-share = 4%. But we can not solve larger problems without introducing heuristics. We will explain about the performance and accuracy of our heuristics in the next sections.

### 5.3.4 Heuristic 1: Rolling horizon and re-scheduling performance

The rolling horizon immediately improves computational times by reducing the exploration space. Thus, the algorithm can handle the fleet assignment problem with higher market-shares. On the other hand, the objective function will increase compared with the optimal situation. Figure 5.10 compares the first heuristic method with the optimal system performance when the market share is 1 percent, considering the objective function value and the simulation time for different numbers of sharing.

To compare the objective functions of different methods, we assume a baseline. The baseline is the objective function of the method which is expected to give the best value. To make it clear, we set the objective function of the baseline equal to zero and then

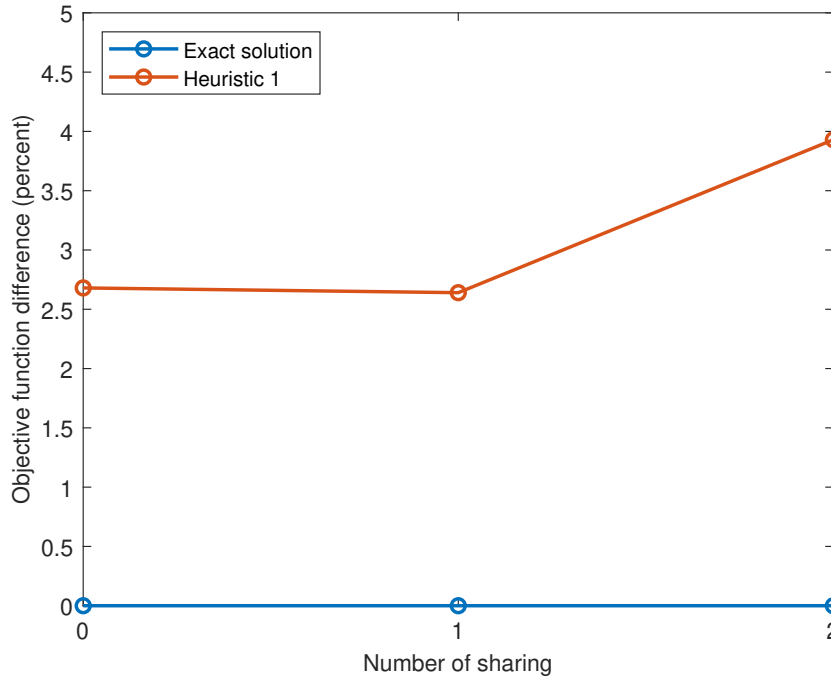


Figure 5.10: Comparison between the exact method and heuristic 1 (market-share = 1%)

nshare	Simulation time (min)	
	Exact method	H1
0	2.8	3.1
1	216.1	3.1
2	2382.8	3.1

Table 5.4: Simulation time for the exact method and heuristic 1

compute the percentage of objective function differences for the second method and the reference method and plot the objective function difference percentages.

The exact method leads to lower objective function values than heuristic 1. Therefore, we assume the exact method as the baseline and plot figure 5.10 considering the percentage increase with heuristic 1. With the rolling horizon approach, the objective function increases by only 2.68% for the number of sharing 0, 2.64% for the number of sharing 1 and 3.93% for the number of sharing 2, compared to the exact method while the simulation time decreases by more than 89% for the number of sharing 1 and 3000% for the number of sharing 2. The increase in the objective function is therefore negligible compared to the improvement in the simulation time with the first heuristic approach.

nshare	Simulation time (min)			
	<i>H1</i>	<i>H2</i> (cs 10)	<i>H2</i> (cs 20)	<i>H2</i> (cs 30)
0	3.2	4.5	5.1	5.4
1	98.0	5.1	9.1	18.9
2	1998.9	14.9	130.7	301.4

Table 5.5: Simulation time for heuristic 1 and heuristic 2

### 5.3.5 Heuristic 2: Clustering method performance

The computation time for the dynamic ride-sharing system simulation with bigger market-shares in the optimal situation is long. In particular, it is not efficient when the number of sharing is 2. The clustering method proposed narrows the search for feasible solutions in the algorithm and makes it fast enough to assign requests to the shared vehicles in a short time and respond to the passengers quickly.

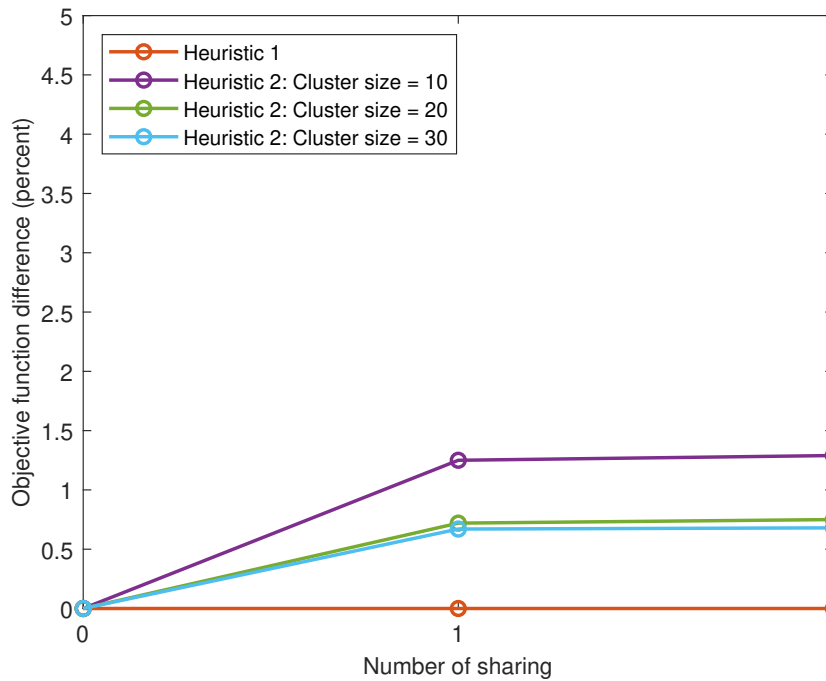


Figure 5.11: Comparing heuristic 1 and heuristic 2 when the market-share = 10%

To examine the performance of the heuristic method proposed, the simulations are compared to the previous results with a market-share of 10% considering the simulation time and the quality of the objective function for different sizes of clusters.

For a market-share of 10% the maximum number of requests at every time step is not

more than 75, so we execute the simulations with cluster sizes of 10, 20, and 30 to have at least 3 clusters at each optimization time step and to be sure that the clustering method will be effective in terms of reducing the computation time.

Figure 5.11 shows the comparison for the objective function values and simulation times. Clustering can significantly improve the computation time with a very small increase in the objective function. This is because we carefully define the clusters by putting the shareable trips together. Therefore, the optimization is not deteriorated, even when the number of sharing is 1 and 2.

In figure 5.11, the reference method is heuristic 1, and the clustering method with three different sizes of clusters is compared to the reference method. When the number of sharing is 0, all the methods give the same solutions. When the number of sharing is 1 and 2, the objective function is 1.25% and 1.29% higher for a cluster size of 10, 0.72% and 0.75% higher for a cluster size of 20 and 0.67% and 0.68% higher for a cluster size of 30.

It is also clear that the clustering method can significantly reduce computation time. When the number of sharing is 1 and 2, the improvement in the assignment computation time makes the clustering computation time negligible. This improvement is considerable when the number of sharing is 2. The clustering method with a cluster size of 30 can reduce the computation time when the number of sharing is 2 from 2001 seconds to 300.4 seconds while increasing the objective function with just 0.68%. The computation time for the clustering method, when the number of sharing is 0, is equal to or greater than the computation time without clustering. This is due to the clustering execution time.

### 5.3.6 Heuristic 3: Force the sharing method (FOSH method) performance

With clustering we can now address large size problems and investigate higher market-shares. Here we focus on the 40% market-share case to determine the influence of the final FOSH heuristic.

When we favor sharing in the algorithm, the assignment gives shorter travel times and

distances for the vehicles and longer travel times and waiting times for the passengers. To keep the waiting time acceptable for passengers, we set the fixed time window length to 1 minute instead of 6 minutes in the simulations. As the time window for passenger pickup and drop off times is restricted, the decrease in the vehicle objective becomes dominant. Figure 5.12 shows the objective functions with and without the FOSH method for different cluster sizes and numbers of sharing when the market-share is 40%. As the number of requests in each time step is higher for higher market-shares, we increase the size of the clusters to 40. We expect a better objective function for bigger cluster sizes. Thus, here, we consider that the reference method is the clustering method (heuristic 2) with a cluster size of 40, and we compute the percentage increase of the objective function for other methods based on this reference method. The FOSH method (heuristic 3) increases the objective function by only 0.13% when the number of sharing is 1 and 0.01% when the number of sharing is 2. The results are almost the same for the other cluster sizes. The objective function increases up to 2.5%. Also, the results show that cluster sizes of 30 and 40 can give very similar solutions. The difference between the cluster size of 30 and 40 is less than 0.5% for both heuristic 2 and heuristic 3.

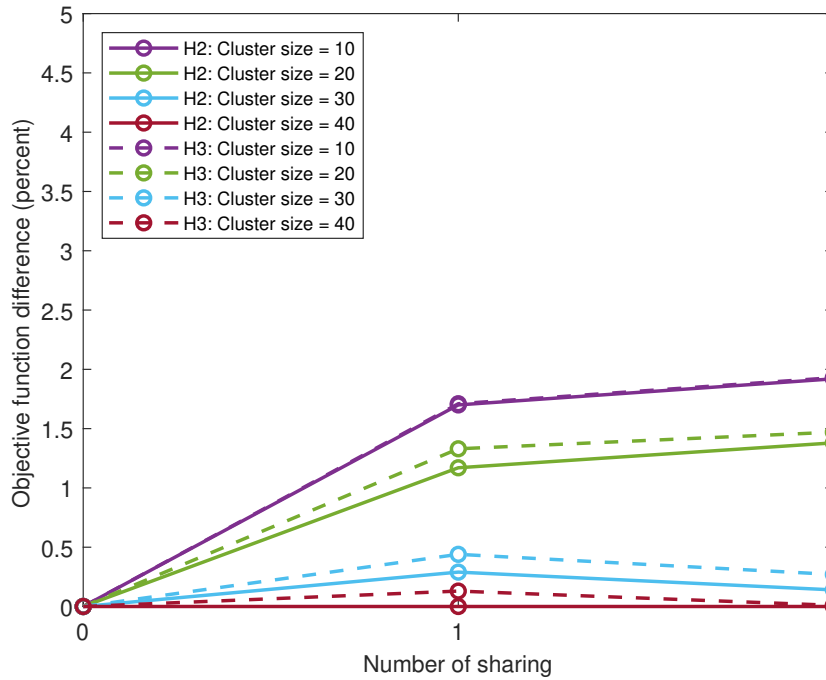


Figure 5.12: Comparing the heuristic 2 and the heuristic 3 when market-share = 40%



nshare	Simulation time (min)	
	<i>H2</i>	<i>H3</i>
cs = 10		
0	43.7	36.7
1	44.6	37.8
2	110.4	38.0
cs = 20		
0	43.7	43.8
1	56.4	46.1
2	598.5	49.7
cs = 30		
0	45.4	45.7
1	89.0	53.1
2	3091.3	94.3
cs = 40		
0	47.0	46.9
1	138.2	61.6
2	3333.3	238.2

Table 5.6: Simulation time for heuristic 2 and heuristic 3

In the FOSH method, the algorithm assigns more trips within each branch every time. Therefore, we expect that this method decreases the simulation time. Table 5.6 shows the simulation time comparison for heuristic 2 and heuristic 3. Heuristic 3 significantly improves computation time. Also, with this method a huge drop is observed when the number of sharing is 2.

When the algorithm assigns more requests to a vehicle, the number of trips needed to serve all the passengers' requests decreases. Thus, with the FOSH method, we expect to use fewer trips and consequently, fewer service vehicles. Figure 5.13 proves this assumption. The number of shared cars is significantly lower with the FOSH method. Heuristic 2 serves 40% of the network demand via 4,257 trips with a cluster size of 30 and the number of sharing at one while heuristic 3 serves the same number of requests via 2,407 trips (each trip starts from a depot and ends at a depot).

As the results show, a cluster size of 30 can give a reasonable trade-off between the computation time and the quality of the solution for this scale. Therefore, in the next experiments, we execute the simulations with a cluster size of 30 to ensure rapidity and also to keep the quality of the objective function at an adequate level.

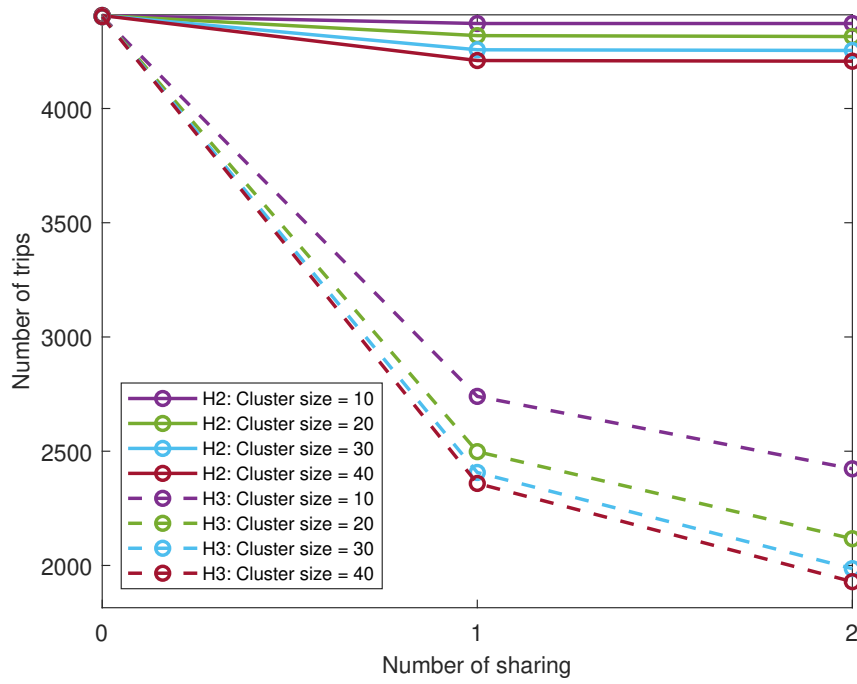


Figure 5.13: Comparing the number of trips for heuristic 2 and heuristic 3 when market-share = 40%

### 5.3.7 Comparing the final algorithm with the existing methods

In this section, we show the quality of the proposed method comparing with the other existing methods.

First, it is important to see the performance of the solution method comparing with the optimal situation when we increase the market-share.

Method	Simulation time (min)	
	nshare 0	nshare 1
Exact solution	54.1	4800.0
Heuristic 1	3.2	98.0
Heuristic 2	5.4	18.9
Heuristic 3	4.4	6.6

Table 5.7: Solution methods comparison (computation time for market-share = 10%)

Figure 5.14 shows the percentage of difference for all the solution method steps compared to the optimal solution when the market-share is 10%, and the size of clusters is 30 and Table 5.7 shows the computation time. The number of requests, in this case, is 1092, and

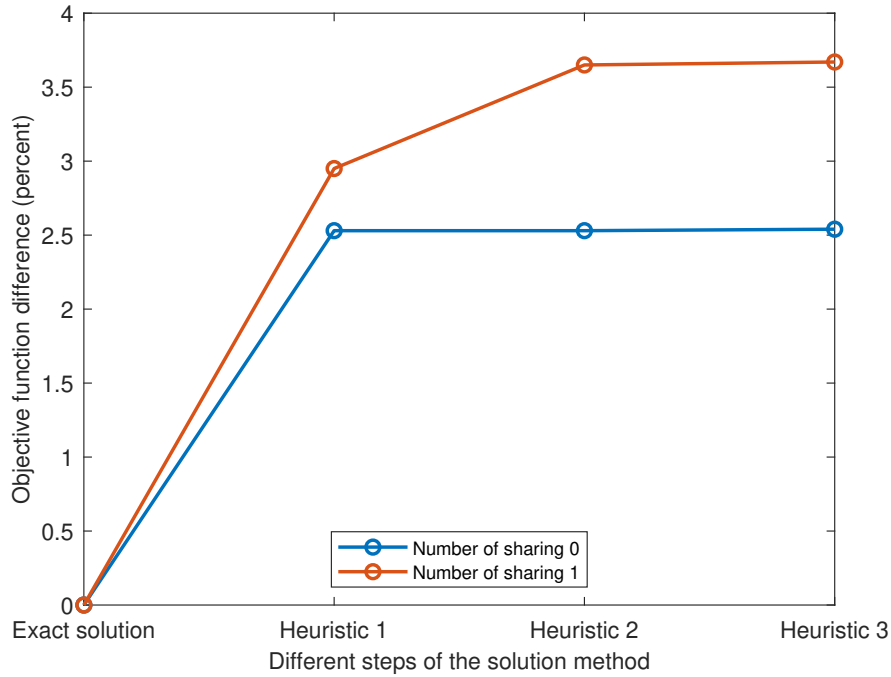


Figure 5.14: Objective function difference for the solution method steps (market-share = 10%)

Table 5.8: Simulation results for heuristic methods (number of sharing = 1)

Configuration	Shared vehicles			Passengers	Personal vehicles	All vehicles
	$N_{trips}$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	$\sum_{i \in N} WT_i(h)$	Travel time (h)	Travel time (h)
MS : 0%					7978.4	7978.4
MS : 10%						
Exact solution	1015	117.6	3744.6	8.2	7904.1	8021.7
Heuristic 1	1041	119.5	3825.1	8.5	7907.5	8027.1
Heuristic 2	1067	120.5	3853.0	6.8	7906.5	8027.0
Heuristic 3	610	101.8	3279.4	20.2	7896.5	7998.3

the computation time for the optimal solution is 80 hours when the number of sharing is 1. The figure shows that the final algorithm increases the objective function for 3.67% while the computation time decreases to 6 minutes (99.9% decrease) when the number of sharing is 1. The results prove the capability of the presented algorithm to provide fast and qualified solutions for the dynamic ride-sharing problem. The optimization time step is 10 minutes, so the rolling horizon method can find near-optimal solutions. Then in heuristic 2, the shareability index can effectively find the potential sharing opportunities and cluster the requests. Therefore, increasing the market-share will not affect the quality of the solution methods, and the results are very close to the optimal situation.

Table 5.8 shows the results for different solution methods when we serve 10% of the internal trips with service cars.

The market-share = 0 is when only personal vehicles serve all the network demand. Using service vehicles increases the travel distance for the vehicles and consequently, the travel time. Serving 10% of the requests with service cars, increases the total travel time for vehicles by 43.3 hours in the optimal situation. The FOSH method reduces this value to 19.9 hours by favoring the sharing but it increases the passengers' waiting time from 8.2 hours in the optimal situation (27.0 seconds for each passenger on average) to 20.0 hours (66.6 seconds for each passenger on average).

The heuristic method proposed here can also outperform the previous methods in the literature. In [35] after an exact method, they propose a lean heuristic algorithm based on Large Neighborhood Search (LNS), to find near-optimal solutions. The lower bounds generated by their exact approaches are on average 3.68% better than the average LNS result and the average computation time is less than 90 s for instances with up to 40 requests, while large-scale instances with up to 100 requests are solved in about 10 min on average. Our heuristic method can solve the problem in less than 20 seconds for 40 requests and less than a minute for 112 requests while it increases the objective function only by 3.19% percent for 112 requests. For market-share = 10% (1092requests) the heuristic method increases the objective function for 3.67%. [173] present a Tabu search heuristic for the pick and delivery problems for ride-sharing. They use a ratio (the objective value of the optimal solution divided by the output objective value of the heuristic method) to compare the heuristic method with optimal situation. For 9 requests, the ratio is 0.94 in average. The final heuristic method proposed in this study can find optimal solution for 9 requests. Increasing the number of requests, keeps this ratio low. For market-share = 10%, this ratio is 0.96 in our method.

Different clustering methods have been implemented in the literature for large-scale problems. They usually divide the space geographically and use a spatial clustering to downsize the problem. We have compared the Shareability clustering with such a spatial clustering method.

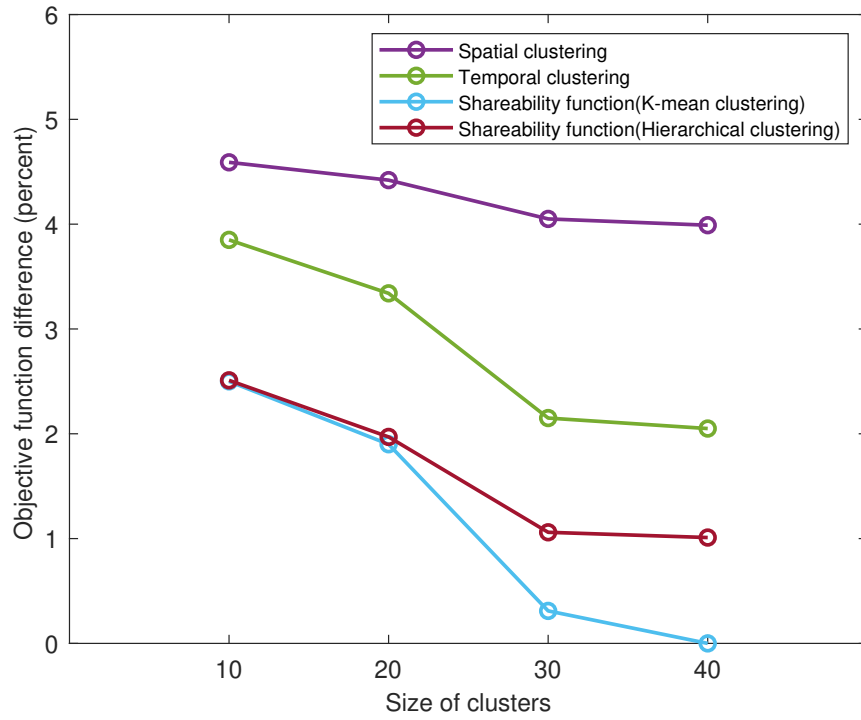


Figure 5.15: Comparing clustering methods' objective function (market-share=50%)

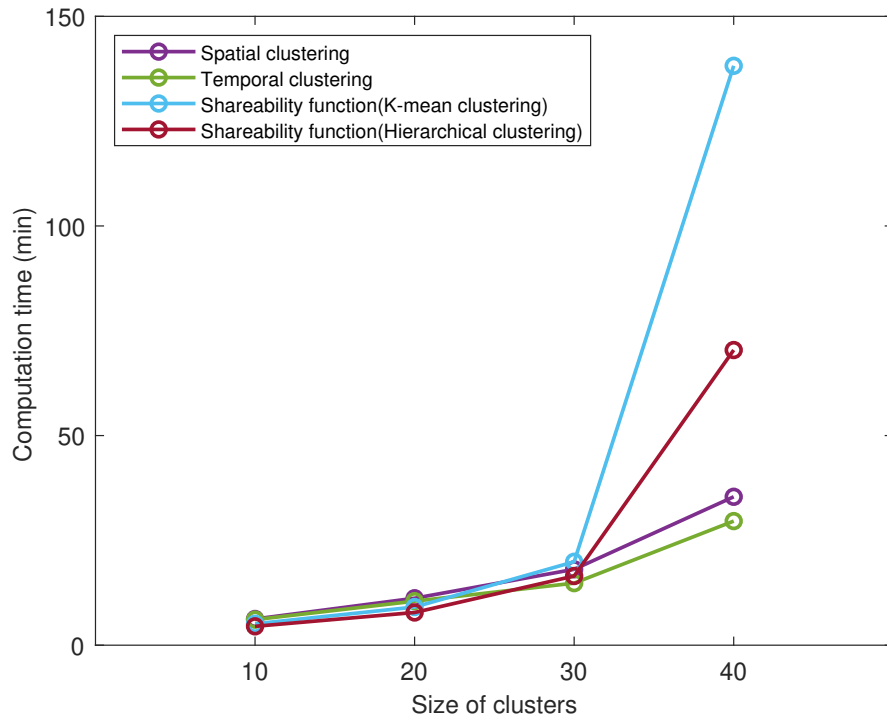


Figure 5.16: Comparing clustering methods' computation time (market-share=50%)

For the spatial clustering, we put the two corresponding trips in the same cluster based

on the distance between their origins. Also, we try to cluster the trips based on the time in a temporal clustering method. For the temporal clustering, we put two trips in the same cluster based on their departure time and their position. Finally, we compare these methods with our proposed method to show the quality of our proposition.

Figures 5.15 and 5.16 show the comparison of different clustering methods considering the objective function and the computation time for four different cluster sizes when the market-share is 50%. The best objective function is provided by our k-means clustering method when the size of the cluster is 40. So we choose this objective function value as a base, and we compute the percentage of difference for other methods considering this basic scenario. As it is clear, the performance of spatial clustering is not acceptable compared to the other methods. In the best situation, the spatial clustering's objective function is 4% more than the k-means clustering. The temporal clustering can perform better than spatial clustering, but it can not outperform our clustering methods. With the cluster size of 40, the objective function for temporal clustering is 2.02% more than k-means clustering. The computation time increases exponentially for the shareability clustering when the cluster size is 40. However, with the cluster size of 30, the algorithm can give a high-quality solution in a short time.

As explained in section 4.4.3, we can do the transportation analysis for medium-scale with the k-means method when the cluster size is 30. For large-scale, we use the hierarchical method.

