DISS. ETH NO.30053

OPTIMIZATION OF SHARED ON-DEMAND TRANSPORTATION

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by

ZAHRA GHANDEHARIOUN
MSc., Technical University of Munich

born on 21. March 1987

accepted on the recommendation of

Prof. Dr. Kay W. Axhausen, examiner
Dr. Anastasios Kouvelas, co-examiner
Prof. Dr. Francesco Corman, co-examiner
Prof. Dr. Monica Menendez, co-examiner
Dr. Alessio Trivella, co-examiner
Dr. Michail Makridis, co-examiner

2024

DOI: 10.3929/ethz-b-000680076

To the loving memory of my father

ABSTRACT

Nations worldwide are experiencing significant urban growth, with over half of the global population currently residing in cities. This urbanization has increased urban commuting, leading to issues like congestion, air and noise pollution, and threats to public health. Recent years have witnessed a transformation in transportation driven by information technology, introducing innovative mobility solutions to tackle urban mobility challenges. These innovations encompass on-demand and shared mobility services, enhancing transportation efficiency and convenience. The integration of these solutions with public transit holds the potential to revolutionize the entire transportation system. Overcoming these challenges requires a comprehensive evaluation of the advantages and disadvantages of shared mobility services, ensuring a shift toward sustainable mobility in the future. This thesis aims to explore the optimization of on-demand transportation services in urban areas by employing methods in three aspects.

The first part of this thesis analyzes historical travel time data from on-demand transport services, like taxis, to gain insights into traffic patterns and estimate arterial travel time precisely. It introduces a novel methodology that uses sparse GPS probe data and considers spatial correlations between network links. This research demonstrates the improved accuracy of travel time estimation by factoring in progressive spatial correlations. A case study in a partial network of New York City, using taxi data, shows enhanced travel time estimation accuracy, benefiting urban traffic optimization and congestion identification.

The second part of this thesis centers on optimizing on-demand services with a real-time shuttle ridesharing algorithm. This novel algorithm efficiently matches ride requests to a fleet of vehicles, using a flexible simulation framework that adapts to different scenarios and incorporates real-time traffic data. By focusing on fleet capacities and tolerance times, the study shows that a reduced number of high-capacity taxis, along with optimized operational policies, significantly reduces waiting times and in-car delays for Manhattan taxi rides.

The final part of this thesis focuses on developing precise short-term demand forecasting models for on-demand services, with an emphasis on deep learning techniques. It seeks to enhance prediction accuracy, investigate data granularity's impact, explore temporal and spatiotemporal

variables, compare the model's performance with traditional and complex machine learning methods, and highlight the benefits of spatiotemporal considerations and vector embedding for improved prediction accuracy.

The research presented in this thesis offers valuable implications for both research and practical applications. First, accurate estimates and predictions of travel times for urban links are crucial for optimizing urban traffic operations and identifying traffic congestion points. Providing precise travel time information offers benefits to users and operators by enabling them to choose better paths within the network and reduce overall travel time. Second, the potential of ridesharing services, optimized in real-time with dynamic traffic data, is shown by the proposed modular framework, together with the novel matching algorithm in Chapter 3. The importance of system parameters and tailored operational policies to improve urban transportation systems gives valuable insight for the design of such services for operators. Moreover, the precise demand prediction can help the operators plan the fleet dispatching more efficiently.

ZUSAMMENFASSUNG

Weltweit erleben die Länder ein erhebliches Wachstum der Städte, und mehr als die Hälfte der Weltbevölkerung wohnt derzeit in Städten. Diese Verstädterung hat zu einer Zunahme des städtischen Pendlerverkehrs geführt, der Probleme wie Staus, Luftverschmutzung und Lärmbelästigung verursacht und die öffentliche Gesundheit gefährdet. In den letzten Jahren hat sich das Verkehrswesen durch die Informationstechnologie verändert, und es wurden innovative Mobilitätslösungen eingeführt, um die Herausforderungen der städtischen Mobilität zu bewältigen. Diese Innovationen umfassen On-Demand- und Shared-Mobility-Dienste, die die Effizienz und den Komfort des Verkehrs erhöhen. Die Integration dieser Lösungen in den öffentlichen Verkehr hat das Potenzial, das gesamte Verkehrssystem zu revolutionieren. Die Bewältigung dieser Herausforderungen erfordert eine umfassende Bewertung der Vor- und Nachteile von gemeinsam genutzten Mobilitätsdiensten, um einen Wandel hin zu einer nachhaltigen Mobilität in der Zukunft zu gewährleisten. Diese Arbeit zielt darauf ab, die Optimierung von On-Demand-Verkehrsdiensten in städtischen Gebieten zu erforschen, indem Methoden in drei Aspekten eingesetzt werden.

Im ersten Teil dieser Arbeit werden historische Reisezeitdaten von On-Demand-Verkehrsdiensten, wie z. B. Taxis, analysiert, um Einblicke in die Verkehrsstrukturen zu gewinnen und die Reisezeit auf den Arterien genau zu schätzen. Es wird eine neuartige Methodik vorgestellt, die spärliche GPS-Sondendaten verwendet und räumliche Korrelationen zwischen Netzverbindungen berücksichtigt. Diese Forschung zeigt die verbesserte Genauigkeit der Reisezeitschätzung durch die Berücksichtigung progressiver räumlicher Korrelationen. Eine Fallstudie in einem Teilnetz von New York City unter Verwendung von Taxidaten zeigt eine verbesserte Genauigkeit der Reisezeitschätzung, die der Optimierung des Stadtverkehrs und der Erkennung von Verkehrsstaus zugute kommt.

Der zweite Teil dieser Arbeit konzentriert sich auf die Optimierung von On-Demand-Diensten mit einem Echtzeit-Shuttle-Ridesharing-Algorithmus. Dieser neuartige Algorithmus ordnet Fahrtwünsche effizient einer Fahrzeugflotte zu und verwendet einen flexiblen Simulationsrahmen, der sich an verschiedene Szenarien anpasst und Echtzeit-Verkehrsdaten einbezieht. Durch die Fokussierung auf Flottenkapazitäten und Toleranzzeiten zeigt die Studie, dass eine reduzierte Anzahl von Taxis mit hoher Kapazität zusam-

men mit optimierten Betriebsrichtlinien die Wartezeiten und Verspätungen im Fahrzeug für Taxifahrten in Manhattan deutlich reduzieren.

Der letzte Teil dieser Arbeit konzentriert sich auf die Entwicklung präziser kurzfristiger Nachfrageprognosemodelle für On-Demand-Dienste, wobei der Schwerpunkt auf Deep-Learning-Techniken liegt. Ziel ist es, die Vorhersagegenauigkeit zu verbessern, die Auswirkungen der Datengranularität zu untersuchen, zeitliche und räumliche Variablen zu erforschen, die Leistung des Modells mit traditionellen und komplexen Methoden des maschinellen Lernens zu vergleichen und die Vorteile räumlicher Überlegungen und der Vektoreinbettung für eine verbesserte Vorhersagegenauigkeit hervorzuheben.