## 5.4 Dynamic simulation component performance

Network congestion has impacts on the dynamic ride-sharing service. When the rides are executed, a gap can exist between the estimated travel times used by the optimization process at the beginning of the time horizon and the travel times experienced during the time horizon in the plant model. So, the objective function when solving the fleet allocation problem at the beginning of the time horizon can be different from the objective function experienced. Then, the current speed is updated based on the new traffic condition for the next step to take into account the impact of congestion and to minimize this

gap.

Table 5.9 shows the estimated objective and the objective function values experienced (normalized values) for the different methods when the market-share is 10%, and the cluster size for heuristic 2 and 3 is 30. The objective function implemented is greater than the estimated objective function for all the methods because of the gap between the predicted and the real travel times. For example, when the number of sharing is 1, the estimated objective function for heuristic 1 is 1,980, but the objective function experienced is 2,030. The differences are small in all the scenarios showing that the prediction model is accurate enough.

Table 5.9: Simulations results for optimal assignment

Method	Estimated objective function			Experienced objective function		
	nshare 0	nshare 1	nshare 2	nshare 0	nshare 1	nshare 2
Heuristic 1	2017	1980	1979	2063	2030	2029
Heuristic 2	2017	1994	1993	2063	2043	2039
Heuristic 3	2017	1994	1993	2063	2008	2007

## Conclusion

In this chapter, we presented the simulation framework for the dynamic ride-sharing problem. We defined the plant model to represent the real traffic conditions and the prediction model to predict the travel time for the optimization algorithm. For the plant model, we use the trip-based MFD, and for the prediction model, we use the mean speed. Accordingly, we can assess the impact of network traffic on the dynamic ride-sharing system performance.

Then we showed the performance of the proposed simulation-based optimization frameworks. We assessed the performance of all the heuristic methods and demonstrated that our heuristic approaches significantly improve computation time with few compromises on optimality. Also, the proposed optimization framework can outperform the previous methods in the literature. In the next chapter, we will evaluate the effect of a dynamic ride-sharing system on medium-scale and large-scale networks.

# Chapter 6

## Transportation Analyses

### Introduction

In chapter 5, we presented the simulation-based optimization frameworks for the dynamic ride-sharing problem that enable us to assess the main goal of this dissertation. In this chapter, we evaluate the impact of the ride-sharing system on network traffic, and we see how it can affect congestion.

We aim to assess the impact of dynamic ride-sharing on both medium-scale and large-scale networks. In megacities and large-scale networks, the performance of ride-sharing in terms of reducing congestion can be different. However, these services are increasingly popular in large cities and small and medium-sized cities. First, we assess the impact of ride-sharing on the network of Lyon 6 + Villeurbanne as a medium-sized network in section 6.1. Then we present the analyses in large-scale in section 6.2 by applying the method on the network of the whole Lyon city in France.

### 6.1 Analysis in medium-scale

To assess the influence of the dynamic ride-sharing system on reducing traffic congestion, we compare the traffic condition for the dynamic ride-sharing system considering different market-shares and the numbers of sharing with the case where the market-share is zero when only personal vehicles serve all the network demand ("No service scenario" in the



figures). We perform sensitivity analysis on the fixed length of the time window, the rolling horizon, and the depot size. The simulation's configuration for medium-scale analysis is explained in section 5.3.1. Table 6.1 shows the simulation results for different market-shares and numbers of sharing. We discuss the results in detail in the next sections.

### 6.1.1 Market-share

We use the vehicle accumulation in the network as a measure of traffic congestion. We compute the vehicle accumulation in the network every second in the simulations. Figure 6.1 shows the vehicle accumulation in the network every 100 seconds for different market-shares when the number of sharing is 0. The service vehicles must travel a certain distance from the depot to the first origin and then from the last destination to the depot. This extra distance makes the car stay longer in the network and leads to more traffic. Hence, when the market-share increases, the accumulation of vehicles increases.

In table 6.1, the travel time for the personal vehicles when the market-share is zero is 7,978.4 hours. Then, with a market-share of 20%, the total travel time for shared vehicles is 251.7 hours, and the total personal vehicle travel time is 7,816.6 hours. Therefore, the total travel time for all the vehicles in the network is 1.13 % higher than the total travel time when there is no service vehicle in the network. Increasing the market-share will increase this extra travel time by 2.27 % , 3.33 % , 4.37 % and 5.50 % for market-shares of 40%, 60%, 80% and 100% .

As shown in table 6.1, sharing decreases the travel distance and the travel time for service vehicles. Hence, the number of sharing 1 can reduce the accumulation of cars driving in the network. Figure 6.2 shows the accumulation of all the vehicles in the network when the number of sharing is 1 for different market-shares. The results show that sharing can reduce traffic congestion for a given market-share. The total travel time for shared vehicles when the market-share is 80% is 1,018.6 hours, which is for 515 vehicles that make 8,963 trips to serve 8,978 requests when there is no sharing. But it falls to 852.4 hours with 404 vehicles via 4,880 trips for the same number of requests when the number of sharing is 1.

Table 6.1: Simulations results

Configuration	Shared vehicles				Passengers		Personal vehicles		All vehicles	
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	$n$	$\sum_{i \in N} TT_i(h)$	$\sum_{i \in N} WT_i(h)$	Travel time (h)	Total travel time (h)	Total travel distance (km)
MS : 0%	-	-	-	-	-	-	-	7978.4	7978.4	268481.0
MS : 20%										
nshare = 0	2235	158	251.7	7893.3	2236	204.0	7.5	7816.6	8068.3	269698.3
nshare = 1	1239	116	209.3	6676.8	2236	218.7	39.0	7797.1	8006.4	268481.8
MS : 40%										
nshare = 0	4482	307	501.7	15452.6	4482	415.7	9.9	7657.8	8159.5	270750.6
nshare = 1	2455	218	419.3	13235.8	4482	450.7	71.9	7603.5	8022.8	268533.8
MS : 60%										
nshare = 0	6731	422	754.7	23016.0	6732	632.2	11.1	7489.6	8244.4	271770.0
nshare = 1	3672	303	632.3	19779.8	6732	687.1	104.0	7417.9	8050.2	268533.8
MS : 80%										
nshare = 0	8963	515	1018.6	30867.6	8978	856.3	12.5	7308.3	8326.9	272887.6
nshare = 1	4880	404	852.4	26513.9	8978	937.5	133.5	7220.6	8073.0	268533.9
MS : 100%										
nshare = 0	11213	578	1276.3	38434.0	11235	1078.6	12.6	7140.6	8416.9	273891.0
nshare = 1	6072	477	1074.1	33099.4	11235	1183.7	168.6	7032.6	8106.8	268556.4
nshare = 2	4993	474	1049.9	32605.2	11235	1283.0	233.3	7021.4	8071.3	268062.2

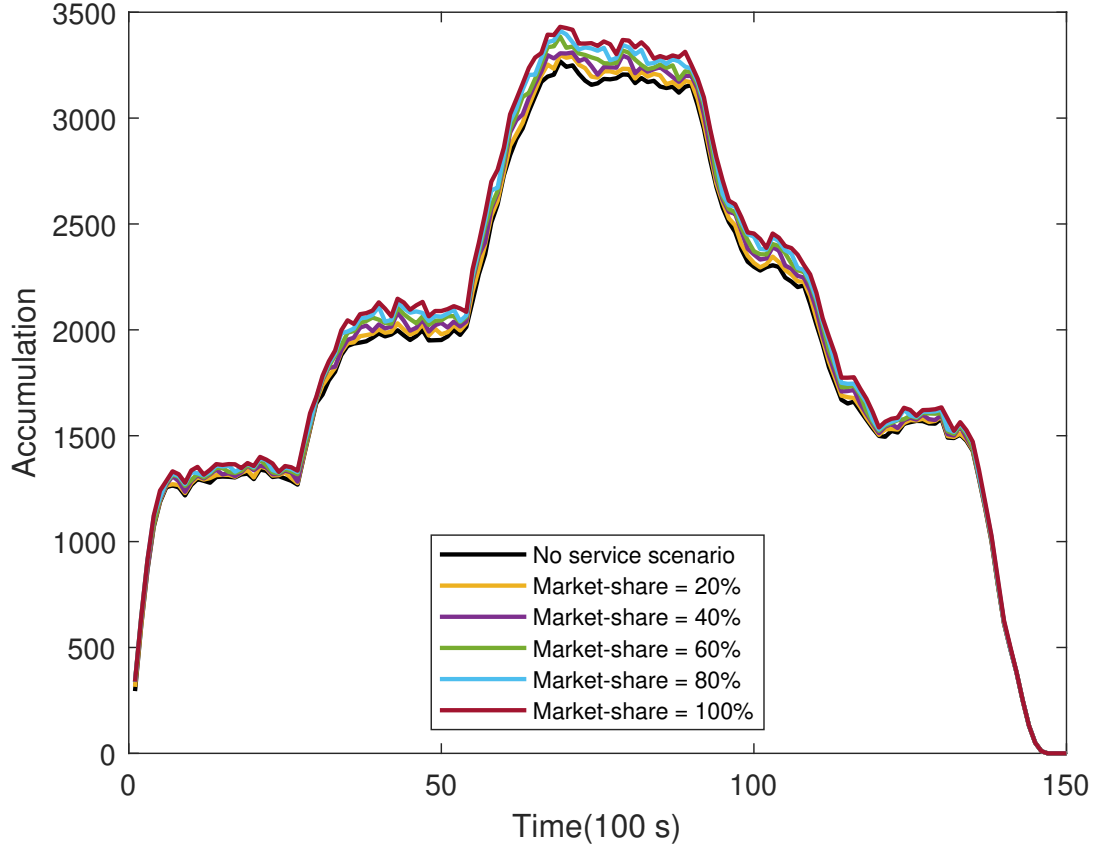


Figure 6.1: Traffic situation for the number of sharing 0 with different market-shares (medium-scale)

However, sharing cannot improve the traffic situation significantly compared to the case when all the trips are made with personal cars. For example, for the market-share of 20 percent and the number of sharing 1, the total travel time for all the vehicles in the network is 8,006.4 hours, which is 0.77% better than the number of sharing 0 but still 0.35% worse than the no service scenario. The total travel time for all the vehicles in the network is 0.56%, 0.90%, 1.19% and 1.61% longer than the no service scenario for market-shares of 40%, 60%, 80%, and 100%.

Figures 6.3, 6.4 show the total travel time and the total travel distance for all the vehicles in the network for the number of sharing 0 and 1 with different market-shares. It is clear that increasing the market-share increases the total travel time and distance for the number of sharing 0 when each passenger is served individually but then, when sharing the trip of just two passengers with the number of sharing 1, the slope of this increasing trend flattens considerably.

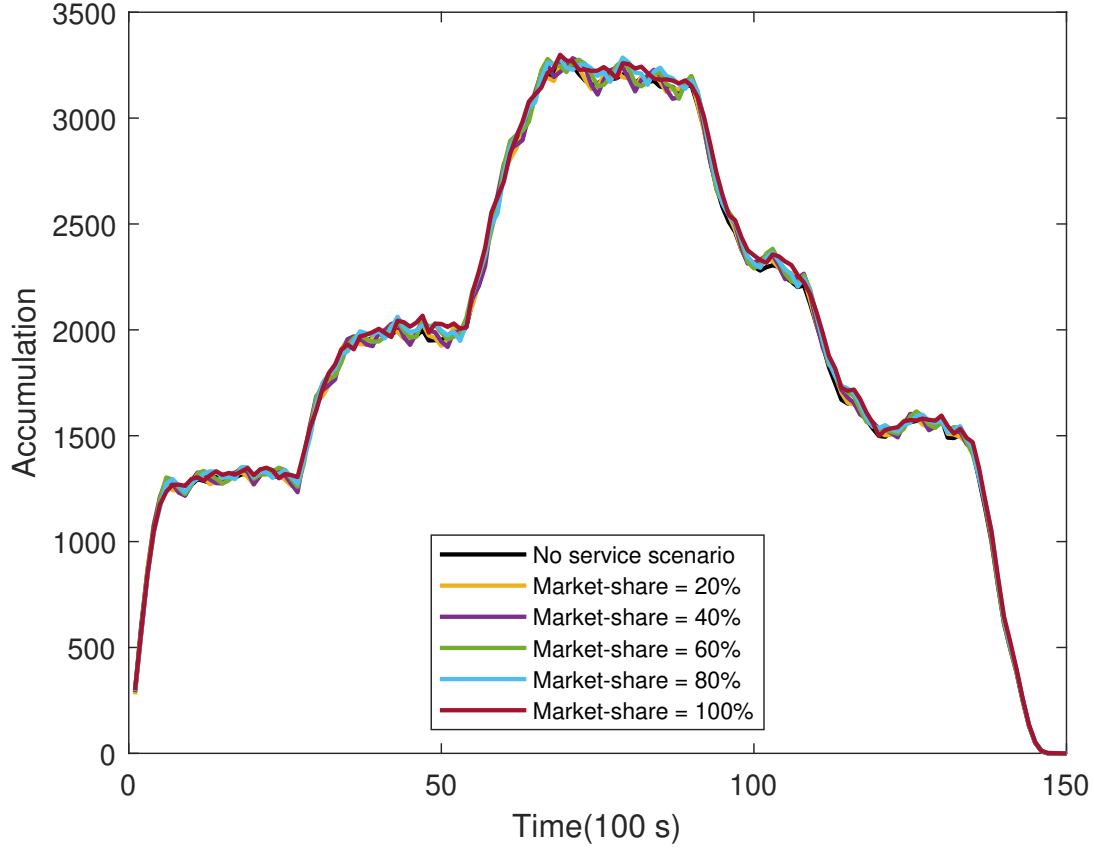


Figure 6.2: Traffic situation for the number of sharing 1 with different market-shares (medium-scale)

To consider the passengers' willingness to share the ride and their satisfaction, we optimize the passengers' waiting time and travel time in addition to the vehicle objectives. Thus, increasing the market-share cannot increase the passengers' objectives so that this leads to their dissatisfaction. As we place a strict constraint on the passenger pickup and delivery time window, the average waiting time for each passenger is not more than 63 seconds. For the market-share of 20% the average waiting time for each passenger is 62.8 seconds when the number of sharing is 1, and it is 54.0 seconds for the market-share of 100%. Also, the average travel time for passengers is 5.9 minutes when the market-share is 20% and the number of sharing is 1, and it increases by 24 seconds for the market-share of 100%.

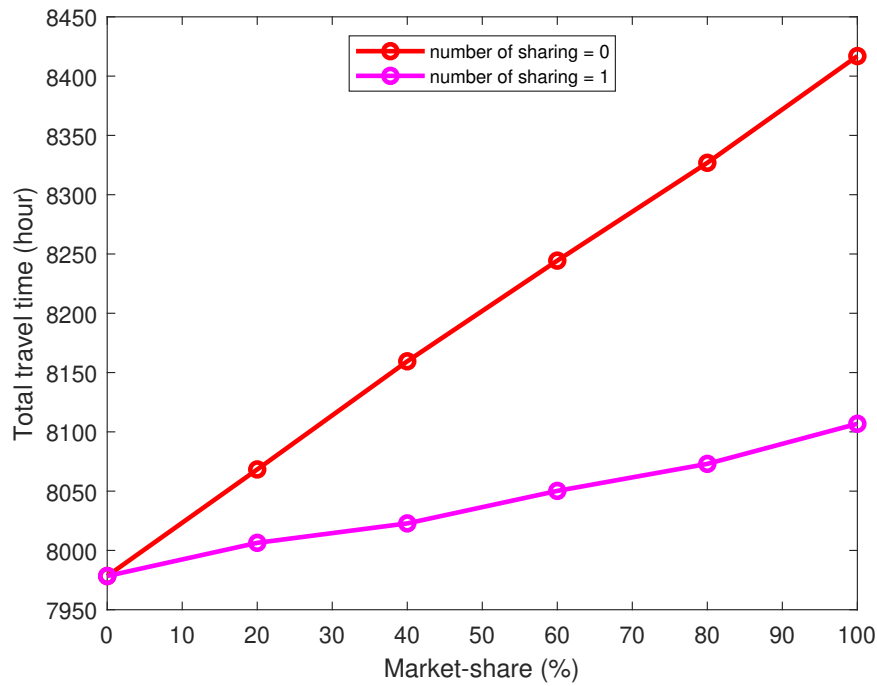


Figure 6.3: Total travel time for all the cars for the number of sharing 0 and 1 with different market-shares (medium-scale)

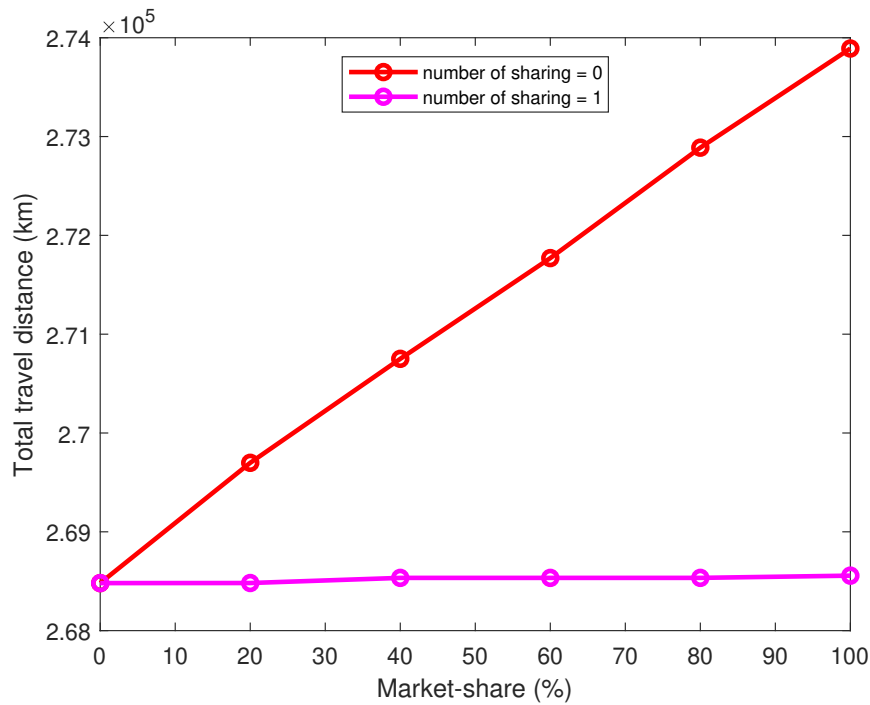


Figure 6.4: Total travel distance for all the cars for the number of sharing 0 and 1 with different market-shares (medium-scale)

### 6.1.2 Number of sharing

Increasing the number of sharing provides the system with greater leeway to decrease the travel distance by reducing the distance between stop points and depots. So, with more

sharing, we expect a better traffic situation and fewer vehicles in the network.

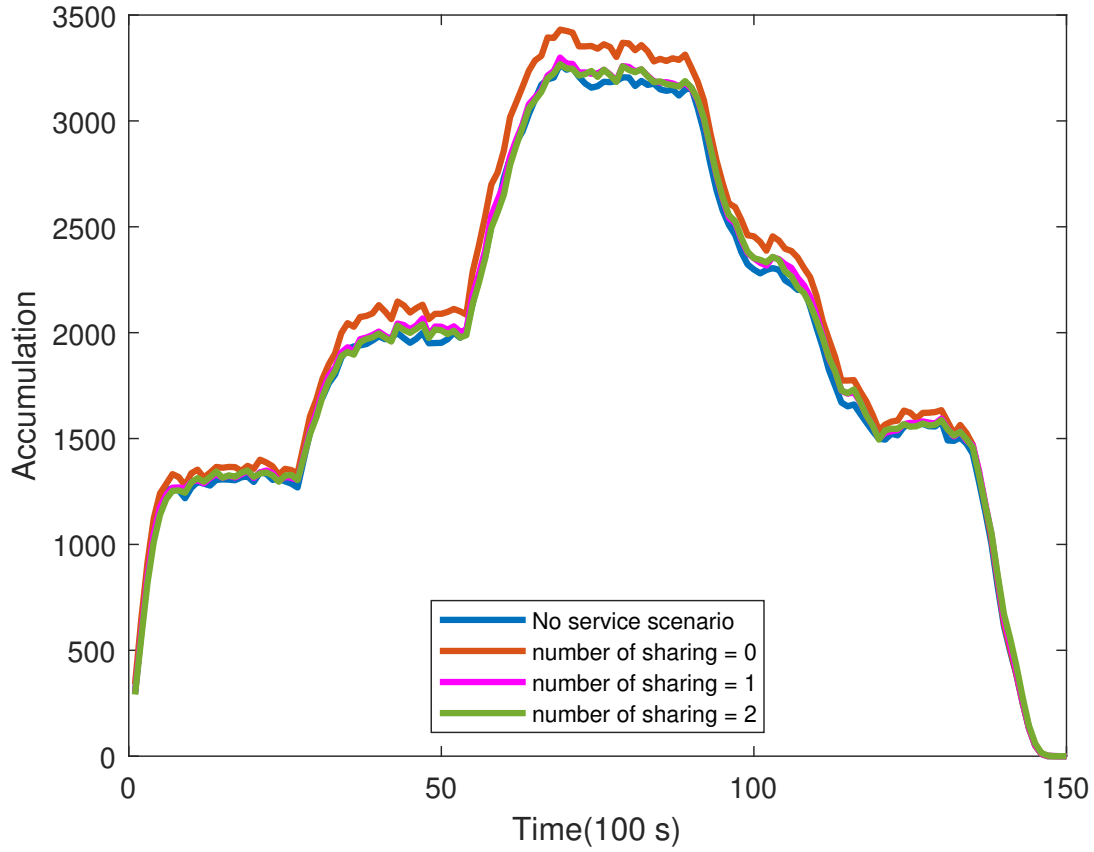


Figure 6.5: Traffic situation for market-share 100% with different numbers of sharing (medium-scale)

Figure 6.5 shows the network traffic when the market-share is 100% for the different numbers of sharing. The results show that increasing sharing can reduce congestion, but it still cannot compete with the no service scenario.

Figure 6.6 shows the average speed every hour in the system for different numbers of sharing. The vehicle speed decreases using service cars without sharing. Sharing can increase the speed, but it still cannot be higher than the speed in the no service scenario. At the onset of congestion, with the number of sharing 2, the speed is  $35.5 \text{ km/h}$  which is 0.16% higher than the number of sharing 1.

Figure 6.7 shows the accumulation differences with the baseline in peak hours. It is clear that sharing can improve congestion compared with the number of sharing 0 (systems like traditional taxis), but it is not better than the no service scenario. At the onset of congestion, sharing can prevail over the no service scenario, especially when the number

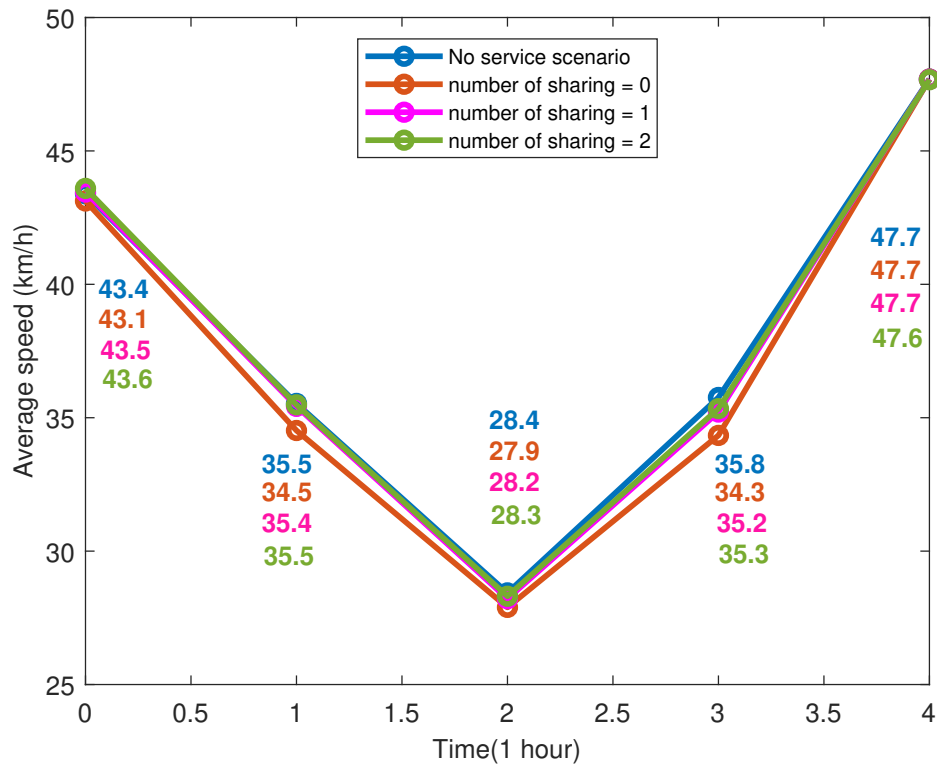


Figure 6.6: Average vehicle speed (medium-scale)

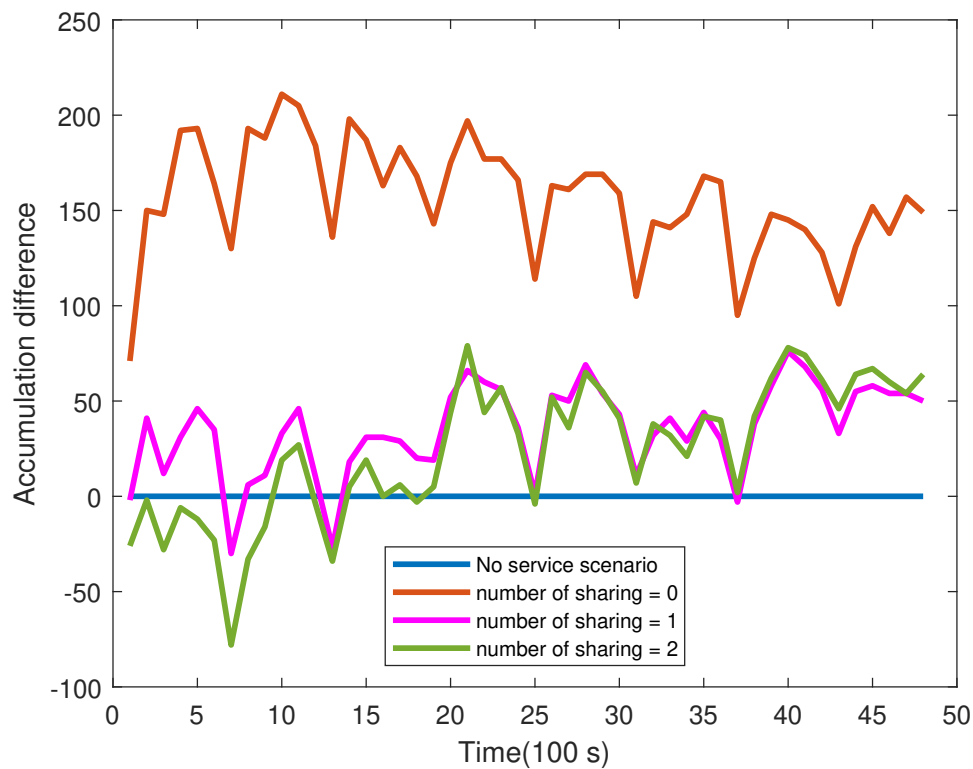


Figure 6.7: Traffic situation at peak hours (medium-scale)

of sharing is 2. Then, as congestion subsides, the sharing scenarios is better than the no service scenario. Finally, the number of sharing 2 is better than when the number of sharing is 1. This is because the vehicles have to travel a longer distance after the peak hour as they have more requests to serve.

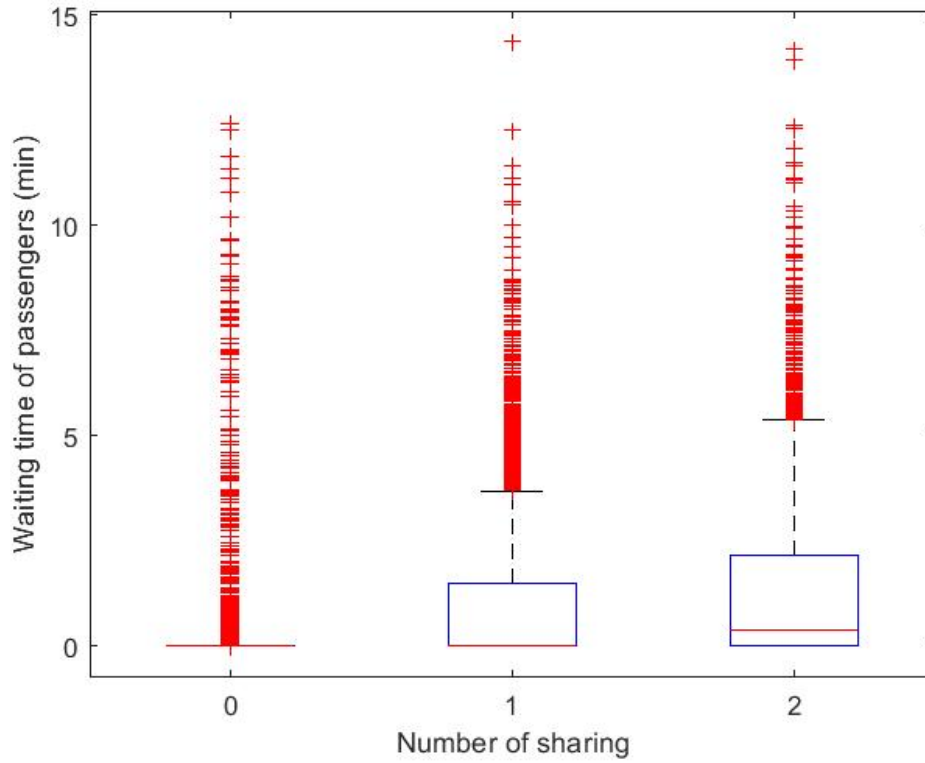


Figure 6.8: Passengers' waiting time for different sharing scenarios when the market-share is 100% (medium-scale)

Increasing the number of sharing will increase the passengers' travel time and waiting time. The average waiting time for the number of sharing at 0 (when there is no sharing) is 4 seconds. It increases to 54 seconds when the number of sharing is one and 74 seconds when the number of sharing is two. This increase is acceptable considering the last heuristic used to force the sharing. This means that the passenger must wait no more than 1 minute to be picked up at the origin when the number of sharing is 1. In the first heuristic, sometimes the algorithm can find a better assignment for the vehicles that are waiting in the depot, and it changes the first schedule. It can increase the passengers' waiting time. However, the waiting time will not be more than 10 minutes. Figure 6.8 shows the variation of passengers' waiting time for different numbers of sharing when the



market-share is 100%. For the number of sharing 0, the median for the waiting time is 0, and 50% of passengers depart at their defined pick up time. For the number of sharing one, the median of waiting time is still 0, and 50% of the passengers can start their trip at their desired time. The 75th percentile is 108 seconds, and the upper adjacent is 4 minutes. For the number of sharing 2, the median increases to 22 seconds, and the upper adjacent is 5 minutes. In section 5.3.7, we showed that the heuristic method gives an error of 3.67% comparing with the optimal solution. The total travel time for all the vehicles in the system is 7978.4 hours in no service scenario. We can estimate the optimal situation considering this error. If we reduce this error from the final results, the total travel time for the number of sharing 1 and market-share = 100% can be reduced to 8068.0, which is still more than the no service scenario. The estimation of the heuristic error when the number of sharing is 2 is 4.66%. The total travel time for the vehicles when the number of sharing is 2 can be around 8022.4 hours in the optimal situation, and it is still more than the no service scenario.

In the next sections, we analyze the time window, rolling horizon, and depot size in the medium-scale network (Lyon 6 + Villeurbanne). We also implement the method with an operational setting on this network to compare it with our proposed system.

### 6.1.3 Time window

In section 5.3.1, we explained that the time window for each request has a fixed-length, and we set this length equal to 1 minute in the simulations. In this section, we assess the impact of the time window on the ride-sharing system performance. Increasing the time window leads to longer trips as the passengers can wait more for the service.

In the current configuration  $Fixed_{TW}$  is equal to 1 minute. We do the simulations for  $Fixed_{TW} = 6$  and  $Fixed_{TW} = 10$  to see the effect of increasing the time window.

Table 6.2 shows the results for different time window lengths when the market-share is 100%, and the number of sharing is one. When the  $Fixed_{TW}$  is 1 minute, the system serves the requests with 6,072 trips. When we increase the  $Fixed_{TW}$  to 6 minutes, the number of trips decreases to 4913, and when the  $Fixed_{TW}$  is 10 minutes, the system just

Table 6.2: Simulation results for different time window lengths

Configuration	Shared vehicles			Passengers	
	$N_{trips}$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	$\sum_{i \in N} WT_i(h)$	$Average_{WT_i}(min)$
MS : 100% nshare = 1					
$Fixed_{TW} = 1$	6072	1074.1	33099.4	168.6	0.9
$Fixed_{TW} = 6$	4913	1128.5	34959.3	448.1	2.4
$Fixed_{TW} = 10$	3960	1173.3	36838.2	839.5	4.5

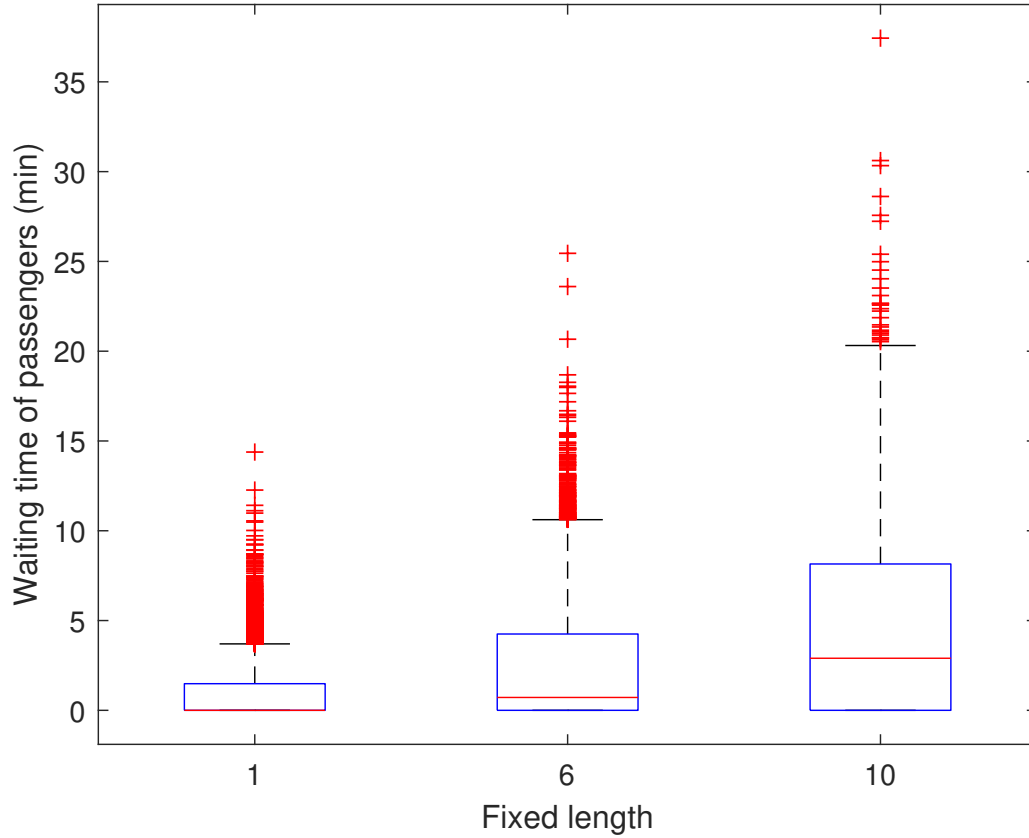


Figure 6.9: Passengers' waiting time for different time window (market-share = 100% , nshare = 1)

makes 3960 trips to serve all the requests. On the other side, increasing the  $Fixed_{TW}$  increases the travel distance and travel time of the vehicles. The total travel distance increases by 5.6% for  $Fixed_{TW} = 6$  and 11.3% for  $Fixed_{TW} = 10$ . Increasing the time window will also increase the passengers' waiting time. Figure 6.9 shows the box-plot for the passengers' waiting time. The average waiting time is 0.9 minute for  $Fixed_{TW} = 1$ . It increases to 2.4 minutes and 4.5 minutes for  $Fixed_{TW} = 6$  and  $Fixed_{TW} = 10$ . When the fixed-length is 10 minutes, some passengers wait more than 20 minutes to get the service

(it happens for the trips that are longer than 10 kilometers). Finally, we can see that even increasing the time window, which should favor the sharing, cannot reduce congestion in Lyon 6 + Villeurbanne's network.

### 6.1.4 Rolling horizon

In section 5.1.2, we explained that we stop the simulations halfway on the rolling horizon ( $\frac{TH}{2}$ ) and solve a new fleet allocation problem over a new full rolling horizon ( $TH$ ). This prevents the system from being myopic to the new demand that may arrive just after the end of a simulation period.

In the current simulations' configuration,  $TH$  is 20 minutes. It means that every 10 minutes, the system optimizes the requests over the next 20 minutes ((10,20) in table 6.3). In this section, we assess the impact of rolling horizon and optimization time step on the system performance. We do the simulations for (5,10) and (20,40) to evaluate the system when we have a shorter and a longer rolling horizon.

Table 6.3: Simulation results for different rolling horizon scenarios

Configuration	Shared vehicles				Personal vehicles
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	Travel time (h)
MS : 100% nshare = 1					
(5,10)	6142	479	1074.2	33325.8	7035.1
(10,20)	6072	477	1074.1	33099.4	7032.6
(20,40)	6049	472	1074.0	33005.7	7030.7

Table 6.3 shows the results for different rolling horizon. Increasing the rolling horizon can increase sharing and reduce the number of trips. Nevertheless, the difference is not very big. With  $TH = 10$  minutes, the system serves the requests with 6142 trips. This number decreases to 6072 for  $TH = 20$  and 6049 for  $TH = 40$ . The total travel distance is reduced by 0.7% when we increase the rolling horizon from 10 to 20 minutes, and it is reduced by 0.3% when we increase the  $TH$  from 20 minutes to 40 minutes. When we increase  $TH$  from 10 to 20, the system can find more sharing opportunities, but when we increase it to 40 minutes, as we have time window constraint for the trips, it is not possible to have much more sharing than 20 minutes.

### 6.1.5 Depot size

Increasing the number of allowed stop locations in the network can increase the accessibility of cars to the closest passengers and reduce the passengers waiting time. It can also decrease the travel distance between the stop location and the first origin. To assess the impact of the number of local depots and the number of vehicles waiting in these locations, we define two different scenarios. We compare them with the basic scenario where we have 237 depots and 142 cars waiting in these depots. The first scenario increases the number of vehicles and puts 1780 vehicles in the stop locations. In the second scenario, we increase the number of depots to 1067 and distribute 1780 vehicles, considering the geographical demand pattern on these locations.

Table 6.4: Simulation results for different depot management scenarios

Configuration	Shared vehicles				Passengers
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	$\sum_{i \in N} WT_i(h)$
MS : 100% nshare = 1					
Basic scenario	6072	477	1074.1	33099.4	54.04
237 depots and 1780 vehicles	6055	475	1068.2	32647.2	50.79
1067 depots and 1780 vehicles	6065	446	986.4	30118.0	52.01

Table 6.4 shows the results for different scenarios. The second scenario can improve the total travel distance by 1.3% compared to the basic scenario recall that we increased the number of vehicles by 1153%. In the third case, with 350% increase in the number of allowed waiting locations for the vehicles, the total travel distance is improved by 9%.

Figure 6.10 shows the percentage of improvement for the three sharing scenario compared to the scenario when we have just personal vehicles in the system. All the sharing scenarios decrease the congestion in peak hour but this improvement is not significant. In the best case, the basic scenario can improve the congestion by 4.1%. Increasing the number of vehicles can further improve it by 0.6% and increasing the number of stop locations by 5 times can make 2.5% improvement compared to the basic scenario.

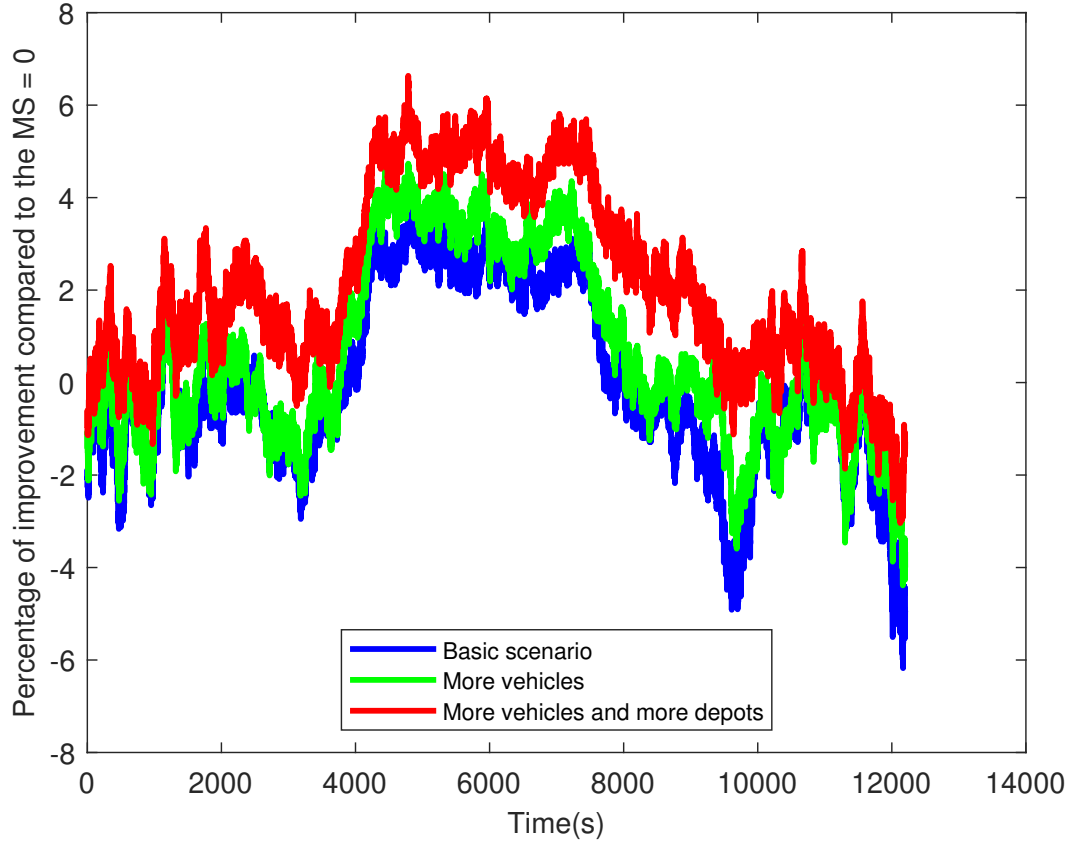


Figure 6.10: Comparing different scenarios for depot management

### 6.1.6 Operational settings

The main goal of this study is to assess the performance of dynamic ride-sharing in terms of reducing congestion. To favor the ride-sharing service, we choose a longer optimization time step comparing with other dynamic methods to guarantee that we can find near-optimal solutions for the matching problem. Hence, to solve the problem dynamically, we apply the method every 10 minutes over the requests received in the next 20 minutes, considering we have a perfect knowledge of all requests over such a time horizon. However, one can argue that today's real operational systems do not have this long horizon and that they answer service requests immediately. Hence, our conclusions would be valid only for the methods that we are using, i.e., with a rolling horizon.

To verify our conclusions in this context, we propose to mimic today's operational settings. To do so, we propose an insertion heuristic method that matches the passengers and vehicles in real-time. The chosen vehicle to insert the new request in its schedule is the

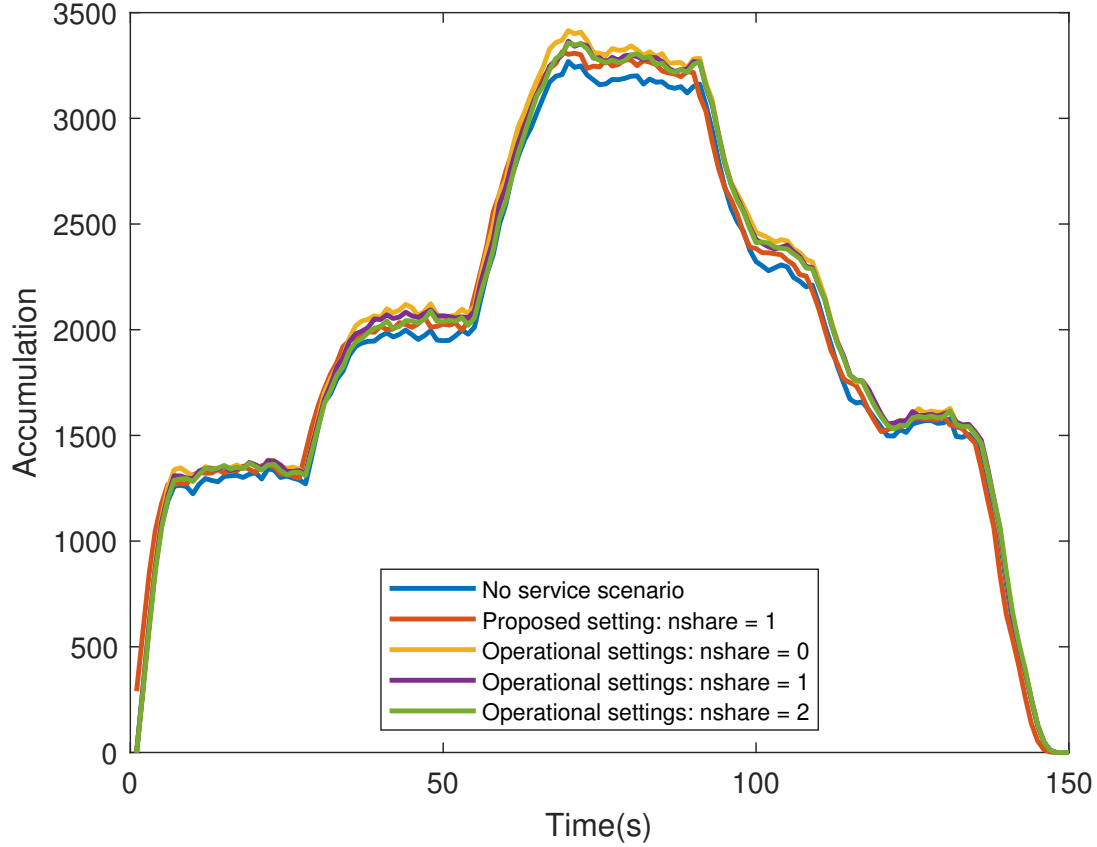


Figure 6.11: Traffic situation for market-share 100% with operational settings

one with the minimal marginal costs increase. If no existing vehicle can insert the request, a new vehicle is dynamically created, and the request is assigned to it. In this situation, the system has immediate response times and matches the received individual requests to the vehicles in real-time.

Figure 6.11 shows the network traffic when the market-share is 100% for different numbers of sharing under operational settings comparing with the no service scenario and our proposed system when the number of sharing is 1. It is clear that the operational settings perform worse than the near-optimal situation proposed in this thesis in terms of reducing congestion even with the number of sharing 2.

Table 6.5 shows the results for operational system settings compared with our proposed settings. The total travel time for service vehicles in the operational setting is 66.8 hours more than the proposed system when the number of sharing is 1. Also, the total travel time for personal vehicles in the network is 47.8 hours more in operational settings. It is

clear that the proposed settings outperform the operational settings.

Table 6.5: Simulation results for operational settings

Configuration	Shared vehicles				Passengers	Personal vehicles
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	$\sum_{i \in N} WT_i(h)$	Travel time (h)
MS : 100%						
Proposed settings						
nshare = 0	11213	578	1276.3	38434.0	12.6	7140.6
nshare = 1	6072	477	1074.1	33099.4	168.6	7032.6
nshare = 2	4993	474	1049.9	32605.2	233.3	7021.4
Operational settings						
nshare = 0	11215	578	1277.9	39559.2	12.5	7141.6
nshare = 1	8243	510	1140.9	36875.9	206.4	7080.4
nshare = 2	7864	515	1140.6	36904.2	240.2	7076.3

## 6.2 Analysis in large-scale

In this section, we assess the impact of dynamic ride-sharing on traffic congestion in a large-scale network, using the proposed simulation-based optimization framework. The Lyon network area is 220% larger than the medium-scale network, and the number of trips is 676% more. Also, there are longer trips in the large-scale test case.

In the previous section, we observed that dynamic ride-sharing can reduce the travel time and distance and improve the traffic congestion compared to the scenario when we use taxis in the system, and we do not share the trips. However, it can not overcome the basic scenario where all the trips are made with personal cars. The impact of ride-sharing would be different in large-scale as the number of trips, and the trip length is much more than in the medium-scale, and the system has more opportunity to match the shareable trips.

### 6.2.1 Simulations' configuration in large-scale

In the rolling horizon approach, every 10 minutes, the system solves the assignment problem for the requests of the next 20 minutes. The simulation time step is 1 second, so the state of the system is updated every second. The fixed-length of the time window was 6 minutes for the medium-scale, and after the FOSH method to keep the passengers'

waiting time acceptable, we reduced it to 1 minute. In large-scale as the trips are longer than in medium-scale, we set the  $Fixed_{TW}$  equal to 6 minutes. We do the simulations for the number of sharing 0,1 and 2 and the market-share from 0 to 100%. The capacity of service vehicles is 4. We have defined 2,263 allowed stop locations for the service vehicles in the system. The number of vehicles waiting on these local depots at the beginning of the simulations is 22630. In addition to the local depots, there are nine central depots in the network that can feed the local depots. As the main goal is to assess ride-sharing performance on congestion, we put 9,000 service vehicles in the central depots to be sure that the system has enough service cars. This is equivalent to not planning an upper bound on the number of available vehicles. We have considered 1 minute as service time for passengers pick up and drop off in the simulations.

### 6.2.2 Optimal size of clusters

In section 4.4.3, we showed that hierarchical clustering is an appropriate clustering method to cluster the requests using our proposed "Shareability function" on a large-scale. We try to have the same size clusters (to avoid too big or too small clusters) to be able to keep the computation time low and have the opportunity for sharing in all the clusters. In the hierarchical method, we can keep the first cluster of the desired size at the bottom of the dendrogram to have the same size clusters [1]. The crucial point here is to find an approximation for the size of clusters in the hierarchical method.

There are different methods in the literature to choose the optimal size of clusters. Here, the quality of the clusters (how similar are points within a cluster) is very important. Furthermore, we have to be sure that the clusters are separated from each others, and the possibility of sharing two trips from two different clusters is minimum. Thus the best way to find the optimal size of clusters is to use the Sum of Squares method (SS)[103]. It is a clustering validation method that chooses the optimal size of clusters by minimizing the Within-cluster Sum of Squares (WSS) (a measure of how tight each cluster is) and maximizing the Between-cluster Sum of Squares (BSS) (a measure of how separated each cluster is from the others). We compute the WSS and BSS for all the clusters in different



periods to evaluate the optimal size of clusters in different demand situations. We analyze the size of clusters from 75 to 300.