Die in dieser Arbeit vorgestellten Forschungsergebnisse haben wertvolle Auswirkungen sowohl auf die Forschung als auch auf praktische Anwendungen. Erstens sind genaue Schätzungen und Vorhersagen von Reisezeiten für städtische Verbindungen von entscheidender Bedeutung für die Optimierung des städtischen Verkehrsbetriebs und die Identifizierung von Staupunkten. Die Bereitstellung präziser Reisezeitinformationen bietet Nutzern und Betreibern Vorteile, da sie bessere Wege innerhalb des Netzes wählen und die Gesamtreisezeit reduzieren können. Zweitens wird das Potenzial von Ridesharing-Diensten, die in Echtzeit mit dynamischen Verkehrsdaten optimiert werden, durch den vorgeschlagenen modularen Rahmen zusammen mit dem neuartigen Matching-Algorithmus in Kapitel 3 aufgezeigt. Die Bedeutung von Systemparametern und maßgeschneiderten Betriebsstrategien für die Verbesserung städtischer Verkehrssysteme gibt den Betreibern wertvolle Hinweise für die Gestaltung solcher Dienste. Darüber hinaus kann die genaue Vorhersage der Nachfrage den Betreibern helfen, die Disposition der Flotte effizienter zu planen.

ACKNOWLEDGEMENTS

In embarking on the journey of doctoral research, one quickly realizes that the path to academic achievement is rarely walked alone. With deep gratitude and humility, I extend my acknowledgments to the individuals and institutions that have played pivotal roles in shaping my academic pursuits and enabling the successful completion of this doctoral thesis. This dissertation stands as a testament to the collaborative spirit that underpins the pursuit of knowledge, and it is my privilege to acknowledge those who have made it possible.

I would like to extend my appreciation to Dr. Anastasios Kouvelas for granting me the opportunity to pursue my doctoral studies as one of his first PhD students. Your belief in me and the guidance you provided throughout these years have been instrumental in my academic journey. I appreciate all the valuable input you shared, which has shaped my research and my growth as a scholar.

I am also thankful to Prof. Axhausen for his support during my PhD, which significantly contributed to my academic development. I extend my sincere thanks to Prof. Francesco Corman for serving as my second advisor and for the invaluable support and mentorship he provided. I would like to thank Dr. Michail Makridis for his feedback and support during the later years of this journey.

A special mention of gratitude goes to Dr. Shima Sadat Mousavi for her support during the most challenging phases of my doctoral journey. Her guidance helped me overcome unforeseen obstacles and ultimately achieve my doctorate.

I would also like to thank Kimia Chavoshi for the enriching conversations that made the difficult days more bearable and the good days even brighter. Additionally, I am grateful to Dr. Alexander Genser for being a supportive colleague throughout these years. To all my colleagues at the institute, I appreciate the engaging discussions and friendship we shared during social events, making this academic journey all the more enjoyable.

I wish to thank my mother for her encouragement, which motivated me to strive harder toward my goals during this journey. Additionally, I want to convey my gratitude to my partner, Patrik. Thank you for being by my side throughout this incredible journey.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

As far as the laws of mathematics refer to reality, they are not certain; as far as they are certain, they do not refer to reality.

— Albert Einstein

1.1 BACKGROUND AND CONTEXT

Nations around the globe are experiencing significant urban growth: Currently, over half of the global population resides in cities, and this figure is projected to climb to 68% by 2050 (United Nations, 2018). The process of urbanization contributes to a rise in urban commuting, thereby resulting in amplified consequences associated with this travel. Urban areas have been experiencing significant rises in congestion, as well as air and noise pollution levels. Such increases pose serious threats to public health (Khreis et al., 2016), as well as diminishing the overall quality of life for city residents (Künzli et al., 2000). The transport mode that plays the most prominent role is the individual private vehicles (Steg & Gifford, 2005), which have a major impact on city planning due to the need for large infrastructure dedicated to parking and driving. Cars have become highly popular due to their flexibility and efficiency, namely their ability to transport individuals quickly and comfortably from their origin to their destination at the time of their choosing. However, flexibility and comfort come at a cost, as private cars contribute to traffic congestion and pollution, making it necessary to consider alternative modes of transportation to alleviate the situation.

One transportation mode that can complement or replace private vehicle use is public transportation, which offers numerous advantages for social welfare, such as promoting public health (Litman, 2012), reducing greenhouse gas emissions (Chester et al., 2013; Ercan et al., 2016; Peng et al., 2015) and creating employment opportunities (Tyndall, 2017; Johnson et al., 2017). Additionally, it allows individuals who cannot afford to own a car to travel. Unfortunately, the availability of public transportation is often limited to time and space, and its routes and schedules may not be optimal for everyone, leading to longer travel times and requiring people to walk

to access the service. However, it is crucial to acknowledge that public transportation can only attract a portion of travelers, In modern societies, cars have become more popular, primarily due to practical and economic reasons. Families often rely on cars for their convenience and the ability to transport multiple members together. Additionally, cars have become a status symbol, representing financial stability and success.

The way people move around cities has been significantly transformed by the emergence of the information technology era, making mobility more convenient and secure than ever before. The last few decades have witnessed a notable increase in innovative mobility solutions such as on-demand services or shared mobility services aimed at addressing current mobility challenges (Fulton et al., 2017). The transportation sector has been revolutionized by the latest advancements in information technology, leading to unprecedented levels of accessibility due to the availability of more on-demand services at different times of the day and improving safety in mobility through tracking services (Sperling, 2018). With the advent of innovative mobility solutions such as Mobility as a Service (MaaS) and the introduction of higher-frequency services and innovative ticketing systems in public transportation, along with the increased availability of on-demand services and car sharing, transportation options have significantly improved to meet the changing needs of people in today's society. These advancements aim to make transportation more efficient and streamlined, enabling individuals to navigate through cities with ease (Sperling, 2018).

Combining innovative mobility solutions (e.g., carpooling, carsharing, bikesharing, and ridesharing [1]) with Mobility as a Service (MaaS) and public transportation has the potential to revolutionize the current transportation landscape and bridge the gaps between existing modes of transportation. By facilitating the sharing of resources, such as vehicles and routes, this approach can reinforce and extend the benefits of public transportation while retaining some of the advantages of private vehicle use. Consequently, sharing resources is an essential strategy for achieving a resilient, efficient, and low-carbon transportation system in the future.