Table 6.6: Sum of Squares method

Number	Size of clusters
1	75
2	100
3	125
4	150
5	175
6	200
7	225
8	250
9	275
10	300

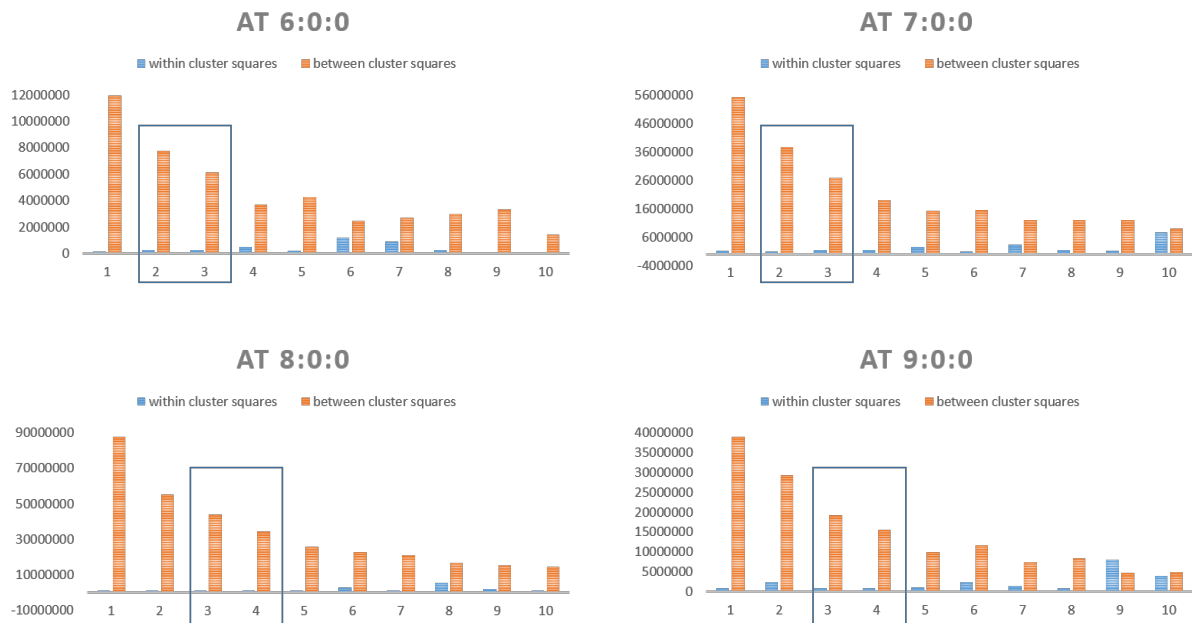


Figure 6.12: Sum of squares method for finding the optimal size of clusters

Figure 6.12 shows the SS method at different times of the simulations. Increasing the size of clusters decreases the BSS. It means that more number of clusters can ensure that the clusters are separate from each others. We have determined the cluster sizes that minimize the WSS and maximize the BSS. At 6 and 7 AM, the cluster sizes 100 and 125 can make this trade-off between WSS and BSS. At 8 and 9 AM, the best cluster sizes are 125 and 150. Therefore, the best cluster size to do the simulations for this scale is 125.

Table 6.7: Simulations results

Configuration	Shared vehicles				Passengers		Personal vehicles		All vehicles	
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} T D_k(km)$	$n$	$\sum_{i \in N} T T_i(h)$	$\sum_{i \in N} W T_i(h)$	Travel time (h)	Total travel time (h)	Total travel distance (km)
MS : 0%	-	-	-	-	-	-	-	75271.1	75271.1	2290280.0
MS : 20%										
nshare = 0	41011	3667	10373.2	312739.0	41043	9439.9	265.7	65770.0	76143.2	2308249.0
nshare = 1	21285	2089	6210.1	188979.0	41043	9726.4	346.9	64850.3	71060.3	2184489.0
nshare = 2	14809	1559	4579.2	139516.0	41043	9894.5	498.4	64505.3	69084.5	2135026.0
MS : 40%										
nshare = 0	82021	7261	20811.6	625745.0	82104	19052.3	449.3	56176.4	76987.9	2325345.0
nshare = 1	42374	4138	12377.0	380495.0	82104	19406.1	661.7	54648.1	67025.0	2080095.0
nshare = 2	29236	3034	9091.8	281067.0	82104	19763.1	974.8	54088.3	63180.1	1980667.0
MS : 60%										
nshare = 0	123053	10632	31309.2	939129.0	123166	28740.8	615.7	46520.6	77829.7	2342399.0
nshare = 1	63566	6067	18456.4	573197.0	123166	28921.7	970.5	44670.3	63126.6	1976467.0
nshare = 2	43610	4368	13459.4	422194.0	123166	29316.1	1476.1	44007.2	57466.6	1825464.0
MS : 80%										
nshare = 0	164079	13740	41708.9	1248210.0	164227	38358.1	765.0	36951.7	78660.6	2359100.0
nshare = 1	84686	7724	24309.3	762330.0	164227	38110.8	1269.0	35037.5	59346.8	1873220.0
nshare = 2	57968	5597	17745.97	564108.0	164227	38577.8	1990.3	34383.1	52128.9	1674998.0
MS : 100%										
nshare = 0	205124	17102	52208.6	1558900.0	205308	48068.1	913.2	27296.3	79504.9	2375940.0
nshare = 1	105745	9489	30075.0	952139.0	205308	47182.2	1564.4	25557.1	55632.1	1769179.0
nshare = 2	72160	6826	21856.8	703947.0	205308	47596.1	2519.0	24982.4	46839.3	1520987.0
nshare = 3	69790	6595	19731.3	688755.0	205308	49595.8	2812.4	24751.2	44482.5	1505795.0

### 6.2.3 Market-share

Table 6.7 shows the results for different market-shares and numbers of sharing in the network of Lyon (large-scale network).

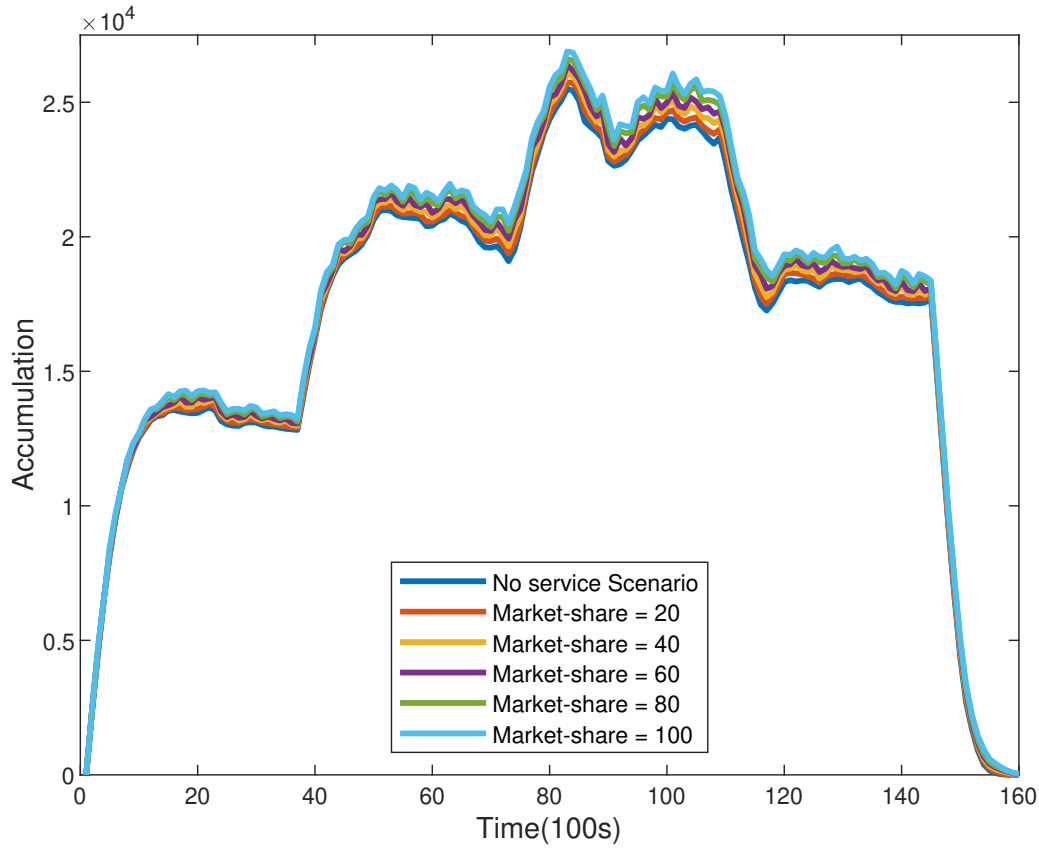


Figure 6.13: Traffic situation for the number of sharing 0 with different market-shares (large-scale)

We mentioned that we use the vehicle accumulation in the network as a measure of traffic congestion. Figure 6.13 shows the traffic situation in a large-scale from 6 AM to 10 AM, every 100 seconds for different market-shares when the number of sharing is zero compared with the no service scenario (when all the trips are made by personal cars). It is clear that increasing market-share increases the number of service vehicles in the system. For example, with market-share = 20%, the system has to serve 41,043 requests with the service vehicles. As we consider the "Sequential trips" in addition to the "Shared trips", even with the number of sharing zero, the system serves this number of requests with fewer trips. It makes 41,011 trips using 3,667 service vehicles to serve the requests.

With market-share = 40%, the number of requests increases to 82,104, and the number of required service vehicles is 7,261. Finally, with the market-share = 100%, the number of requests is 205,308. The system serves this number of requests, with 17,102 vehicles in 205,124 trips. Accordingly, the increase in the accumulation plot is not very huge. The total travel distance for all the trips increases by 0.8%, 1.5%, 2.3%, 3.0% and 3.7% for the number of sharing 20%, 40%, 60%, 80% and 100% respectively.

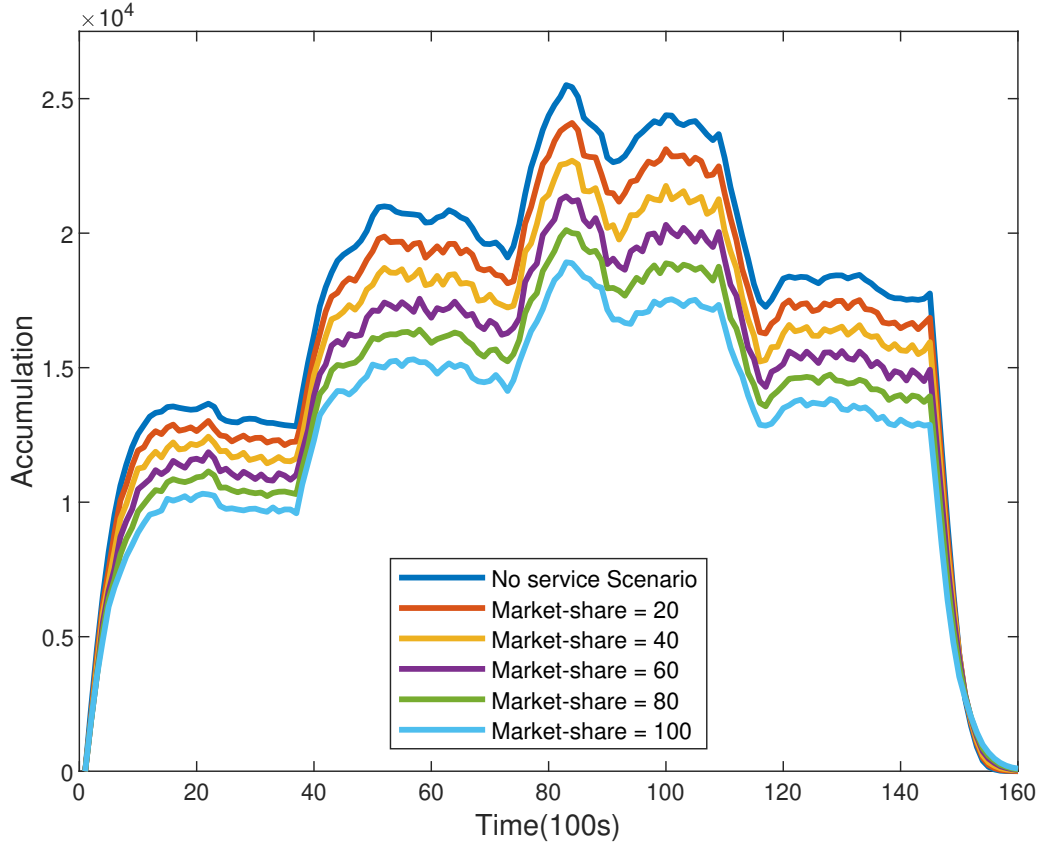


Figure 6.14: Traffic situation for the number of sharing 1 with different market-shares (large-scale)

Sharing two passengers' trip with the number of sharing 1 with our proposed system can make big progress in reducing congestion in large-scale. Figure 6.14 shows this fact. With market-share = 20%, the number of service trips decreases for 48.1%, and the system can make these trips with 1,578 fewer cars. For all the market-shares, with the number of sharing 1, the number of trips to serve the same number of requests is almost 50% less than the number of sharing 0. The system serves the requests with 3,123 fewer cars with market-share = 40%, 4,565 fewer cars with market-share = 60%, 6,016 fewer cars with

market-share = 80% and 7,613 fewer car with market-share = 100%.

Increasing the number of sharing to 2, can make a more remarkable improvement in the large-scale network traffic. Figure 6.15 shows the traffic situation in large-scale when the number of sharing is 2 for different market-shares. The number of trips is reduced by 63.9% with the market-share 20%, 64.4% with the market-share 40%, 64.6% with the market-share 60%, 64.7% with the market-share 80% and 64.8% with the market-share 100%. Also, the number of sharing 2 can decrease the needed cars for 57.5%, 58.2%, 58.9%, 59.3%, 60.1% with the market-share 20%, 40%, 60%, 80% and 100% respectively.

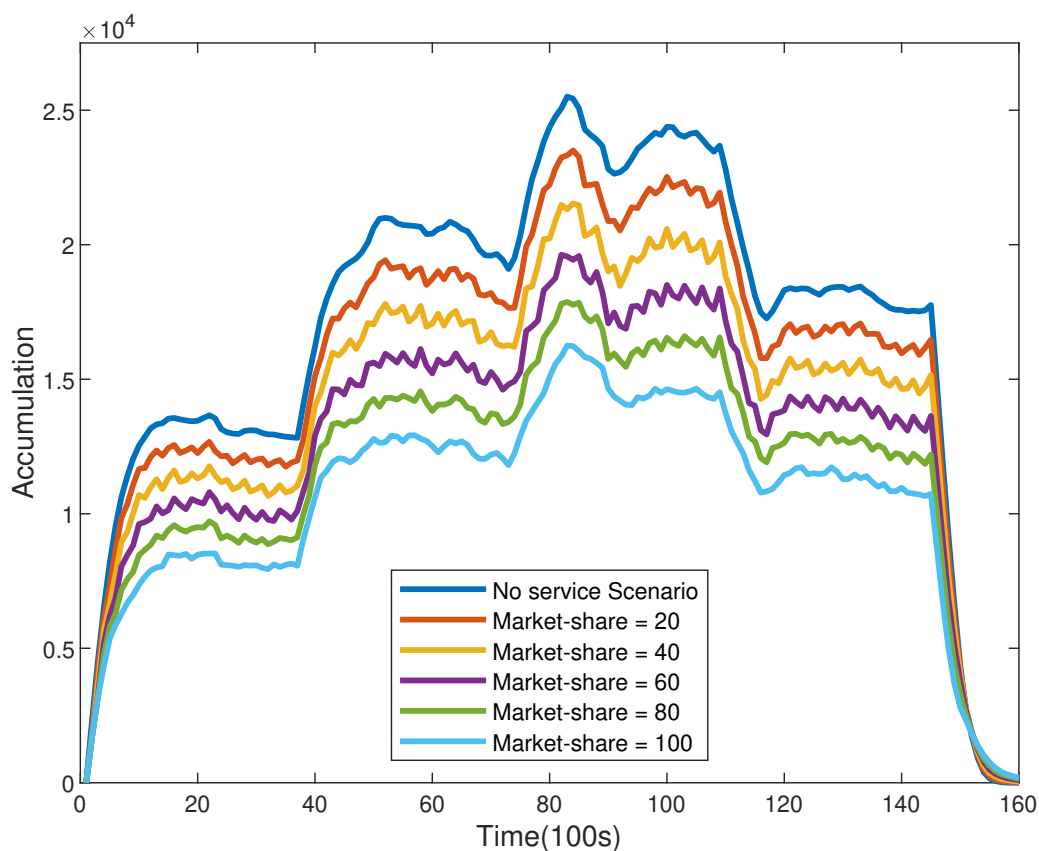


Figure 6.15: Traffic situation for the number of sharing 2 with different market-shares (large-scale)

Table 6.7 shows that sharing can significantly decrease total travel distance and total travel time of the service vehicles and total travel time of the personal vehicles. Sharing the trips can reduce the number of moving vehicles in the network and, consequently, increase the vehicles' speed. So with sharing, even personal cars can move faster and have shorter travel times.

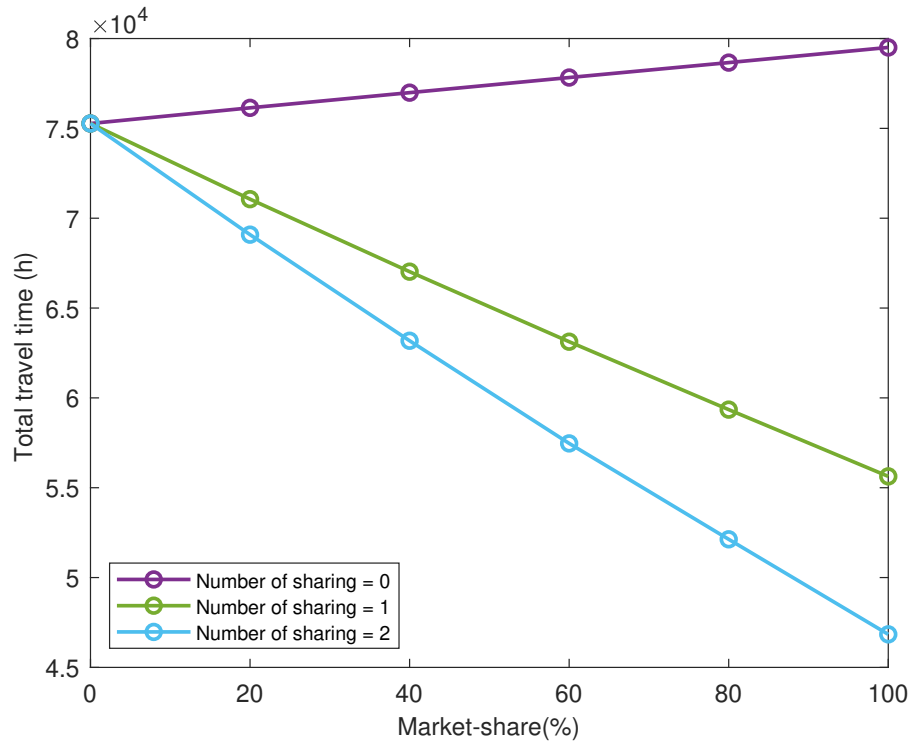


Figure 6.16: Total travel time for all the cars for the number of sharing 0, 1 and 2 with different market-shares (large-scale)

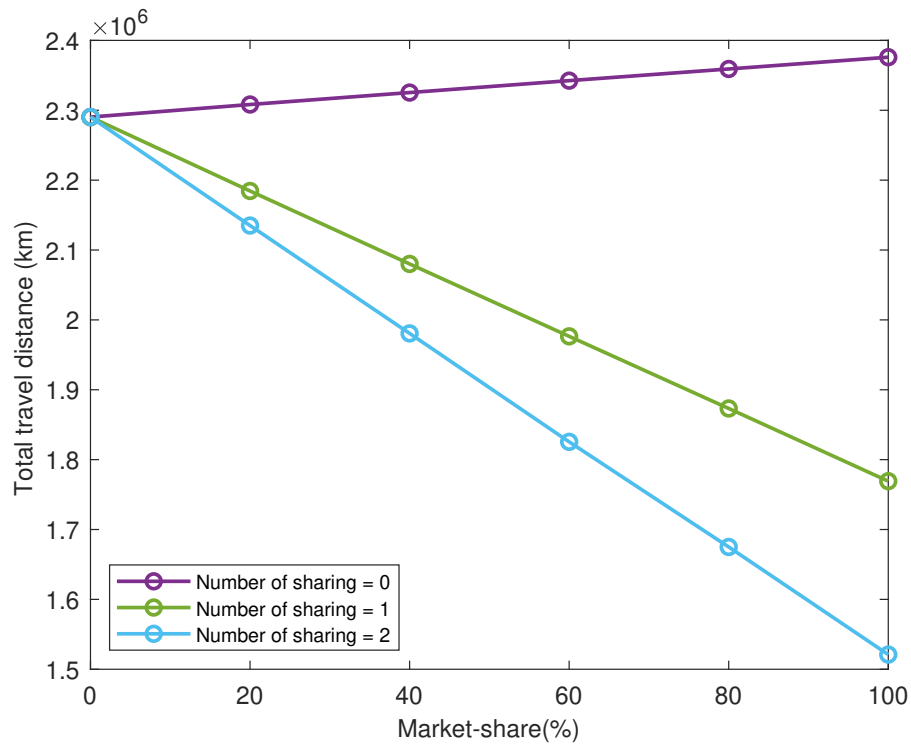


Figure 6.17: Total travel distance for all the cars for the number of sharing 0, 1 and 2 with different market-shares (large-scale)

Figure 6.16 and figure 6.17 shows the total travel time and distance for all the service and personal cars in the network for different market-shares when the number of sharing is 0, 1, and 2. It is clear that with the number of sharing zero, total travel time and distance increases with increasing the market-share. Market-share = 100% with the number of sharing zero can increase the total travel time by 5.6% and the total travel distance by 3.7%. Then, sharing can fix this problem by reducing the total travel time by 30.0% with the number of sharing 1 and 41.1% with the number of sharing 2 compared to the number of sharing 0. Furthermore, the total travel distance is reduced by 25.5% with the number of sharing 1 and 36.0% with the number of sharing 2.

#### 6.2.4 Number of sharing

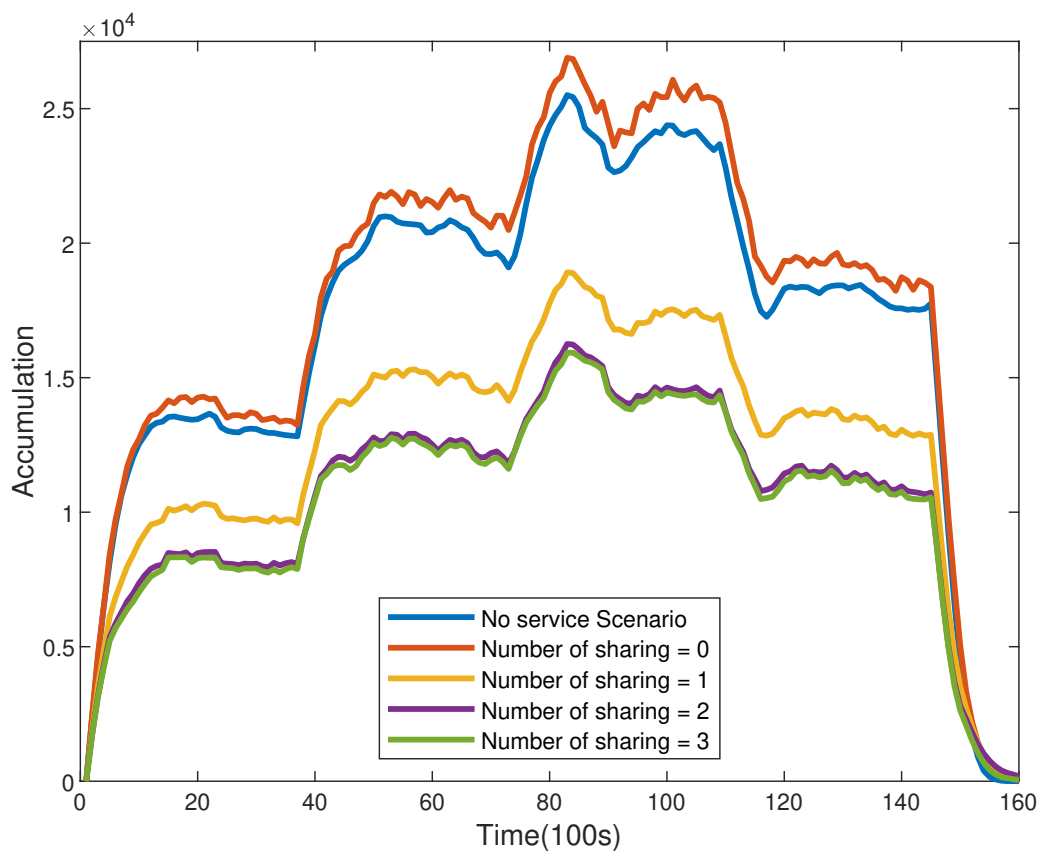


Figure 6.18: Traffic situation for market-share 100% with different numbers of sharing (large-scale)

As we mentioned, increasing the number of sharing provides the system with greater leeway to decrease the travel distance by reducing the distance between stop points and depots. Figure 6.18 shows how increasing the number of sharing can reduce congestion in large-scale (for market-share = 100%). The system serves 205,308 requests, with 205,124 trips using 17,102 vehicles when the number of sharing is zero. With the number of sharing 1, the requests are served with 105,745 trips using 9,489 vehicles. The number of sharing 2 reduces the number of trips to 72,160 using 6,826 vehicles. Finally, if we use all the car capacity and share each trip with a maximum of 3 other passengers, the system can serve the requests with 69,790 trips using 6,595 vehicles.