Sperling, 2018 highlighted that the advent of electrification and automation could lead to negative consequences in the mobility sector unless they

1 In this thesis Carpooling means sharing one's private car to a commitment where two or more drivers agree to drive all members of the carpool on alternating days for a period of time (Julagasigorn et al., 2021). Ridesharing is an arrangement where passengers can book rides with drivers in their taxis or private vehicles, typically using an online platform. A formal agreement, for example for splitting travel costs, may or may not exist between ridesharing participants, and this mode of commuting may be used on a regular or occasional basis (Amirkiaee & Evangelopoulos, 2018).

are appropriately integrated with sharing policies and incentives. Conversely, a successfully shared mobility revolution with pooling as a core element could substantially enhance the efficiency of transportation systems, reducing the number of vehicles needed, parking requirements, and greenhouse gas emissions (Simonetto et al., 2019; Alonso-Mora et al., 2017; Greenblatt & Saxena, 2015). The significance of shared mobility is well-recognized by cities and major mobility actors, such as car manufacturers, who are now focusing on developing policies, conducting experiments, and research to accomplish this objective. However, the widespread adoption of shared mobility requires individuals to abide by the regulations of these new systems and alter their transportation habits. Moreover, they need to be willing to relinquish their private vehicles and accept that doing so might entail sacrificing some of the comfort, privacy, safety, and flexibility that come with them. Overcoming these challenges necessitates a comprehensive evaluation of the benefits, drawbacks, and trade-offs of shared mobility services to ensure the transition towards sustainable mobility in the future (Hyland & Mahmassani, 2020).

This thesis aims to explore the optimization of shared on-demand transportation services in modern cities by utilizing various approaches. Chapter 2 investigates what can be achieved through the analysis of historical travel time data, specifically obtained from on-demand transport services, particularly taxi companies. The goal is to gain insights into traffic patterns and obtain precise estimates of arterial travel time. Precise travel time estimations can contribute to better planning and operation of various mobility services, as well as provide more convenience for the users. With the results from the second Chapter, we make a groundwork for Chapter 3. In Chapter 3 a real-time shuttle ridesharing algorithm is developed to optimize on-demand ridesharing needs by considering real-time traffic information, based on the precise travel time estimation method introduced in Chapter 2. The ridesharing algorithm introduces a new formulation for optimally matching the ride requests to a fleet of vehicles. Finally, Chapter 4 explores a range of methods, including deep learning approaches, for forecasting short-term passenger demand for on-demand transportation services, which can facilitate efficient traffic supply and demand coordination. The fourth chapter's demand prediction methods can enhance the accuracy of short-term passenger demand forecasting for ridesharing platforms and can contribute to optimal planning of the fleet (dispatching) for the algorithm introduced in Chapter 3. In the following, the main topic of each chapter is addressed in more detail.

1.1.1 *Travel time estimation*

Over the past five decades, a variety of sensors have been developed to capture different types of traffic information. In general, traffic data comprises flows, which measure the number of vehicles passing through a given location in a specific time frame; density, which records the number of vehicles per unit of distance; occupancy, which represents the percentage of time that a vehicle occupies a specific location and is closely linked to density; velocity, which quantifies the distance traveled by a vehicle in a specific time period; and travel time, which measures the time it takes to move from one location to another. Vehicle trajectories, which represent the sequence of discrete-time and location pairs for each vehicle, can also provide valuable data. With a location-reporting frequency of several seconds or less, travel times and short-distance velocities can be calculated directly from vehicle trajectory data. However, when the location-reporting frequency is greater than ten seconds, accurately measuring travel times and velocities becomes challenging.

Accurate estimation and forecasting of travel times for urban links are essential for optimizing traffic operations and identifying major bottlenecks in the traffic network. Providing precise travel time information can benefit users and operators by allowing for improved path selection within the network and reducing total trip traveling time. To correctly estimate link travel times, real-time information from in-road sensors such as loop detectors, microwave sensors, or roadside cameras, mobile sensors (e.g., floating vehicles), or global positioning system (GPS) devices are necessary.

At present, the most ubiquitous data source for traffic information on arterial networks is sparsely-sampled probe GPS data. This term refers to the situation where probe vehicles send their current GPS location at a predetermined frequency, which is insufficient to measure velocities or link travel times directly (i.e. when the sampling frequency is greater than 10 seconds). GPS-equipped taxicabs have become increasingly ususal in metropolitan regions in recent years. While these cabs have many benefits, such as providing accurate directions to passengers, they also serve as valuable real-time probes for the traffic network. Taxis equipped with GPS devices collect a vast amount of data over days and months, providing a rich data supply for calculating network-wide performance indicators. This type of data poses several challenges. Firstly, GPS measurements must be matched to the road network representation used by the traffic information system, requiring the determination of both the precise location on the road

and the path between successive measurements. This is referred to as map matching and path inference. Secondly, probe vehicles may travel through multiple links between measurements when the sampling frequency is low, making it necessary to infer the probable travel times for each link of the path. This is a component of the traffic estimation algorithm explained in Chapter 2.

1.1.2 *Ridesharing optimization*

Urban transportation networks are under increasing strain, and they need creative ways to boost their efficiency. With the quick adoption of innovative mobility services, intelligent transportation systems have recently changed the way conventional transportation is provided. Ridesharing is one of these services that is gaining popularity. Over the years, the phrase "ridesharing" has been defined in a variety of ways. One of the first definitions was given by the State of Virginia in the United States in 1989: “Ridesharing 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.” (Code of Virginia, 2015). Following that, other scientific studies described and investigated ridesharing systems (Haselkorn et al., 1994; Burris & Winn, 2006; Kelly, 2007).

Ridesharing is defined by Agatz et al., 2010 as a system that tries to connect passengers with matching routes and timetables. They emphasize that for ridesharing to be extensively used, it must be simple, safe, adaptable, effective, and affordable. It must also be able to compete with one of the primary preferences of private users, particularly immediate access to door-to-door transportation. Wang et al., 2018 presented one of the most recent definitions" Ridesharing is an emerging transport mode that harnesses both private cars and taxis to combine two (groups) travelers into the same vehicle if all or part of the two groups’ travels is overlapped in space and time".

The Information Technology (IT) revolution is transforming many facets of modern life, including transportation (Golob & Regan, 2001). All transportation providers are now able to adjust their transportation supply to passenger demand in real time because of the general ownership of mobile devices and the advances in the global positioning system. Both taxis and other transportation modes have undergone massive change as a result of these new technologies (Srinivasan & Raghavender, 2006). With

the help of these features, it will be possible to match ridesharing partners in real-time and to have access to a vehicle's position at any moment. These possibilities have sparked the growth of a brand-new ridesharing model known as dynamic ridesharing. Dynamic ridesharing, which is also known as real-time ridesharing, and real-time peer-to-peer ridesharing, is a transportation mode that offers rides for single, one-way trips. In dynamic ridesharing, the sharing is set up for each journey rather than for trips that are taken on a regular basis (Casey et al., 2000). In order to match users, dynamic ride-sharing systems must enable arbitrary locations and travel times (Siddiqi & Buliung, 2013; Dailey et al., 1999). Amey, 2010 offers the most recent definition as: "A single or recurring rideshare trip with no fixed schedule, organized on a one-time basis, with the 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." [2]

The passenger, the ride provider, and the matching algorithm are the three primary components of this dynamic ride-sharing system. The customer requests a ride that will pick her or him up at the point of origin and drop him or her off at the final destination in a certain amount of time. The transportation service has a fleet of vehicles (taxi, van, self-driving car, etc.) available to accommodate the clients' needs. The matching algorithm seeks for the best matches quickly after receiving requests and fleet data.