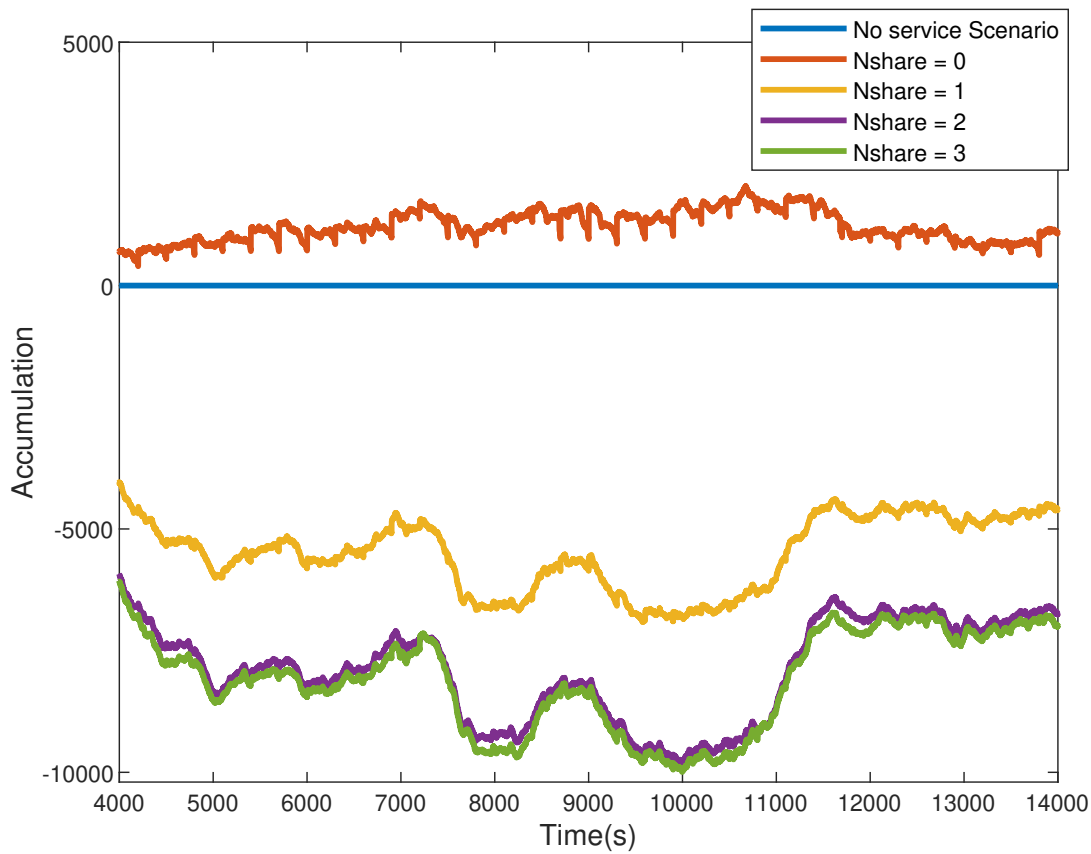


Figure 6.19: Traffic situation at peak hour (large-scale)

This reduction in the number of trips and service cars is significantly more effective in reducing congestion during peak hours. Figure 6.19 shows how our proposed system can reduce traffic in the morning peak hour. The accumulation of cars at the busiest time of the day can be reduced by 14,908 with the number of sharing 2 compared to the number



of sharing 0.

The difference between the number of sharing 1 and 2 is more significant than the difference between sharing 2 and 3. With the number of sharing 3, the vehicles have longer travel distance and remain more in the system. So the increase in the vehicles' speed with the number of sharing 3 is not comparable with the number of sharing 1 and 2. Figure 6.20 shows the average vehicle speed every hour in the network. In the peak hour, the average speed is  $29.1 \text{ km/h}$  for the "no service scenario" when we have just personal cars in the system. When we have service cars in the network without sharing, the average speed in peak hour is  $28.7 \text{ km/h}$ . The average speed increases to  $31.2 \text{ km/h}$  with the number of sharing 1,  $32.0 \text{ km/h}$  with the number of sharing 2 and  $32.1 \text{ km/h}$  with the number of sharing 3.

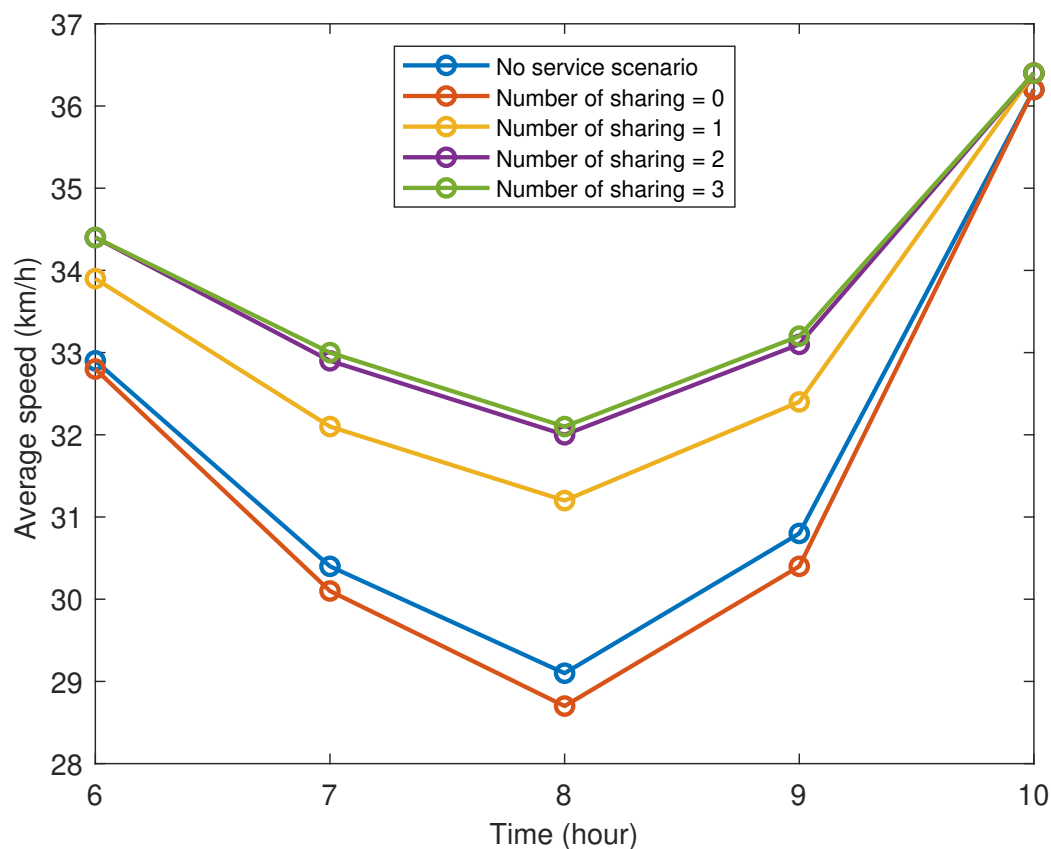


Figure 6.20: Average vehicle speed (large-scale)

Increasing the number of sharing increases the passenger waiting time, but the waiting time remains acceptable for passengers with different numbers of sharing. This is because

we are targeting to minimize the passengers' waiting time as an objective function in our proposed method. Also, in the large-scale, the accumulation of demands for the service is very high, and we have very close origin points. So the system can share these trips without deteriorating the passengers' waiting time. The average waiting time for the passengers is 16.0 seconds for the number of sharing 0. It increases to 27.4 seconds, 44.2 seconds, and 49.3 seconds for the number of sharing 1, 2, and 3.

### 6.2.5 Vehicle capacity

Ride-sharing allows travelers to share a ride and can include several ways of sharing a ride. One of these ways is van-pooling. In the van-pooling, the capacity of vehicles is higher than regular vehicles. In our evaluations, it is interesting to see the impact of different vehicles' capacity on network congestion.

In the current configurations, the capacity of cars is 4. In the previous section, we figured out the impact of ride-sharing when using the full capacity of service cars with four capacity, on the network congestion. In this section, we assess the system when we use vehicles with 6 capacity, vans with a capacity of 10 passengers and also shuttles with capacity of 20.

The goal is to use the full capacity of cars. Thus, for capacity = 4, the number of sharing is 3, for capacity = 6, the number of sharing is 5, for capacity = 10, the number of sharing is 9 and for capacity = 20, the number of sharing is 19.

In the literature, different researches on ride-sharing try to find optimal pick-up or drop off points for the passengers to be able to maximize the car occupancy [68, 32]. The passengers need to walk a distance usually from 400 meters to 1 kilometer to get to these fixed points to be picked up. In our method, to make the assignment faster and to increase the service vehicles' occupancy, we merge two trips if both the origins and destinations of the trips are close together such that the passenger needs to walk less than a determined distance. To make more comfort for the passengers and reduce the walking distance, we fix this distance to be 200 meters for capacity = 6 and 500 meters for vans and shuttles. This merging process shares two trips in a cluster. So the size of the cluster decreases

before the optimization component starts to work. Thus, we increase the size of clusters when we increase the car capacity. For capacity = 6, we increase the clusters' size to 150, for capacity = 10, we increase the size to 200 and for capacity = 20, we increase the size to 250.

Table 6.8: Simulation results for different vehicle's capacity

Configuration	Shared vehicles				Personal vehicles
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	Travel time (h)
MS : 100%					
Capacity = 4	69790	6595	19731.3	688755.0	24751.2
Capacity = 6	63304	5714	15201.5	501473.0	24576.2
Capacity = 10	46448	4253	11102.2	370030.0	24307.2
Capacity = 20	30004	2785	7253.1	244154.0	24055.2

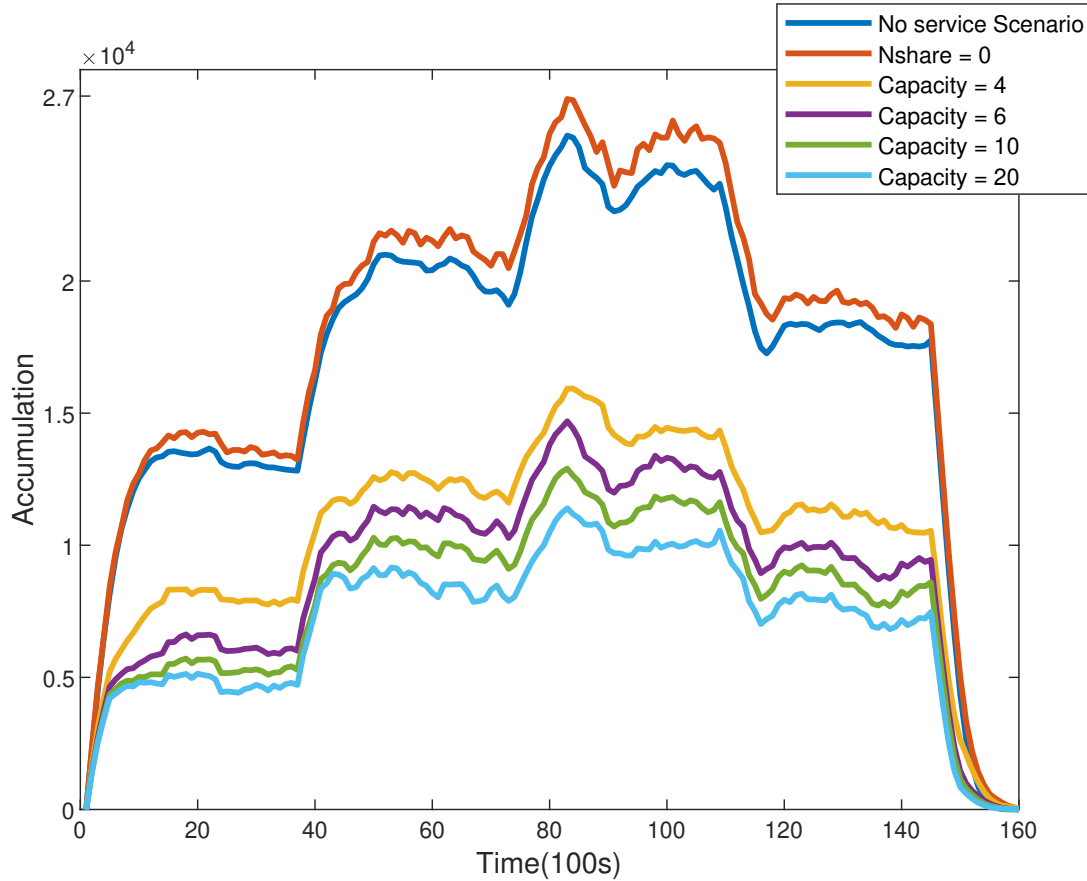


Figure 6.21: Traffic situation for different vehicle capacity (market-share = 100%)

Table 6.8 compares the results for different vehicles' capacity. Increasing the capacity of cars can significantly improve the system performance in terms of reducing the total travel

time and distance and reducing the number of trips and needed vehicles. The total travel time for service cars is reduced by 23.0% for capacity = 6, 43.7% for vans and 63.2% for shuttles. Also, the total travel time for personal cars can be reduced for 175 hours, 444 hours and 696 hours for capacity = 6, van-pooling and shuttle-sharing.

Figure 6.21 shows the traffic situation for the different vehicles' capacity compared to the no service scenario and the number of sharing zero (when we have service cars in the system without sharing). It is clear that increasing the vehicles' capacity can improve the traffic in the network. With capacity = 6, the number of trips is reduced by 9.3% compared to the scenario when the car capacity is 4. Also using vans can reduce the number of trips by 33.4% and the number of needed cars by 35.5%. Shuttle-sharing can make a more significant improvement in improving the traffic situation by reducing the number of trips by 57.0% and the number of needed vehicles by 57.8%.

### 6.2.6 Parking effect

In the previous simulations, we compared the different configurations of the sharing system with a baseline, which shows a market-share of 0. A zero market-share is the case when personal cars serve all the trips in the network. We consider that each personal trip is precisely from the origin point to the destination point in our simulations. But when people use their car for a trip, at their destinations, they have to drive randomly until a free parking place is reached to park their vehicles. During peak hours, such trip-endings increase the total number of vehicles driving and decrease the service level for other drivers ([106]). [23] based on surveys in different European cities showed that between 4.9% and 40.3% of the total travel time in a city is dedicated to searching for parking spaces. Therefore, in this section, to be more realistic, we define a new baseline with a market share 0 when we consider a specific time for parking searches. We consider 5 percent of the total travel time for parking searches and increase it to 25% in peak hours. Table 6.9, compares the results with and without considering searching for parking spaces. Searching for parking space can increase the personal cars' travel time in the "no service scenario" by 12.7%. Then using the service cars to serve a part of the trips requests

Table 6.9: Simulation results for different vehicle's capacity

Configuration	Shared vehicles				Personal vehicles
	$N_{trips}$	$m$	$\sum_{k \in M} T_k(h)$	$\sum_{k \in M} TD_k(km)$	Travel time (h)
Without Parking					
No service scenario	-	-	-	-	75271.1
Number of sharing = 0	205124	17102	52208.6	1558900.0	27296.3
Number of sharing = 1	105745	9489	30075.0	952139.0	25557.1
With Parking					
No service scenario	-	-	-	-	84835.6
Number of sharing = 0	205141	17683	52970.0	1558990.0	31021.9
Number of sharing = 1	105832	9687	30367.2	951469.0	28880.8

without sharing decreases the total travel time of all the cars by 1.0%. When we do not consider the parking, sharing can improve the total travel time by 26.1%. In the parking scenario, sharing can reduce the total travel time by 30.2%.

## Conclusion

A critical question about dynamic ride-sharing services is whether they can reduce traffic congestion. In this chapter, we implemented the proposed simulation-based optimization framework for the dynamic ride-sharing on two different scales to find the answer to this question. We performed an extensive simulation study (based on real-world traffic patterns) to assess the influence of dynamic ride-sharing systems on traffic congestion. Different situations (five different market-shares and three numbers of sharing) were investigated in terms of traffic conditions. We compared these situations with a baseline traffic situation where all the trips are served with personal cars.

In the medium-scale (Lyon 6 + Villeurbanne) contrarily to what we expected, the results showed that ride-sharing could not make a considerable improvement to the traffic situation. Ride-sharing can reduce congestion compared to traditional taxi services and dial-a-ride services. However, high levels of market-share add extra travel distance and travel time to the trips and lead to more traffic in the network. Then we observed that ride-sharing services in large cities are completely different from those in small and medium-sized cities. In large-scale (Lyon) simulations, the proposed dynamic ride-sharing

system can significantly improve traffic conditions, especially during peak hours. Increasing the market-share and the number of sharing can enhance this improvement. Also, we showed that the system could perform better with bigger cars and vans. Furthermore, we investigated the influence of personal vehicle parking space searches on the performance of the dynamic system in terms of reducing congestion. When we considered parking space searching by personal cars, we found that the dynamic ride-sharing system can make bigger improvements to traffic congestion during peak hour.

In the next chapter, we summarize the critical findings of this dissertation, we draw general conclusions and describe our future research.

## Part IV

What are the major outcomes of this  
thesis?

In previous parts, we answered two main questions about dynamic ride-sharing service by proposing simulation-based optimization frameworks for such a service. In this part, we summarize the critical points of this dissertation, and we discuss the conclusions.



# Chapter 7

## Conclusion

### 7.1 Summary and discussions

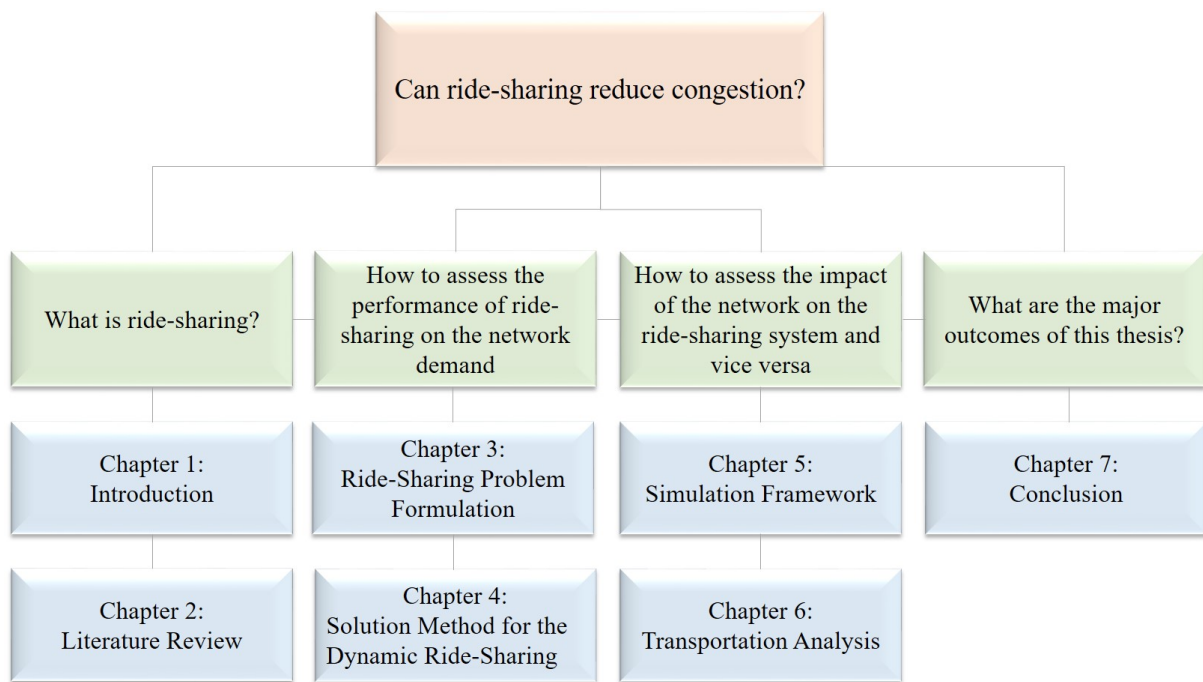


Figure 7.1: Thesis summary

The thesis has four parts. In part 1 (chapter 1 and 2) we have defined the problem and provided a literature review on ride-sharing. In part 2, we have answered the first important sub-problem of the dynamic ride-sharing problem: How to assess the performance of ride-sharing on the network demand? To answer this question, first in chapter 3, we have expressed the ride-sharing problem into a mathematical problem. Then in chapter 4, we

have presented the solution method to solve the mathematical problem. In part 3, we have answered the second sub-problem: How to assess the impact of the network traffic conditions on the ride-sharing system and vice versa? In chapter 5 we have presented a simulation framework for dynamic ride-sharing that can enable us to evaluate the impact of the network on ride-sharing. In chapter 6 using this framework, we analyze the impact of dynamic ride-sharing on both medium-scale and large-scale networks. Finally, in part 4, we present the major outcomes of the thesis. In this chapter, we summarize the thesis in section 7.1.1, 7.1.2 and 7.1.3. In section 7.2, we present the conclusions, and in section 7.3, we give an outlook for future directions that can improve the dynamic ride-sharing systems.

### 7.1.1 Dynamic ride-sharing optimization method

We define a system for dynamic ride-sharing with two main components: the fleet management component and the simulation component.

The fleet management component solves a mathematical problem to assign the vehicles to the trip requests. The mathematical model is rigorously formulated to minimize both passengers' and service providers' objectives. According to the state-of-the-art, the most important operation objective for the service provider is to minimize the total travel time and the total travel distance of vehicles. Also, the passengers need to get to the destination on time and have the minimum waiting time. The main model constraints are capacity constraints, time constraints, and assignment constraints. Also, we have considered limitations on the number of sharing, and we made sure that there is a sufficient number of vehicles in the fleet. The mathematical problem is NP-hard, and even very small instances of this problem give a number of variables and constraints to compute that encompasses the acceptable dimension of a linear programming problem. To be able to solve such a complex problem, we have presented a solution method with multiple steps that starts from finding the exact solution for small instances. Furthermore, we have introduced extensions that speed up the solution method and address bigger networks, even large-scale networks, while assessing the difference in quality at each step. We

showed that the proposed heuristics can keep the quality of solutions at an acceptable level (near-optimal solution) while significantly decreasing the computation time.

In the beginning, to find the exact optimal solution to the problem, we introduced an exact method based on the branch-and-cut concept. The method creates branches of routes and then chooses the optimal solution among the feasible routes. Then we introduce a rolling horizon method as the first heuristic step. The requests are assumed known only over the next rolling horizon (20 minutes). The corollary is that we have to introduce a new process to handle traveling cars that have not yet reached their maximal occupancy because of the car or the passenger constraints. We, therefore, introduce a specific algorithm to assign the new requests in priority to en-route vehicles. The exact algorithm handles the remaining requests.

The algorithm must try all the different arrangements of stop points to find the optimal assignment. Thus, the computation time increases exponentially as the number of requests increases. Restricting the exploration of the feasible area to the branches that are more likely to create the optimal assignment can narrow the search for feasible solutions. To overcome this limitation, in the second heuristic method, we define a clustering method to make clusters of the requests which are more likely to be shared. Then the algorithm is executed within each cluster independently. We define a "Shareability Index" based on three different trip situations (shareable and in sequence) to do the clustering. After computing the shareability function, we have the function value for each pair of requests that creates the shareability matrix. The shareability matrix is a kind of similarity matrix for the requests received that can be used in the clustering process. After computing the shareability function for all the pairs of requests, we do the clustering using the computed similarity matrix. When we make clusters based on the SF, we put the trip requests that have more potential to be shared in the same cluster. We use k-means clustering for medium-sized problems and hierarchical clustering for large-scale problems to create clusters based on the shareability matrix.

The optimizer works to minimize the objective function, which combines both passengers' and operators' objectives. Therefore, the algorithm may choose a branch with less sharing

than other feasible branches in the tree built by the algorithm. Thus, in the last step, We propose the third heuristic method to force the algorithm to favor the longest possible route, which is in favor of more sharing. The final algorithm can solve a dynamic ride-sharing problem with high quality in a short time comparing with other methods in the literature.

### 7.1.2 Dynamic ride-sharing simulation framework

The ride-sharing service's optimization system uses estimates for the predicted travel time obtained from a so-called "prediction model". When the fleet management plan is executed, a gap usually exists between the estimation and the real traffic condition. The so-called "plant model" represents the real traffic condition, and it may require dynamic adjustment of the initial assignment to fit with the conditions observed.

In the proposed simulation component of the dynamic ride-sharing system, we accurately distinguish the prediction and the plant models to provide a realistic service. The trip-based MFD is used as the plant model to consider individual trips while keeping a very simple description of traffic dynamics. To carry out travel time prediction for the optimization part, the traffic situation is predicted for the next assignment time horizon in the prediction model, and we assign the passengers to the cars based on this prediction. Accordingly, the final simulation-based optimization frameworks can simulate a dynamic ride-sharing problem's performance.

Using the simulation-based optimization frameworks, in chapter 5 we have evaluated the performance of different solution method steps, and we have demonstrated that our heuristic approaches significantly improve computation time with few compromises on optimality. Also, the proposed optimization framework outperforms the previous methods in the literature.

### 7.1.3 Impact of dynamic ride-sharing on network congestion

In this dissertation, we aimed to assess the impact of dynamic ride-sharing on both medium-scale and large-scale networks.

To assess the influence of the dynamic ride-sharing system on reducing traffic congestion, we compared the traffic condition for the dynamic ride-sharing system considering different market-shares and the numbers of sharing with the case where the market-share is zero when only personal vehicles serve all the network demand. We increased the market-share from 20% to 100%. We also did the simulations with the number of sharing 0, 1, 2, and 3. The number of sharing 0 is when each car serves only one passenger request without sharing, and the number of sharing 3 is when the system uses the full capacity of cars (when the car capacity is 4).

First, we assessed the impact of ride-sharing on the network of Lyon 6 + Villeurbanne as a medium-sized network:

Using the service cars in the system without sharing (when the number of sharing is 0), increases the total travel time and total travel distance. When the market-share is 20%, the total travel time for all the vehicles in the network is 1.13% higher than the total travel time when there is no service vehicle in the network. Increasing the market-share will increase this extra travel time by 2.27% , 3.33%, 4.37% and 5.50% for market-shares of 40%, 60%, 80% and 100%. Sharing can reduce the accumulation of cars in the network and the total travel time and distance. However, sharing cannot improve the traffic situation significantly compared to the case when all the trips are made with personal cars. For example, for the market-share of 20 percent and the number of sharing 1, the total travel time for all the vehicles in the network is 8,006.4 hours, which is 0.77% better than the number of sharing 0 but still 0.35% worse than the no service scenario. The total travel time for all the vehicles in the network is 0.56%, 0.90%, 1.19% and 1.61% longer than the no service scenario for market-shares of 40%, 60%, 80%, and 100%. Increasing the number of sharing to 2 in medium-scale can not make a big progress compared to the number of sharing 1. The speed of vehicles only increases by 0.16% fro the number of sharing 2 compared to the number of sharing 1.

As in the optimization framework, we try to minimize the passenger's travel time and waiting time, the system keeps the passenger waiting time and travel time in an acceptable level. The average waiting time for the number of sharing 0 (when there is no sharing)

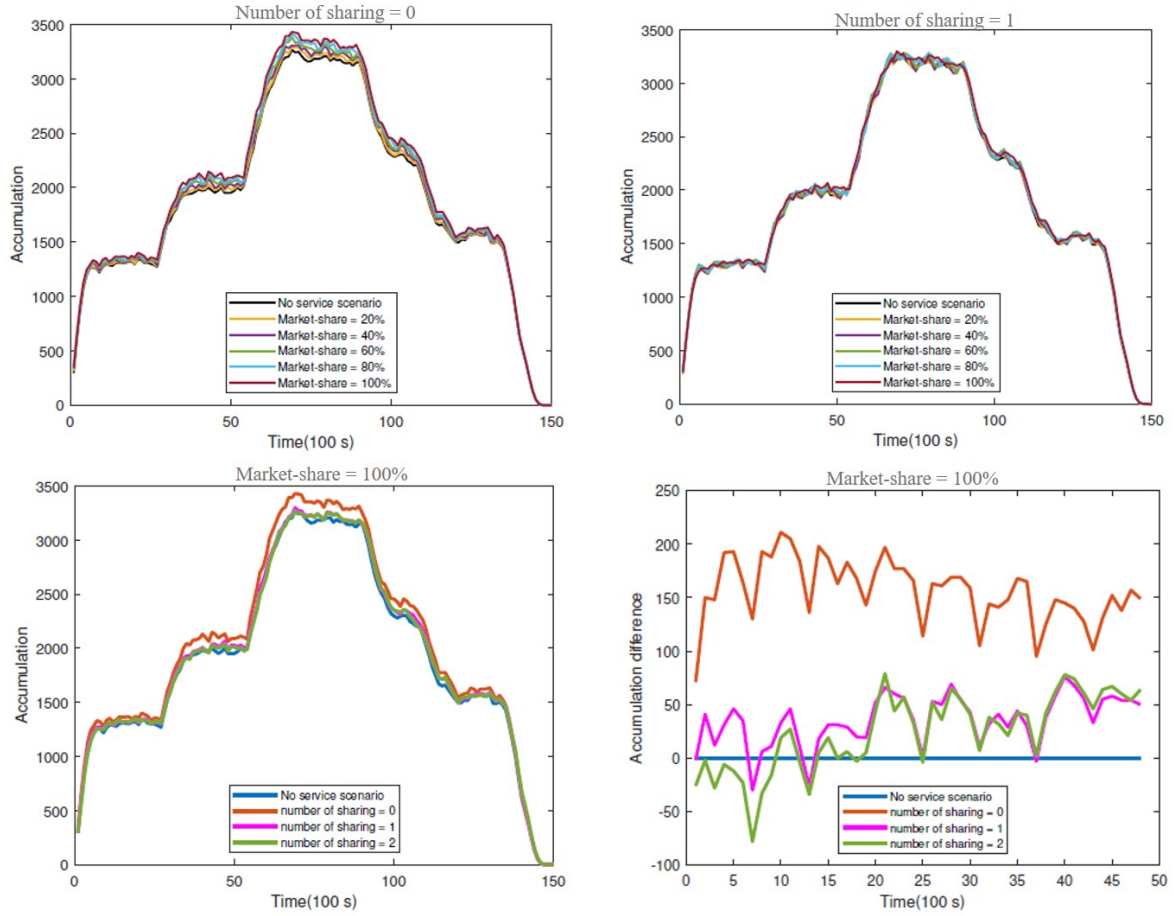


Figure 7.2: Medium-scale results

is 4 seconds. It increases to 54 seconds when the number of sharing is 1 and 74 seconds when the number of sharing is 2. This means that the passenger must wait no more than 1 minute to be picked up at the origin when the number of sharing is 1.

We perform sensitivity analysis on the fixed length of the time window, the rolling horizon, and the depot size in medium-scale. But still no configuration can overcome the "no service scenario" in medium-scale.

To assess the impact of dynamic ride-sharing on traffic congestion in large-scale, we used the simulation-based optimization framework to assess the system in the network of Lyon city in France. The Lyon network area is 220% larger than the medium-scale network, and the number of trips is 676% more:

The proposed system for dynamic ride-sharing performance in reducing congestion in large-scale is terrific. It can significantly reduce traffic congestion.

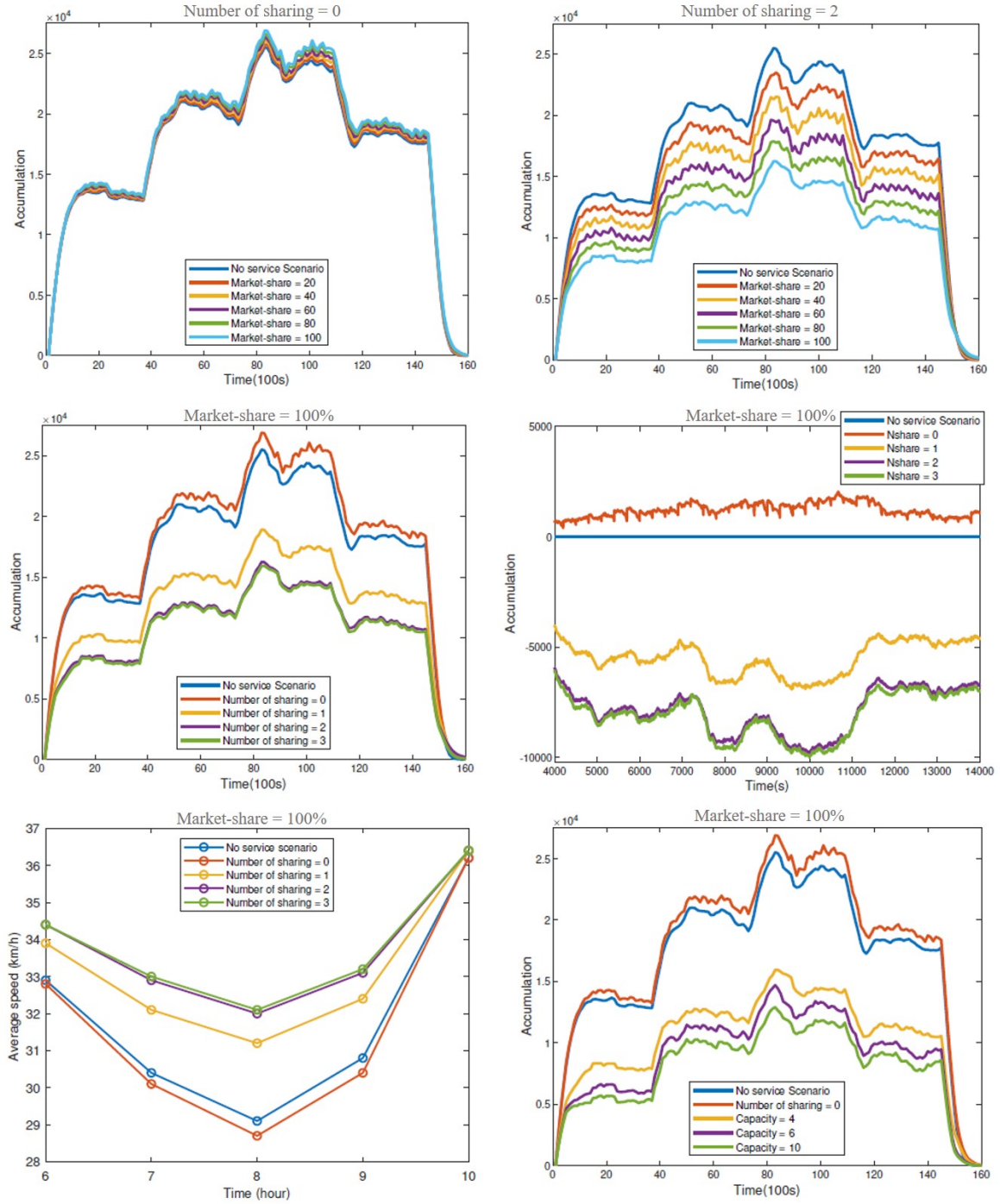


Figure 7.3: Large-scale results

Without sharing, the total travel distance for all the trips increases by 0.8%, 1.5%, 2.3%, 3.0% and 3.7% for the number of sharing 20%, 40%, 60%, 80% and 100% respectively. Then with sharing just two passengers' trips (number of sharing 1), the number of trips reduces by almost 50% for all the market-shares. Increasing the number of sharing to 2 can even make a better improvement. The number of trips is reduced by 63.9% with

the market-share 20%, 64.4% with the market-share 40%, 64.6% with the market-share 60%, 64.7% with the market-share 80% and 64.8% with the market-share 100%. Also, the number of sharing 2 can decrease the needed cars to serve the trips for 57.5%, 58.2%, 58.9%, 59.3%, 60.1% with the market-share 20%, 40%, 60%, 80% and 100% respectively. When the market-share is 100%, sharing can reduce the total travel time by 30.0% with the number of sharing one and 41.1% with the number of sharing two compared to the number of sharing 0. Furthermore, the total travel distance is reduced by 25.5% with the number of sharing 1 and 36.0% with the number of sharing 2.

Increasing the number of sharing increases the passenger waiting time, but the waiting time remains acceptable for passengers with different numbers of sharing. For market-share = 100%, the average waiting time for the passengers is 16.0 seconds for the number of sharing 0. It increases to 27.4 seconds, 44.2 seconds, and 49.3 seconds for the number of sharing 1, 2, and 3.

We also assessed the proposed framework for van-pooling and higher car capacity (capacity = 6 and 10) when we use the full vehicles' capacity (number of sharing = capacity - 1): The system can perform better with bigger cars and vans. With capacity = 6, the number of trips is reduced by 9.3% compared to the scenario when the car capacity is 4. Also using vans can reduce the number of trips by 33.4% and the number of needed cars by 35.5%.

Furthermore, we investigated the influence of personal vehicle parking space searches on the performance of the dynamic system in terms of reducing congestion. When we considered parking space searching by personal cars, we found that the dynamic ride-sharing system can make bigger improvements to traffic congestion during peak hour.

## 7.2 Conclusion

The main goal of this dissertation is to answer one of the critical questions about dynamic ride-sharing services:

Can dynamic ride-sharing reduce congestion?

To answer this question, we proposed a simulation-based optimization framework and



then we implemented the dynamic ride-sharing framework on two different scales to find the answer to this question. The proposed simulation-based optimization framework in this thesis outperforms the existing methods in the literature. The optimization algorithm can provide high-quality solutions in a short time. In the simulation component, a "Plant Model" is applied based on the "Trip-based Macroscopic Fundamental Diagram (MFD)" to represent the traffic dynamics reality and a "Prediction Model" is applied based on the mean-speed to be used during the assignment process.

We performed an extensive simulation study (based on real-world traffic patterns) to assess the influence of dynamic ride-sharing systems on traffic congestion.

In the medium-scale, the results showed that ride-sharing could not significantly improve the traffic situation, contrarily to what we expected. Ride-sharing can reduce congestion compared to traditional taxi services and dial-a-ride services in medium-scale.

To reduce travel times significantly during peak hours, we expect a remarkable reduction in the number of vehicles on the road network. However, high levels of market-share add extra travel distance and travel time to the trips and lead to more traffic in the network. Thus, dynamic ride-sharing can not be a proper solution for reducing traffic in medium and small-scale cities.

In large cities, the results are entirely different from those in small and medium-sized cities. In large-scale (Lyon) simulations, the proposed dynamic ride-sharing system can significantly improve traffic conditions, especially during peak hours. Increasing the market-share and the number of sharing can enhance this improvement. Also, in the large-scale, the accumulation of demand for the service is very high, and we have very close origin points. So the system has more sharing opportunities and can share these trips without deteriorating the passengers' waiting time.

Therefore, the proposed dynamic ride-sharing system is a viable option, alleviating stress on existing public transport, to reduce the network traffic in populated and large-scale cities.

## 7.3 Future work

In the future we will consider the following extensions to our research:

- The proposed system for ride-sharing provides a vast opportunity to assess new mobility services. In the future, we can use extensions of this framework to assess other shared services such as car-sharing or bike-sharing. Also, we can evaluate the performance of new mobility services integrated with public transportation in cities.
- In the optimization component, we can add other heuristic steps that can make the assignment algorithm fast while keeping the quality of the solution acceptable to implement the method on more than two million requests. In the clustering method, a primary clustering can be added to first cluster the requests based on their geographical position, and in the next step, cluster them based on the shareability index defined here. Also, we can put limitations on the number of branches considering detour conditions.
- In the simulation component, for the large-scale network, we use one trip-based MFD to represent the traffic dynamics reality. To be more realistic, we can divide the network into different reservoirs and consider a unique MFD function inside each reservoir.
- In the mathematical model objective function, we consider minimization of the passengers' waiting time and travel time to increase passengers' satisfaction and willingness. Another important point for the passengers is the trip fare when they share their ride. In the future, we can define a pricing scheme for the dynamic ride-sharing system to compute the trip cost depending on the number of passengers sharing a trip and trip conditions.
- We may also work on different market configurations : Monopoly as here, or several companies competing on every request. The analysis could complement the previous point fares analysis

# Bibliography

- [1] Clustering into same size clusters. <http://jmonlong.github.io/Hippocampus/2018/06/09/cluster-same-size/>.
- [2] Code of virginia. <https://law.lis.virginia.gov/vacode/title46.2/chapter14/section46.2-1400/>.
- [3] European commission: Mobility and transport. [https://ec.europa.eu/transport/themes/urban/urban\\_mobility\\_en](https://ec.europa.eu/transport/themes/urban/urban_mobility_en).
- [4] European environment agency. <https://www.eea.europa.eu/data-and-maps/indicators/occupancy-rates-of-passenger-vehicles>.
- [5] AGATZ, N., ERERA, A., SAVELSBERGH, M., AND WANG, X. Sustainable passenger transportation: Dynamic ride-sharing.
- [6] AGATZ, N., ERERA, A., SAVELSBERGH, M., AND WANG, X. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research* 223, 2 (2012), 295–303.
- [7] AGATZ, N. A., ERERA, A. L., SAVELSBERGH, M. W., AND WANG, X. Dynamic ride-sharing: A simulation study in metro atlanta. *Transportation Research Part B: Methodological* 45, 9 (2011), 1450–1464.
- [8] ALISOLTANI, N., LECLERCQ, L., AND ZARGAYOUNA, M. Network performance under different levels of ride-sharing: A simulation study. In *TRISTAN X, The Tenth Triennial Symposium on Transportation Analysis* (2019), p. 5p.

- [9] ALISOLTANI, N., LECLERCQ, L., AND ZARGAYOUNA, M. Real-time ride-sharing systems performance considering network congestion. In *hEART 2019, 8th Symposium of the European Association for Research in Transportation* (2019), p. 6p.
- [10] ALISOLTANI, N., LECLERCQ, L., ZARGAYOUNA, M., AND KRUG, J. Optimal fleet management for real-time ride-sharing service considering network congestion. In *TRB 2019, Transportation Research Board 98th Annual Meeting* (2019), p. 22p.
- [11] ALISOLTANI, N., ZARGAYOUNA, M., AND LECLERCQ, L. Data-oriented approach for the dial-a-ride problem. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)* (2019), IEEE, pp. 1–6.
- [12] ALISOLTANI, N., ZARGAYOUNA, M., AND LECLERCQ, L. A multi-agent system for real-time ride sharing in congested networks. In *Agents and Multi-agent Systems: Technologies and Applications 2019*. Springer, 2020, pp. 333–342.
- [13] ALISOLTANI, N., ZARGAYOUNA, M., AND LECLERCQ, L. Real-time autonomous taxi service: An agent-based simulation. In *Agents and Multi-Agent Systems: Technologies and Applications 2020*. Springer, 2020, pp. 199–207.
- [14] ALISOLTANI, N., ZARGAYOUNA, M., AND LECLERCQ, L. A sequential clustering method for the taxi-dispatching problem considering traffic dynamics. *IEEE Intelligent Transportation Systems Magazine* 12, 4 (2020), 169–181.
- [15] ALTSHULER, T., KATOSHEVSKI, R., AND SHIFTAN, Y. Ride sharing and dynamic networks analysis. *arXiv preprint arXiv:1706.00581* (2017).
- [16] AMELI, M. *Heuristic Methods for Calculating Dynamic Traffic Assignment*. PhD thesis, Université de Lyon, 2019.
- [17] AMELI, M., LEBACQUE, J.-P., AND LECLERCQ, L. Cross-comparison of convergence algorithms to solve trip-based dynamic traffic assignment problems. *Computer-Aided Civil and Infrastructure Engineering* 35, 3 (2020), 219–240.

- [18] AMELI, M., LEBACQUE, J.-P., AND LECLERCQ, L. Simulation-based dynamic traffic assignment: Meta-heuristic solution methods with parallel computing. *Computer-Aided Civil and Infrastructure Engineering* 35, 10 (2020), 1047–1062.
- [19] AMEY, A., ATTANUCCI, J., AND MISHALANI, R. Real-time ridesharing: opportunities and challenges in using mobile phone technology to improve rideshare services. *Transportation Research Record: Journal of the Transportation Research Board*, 2217 (2011), 103–110.
- [20] AMEY, A. M. *Real-time ridesharing: exploring the opportunities and challenges of designing a technology-based rideshare trial for the MIT community*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [21] ANDRÉASSON, I. Ride-sharing on prt. In *Automated People Movers 2005: Moving to Mainstream*. 2005, pp. 1–7.
- [22] ATAHARAN, A., LENTÉ, C., AND T’KINDT, V. A multicriteria dial-a-ride problem with an ecological measure and heterogeneous vehicles. *Journal of Multi-Criteria Decision Analysis* 21, 5-6 (2014), 279–298.
- [23] AXHAUSEN, K. W., POLAK, J., BOLTZE, M., AND PUZICHA, J. Effectiveness of the parking guidance information system in frankfurt am main. *Traffic Engineering+ Control* 35, 5 (1994), 304–9.
- [24] BALDACCI, R., MANIEZZO, V., AND MINGOZZI, A. An exact method for the car pooling problem based on lagrangean column generation. *Operations Research* 52, 3 (2004), 422–439.
- [25] BAN, X. J., DESSOUKY, M., PANG, J.-S., AND FAN, R. A general equilibrium model for transportation systems with e-hailing services and flow congestion. *Transportation Research Part B: Methodological* 129 (2019), 273–304.
- [26] BARD, J. F., AND JARRAH, A. I. Large-scale constrained clustering for rationalizing pickup and delivery operations. *Transportation Research Part B: Methodological* 43, 5 (2009), 542–561.

- [27] BERBEGLIA, G., CORDEAU, J.-F., AND LAPORTE, G. Dynamic pickup and delivery problems. *European journal of operational research* 202, 1 (2010), 8–15.
- [28] BERBEGLIA, G., CORDEAU, J.-F., AND LAPORTE, G. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing* 24, 3 (2012), 343–355.
- [29] BICOCCHI, N., AND MAMEI, M. Investigating ride sharing opportunities through mobility data analysis. *Pervasive and Mobile Computing* 14 (2014), 83–94.
- [30] BISCHOFF, J., AND MACIEJEWSKI, M. Simulation of city-wide replacement of private cars with autonomous taxis in berlin. *Procedia computer science* 83 (2016), 237–244.
- [31] BLOCHO, M. Exact algorithms for solving rich vehicle routing problems. In *Smart Delivery Systems*. Elsevier, 2020, pp. 93–99.
- [32] BOFFEY, B., AND NARULA, S. C. Multiobjective covering and routing problems. In *Essays in Decision Making*. Springer, 1997, pp. 342–369.
- [33] BONGIOVANNI, C., KASPI, M., AND GEROLIMINIS, N. The electric autonomous dial-a-ride problem. *Transportation Research Part B: Methodological* 122 (2019), 436–456.
- [34] BOTSMAN, R., AND ROGERS, R. Beyond zipcar: Collaborative consumption. *Harvard Business Review* 88, 10 (2010), 30.
- [35] BRAEKERS, K., AND KOVACS, A. A. A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological* 94 (2016), 355–377.
- [36] BURRIS, M. W., AND WINN, J. R. Slugging in houston—casual carpool passenger characteristics. *Journal of Public Transportation* 9, 5 (2006), 2.
- [37] CAI, H., WANG, X., ADRIAENS, P., AND XU, M. Environmental benefits of taxi ride sharing in beijing. *Energy* 174 (2019), 503–508.

- [38] CALVO, R. W., DE LUIGI, F., HAASTRUP, P., AND MANIEZZO, V. A distributed geographic information system for the daily car pooling problem. *Computers & Operations Research* 31, 13 (2004), 2263–2278.
- [39] CASEY, R. F., LABELL, L. N., MONIZ, L., ROYAL, J. W., SHEEHAN, M., SHEEHAN, T., BROWN, A., FOY, M., ZIRKER, M. E., SCHWEIGER, C. L., ET AL. Advanced public transportation systems: The state of the art update 2000. Tech. rep., United States. Federal Transit Administration, 2000.
- [40] CAULFIELD, B. Estimating the environmental benefits of ride-sharing: A case study of dublin. *Transportation Research Part D: Transport and Environment* 14, 7 (2009), 527–531.
- [41] CHAN, N. D., AND SHAHEEN, S. A. Ridesharing in north america: Past, present, and future. *Transport Reviews* 32, 1 (2012), 93–112.
- [42] CHEN, S., WANG, H., AND MENG, Q. Solving the first-mile ridesharing problem using autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering* 35, 1 (2020), 45–60.
- [43] CICI, B., MARKOPOULOU, A., FRÍAS-MARTÍNEZ, E., AND LAOUTARIS, N. Quantifying the potential of ride-sharing using call description records. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications* (2013), pp. 1–6.
- [44] COHEN, B., AND KIETZMANN, J. Ride on! mobility business models for the sharing economy. *Organization & Environment* 27, 3 (2014), 279–296.
- [45] CONTARDO, C., CORDEAU, J.-F., AND GENDRON, B. A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optimization* 10, 4 (2013), 263–295.
- [46] CORDEAU, J.-F., AND LAPORTE, G. The dial-a-ride problem: models and algorithms. *Annals of operations research* 153, 1 (2007), 29–46.

- [47] CRAMER, J., AND KRUEGER, A. B. Disruptive change in the taxi business: The case of uber. *American Economic Review* 106, 5 (2016), 177–82.
- [48] DAI, G., HUANG, J., WAMBURA, S. M., AND SUN, H. A balanced assignment mechanism for online taxi recommendation. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)* (2017), IEEE, pp. 102–111.
- [49] DAILEY, D. J., LOSEFF, D., AND MEYERS, D. Seattle smart traveler: dynamic ridematching on the world wide web. *Transportation Research Part C: Emerging Technologies* 7, 1 (1999), 17–32.
- [50] DAVIS, N., RAINA, G., AND JAGANNATHAN, K. Taxi demand forecasting: A hedge-based tessellation strategy for improved accuracy. *IEEE Transactions on Intelligent Transportation Systems* 19, 11 (2018), 3686–3697.
- [51] DEAKIN, E., FRICK, K. T., AND SHIVELY, K. M. Markets for dynamic ridesharing? case of berkeley, california. *Transportation Research Record* 2187, 1 (2010), 131–137.
- [52] DEWAN, K. K., AND AHMAD, I. Carpooling: a step to reduce congestion. *Engineering Letters* 14, 1 (2007), 61–66.
- [53] D’OREY, P. M., AND FERREIRA, M. Can ride-sharing become attractive? a case study of taxi-sharing employing a simulation modelling approach. *IET Intelligent Transport Systems* 9, 2 (2014), 210–220.
- [54] DREXL, M. Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. *Transportation Science* 46, 3 (2012), 297–316.
- [55] FAGNANT, D. J., AND KOCKELMAN, K. M. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas. *Transportation* 45, 1 (2018), 143–158.
- [56] FARIN, N. J., RIMON, M. N. A. A., MOMEN, S., UDDIN, M. S., AND MANSOOR, N. A framework for dynamic vehicle pooling and ride-sharing system. In



- 2016 International Workshop on Computational Intelligence (IWCI)* (2016), IEEE, pp. 204–208.
- [57] FERGUSON, E. The rise and fall of the american carpool: 1970–1990. *Transportation* 24, 4 (1997), 349–376.
- [58] FINK, M., DESAULNIERS, G., FREY, M., KIERMAIER, F., KOLISCH, R., AND SOUMIS, F. Column generation for vehicle routing problems with multiple synchronization constraints. *European Journal of Operational Research* 272, 2 (2019), 699–711.
- [59] FIRNKORN, J., AND MÜLLER, M. What will be the environmental effects of new free-floating car-sharing systems? the case of car2go in ulm. *Ecological economics* 70, 8 (2011), 1519–1528.
- [60] FISHER, M. L. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research* 42, 4 (1994), 626–642.
- [61] FONTARAS, G., ZACHAROF, N.-G., AND CIUFFO, B. Fuel consumption and co2 emissions from passenger cars in europe–laboratory versus real-world emissions. *Progress in Energy and Combustion Science* 60 (2017), 97–131.
- [62] FURUHATA, M., DESSOUKY, M., ORDÓÑEZ, F., BRUNET, M.-E., WANG, X., AND KOENIG, S. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological* 57 (2013), 28–46.
- [63] GANGANATH, N., CHENG, C.-T., AND CHI, K. T. Data clustering with cluster size constraints using a modified k-means algorithm. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (2014), IEEE, pp. 158–161.
- [64] GARGIULO, E., GIANNANTONIO, R., GUERCIO, E., BOREAN, C., AND ZENEZINI, G. Dynamic ride sharing service: are users ready to adopt it? *Procedia Manufacturing* 3 (2015), 777–784.