The efficiency of such a system is greatly influenced by the performance of the matching algorithm. Using effective algorithms, it is feasible to deliver the best real-time match between the vehicle and the passenger. Numerous investigations, like those by (Alonso-Mora et al., 2017; Simonetto et al., 2019), concentrated on finding such an algorithm.

Congestion reduction is one of the primary benefits of ride-sharing that has been discussed by the majority of research in this field (Carey, 2016). The crucial factor that has not been extensively studied is how dynamic traffic conditions in the network can have significant impacts on the ride-sharing services as well. In Chapter 3 of this thesis a new formulation of real-time ridesharing operation, considering dynamic travel time information is described.

2 In static ridesharing, all driver and rider requests are known upfront, while in dynamic ridesharing, new requests can emerge throughout the designated time period (Javidi et al., 2021).

1.1.3 *Demand prediction*

Forecasting the future has always been a challenge for humanity, but through the analysis of past data and events, it is possible to predict the possible future values of a phenomenon. While sciences like physics and engineering can establish universal laws with certainty, economic and social sciences, including transportation, face uncertainty, and subjectivity. Transportation systems enable the movement of people and goods across various distances, facilitating work, trade, and exploration. Transport demand models are designed to forecast the demand for transportation services, taking into account various factors such as price, travel time, convenience, and individual preferences. These models consider both objective aspects (technical and physical characteristics) and subjective aspects (economic, social, and psychological factors) to establish correlations between demand and its influencing factors. By understanding and forecasting transport demand, authorities and service providers can optimize their operations and enhance overall efficiency (Profillidis & Botzoris, 2018).

Transport demand modeling relies on two fundamental computational tools: statistics and computational intelligence. Statistics enable the analysis of past data and the formulation of equations that accurately describe the evolution of this data, facilitating the forecast of future transport demand. The equations in a transport demand model can consider various factors beyond time that influence demand. Computational intelligence, on the other hand, utilizes artificial intelligence techniques such as neural networks and fuzzy methods when statistical techniques fall short in accurately simulating a problem. As a relatively new field within transport science, demand modeling has evolved alongside the development of large infrastructure projects since the 1960s. While qualitative and simple statistical methods were once the norm, advancements in computer science and mathematical thinking have expanded the range of methods and techniques available for analyzing transport data and forecasting their future trends. Consequently, transport forecasters have a wide array of methods at their disposal to address any transport demand-related problem.

Accurate transport demand forecasts are essential for effective planning, investment, and operation of transportation systems. In today's competitive economic environment, reliable predictions of transport demand are crucial. Without relying on precise forecasts, decisions regarding infrastructure construction and operation of transport services are at risk of becoming uneconomic ventures and potentially leading to financial disaster. There-

fore, the availability of the most accurate transport demand forecasts is paramount for successful and sustainable transportation-related activities.

On-demand transportation systems, with their app-based platforms and real-time data collection, can significantly be advantageous for demand prediction. These systems generate vast amounts of valuable data, including trip origins and destinations, pick-up and drop-off times, travel distances, and user preferences. By analyzing this rich dataset, demand prediction models can gain insights into passenger behavior, travel patterns, and trends. This information can then be utilized to make accurate forecasts of future demand for transportation services. The availability of real-time data also allows for dynamic adjustments in service provision, enabling transportation providers to optimize resource allocation, fleet management, and routing decisions. Additionally, on-demand transportation systems foster increased user engagement and participation, as customers actively interact with the platform to request rides and provide feedback. This user engagement further enriches the data pool and improves the accuracy of demand prediction models, ultimately leading to more efficient and responsive transportation services.

Chapter 4 of this thesis focuses on exploring short-term demand prediction methods with an emphasis on machine learning approaches. It is presented that a short-term demand forecast helps coordinate traffic supply and demand. It is crucial for on-demand transport services to predict short-term demand, as it encourages relocation of empty cars from oversupplied to under-supplied areas. However, forecasting short-term passenger demand can be challenging due to spatial, temporal, and external dependencies.

1.2 RESEARCH OBJECTIVES

The objective of this thesis is to optimize on-demand shared transportation services in contemporary urban areas by employing three distinct methodologies. This initiative is driven by the growing need for environmentally sustainable, economically viable, and readily available services. Furthermore, the availability of vast amounts of data observations can offer unique opportunities for better understanding, prediction, operation, and control of traffic-related issues. The research focuses on addressing three aspects:

(a) Supply level: Analyzing historical data to gain insights into traffic patterns and accurately estimating travel times between different locations by considering spatial correlations.

(b) Operational level: Introducing a novel formulation of a real-time on-demand ridesharing algorithm that incorporates real-time traffic information.

(c) Demand level: Exploring methods for short-term demand forecasting for on-demand services, taking into account spatial, temporal, and external dependencies.

The thesis research objectives regarding the identified aspects are as follows:

- **Objective 1**: **Improving Link Travel Time Estimation Using Sparse GPS Probe Data and Spatial Correlations**
  The development of a method for understanding and predicting complicated city traffic patterns using sparse GPS probe data obtained from on-demand services. The focus is on allocating travel time data to different links traveled between GPS observations, taking into account the progressive spatial correlations within the network. The main goal is to demonstrate how considering these spatial correlations can lead to more realistic and improved results compared to existing parametric methods.

- **Objective 2**: **Development of an Efficient Real-Time Matching Algorithm for On-Demand Ridesharing: Optimizing Urban Mobility and Congestion Reduction**
  The development of a computationally efficient and real-time matching algorithm for on-demand ridesharing in urban mobility. Proposing a simulation framework for testing and evaluating the algorithm's performance, taking into account dynamic congestion by updating travel times of road segments during the simulation. The proposed algorithm aims to solve the ridesharing assignment problem as a combinatorial optimization task, with a focus on reducing computational complexity and search space through the introduction of heuristics.

- **Objective 3**: **Accurate Short-Term Passenger Demand Forecasting for On-Demand Transportation: Exploring Deep Learning Approaches and Spatiotemporal Dependencies**
  To explore and investigate various methods, particularly deep learning approaches, for accurately forecasting short-term passenger demand in on-demand transportation service platforms. The focus is on addressing the challenges posed by spatial, temporal, and exogenous dependencies that make short-term demand forecasting complex. Analysis of more than twenty methods on a taxi data set and exam-

ining different levels of temporal aggregation and their impact on architectural configurations.

1.3 RESEARCH CONTRIBUTIONS

The research contributions of this thesis are presented in accordance with the aforementioned objectives.

Contributions to achieving Objective 1:

- **Contribution 1:** Proposing a method that utilizes sparse GPS probe data to allocate travel time data to different links between GPS observations.
- **Contribution 2:** Consideration of progressive spatial correlations and a demonstration of the benefits of considering correlations and how they can enhance the results compared to existing parametric methods.
- **Contribution 3:** Application of the methodology to a partial network of New York City, utilizing data collected from taxicabs. Through the estimation of link travel times using our proposed method, the enhanced travel time estimation accuracy when compared to conventional parametric approaches is presented.