- [65] GHILAS, V., CORDEAU, J.-F., DEMIR, E., AND WOENSEL, T. V. Branch-and-price for the pickup and delivery problem with time windows and scheduled lines. *Transportation Science* 52, 5 (2018), 1191–1210.
- [66] GHOSEIRI, K., HAGHANI, A. E., HAMED, M., AND CENTER, M. *Real-time rideshare matching problem*. Mid-Atlantic Universities Transportation Center Berkeley, 2011.
- [67] GOEL, P. *Private personalized dynamic ride sharing*. PhD thesis, 2016.
- [68] GOEL, P., KULIK, L., AND RAMAMOHANARAO, K. Optimal pick up point selection for effective ride sharing. *IEEE Transactions on Big Data* 3, 2 (2017), 154–168.
- [69] GOLOB, T. F., AND REGAN, A. C. Impacts of information technology on personal travel and commercial vehicle operations: research challenges and opportunities. *Transportation Research Part C: Emerging Technologies* 9, 2 (2001), 87–121.
- [70] GONZALEZ, M. C., HIDALGO, C. A., AND BARABASI, A.-L. Understanding individual human mobility patterns. *nature* 453, 7196 (2008), 779.
- [71] GU, Q.-P., LIANG, J. L., AND ZHANG, G. Algorithmic analysis for ridesharing of personal vehicles. *Theoretical Computer Science* 749 (2018), 36–46.
- [72] GURUMURTHY, K. M., AND KOCKELMAN, K. M. Modeling americans’ autonomous vehicle preferences: A focus on dynamic ride-sharing, privacy & long-distance mode choices. *Technological Forecasting and Social Change* 150 (2020), 119792.
- [73] HAMARI, J., SJÖKLINT, M., AND UKKONEN, A. The sharing economy: Why people participate in collaborative consumption. *Journal of the association for information science and technology* 67, 9 (2016), 2047–2059.
- [74] HANDKE, V., AND JONUSCHAT, H. *Flexible ridesharing: new opportunities and service concepts for sustainable mobility*. Springer Science & Business Media, 2012.

- [75] HASELKORN, M., SPRYIDAKIS, J., BLUMENTHAL, C., MICHALAK, S., GOBLE, B., GARNER, M., COMMISSION, W. S. T., ET AL. Bellevue smart traveler: Design, demonstration, and assessment. Tech. rep., United States. Joint Program Office for Intelligent Transportation Systems, 1995.
- [76] HASELKORN, M., SPYRIDAKIS, J., GOBLE, B., AND MICHALAK, S. Bellevue smart traveler: An integrated phone and pager system for downtown dynamic ride sharing. In *Moving Toward Deployment. Proceedings of the IVHS America Annual Meeting. 2 Volumes IVHS America* (1994), no. Volume 1.
- [77] HENAO, A. *Impacts of Ridesourcing-Lyft and Uber-on Transportation Including VMT, Mode Replacement, Parking, and Travel Behavior*. University of Colorado at Denver, 2017.
- [78] HENSHER, D. A. Climate change, enhanced greenhouse gas emissions and passenger transport—what can we do to make a difference? *Transportation Research Part D: Transport and Environment* 13, 2 (2008), 95–111.
- [79] HERBAWI, W., AND WEBER, M. Ant colony vs. genetic multiobjective route planning in dynamic multi-hop ridesharing. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on* (2011), IEEE, pp. 282–288.
- [80] HERBAWI, W., AND WEBER, M. The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm. In *Evolutionary Computation (CEC), 2012 IEEE Congress on* (2012), IEEE, pp. 1–8.
- [81] HERBAWI, W. M., AND WEBER, M. A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation* (2012), ACM, pp. 385–392.
- [82] HILDERMEIER, J., AND VILLAREAL, A. Two ways of defining sustainable mobility: Autolib’and bemobility. *Journal of Environmental Policy & Planning* 16, 3 (2014), 321–336.

- [83] HO, S. C., SZETO, W., KUO, Y.-H., LEUNG, J. M., PETERING, M., AND TOU, T. W. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological* 111 (2018), 395–421.
- [84] HOSNI, H., NAOUM-SAWAYA, J., AND ARTAIL, H. The shared-taxi problem: Formulation and solution methods. *Transportation Research Part B: Methodological* 70 (2014), 303–318.
- [85] HUANG, Y., JIN, R., BASTANI, F., AND WANG, X. S. Large scale real-time ridesharing with service guarantee on road networks. *arXiv preprint arXiv:1302.6666* (2013).
- [86] HYLAND, M., AND MAHMASSANI, H. S. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign avs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies* 92 (2018), 278–297.
- [87] HYLAND, M. F., AND MAHMASSANI, H. S. Sharing is caring: Dynamic autonomous vehicle fleet operations under demand surges. Tech. rep., 2018.
- [88] JACOBSON, S. H., AND KING, D. M. Fuel saving and ridesharing in the us: Motivations, limitations, and opportunities. *Transportation Research Part D: Transport and Environment* 14, 1 (2009), 14–21.
- [89] JIA, Y., XU, W., AND LIU, X. An optimization framework for online ride-sharing markets. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on* (2017), IEEE, pp. 826–835.
- [90] JIANG, S., CHEN, L., MISLOVE, A., AND WILSON, C. On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi. In *Proceedings of the 2018 World Wide Web Conference* (2018), pp. 863–872.
- [91] JIN, S. T., KONG, H., WU, R., AND SUI, D. Z. Ridesourcing, the sharing economy, and the future of cities. *Cities* 76 (2018), 96–104.

- [92] JUNG, J., JAYAKRISHNAN, R., AND PARK, J. Y. Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. *Computer-Aided Civil and Infrastructure Engineering* 31, 4 (2016), 275–291.
- [93] KAAAN, L., AND OLINICK, E. V. The vanpool assignment problem: Optimization models and solution algorithms. *Computers & Industrial Engineering* 66, 1 (2013), 24–40.
- [94] KATZEV, R. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy* 3, 1 (2003), 65–86.
- [95] KELLY, K. L. Casual carpooling-enhanced. *Journal of Public Transportation* 10, 4 (2007), 6.
- [96] KIRCHLER, D., AND CALVO, R. W. A granular tabu search algorithm for the dial-a-ride problem. *Transportation Research Part B: Methodological* 56 (2013), 120–135.
- [97] KIRSHNER, D. Definition of dynamic ridesharing. *Retrieved April 7* (2008), 2009.
- [98] KLEINER, A., NEBEL, B., AND ZIPARO, V. A. A mechanism for dynamic ride sharing based on parallel auctions. In *Twenty-Second International Joint Conference on Artificial Intelligence* (2011).
- [99] KOOTI, F., GRBOVIC, M., AIELLO, L. M., DJURIC, N., RADOSAVLJEVIC, V., AND LERMAN, K. Analyzing uber’s ride-sharing economy. In *Proceedings of the 26th International Conference on World Wide Web Companion* (2017), pp. 574–582.
- [100] KOWSHIK, R., JOVANIS, P. P., AND KITAMURA, R. Evaluation of the sacramento-area real-time rideshare matching field operational test final report, 1996.
- [101] KRUEGER, R., RASHIDI, T. H., AND ROSE, J. M. Preferences for shared autonomous vehicles. *Transportation research part C: emerging technologies* 69 (2016), 343–355.

- [102] KRUG, J., BURIANNE, A., AND LECLERCQ, L. Reconstituting demand patterns of the city of lyon by using multiple gis data sources. Tech. rep., University of Lyon, ENTPE, LICIT, 2017.
- [103] KRZANOWSKI, W. J., AND LAI, Y. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics* (1988), 23–34.
- [104] LAMOTTE, R., AND GEROLIMINIS, N. The morning commute in urban areas: Insights from theory and simulation. Tech. rep., 2016.
- [105] LECLERCQ, L., CHIABAUT, N., AND TRINQUIER, B. Macroscopic fundamental diagrams: A cross-comparison of estimation methods. *Transportation Research Part B: Methodological* 62 (2014), 1–12.
- [106] LECLERCQ, L., SÉNÉCAT, A., AND MARIOTTE, G. Dynamic macroscopic simulation of on-street parking search: A trip-based approach. *Transportation Research Part B: Methodological* 101 (2017), 268–282.
- [107] LEHUÉDÉ, F., MASSON, R., PARRAGH, S. N., PÉTON, O., AND TRICOIRE, F. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society* 65, 7 (2014), 983–1000.
- [108] LEVIN, M. W., KOCKELMAN, K. M., BOYLES, S. D., AND LI, T. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. *Computers, Environment and Urban Systems* 64 (2017), 373–383.
- [109] LEVOFSKY, A., AND GREENBERG, A. Organized dynamic ride sharing: The potential environmental benefits and the opportunity for advancing the concept. In *Transportation Research Board 2001 Annual Meeting* (2001), pp. 7–11.
- [110] LI, B., KRUSHINSKY, D., VAN WOENSEL, T., AND REIJERS, H. A. The share-a-ride problem with stochastic travel times and stochastic delivery locations. *Transportation Research Part C: Emerging Technologies* 67 (2016), 95–108.

- [111] LI, Z., HONG, Y., AND ZHANG, Z. Do ride-sharing services affect traffic congestion? an empirical study of uber entry. *SSRN Electronic Journal*, 2002 (2016), 1–29.
- [112] LI, Z., HONG, Y., AND ZHANG, Z. An empirical analysis of on-demand ride sharing and traffic congestion. In *Proc. International Conference on Information Systems* (2016).
- [113] LINARES, M. P., MONTERO, L., BARCELÓ, J., AND CARMONA, C. A simulation framework for real-time assessment of dynamic ride sharing demand responsive transportation models. In *Winter Simulation Conference (WSC), 2016* (2016), IEEE, pp. 2216–2227.
- [114] LIU, M., LUO, Z., AND LIM, A. A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological* 81 (2015), 267–288.
- [115] LIU, X., YAN, X., LIU, F., WANG, R., AND LENG, Y. A trip-specific model for fuel saving estimation and subsidy policy making of carpooling based on empirical data. *Applied Energy* 240 (2019), 295–311.
- [116] LIU, Y., LIU, J., LIAO, Z., TANG, M., AND CHEN, J. Recommending a personalized sequence of pick-up points. *Journal of computational science* 28 (2018), 382–388.
- [117] LOKHANDWALA, M., AND CAI, H. Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of nyc. *Transportation Research Part C: Emerging Technologies* 97 (2018), 45–60.
- [118] MA, S., ZHENG, Y., AND WOLFSON, O. T-share: A large-scale dynamic taxi ridesharing service. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on* (2013), IEEE, pp. 410–421.
- [119] MA, S., ZHENG, Y., AND WOLFSON, O. Real-time city-scale taxi ridesharing. *IEEE Transactions on Knowledge and Data Engineering* 27, 7 (2015), 1782–1795.

- [120] MA, T.-Y. On-demand dynamic bi-/multi-modal ride-sharing using optimal passenger-vehicle assignments. In *Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), 2017 IEEE International Conference on* (2017), IEEE, pp. 1–5.
- [121] MACIEJEWSKI, M., AND NAGEL, K. Simulation and dynamic optimization of taxi services in matsim. *VSP Working Paper 13-0. TU Berlin, Transport Systems Planning and Transport Telematics, 2013* (2013).
- [122] MAHMOUDI, M., AND ZHOU, X. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. *Transportation Research Part B: Methodological* 89 (2016), 19–42.
- [123] MARIOTTE, G., AND LECLERCQ, L. Flow exchanges in multi-reservoir systems with spillbacks. *Transportation Research Part B: Methodological* 122 (2019), 327–349.
- [124] MARIOTTE, G., LECLERCQ, L., AND LAVAL, J. A. Macroscopic urban dynamics: Analytical and numerical comparisons of existing models. *Transportation Research Part B: Methodological* 101 (2017), 245–267.
- [125] MARTINEZ, L. M., CORREIA, G. H., AND VIEGAS, J. M. An agent-based simulation model to assess the impacts of introducing a shared-taxi system: an application to lisbon (portugal). *Journal of Advanced Transportation* 49, 3 (2015), 475–495.
- [126] MASOUD, N., LLORET-BATLLE, R., AND JAYAKRISHNAN, R. Using bilateral trading to increase ridership and user permanence in ridesharing systems. *Transportation Research Part E: Logistics and Transportation Review* 102 (2017), 60–77.
- [127] MITCHELL, J. E. Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization* 1 (2002), 65–77.
- [128] MOLENBRUCH, Y., BRAEKERS, K., AND CARIS, A. Typology and literature review for dial-a-ride problems. *Annals of Operations Research* 259, 1-2 (2017), 295–325.



- [129] MORENCY, C. The ambivalence of ridesharing. *Transportation* 34, 2 (2007), 239–253.
- [130] MOURAD, A., PUCHINGER, J., AND CHU, C. A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological* (2019).
- [131] MURTAGH, F. A survey of recent advances in hierarchical clustering algorithms. *The computer journal* 26, 4 (1983), 354–359.
- [132] NADAL, L. Bike sharing sweeps paris off its feet. *Sustainable transport*, 19 (2007).
- [133] NAOUM-SAWAYA, J., COGILL, R., GHADDAR, B., SAJJA, S., SHORTEN, R., TAHERI, N., TOMMASI, P., VERAGO, R., AND WIRTH, F. Stochastic optimization approach for the car placement problem in ridesharing systems. *Transportation Research Part B: Methodological* 80 (2015), 173–184.
- [134] ORDÓÑEZ, F., AND DESSOUKY, M. M. Dynamic ridesharing. In *Leading Developments from INFORMS Communities*. INFORMS, 2017, pp. 212–236.
- [135] OTA, M., VO, H., SILVA, C., AND FREIRE, J. Stars: Simulating taxi ride sharing at scale. *IEEE Transactions on Big Data* 3, 3 (2017), 349–361.
- [136] ÖZDAMAR, L., AND DEMIR, O. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics and Transportation Review* 48, 3 (2012), 591–602.
- [137] PAEA, S., AND BAIRD, R. Information architecture (ia): Using multidimensional scaling (mds) and k-means clustering algorithm for analysis of card sorting data. *Journal of Usability Studies* 13, 3 (2018).
- [138] PILLAC, V., GENDREAU, M., GUÉRET, C., AND MEDAGLIA, A. L. A review of dynamic vehicle routing problems. *European Journal of Operational Research* 225, 1 (2013), 1–11.

- [139] PISARSKI, A. *Commuting in America III: The third national report on commuting patterns and trends*, vol. 550. Transportation Research Board, 2006.
- [140] POWELL, J. W., HUANG, Y., BASTANI, F., AND JI, M. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *International Symposium on Spatial and Temporal Databases* (2011), Springer, pp. 242–260.
- [141] PRATSCH, L. *Carpool and buspool matching guide*. US Federal Highway Administration, 1975.
- [142] QI, H., AND LIU, P. Mining taxi pick-up hotspots based on spatial clustering. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI)* (2018), IEEE, pp. 1711–1717.
- [143] QIAN, X., ZHANG, W., UKKUSURI, S. V., AND YANG, C. Optimal assignment and incentive design in the taxi group ride problem. *Transportation Research Part B: Methodological* 103 (2017), 208–226.
- [144] QIANG, X., AND SHUANG-SHUANG, Y. Clustering algorithm for urban taxi carpooling vehicle based on data field energy. *Journal of Advanced Transportation* 2018 (2018).
- [145] RAYLE, L., DAI, D., CHAN, N., CERVERO, R., AND SHAHEEN, S. Just a better taxi? a survey-based comparison of taxis, transit, and ridesourcing services in san francisco. *Transport Policy* 45 (2016), 168–178.
- [146] REDDY, C. K., AND VINZAMURI, B. A survey of partitional and hierarchical clustering algorithms. *Data clustering: Algorithms and applications* 87 (2013).
- [147] RENO, A. T., GELLERT, W. A., AND VERZOSA, A. *Evaluation of Springfield instant carpooling*. No. 1212. 1989.

- [148] ROPKE, S., AND CORDEAU, J.-F. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science* 43, 3 (2009), 267–286.
- [149] ROSS, G. T., AND SOLAND, R. M. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming* 8, 1 (1975), 91–103.
- [150] SÁEZ, D., CORTÉS, C. E., AND NÚÑEZ, A. Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers & Operations Research* 35, 11 (2008), 3412–3438.
- [151] SANTI, P., RESTA, G., SZELL, M., SOBOLEVSKY, S., STROGATZ, S. H., AND RATTI, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences* 111, 37 (2014), 13290–13294.
- [152] SANTOS, D. O., AND XAVIER, E. C. Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications* 42, 19 (2015), 6728–6737.
- [153] SCHREIECK, M., SAFETLI, H., SIDDIQUI, S. A., PFLÜGLER, C., WIESCHE, M., AND KRCMAR, H. A matching algorithm for dynamic ridesharing. *Transportation Research Procedia* 19 (2016), 272–285.
- [154] SCHWANEN, T., BANISTER, D., AND ANABLE, J. Scientific research about climate change mitigation in transport: A critical review. *Transportation Research Part A: Policy and Practice* 45, 10 (2011), 993–1006.
- [155] SCHWEIGER, C. L., LABELL, L. N., KIHLE, M., ET AL. Advanced public transportation systems: The state of the art, update’94. Tech. rep., United States. Federal Transit Administration, 1994.
- [156] SHEN, W., AND LOPES, C. Managing autonomous mobility on demand systems for better passenger experience. In *International Conference on Principles and Practice of Multi-Agent Systems* (2015), Springer, pp. 20–35.

- [157] SHERIF, A. B., RABIEH, K., MAHMOUD, M. M., AND LIANG, X. Privacy-preserving ride sharing scheme for autonomous vehicles in big data era. *IEEE Internet of Things Journal* 4, 2 (2016), 611–618.
- [158] SIDDIQI, Z., AND BULIUNG, R. Dynamic ridesharing and information and communications technology: past, present and future prospects. *Transportation Planning and Technology* 36, 6 (2013), 479–498.
- [159] SINDHU, M., AND RACHNA, S. Clustering algorithm technique. *International Journal of Enhanced Research in Management and Computing Applications* 2, 3 (2013).
- [160] SRINIVASAN, K. K., AND RAGHAVENDER, P. N. Impact of mobile phones on travel: Empirical analysis of activity chaining, ridesharing, and virtual shopping. *Transportation Research Record* 1977, 1 (2006), 258–267.
- [161] STIGLIC, M., AGATZ, N., SAVELSBERGH, M., AND GRADISAR, M. The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological* 82 (2015), 36–53.
- [162] STIGLIC, M., AGATZ, N., SAVELSBERGH, M., AND GRADISAR, M. Making dynamic ride-sharing work: The impact of driver and rider flexibility. *Transportation Research Part E: Logistics and Transportation Review* 91 (2016), 190–207.
- [163] STIGLIC, M., AGATZ, N., SAVELSBERGH, M., AND GRADISAR, M. Enhancing urban mobility: Integrating ride-sharing and public transit. *Computers & Operations Research* 90 (2018), 12–21.
- [164] SUI, Y., ZHANG, H., SONG, X., SHAO, F., YU, X., SHIBASAKI, R., SUN, R., YUAN, M., WANG, C., LI, S., ET AL. Gps data in urban online ride-hailing: A comparative analysis on fuel consumption and emissions. *Journal of Cleaner Production* 227 (2019), 495–505.

- [165] TACHET, R., SAGARRA, O., SANTI, P., RESTA, G., SZELL, M., STROGATZ, S., AND RATTI, C. Scaling law of urban ride sharing. *Scientific reports* 7 (2017), 42868.
- [166] TAHMASSEBY, S., KATTAN, L., AND BARBOUR, B. Dynamic real-time ridesharing: A literature review and early findings from a market demand study of a dynamic transportation trading platform for the university of calgary's main campus. Tech. rep., 2014.
- [167] TEN BRINK, P. Mitigating co2 emissions from cars in the eu (regulation (ec) no. 443/2009). *The new climate policies of the European Union: Internal legislation and climate diplomacy* 15 (2010), 179.
- [168] TEUBNER, T., AND FLATH, C. M. The economics of multi-hop ride sharing. *Business & Information Systems Engineering* 57, 5 (2015), 311–324.
- [169] VAZIFEH, M. M., SANTI, P., RESTA, G., STROGATZ, S., AND RATTI, C. Addressing the minimum fleet problem in on-demand urban mobility. *Nature* 557, 7706 (2018), 534.
- [170] VORA, P., OZA, B., ET AL. A survey on k-mean clustering and particle swarm optimization. *International Journal of Science and Modern Engineering* 1, 3 (2013), 24–26.
- [171] WANG, Q., AND BOYER, K. L. Feature learning by multidimensional scaling and its applications in object recognition. In *2013 XXVI Conference on Graphics, Patterns and Images* (2013), IEEE, pp. 8–15.
- [172] WANG, S., LI, L., MA, W., AND CHEN, X. Trajectory analysis for on-demand services: A survey focusing on spatial-temporal demand and supply patterns. *Transportation Research Part C: Emerging Technologies* 108 (2019), 74–99.
- [173] WANG, X., AGATZ, N., AND ERERA, A. Stable matching for dynamic ride-sharing systems. *Transportation Science* 52, 4 (2018), 850–867.

- [174] WANG, X., DESSOUKY, M., AND ORDONEZ, F. A pickup and delivery problem for ridesharing considering congestion. *Transportation letters* 8, 5 (2016), 259–269.
- [175] WANG, Y., WINTER, S., AND TOMKO, M. Collaborative activity-based ridesharing. *Journal of Transport Geography* 72 (2018), 131–138.
- [176] YIN, B., LIU, L., COULOMBEL, N., AND VIGUIE, V. Appraising the environmental benefits of ride-sharing: The paris region case study. *Journal of cleaner production* 177 (2018), 888–898.
- [177] YUAN, N. J., ZHENG, Y., ZHANG, L., AND XIE, X. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on knowledge and data engineering* 25, 10 (2012), 2390–2403.
- [178] ZARGAYOUNA, M., AND ZEDDINI, B. Fleet organization models for online vehicle routing problems. In *Transactions on Computational Collective Intelligence VII*. Springer, 2012, pp. 82–102.
- [179] ZOU, H., AND DESSOUKY, M. M. A look-ahead partial routing framework for the stochastic and dynamic vehicle routing problem. *Journal on Vehicle Routing Algorithms* 1, 2-4 (2018), 73–88.

# Appendix A

Literature review based on transportation demand and service characteristics: We have evaluated 15 papers using the transportation demand and service characteristics defined in section 2.2 in chapter 2. Table A.1 shows the list of these 15 papers. Table A.2 shows the list of characters and their numbers. Then the next tables show the characteristics that are assumed in each paper and the explanations about them.