Contributions to achieving Objective 2:

- **Contribution 1:** Development of a highly modular real-time simulation framework specifically designed to address the complexities of the capacitated ridesharing problem, allowing for flexible and customizable simulations that can accommodate various scenarios and system configurations.
- **Contribution 2:** Formulation of the ridesharing problem as a dynamic deterministic on-demand matching problem, considering the inclusion of tolerance times to enhance the matching process.
- **Contribution 3:** Implementation of dynamic congestion within the simulation framework. This is achieved by regularly updating the travel times of links in the network during the simulation horizon, taking into account the evolving traffic conditions and their impact on the ridesharing operations.

- **Contribution 4:** To solve the optimization problem posed by the ridesharing problem in an online manner, a combination of heuristic algorithms and commercial solvers is employed.
- **Contribution 5:** Modeling of multiple objectives and the design of policies that aim to achieve efficient and mutually beneficial solutions for all stakeholders involved in the ridesharing system.

Contributions to achieve Objective 3:

- **Contribution 1:** Developing accurate forecasting models for short-term demand prediction specifically for on-demand services.
- **Contribution 2:** Investigating various levels of data aggregation within the input data and examining the impact of these levels on the prediction outcomes, shedding light on the relationship between data granularity and prediction accuracy.
- **Contribution 3:** Considering both independent and dependent temporal and spatiotemporal variables, recognizing their significance in accurately predicting demand patterns. The characteristics of demand prediction are thoroughly considered in the analysis.
- **Contribution 4:** Introducing a representation of time in the form of vector embedding, enabling automated feature engineering and enhancing the model's ability to capture and comprehend temporal patterns effectively.
- **Contribution 5:** Comparison with classical machine learning methods, providing empirical evidence of the performance and demonstrating the superiority in demand prediction accuracy.

1.4 THESIS OUTLINE

The primary aim of this thesis is to optimize shared transportation services in modern cities through three different approaches. This research is motivated by the increasing demand for services that are environmentally friendly, cost-efficient, and readily available. The study focuses on three key areas. Firstly, at the supply level, historical data will be analyzed to gain valuable insights into traffic patterns and accurately estimate travel times between different locations, considering spatial correlations. Secondly, at the operational level, a new real-time on-demand ridesharing algorithm is

developed, integrating real-time traffic information. Lastly, at the demand level, various techniques will be explored to forecast short-term demand, considering spatial, temporal, and external factors that influence it. The research conducted on the supply level serves as the foundation for the operational level, allowing the improvement of a real-time ridesharing algorithm. Furthermore, the outcomes on the demand level contribute to enhancing the accuracy of short-term passenger demand forecasting for ridesharing platforms. By addressing these aspects, the thesis contributes to the enhancement of shared on-demand transportation services in cities. The thesis outline is illustrated in Figure 1.1.

Chapter 2 proposes a novel approach that leverages sparse GPS probe data to allocate travel time information to different links between GPS observations. This method enables the estimation of travel times on specific routes using limited data points. The research took into account progressive spatial correlations and highlighted the advantages of considering these correlations. The results demonstrated how incorporating correlations can improve the accuracy of travel time estimation compared to traditional parametric methods. The proposed methodology is applied to a partial network of New York City, using data collected from taxicabs. The study showcased the enhanced accuracy of link travel time estimation achieved through the proposed method, surpassing the performance of conventional parametric approaches.

In Chapter 3 a modular real-time simulation framework has been developed specifically to address the complexities of the capacitated ridesharing problem. This framework enables flexible and customizable simulations, capable of accommodating various scenarios and system configurations. The ridesharing problem is formulated as a dynamic deterministic on-demand matching problem, incorporating tolerance times to enhance the matching process. To account for dynamic congestion, the simulation framework implements regular updates of travel times for network links, considering evolving traffic conditions and their impact on ridesharing operations. In solving the optimization problem posed by the ridesharing system, a combination of heuristic algorithms and commercial solvers is employed in an online fashion. The simulation also models multiple objectives and designs policies that strive to achieve efficient and mutually beneficial solutions for all stakeholders involved in the ridesharing system. The algorithm, when tested on the New York City taxi dataset, demonstrates a distinct advantage compared to the current taxi fleet in terms of service rate.

Chapter 4 focuses on the development of precise forecasting models specifically designed for predicting short-term demand in on-demand services. It explores the impact of different levels of data aggregation on prediction outcomes, highlighting the relationship between data granularity and accuracy. Both independent and dependent temporal and spatiotemporal variables are considered, recognizing their significance in accurately predicting demand patterns. The analysis thoroughly examines the characteristics of demand prediction. Furthermore, a novel approach is introduced, utilizing vector embedding to represent time. This approach automates feature engineering and improves the model's ability to capture and comprehend temporal patterns effectively. The research also includes an empirical comparison with traditional machine learning methods, demonstrating superior performance in predicting demand accuracy.

Chapter 5 provides a summary of the contributions and implications of this thesis, the limitations of the presented methods, and an outlook on promising future research directions.

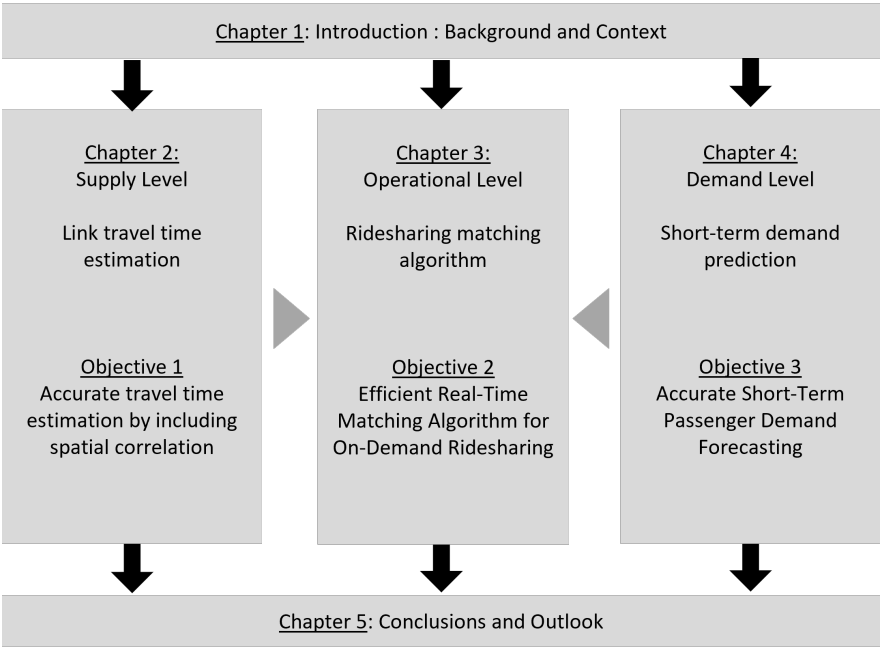


FIGURE 1.1: Thesis outline

2

LINK TRAVEL TIME ESTIMATION

The chapter is based on the following publications:

- Ghandeharioun, Z., and Kouvelas, A. (2022). "Link Travel Time Estimation for Arterial Networks Based on Sparse GPS Data and Considering Progressive Correlations," in *IEEE Open Journal of Intelligent Transportation Systems, vol. 3, pp. 679-694.*
- Ghandeharioun, Z., Rau, M., and Kouvelas, A. (2022). "Travel time estimation for urban arterials based on origin-destination data and spatial correlations." paper presented at the *101th Annual Meeting Online, Transport Research Board. TRB.*

2.1 INTRODUCTION

For the purpose of optimizing urban traffic operations and identifying major bottlenecks in the traffic network, accurate estimates, and forecasts of urban link travel times are critical. User advantage may also be gained by giving precise travel time information, allowing for improved path selection within the network, and reducing total trip traveling time. The use of real-time information from either in-road sensors such as loop detectors, microwave sensors, or roadside cameras, or mobile sensors (e.g., floating vehicles), or global positioning system (GPS) devices is required in order to estimate link travel times (e.g., cell phones) correctly. While there is little information available about the speed or the location of the connection in most of these instances, it is necessary to establish suitable methods for correctly estimating the performance measure of interest at the link, path, or network level.

There has been an increasing trend for GPS-equipped taxicabs in metropolitan regions in recent years. While GPS-equipped cabs have many benefits, they also act as valuable real-time probes for the traffic network. Taxis equipped with a GPS device collect a large quantity of data over days and months, offering a rich data supply for calculating network-wide performance indicators. Within this context, we present a technique based on sparse GPS probe data, and that is concerned with how to assign trip time

data to the various links traversed between GPS observations to improve accuracy. The spatial correlations between the connections in a network are taken into account by this approach. Ultimately, the purpose of this study is to demonstrate that by modifying the previously established techniques, we may include spatial correlations in our calculations and enhance our findings more realistically.

The present chapter is organized as follows: first, the problem at hand is described briefly. In Section 2.2, we review related publications and briefly discuss several approaches to similar problems. The subsequent section introduces the detailed methodology of the work presented in this chapter. In Section 2.4 proposed modification is explained in detail. The result of applied methods on a case study is presented in Section 2.5, with more details on the initial assumption and estimation results. The chapter is then closed with conclusions and final remarks in Section 2.6.

2.1.1 *Problem statement*

Urban travel time estimation based on GPS probe data has attracted many researchers recently (Bertsimas et al., 2019; Chen & Chien, 2001; El Esawey & Sayed, 2011; Hunter et al., 2009; Jenelius & Koutsopoulos, 2013). The goal is to determine the urban link travel time based on the large amount of reported trip data for a network. Taxi trip data consist of the following information: exact coordinates of origin and destination with the trip distance and travel time. In most of the available data sets, the precise trajectory of the taxi trip is unknown, and different assumptions are made to discover the most probable path for a given origin and destination of trip data. In order to estimate the link travel time, the following problems should be solved:

1. Represent the network in a digital form.
2. Match the recorded geographic coordinates of the trip origin and destination on the produced digital network.
3. Discover the most probable path for the given trip.
4. Allocate the travel time to the links belonging to the discovered path.
5. Estimate the travel time of the link based on the observed travel times.

The first two steps are usually solved in similar ways by different researchers (Hunter et al., 2009; Yuan et al., 2010). For the third step, most

researchers benefit from applying the k-shortest path algorithms to minimize the difference between the observed path and the assumed one (Hunter et al., 2009; Herring et al., 2010; Wang et al., 2019; Yuan et al., 2010). The methodologies used in the fourth and fifth steps can be classified into three categories:

a) Parametric approaches rely on statistical models and, based on mathematical assumptions, estimate the travel times. The majority of the parametric approaches assume that the link travel time is spatially and temporally independent of the rest of the network (Herring et al., 2010). However, in reality, travel time on different road segments and at other times of day are spatially and temporally associated with one another (Sen et al., 1997). Incorporating information on the spatio-temporal correlations of trip times may improve the estimation performance.

b) Non-parametric approaches are based on data-driven methods such as machine learning and neural networks. These approaches are free from assumptions and highly dependent on the amount of input data. The fusion of parametric and non-parametric approaches is classified as a third category called:

c) Hybrid approaches, which utilize a combination of both statistical models and data-driven methodologies. The details of the relevant works regarding this classification are presented in Section 2.2.

Based on the classification mentioned above, in the current work, we focus on extending a parametric approach, introducing static and progressive spatial correlations between the links on the network, and modifying a statistically proven method to have more realistic travel time estimations.

2.1.2 *Contributions*

In the current chapter, the main contribution is as follows:

- We propose a method based on sparse GPS probe data that focuses on allocating travel time data to the different links traveled between GPS observations. This model incorporates the spatial correlations between the links in a network.
- The main goal of this work is to show how we can consider progressive spatial correlations and improve our results more realistically with a simple adjustment in the previously known parametric methods.

- The methodology is applied to a case study for the partial network of New York City; based on the data collected from the taxicabs in New York City. By estimating link travel times with the proposed method, we show that travel time estimation accuracy is improved compared to the previously known parametric approaches.

2.2 LITERATURE REVIEW

In this section, we investigate the related works focusing on two topics. First, we review the works contributing to different travel time estimation methods. Second, we explore the literature considering the travel time correlation between the links.

2.2.1 *Travel time estimation*

Urban travel time estimation methods depend on the technologies deployed. The majority of the studies are based on data from technologies requiring extensive investment in sensor installation and maintenance, such as loop detectors in the following works: (Coifman, 2002; Zheng & Van Zuylen, 2013; Wu et al., 2004); Automated Vehicle Identification (AVI) in (Park & Rilett, 1998; Li & Rose, 2011; Sherali et al., 2006); video cameras in Yeon et al., 2008. Therefore, travel time estimation becomes expensive depending on the network coverage and the accuracy of the sensors.

An alternative approach is to develop methods of estimation based on emerging large-scale data sources, such as GPS devices in either a dedicated fleet of vehicles, available from taxis, transit, commercial vehicles, and service vehicles or even users' mobile phones. Herring et al., 2010 used GPS trace data from a fleet of around 500 taxis in San Francisco, USA, to estimate and predict traffic conditions. Bertsimas et al., 2019, Wang et al., 2019 and Zhan et al., 2013 utilize methods that are based on OD data, such as the New York City data set. Hunter et al., 2009 proposed a statistical approach for path and travel time inference using GPS probe vehicle trajectory data. Furthermore, Liu and Ma, 2009 states that reliable traffic estimation based on taxi data is provided when an adequate historical traffic database is available and the data covers long road segments sufficiently. Nevertheless, more complex approaches are needed to generate valuable output compared to the methods for traditional sensors stated in Leduc et al., 2008.

The methodologies based on GPS data introduced in different approaches can be categorized as follows:

Parametric approaches rely on mathematical and statistical equations. These approaches are limited by the assumptions made in the analytical and statistical models. However, they are proven mathematically correct and less computationally expensive (Hunter et al., 2009). Yeon et al., 2008 developed a model that can estimate travel time on a freeway using Discrete Time Markov Chains (DTMC), where the states correspond to whether or not the link is congested. Ramezani and Geroliminis, 2012 also used a Markov chain approach to estimate arterial trip travel time distributions by capturing the spatial correlations using a Transition Probability Matrix (TPM) calibrated from historical data.