Table A.1: List of papers

Number	Title	Author	Journal	Year
P1	Dynamic ride-sharing: A simulation study in metro Atlanta	Agatz et al.	Transportation Research Part B	2011
P2	Optimal assignment and incentive design in the taxi group ride problem	Qian et al.	Transportation Research Part B	2017
P3	Stochastic optimization approach for the car placement problem in ridesharing systems	Naoum-Sawaya et al.	Transportation Research Part B	2015
P4	Making dynamic ride-sharing work: The impact of driver and rider flexibility	Stiglic et al.	Transportation Research Part E	2016
P5	Stable matching for dynamic ride-sharing systems	Wang et al.	Transportation Science	2018
P6	STaRS: Simulating Taxi Ride Sharing at Scale	Ota et al.	IEEE Transactions on Big Data	2017
P7	Can ride-sharing become attractive?			
P8	A case study of taxi-sharing employing a simulation modelling approach	D'Orey and Ferreira	IET Intelligent Transport Systems	2014
	Real-time city-scale taxi ridesharing	Ma et al.	IEEE Transactions on Knowledge and Data Engineering	2015
	T-share: A large-scale dynamic taxi ridesharing service	MA et al.	Data Engineering (ICDE)	2013
P9	A simulation framework for real-time assessment of dynamic ride sharing demand responsive transportation models	Linares et al.	IEEE(WSC)	2016
P10	A genetic and insertion heuristic algorithm for solving the dynamic ride-matching problem with time windows	Herbawi and Weber	Conference on Genetic and evolutionary computation	2012
	Ant colony vs. genetic multiobjective route planning in dynamic multi-hop ridesharing	Herbawi and Weber	ICTAI	2012
P11	Scaling law of urban ride sharing	Tachet et al.	Scientific reports 7	2017
P12	Quantifying the benefits of vehicle pooling with shareability networks	Sauti et al.	Proceedings of the National Academy of Sciences	2014
P13	Optimal pick up point selection for effective ride sharing	Goel et al.	IEEE Transactions on Big Data	2017
P14	A mechanism for dynamic ridesharing based on parallel auctions	Kleirner et al.	International Joint Conference on Artificial Intelligence	2011
P15	On-demand dynamic bi-/multi-modal ride-sharing using optimal passenger-vehicle assignments	Ma, Liser	IEEEIC/ICPS Europe	2017
	A framework for dynamic vehicle pooling and ride-sharing system	Farn et al.	IEEE(IWCI)	2016



Transportation demand	Trip Planning	Time Dependent	1
		Real time demand	2
		Trip hailing	3
		Authentication System	4
		Trip Motivation	5
		Special passenger	6
		Origin	7
		Destination	8
		Seat Demand	9
	Trip	Max waiting time	10
		Max Travel time	11
		Max Detour	12
		Max Fare	13
		Time window	14
		Desired pick up time	15
		Desired arrival time	16
		Max delay	17
		Max number of sharing	18
		Distance from origin to pick up	19
		Departure time	20
		Waiting time	21
		Number of Sharing	22
		Detour	23
		Arrival time	24
		Time for picking up and dropping off	25
		Fare	26
Transportation Service	Vendors	27	
	Fixed Fleet	28	
	Time table	29	
	Fixed stops (pick up or drop off locations)	30	
	Waiting locations	31	
	Car Specifications (homogeneous/heterogeneous)	32	
	Supply type (pro/not pro)	33	
	Car capacity	34	
	Pricing scheme	35	
Algorithm	Objective function	36	
	Dispatching	37	
	Searching	38	
	Scheduling	39	
	Monitoring	40	
	Method	41	
	Shortest path	42	
	Empty vehicle behavior	43	
	Accept/reject behavior of riders	44	
Traffic	45		
Application	Network	Characteristics	46
	Scale	Time	47
		Space	48
		Demand	49
		Supply	50
Criteria	Passengers' objectives	Waiting time	51
		Service Time	52
		Total travel time/distance	53
		Trip fare	54
		Rejected Requests	55
		Max delay	56
	Providers' objectives	Total travel time/distance	57
		Number of served requests	58
		Number of required vehicles	59
		Taxi stand departures	60
		Revenue per travel distance	61
		Average number of persons per vehicle	62
		Vehicle emissions	63
		Saved total travel distance percentage	64

Table A.2: List of characteristics

	P1	P2
1	Yes (dynamic)	Yes (dynamic)
3	Smart phones	
4	Yes	
5	Trips from home to work and return trips from work to home	
6	No(participants can be driver or rider)	
7		start location of the passenger
8		end location of the passenger
10	The participant provides an earliest time. So max waiting time is the difference between announcement time and earliest departure time	
11	The participant provides a time flexibility $f(s)$ that specifies the difference between earliest departure time and the latest time he would like to depart	
12	To minimize the inconvenience of the participants, in this research they limit the attention to systems where at most one pickup and delivery can take place during the trip	
14	Time window can be extracted	
15	Earliest departure time	Start time of the passenger
16	Time flexibility that specifies the difference between earliest departure time and the latest arrival time	End time of the passenger
26		Trip fare of the passenger
27		Taxis
28	No	Yes
30		No
32		Homogeneous
33	Not pro	Professional
35		To frame the pricing policy, they measure the participating willingnesses of drivers and passengers based on different levels of incentives, and evaluate the corresponding benefits from resulting IGRGs
36	The ride-share provider seeks to minimize total system-wide vehicle-miles, the total vehicle miles driven by all potential participants traveling to their destinations, either in a ride-share or driving alone if unmatched.	Maximizing the total saved travel mileage
40	No (only one sharing is possible)	
41	1- Fixed driver, rider roles: they can represent the ride-share problem using a maximum-weight bipartite matching model and then solve the problem using any linear programming or network optimization code. 2- Driver, rider role assignment.they use a general graph matching model. They compare the matching and integer programming methods described earlier with a strawman greedy algorithm. The greedy matching algorithm is a straightforward rule based approach that a ride-share provider could use to match riders and drivers without requiring more sophisticated optimization software	They provide answers to two essential questions of Taxi Group Ride (TGR) problem: (1) how much incentive should be provided to passengers and drivers to maximize the benefits of TGR, and (2) how to optimally group passengers who participate in TGR? Consequently, the study involves developing operation strategies as well as pricing policies for the TGR problem. they develop a fast exact algorithm for solution quality and a heuristic algorithm to compute the solutions for large scale problems. To frame the pricing policy, they measure the participating willingnesses of drivers and passengers based on different levels of incentives, and evaluate the corresponding benefits from resulting IGRGs
44	No	
45	They ignore any time-dependency in travel time caused by congestion	
46	Atlanta metropolitan region	
47		Morning peak hours (8:00–8:30 a.m.), off-peak hours (1:00–1:30 p.m.), and evening peak hours (6:00–6:30 p.m.) on both weekday (Wednesday) and weekend (Saturday).
53	*	
54	*	
57	*	
61		*

Figure A.1: Paper 1 and Paper 2

	P3	P4
1	No	Yes
2	No	
3		Smart phone
4		Yes
5	Common destination which occurs in large organizations such as companies, universities, hospitals, and government organizations	
7	Participants home	Origin location of the trip announcement
8	A common destination like university or company	Destination location of the trip announcement
10		Announcements have announcement time and latest possible departure time
14	The participants may indicate desired travel time windows	Earliest departure time and latest arrival time
15	Earliest departure time	
16	Latest arrival time	
17		Announcements have a latest possible arrival time at their destination
18	There is a constraint in their model to ensure that each vehicle is not assigned to more than 1 individual	
23		Detour has flexibility: Detour flexibility is the willingness of drivers to make a detour in order to accommodate riders. We assume the detour flexibility of driver is a function of the duration of his direct trip
25		This value reflects the time necessary to make a stop, getting in and out of the car, validation of the identity of the rider and the driver
28	Yes	No
30		No
31		No
32		Homogeneous
33		Not professional
34	4	
36	Minimizing the expected travel time	Maximize the number of matches and maximize the system-wide vehicle miles savings
39		Scheduling has flexibility in this research. Scheduling flexibility refers to the time available for a match to be found. The scheduling flexibility of a trip is the difference between the latest departure time and the announcement time. In the period between its announcement time and its latest departure time, a trip announcement can be matched. If no match has been found at the latest departure time the trip announcement leaves the system without a match
40		They do not allow en route matching so a match needs to be established before the driver departs
41	They want to find the minimum number of vehicles that is needed and also the areas that shared vehicles should be located and the tradeoff between the number of available vehicles and the savings in travel time. So they have modeled a stochastic optimization model for the car placement problem and then they propose a heuristic that can find solutions for large scale instances of the problem in limited computational time.	They introduce and define three different types of participant flexibility that are relevant in the context of dynamic ride-sharing: matching flexibility, detour flexibility, and scheduling flexibility; Then they quantify the impact of these types of flexibility on system performance by conducting an extensive computational study; and they investigate the required level of additional flexibility that is required to improve the effectiveness of a ride-share system
43		Participants are drivers and riders. they assume that drivers that are not matched, drive to their destination alone.
44	The participants that are assigned to the shared vehicles are encouraged to participate in the system by receiving a guarantee for a free taxi ride in the event where the vehicle that they are assigned to does not pick them up	
46	Rome	
48		500 to 5000 trip announcement
49	10 to 50	
50	4 to 20	
53	*	

Figure A.2: Paper 3 and Paper 4

	P5	P6
1	Yes	Yes
2		Yes
3	Smart phone	Calling a control center, using a mobile app, street hailing
4	Yes	No
7	A trip announcement specifies an origin	Passenger defines his own origin point
8	A trip announcement specifies an origin	Passenger defines his own destination point
9		Passenger defines the number of seats
10		Passenger group defines the max distance between his origin and pick up point
14	Each announcement provides an earliest departure time and a time flexibility that specifies the difference between earliest departure time and the latest departure time	The passenger defines the time window
15	earliest departure time	
16	latest arrival time	
18		Passenger group defines the max number of trips to be shared
19		Passenger group defines the max distance between his origin and pick up point
20		Computed by the algorithm
23		Passenger group defines the max distance that is willing to tolerate in addition to the actual trip time
24		Computed by the algorithm
27		Fitting well the models using different vendors
28		No (taxi sharing)
29		Schedule
32		Homogeneous: Passenger capacity, maximum number of shared trips, max wait time for pick up, extra time for drop off
33		Professional
34		C for all cabs (but it is possible to have different capacities)
36	Minimize total system-wide vehicle-miles, the total vehicle-miles driven by all potential participants traveling to their destinations	$f(r,c)$ :cost (metric) function that given a taxi $c$ and a pick-up request $r$ , returns the cost of accommodating $r$ with $c$ .
37		Dynamic assignment: consider each of the $n$ trips in chronological order. For each pick-up request $r_i$ , go through all the cabs in the fleet and update their states based on the time elapsed from the last trip's pick-up time. For each cab $c_j$ , they also compute additional cost $f(r_i, c_j)$ and assign $r_i$ to the cab with the lowest additional cost.
39		After dispatching, each cab has a set of stop points that should pass. After each drop off the state of the cab is updated and if a new request matches the extra capacity, the request will be added to the cabs schedule after assignment algorithm. Cabs have dynamic schedule in this case.
40		Yes (All the cabs are assigned in the same time to the requests and as a request added, the state of the cabs will be updated to assign to the new requests)
41	They introduce the concept of stability in the context of dynamic ride sharing and to provide mathematical programming approaches to solve stable and nearly stable ride-share matching problems. They experimentally quantify the impact of enforcing stability in a dynamic ride-sharing setting using simulations of representative work based trips for a major U.S. metropolitan area. To assess the stability of a matching solution, they report the following statistics: Participants in blocking pairs-Blocking pairs per participant- Unrealized savings	Using a heuristic to find a best route for $c_j$ to accommodate $r_i$ . (The heuristic name is not mentioned)
42	Travel time and travel distance between each pair of locations are known and constant	
43	Drivers and/or riders may reject a match if they believe they can establish a better match on their own. This is related to the notion of stability in cooperative game theory. In this terminology, a set of matches between riders and drivers is defined as stable if no rider and driver, currently matched to others or unmatched, would prefer to be matched together. If a driver and a rider form such a pair, they are called a blocking pair. If there are no blocking pairs in the matching solution, they call it a stable ride-share matching.	Taxi is a dynamic object and is always on the move. Even without any passenger, the taxi still has a destination where it drives to, for example to look for new passengers. Each taxi can define favorite locations at certain times of the day that it would drive to when it is not occupied.
45		Traffic conditions can be easily incorporated in the model by introducing weights on edges of the graph
46	Atlanta	NYC
47		2 Years (2011 and 2012)
48	6500 square miles	Manhattan and the two international airport
49		260 million rides
50		13237 taxis
61		*

Figure A.3: Paper 5 and Paper 6



	P7	P8
1	Yes	Yes
2		Yes
3	wireless communication and smartphones	Mobile devices
7	passenger current location	
8	passenger defines his own destination point	
9	passenger defines the number of seat demands	
10		the passenger defines his pick up time window
13		They have defined monetary constraints
14		The passenger defines the time window
19	Passenger define the service degradation factor which makes limitations on distance from his origin	
21	time elapsed between passenger's request and vehicle arrival to client's origin point	
22	the algorithm assigns as many sharing trips as possible (according to the capacity and all passengers service degradation factor)	
23	Passenger define the service degradation factor which makes limitations on detour	
24	passenger defines the service degradation factor (It is the sum of waiting time plus service time)	
26	Passenger can choose the min fare	
27	Not mentioned (just taxi)	Taxis
28	No (taxi sharing)	No
29		No
32	homogeneous: fleet size, capacity	homogeneous
33	professional	professional
34	C for all cabs	not mentioned (they have not consider the capacity)
35	the passenger pays a proportion (e.g. 0.6) of the variable charge for the trip whenever sharing a ride with other passengers. The sharing proportion can be defined according to the number of on-board passengers	They considered a pricing scheme based on two properties: (i) taxi fare per mile is higher for multiple passengers than for a single passenger; (ii) the taxi fare of shared distances is evenly split among the riding passengers.
36	minimisation of the total additional travel distance	The primary purpose is to investigate the potential of taxi ridesharing in serving more taxi ride queries by comparing with the case where no ridesharing is conducted. Additionally, they try to reduce the total travel distance of these taxis (so as to reduce the energy consumption) when doing the ridesharing
37	Dynamic assignment: the client makes a pickup solicitation using an application on a smartphone. The request is then received by all or a subset of moving taxis that compute the distributed vehicle algorithm and communicate the trip cost back to the user. This distributed algorithm is responsible for checking the route cost of associating a new request to the current service(s) for each taxi. The end user terminal receives the request responses, determines the least-cost admissible solution for taxi-sharing and selects a taxi for service. If there is no admissible sharing solution, the nearest available taxi is assigned.	On receiving a new query, the operation centre will dispatch an "appropriate" taxi which is able to satisfy both the new query and the trips of existing passengers who are already assigned to the taxi. The updated schedules and routes will be then given to the corresponding taxi driver and passengers.
38		They propose a taxi searching algorithm using a spatio-temporal index to quickly retrieve candidate taxis that are likely to satisfy a user query. They proposed two methods for searching: single-side taxi searching and dual-side taxi searching. the single-side searching algorithm simply tests all grid cells in the order preserved list and finds the first grid cell which fails to be before the late bound of the pick up window using the travel time between the two grid cells. this algorithm only considers taxis currently "near" the pickup point of a query. the dual-side searching is a bi-directional searching process which selects grid cells and taxis from the origin side and the destination side of a query simultaneously
39	Realistic input data is provided by the service scheduler, which emits service requests at the defined timestamp. The system's basic underlying block is a microscopic traffic model where the individual behaviour of each vehicle is mimicked. The other model named 'taxi operation' mimics the functioning of each vehicle in a taxi fleet.	Given the set of taxi statuses {Taxis} retrieved for a query Q by the taxi searching algorithm, the purpose of the scheduling module is to insert Q into the schedule of the taxi which satisfies Q with minimum additional travel distance. They use insertion feasibility check to insert a query Q into the schedule of a taxi status V.
40	No	Yes (a taxi uploads its status to the operation centre when joining in the ridesharing service, or when a passenger gets on or off the taxi, or at a frequency)
41	a 'travelling salesman problem (TSP)' with constraints (vehicle capacity and service degradation constraints) considering the current position as the start point and the final position as the last customer drop-off	They partition the road network using a grid. within each grid cell, the road network node which is closest to the geographical centre of the cell as the anchor node of the cell is chosen. The anchor node of a grid cell $g_i$ is thereafter denoted by $c_i$ . They pre-compute the distance, denoted by $d_{ij}$ , and travel time, denoted by $t_{ij}$ , of the shortest path on the road network for each anchor node pair $c_i$ and $c_j$ . The distance and travel time results are saved in grid distance matrix. Then the distance between any two arbitrary points equals to the distance between two corresponding anchor nodes.
42		The grid distance matrix provides an approximated distance of the road network shortest path between any two geographical points at the grid cell level.
43	cars wait in the fleet to be assigned (taxis usually return to the closest taxi stand after finishing a service where they wait for the next passenger.)	
44		
45		They suggest that a few advanced travel time prediction techniques (e.g., incorporating real time traffic conditions) can be applied to estimate the travel time.
46	The city of Porto	Beijing
47	24 hours	3 months
48	Area of 41,3 km <sup>2</sup> and road network size of 965km	
49	4417 rides	10 million queries
50	441 taxis	33000 taxis
51	*	
52	*	
53	*	*
54	*	
57	*	*
58	*	
59	*	

Figure A.4: Paper 7 and Paper 8

	P9	P10
1	Yes	Yes
2	Yes	No
3	E-hailing	
7	The system identifies the passenger location	Defined by the rider (passenger)
8	the passenger indicates his destination	Defined by the rider (passenger)
9	(They considered that each request is just one demand)	1
10	The passenger defines how long he will accept waiting to be picked up	included in trip time
11		rider's maximum travel time is specified
14	10 , 15 , 30 min	Computed denoting the earliest and latest pick up time and the earliest and latest delivery time
15		Each participant specifies an earliest departure time (both rider and driver)
16	passenger defines his desired arrival time	Each participant specifies an latest arrival time (both rider and driver)
23		Drivers define maximum travel time and distance for their trips and they are willing to detour to take some riders (there is no constraint on number of detours)
27	shared vehicles	
28	Yes (with size n)	
30	the vehicles can use bus stops and other stops	No
31	Yes (designated locations to wait for new request)	No
32	heterogeneous (but with the same capacity)	Homogenous
33	not professional	Not professional
34	p=8	Ck
35	No	
36	The goal of the paper is to evaluate the new transport mode in an urban area, particularly in regard to its impacts on urban traffic while taking into account a very detailed description of the mobility service's characteristics.	They model a multi objective optimization problem. The objectives are: 1. The total distance of vehicles' trips (to be minimized)-2. The total time of vehicles' trips (to be minimized)- 3. The total time of the trips of the matched riders' requests (to be minimized)- 4. The number of matched (served) riders' requests
37	In order to serve passenger requests, a fleet of multiple passenger ride-sharing units (MPRS units) are moved over the network according to the dynamic routing and scheduling.	
38	When the system receives the new service request, it checks the status of the vehicle fleet to determine whether any of the vehicles that are currently in service or idle can feasibly provide the new service. If in service, it determines if a detour can be made to collect the new customer and satisfy his/her conditions without violating the requirements of the customers currently on board.	
39	In the case that there is more than one vehicle available for that service, a Decision Support System determines which is the most profitable, both from the customer's point of view in terms of the quality of the system and from the system's point of view in terms of a cheaper service. Then the system assigns the service to the selected vehicle along with the most efficient route for providing it, including potential detours and changes in the previous routes	
40	Yes (The microscopic traffic simulator model updates the model state every half a second, specifically the positions of all vehicle classes considered: passenger cars, buses and ride-sharing vehicles)	
41	They focus on the use of dynamic insertion heuristic solutions based on feasibility and profitability concepts to provide online solutions to the proposed real time vehicle routing problem. This module contains three modules: Initial routing plan: to find the initial routing plan for the pre-booked demand - New customer request insertion algorithm - Feasibility check: to check if the previously assigned routes are feasible after a traffic condition update by recalculating the routes	They model the ride-matching problem with time windows in dynamic ridesharing and propose a genetic algorithm for solving it.
42	They have implemented a variation of the DOT (Decreasing Order of Time) time dependant shortest path algorithm proposed by Barceló et al. (2013). The shortest path problem considered in this research differs from the usual shortest path approaches in that they do not consider link travel times to be constant but instead time dependent; in other words, the link cost depends on the arrival time at the origin of the link, meaning that the considered shortest paths are dynamic or time-dependent (TDSP).	Haversin formula
43	When a new mode's unit is not operating (i.e., it is empty and waiting for the arrival of a new request), it goes and parks in the allowed zones until a new requested demand has to be served	
45	The simulator to be used must be able to realistically emulate the time evolution of traffic flows on the road network by properly describing the dynamics of individual vehicles.	
46	Barcelona	Northeastern Illinois
47	30 min	6 day
48	7.46 km <sup>2</sup>	250 drivers and 448 riders
49	1000 requests	
50	250 , 500 , 750, 1000	
51	*	
53	*	
55	*	
57	*	
60	*	

Figure A.5: Paper 9 and Paper 10



	P11	P12
1	No (static)	Yes
2		Yes
3	Smart phones (e-hailing application)	
6		
7	Passenger defines the origin	The passengers register their trip requirements as a pair of start and destination PuPs rather than their exact locations
8	Passenger defines the destination	
13		Lowest cost that each passenger has in mind
14	A parameter for the quality of service (delta)	
18		A driver can only pick up three additional passengers (assuming the car capacity of every driver is 4)
20		Negotiable
22	The shareability parameter (k)	
23		Every driver makes a detour while picking up passengers as long as the new trip is within the time constraints
24	The travel time between arbitrary intersections is estimated using an ad hoc heuristic	The sum of the time taken to reach the destination using the shortest path (with an overhead randomly chosen of 10-20 percent to reflect different traffic situations)
26		Negotiable
27	Taxis	Personal cars
30	No	Yes (a model for selection of fixed predefined locations for ride sharing which guarantees anonymity while facilitating ride sharing)(the first research with this
32	Homogeneous	
33	Professional	Not pro
36	Two optimization criteria: (i) maximizing the number of shared trips, or (ii) minimizing the cumulative time needed to accommodate all trips (Their approach is to find a mathematical framework for the systematic understanding of the tradeoff between collective benefits of sharing and individual passenger discomfort)	The goal is to find an optimal number and placement of pick up points such that it maximizes coverage while facilitating ride sharing with maximum privacy for users. Since coverage and anonymity (privacy) are conflicting objectives we evaluate different coverage solutions for different levels of anonymity and compute Pareto solution sets
40	No (the method only concerns the sharing of nonvacant trips)	
41	With the shareability network, classical algorithms for solving maximum matching on graphs can be used to determine the best trip-sharing strategy according to two optimization criteria. it is sufficient to compute a maximum matching or a weighted maximum matching on the shareability network	They have used greedy randomized adaptive search procedure (GRASP). It is an iterative procedure consisting of two phases. The first phase is a greedy randomized construction phase where a feasible solution is created. The second phase is a local search phase where the neighborhood of the solution created in phase one is investigated to find a local minima. The best overall solution is kept after every iteration. In every iteration of the procedure GRASP, they compare the solution with the solutions having the same number of PuPs and keep the one with the best coverage.
45		Just to compute the end time for a driver. It is the sum of the time taken to reach the destination using the shortest path (with an overhead randomly chosen of 10-20 percent to reflect different traffic situations) and the extra time the driver is willing to travel to pick up passengers.
46	New york, San Francisco, Vienna, Singapore	Melbourne
47	2011 (one year)	10 different simulation (each of them 2 h)
49	150 million trips	15000 passengers (for the first simulation)
50	13586 taxis, 500 taxis, 16000 taxis, 1000 taxis	5000 drivers (for the first simulation)
56	*	
57		Average vehicle travel distance (VTD)
60		Ridesharing occupancy (occ)

Figure A.6: Paper 11 and Paper 12

	P13	P14	P15
1	No (static)	Yes	Yes
2		Yes	Yes
3		Smart phone	Web browser and smartphones
5	Serving passengers and parcels		
7	Passenger defines the origin		Passenger current location
8	Passenger defines the destination		Passenger defines his own destination point
9	Passenger defines the number of seat demand		Passenger defines the number of seat demands
10		The user defines the max waiting time. 3 , 5 , 7 min	
12	There are constraints in the model to avoid long detour for the passenger due to parcel delivery		
14	Each request has a time window (20 min)		
15		The user defines the request time or desired pick up time	
19			No constraint
22			The algorithm assigns as many sharing trips as possible
23	There are constraints in the model to avoid long detour for the passenger due to parcel delivery		No constraint
24	It is insured by the travel time constrains (penalty will be added when excess the direct travel time)		No constraint
26			Passenger can choose the min fare
27	Taxis		All types of vehicle (car, boss, lorry, taxi, motorbike...)
28	No	Initial departure depots	No
29	Yes (statistic)	Yes (for mass rapid transit (MRT) system)	
30	No		
31			
32	Homogeneous	Homogeneous	Heterogeneous : Type, Capacity
33	Professional		Professional
34	5	K	
35	A fixed price and a variable price per kilometers		User royalty ( user earns royalty points for using the system that can use them for payment purpose), a prepaid system (rider has to ensure payment before getting into the ride)
36	Objective function is in two stage. The first stage for share a ride problem with stochastic travel time, maximizes the total profit obtained from people and parcel delivery, and includes three parts: (i) the profit obtained from passengers; (ii) the profit obtained from parcels; and (iii) the cost of the distance traveled and with the optimal value of the second stage. the second stage function sums the expected costs of violating the time windows and ride time constraints.	Minimizing the total waiting times and travel time delays of the entire bi-/multi-modal trips	Min waiting time and min traffic
37		When the dispatching center receives a request, it transforms the request to a pre-transit trip request at a first step. The pre-transit trip request, is defined by its origin $oi$ , the entry station $si'in$ and the desired pickup time $tr$ . As soon as a pre-transit trip request is served, the dispatching center estimates passenger's arrival times at his exit station $si'out$ , and generates a post-transit trip request, where $si'out$ is the exit station (pick-up points), $di$ is user's destination, and $ti'out$ is user's desired pick-up time, assuming to be expected arrival time at next pickup points (exit stations).	Dynamic assignment: the client makes an account and sends the request the server. The server computes the euclidean distance between requested type of vehicles and user. The server shows the closest vehicles to the client. Passenger can choose the vehicle. passenger requests are shown to the drivers. the driver chooses the passenger among request. the server assigns the vehicle to the passenger and passenger has to pay for the ride
39			When the driver see the set of requests in the server, he can choose among the requests and then the travels sequence is the requests that are added by the driver and are selected by the passengers too.
40	No		No
41	The solution methodology integrates an adaptive large neighborhood search heuristic and three sampling strategies for the scenario generation (fixed sample size sampling, sample average approximation, and sequential sampling procedure)		
46	San Francisco	An area between luxembourg city and its french-side cross-border area	Dhaka city (Bangladesh)
47	8 hours		
49	The instances include 30-75 requests		
51			*
59	*		*

Figure A.7: Paper 13, Paper 14 and Paper 15