Most parametric estimations assume the spatially or temporally independent link travel time (Herring et al., 2010; Yeon et al., 2008; Hunter et al., 2009). Bertsimas et al., 2019 introduce the general approach for travel time estimation based on OD data that can recover interpretable city traffic and routing information from potentially noisy and incomplete data. Zhan et al., 2013 combine the statistical model with MNL for path selection and minimize the least square error between the observed and expected path travel times.

Among the parametric approaches, only a few consider spatial correlation; the model presented in Jenelius and Koutsopoulos, 2013 separates trip travel times into link travel times and intersection delays and allows the correlation between travel times on different network links based on a spatial moving average (SMA) structure. Tang et al., 2018 develop a tensor-based Bayesian probabilistic model for citywide and personalized travel time estimation using the large-scale and sparse GPS trajectories generated by taxicabs in Beijing. His model incorporates both the spatial and temporal correlation between different road segments and the person-specific variation between different drivers. Ma et al., 2017 propose a generalized Markov chain approach for estimating the probability distribution of trip travel times from link travel time distributions and take into consideration correlations in time and space.

Non-parametric approaches rely on data-driven methods such as machine learning and neural network (Rahmani et al., 2015). These methods are free of assumptions but highly dependent on the amount of input data and, therefore, computationally expensive. Wang et al., 2019 introduce a neighbor-based approach and considers a dynamic traffic condition using temporal speed references. Furthermore, Zheng and Van Zuylen, 2013 developed a method based on artificial neural networks to estimate the

complete link travel time for an individual probe vehicle traversing the link, using the low-frequency data collected by probe vehicles.

Hybrid approaches utilize a combination of data-driven methods and statistical models. The fusion of parametric and non-parametric methods is generally more precise than the methods mentioned earlier. Allström et al., 2016 combine parametric and non-parametric traffic state prediction techniques through assimilation in an ensemble Kalman filter. For a non-parametric prediction, a neural network method is adopted; the parametric prediction is carried out with a cell transmission model with velocity as the state. Hofleitner et al., 2012 similarly, benefit from a hybrid approach and develop a model on traffic flow through signalized intersections and combines it with a machine learning framework to both learn static parameters of the roadways as well as to estimate and predict travel times through the arterial network.

2.2.2 *Travel time correlation*

Correlation between travel times of links in a network or a path is empirically and theoretically discussed in many previous studies (Hall, 1986, Sen et al., 1997, Rilett and Park, 2001, Chen and Chien, 2001, Eisele and Rilett, 2002, Gajewski and Rilett, 2005, Chan et al., 2009). The problem of how to estimate the travel time correlation between links on a corridor was also introduced by Sen et al., 1997. The theoretical analysis of this correlation is presented in Hall, 1986 and Fu and Rilett, 1998. Rilett and Park, 2001 developed a one-step approach using artificial neural networks (ANN) to predict corridor travel times directly and consider inter-correlation between link travel times. The authors suggested that using a separate model to predict the travel time on each link without considering the covariance with other links can lead to significant errors. Zeng et al., 2015 extended the Lagrangian relaxation algorithm by representing travel time correlations based on the Cholesky decomposition. Chen et al., 2016 further extended the multi-criteria A* algorithm to consider travel time correlations among adjacent K links. In addition, they show that adjacent link travel times are strongly correlated. For example, traffic accidents on a link may also lead to serious travel delays on its upstream links. In their work Gajewski and Rilett, 2005 also estimate the link travel time correlation in the range of -1 to +1 by using a nonparametric regression technique based on Bayesian natural cubic splines. Rachtan et al., 2013 developed three regression models to describe the correlation variation by considering various combinations of

variables such as spatial distance, temporal distance, traffic state, and the number of lanes. They found that the primary factor in the correlation is spatial distance.

Based on the literature above and the logic presented as Tobler's first law of Geography, that 'all things are related, but nearby things are more related than distant things' (Tobler, 1970), we introduce the spatial correlation formulation to incorporate it with the previously proven historic model of traffic introduced by Herring, 2010. Furthermore, El Esawey and Sayed, 2011 show that the correlation is usually very low for links that are spatially distant, even on the same street. Also, they show for the determination of the correlation coefficient between the links, using the exponential model form outperformed the linear and power model forms under the chosen acceptance limits for the goodness of fit criteria.

2.3 METHODOLOGY – ESTIMATION WITHOUT SPATIAL CORRELATIONS

In this section, we present the methodology based on the steps introduced in Section 2.1.1, and explain how we have approached each problem. It is worth mentioning that, the core of our methodology is built on the work presented in Herring, 2010. However, our approach addresses the gap of considering spatial correlations between network links and modifies the aforementioned work.

2.3.1 *Network model*

Basis of this work is a digital representation of a physical network. A directed graph $G(L, N)$ is generated utilizing Open Street Map, where links (L) and nodes (N) represent roads and intersections, respectively. For example, if a road is a two-way street, two links will be defined for that segment. The weight of the links in this graph is the length of the link in the real network.

2.3.2 *Map matching and path inference*

In this work, we benefit from the origin destination of trips reported by a reliable source in NYC Taxi and Limousine Commission, 2019, which has been used in many previous works Alonso-Mora et al., 2017; Bertsimas et al., 2019; Zhan et al., 2013. This type of data is usually reported in GPS format. We know the exact geographical coordinates of the origin and destination

of each trip. If the origin or destination location of a trip is in the middle of a link, it is projected to the nearest node/intersection. This step is a source of error at two levels. On the one hand, GPS data are unavoidably inaccurate, and on the other hand, it is neglected that trips generally do not start and end at intersections. However, the consequences of the latter are not significant, if the trips reported are sufficiently long.

Since we are not aware of the exact path that the taxi has taken in this type of data, we apply the k-shortest path algorithm based on Yen's algorithm explained in Yen, 1970 to determine the inferred path as the one that minimizes the difference between the inferred and observed path distance. Since the k-shortest path is a computationally expensive task, defining k depends on the available resources for each study. After this step, the observations that violate the following inequality are removed.

$$0.5 \times \text{observed distance} < k \text{ shortest path distance} < 1.5 \times \text{observed distance} \quad (2.1)$$

After this step, the data are in the form of path observations. The set of all available path observations for time interval t, is denoted as P_t and a single path as p.

2.3.3 *Travel time estimation model*

The proposed travel time estimation methodology is built on Herring, 2010 methodology, and requires path observations as input data. This work is based on the following assumptions:

- The travel time distribution for each network link is independent of all other network links. Therefore, the set of all network links, that we have observations for is denoted as L.
- Any given moment in time belongs to exactly one historical time period, during which, traffic conditions are assumed to be constant.
- All travel time observations from a specific link l are independent and identically distributed within a given time period t.
- Sparse probe measurements are the only data available to the model.

Admittedly, the first and second assumptions are very strong and proven incorrect. Spatial correlations exist at both the local and non-local levels.

Temporal dependencies exist in a short-term neighboring and long-term periodic timescale (Zheng & Van Zuylen, 2013). While that might hold true, capturing these spatial-temporal dependencies is challenging, independent of whether you try to estimate them or incorporate literature values into the model, given that they even exist. In this approach, we explain the solution with independent variables and try to consider the dependencies of the link and improve the Herring, 2010 approach to a more realistic one.

2.3.3.1 *Probabilistic setting*

The random variable capturing the link travel time for link l in time period t is denoted as $X_{l,t}$, where l can be any element of L. The set of links lying on path p is denoted as L_p, so let $Y_{p,t}$ be the random variable representing the path travel time for path p in time period t. Then, the path travel time $Y_{p,t}$ can be represented as follows

$$Y_{p,t} = \sum_{l \in L_p} X_{l,t}. \tag{2.2}$$

It is assumed that all link travel times in the network follow some probabilistic distribution. This generally can be any probability distribution function for any link l. In the current work, we assume that all link travel times follow Gaussian distributions, and we define $\mu_{l,t}$ as mean value and $\sigma^2_{l,t}$ as variance, thus: $X_{l,t} \sim N(\mu_{l,t}, \sigma^2_{l,t}),\ \forall l \in L_p$.

The parameters describing the distribution for link l and time period t are denoted as $Q_{l,t}$.

The link travel time probability density function for link l during time period t is denoted as $G_{Q_{l,t}}(X_{l,t})$. Path travel time probability density function is denoted as $G_{Q_{L_{p,t}}}(Y_{p,t})$, where the indices $Q_{L_{p,t}}$ denote the parameters of the links along the path p in time period t. The probability distribution of the sum of two or more independent random variables is the convolution of their individual distributions. Therefore, $G_{Q_{L_{p,t}}}(Y_{p,t})$ is the convolution of the link travel time distributions along the path p. In this case, all link travel times are assumed to be independent from one another and to follow Gaussian distributions. Hence, for a path observation, it holds, $Y_{p,t} \sim N(\Sigma_{l \in L_p} \mu_{l,t}, \Sigma_{l \in L_p} \sigma^2_{l,t})$.

The goal is to find the parameter values $Q_{l,t}$ for each link and time period, which make the observed data most probable. This is achieved by

maximizing the likelihood function, which can be written in a general case as follows:

$$\arg\max_{Q_t} \prod_{p \in P_t} G_{Q_{L_{p,t}}}(Y_{p,t}). \tag{2.3}$$

To transfer the product into a sum, the logarithm of the function is calculated. The maximum still occurs at the same parameter values since the logarithm is a monotonic function.

$$\arg\max_{Q_t} \sum_{p \in P_t} \ln(G_{Q_{L_{p,t}}}(Y_{p,t})). \tag{2.4}$$

Given the assumption that all link travel times follow Gaussian distributions, problem (2.4) can be reformulated with optimization problem (2.5).

$$\arg\max_{Q_t} \sum_{p \in P_t} \ln\left(f\Big(\sum_{l \in L_p} \mu_{l,t}, \sum_{l \in L_p} \sigma^2_{l,t}\Big)\right), \; p \in P_t, \tag{2.5}$$

where $f\Big(\sum_{l \in L_p} \mu_{l,t}, \sum_{l \in L_p} \sigma^2_{l,t}\Big)$ denotes the Gaussian probability density function as a function of $\mu_{l,t}$ and $\sigma^2_{l,t}$ for a given $Y_{p,t}$.

This optimization problem is challenging on two levels. On the one hand, it simultaneously solves for the mean and variance. On the other hand, the number of variables is large, particularly in a network-wide study. The number of variables can be calculated as the number of links multiplied by the number of parameters per link.

Herring et al., 2010 explained that the methodology can be extended to cases beyond the Gaussian distribution but leads to more complex optimization problems because it simultaneously solves for the mean and variance of every link in the network. It is possible to solve this problem directly if using a commercial-grade non-linear optimization engine with a lot of computational power. However, it is assumed that such resources may not be available, and an alternative solution strategy is proposed. The Gaussian case is presented here to show an example of the algorithm from start to finish in complete detail.

Since we extend the Herring methodology to a correlated version, we present the work by considering the Gaussian distribution. In general, the choice of a Gaussian distribution restricts the model's flexibility to capture unique traffic characteristics, but it is also far more tractable to solve in practice (Herring, 2010). When using this model with certain classes of link travel time distributions, the travel time allocation problem

is efficient, even for large amounts of data, such distributions include the standard distributions like Gaussian, Log-Normal, Gamma (Nielsen, September 1997). The parameter estimation problem is also efficient for the same set of distributions listed above (Nielsen, September 1997).

Furthermore, recent empirical studies based on field observations show that the use of normal distributions appears to reflect observed path travel time distributions (Rakha et al., 2006). In addition, Chen et al., 2016 found that the normal distribution can reasonably approximate the path travel time distribution. The normal distribution approximation can achieve 98.3% and 94.9% accuracy at the 10th and 90th percentiles. Also, Zeng et al., 2015, in their work, used the empirical link and path travel time data from probe vehicles to characterize travel time distributions at the link and path level. Several typical distributions are tested, such as normal, lognormal, truncated normal, and truncated lognormal. Further, he explains the observed data distribution is approximated by a normal distribution, which is more computationally tractable and has an acceptable compromise on accuracy.

Herring, 2010, suggests an intuitive decomposition scheme reaches near-optimal solutions efficiently. Also, note that for each time interval t, the problem can be solved separately, given the assumption, each time interval is independent.

2.3.3.2 *Decomposition scheme*

The core concept is to decouple the optimization problem into two more manageable sub-problems and iterate between these two until converging to an optimal solution. These two sub-problems are travel time allocation and parameter optimization. Herring's explanation of why his decomposition scheme makes sense, though it cannot be derived mathematically, goes as follows. It would be straightforward to estimate the link parameters if it was known how much time each probe vehicle spent on each link on its path. However, in the case of sparsely sampled and OD data, this information is not available. Instead, one could try to determine the most likely link travel times, which depend on the link travel time parameters that in turn need to be estimated with the most likely link travel times. This is a chicken-and-egg type of problem. It is solved by assuming some initial link parameters, which are then used to determine the most likely link travel times. Following, the most likely link travel times are used to update the link parameters, which then are utilized to determine the most likely travel times again. This iterative process is repeated until convergence is

reached. By reaching the convergence, the algorithm's output is $X_{l,t}$ variable that contains all the individual travel times allocated in an optimal manner to the links $l \in L$ for time period t. This $X_{l,t}$ can be used to compute our final set of parameters Q_t (Herring, 2010).

2.3.3.3 *Travel time allocation*

The travel time allocation determines the most likely link travel times corresponding to a path p. To solve this problem, estimates of the link parameters must be available for all links in time period t, $l \in L_t$. This means that all link parameters are fixed for this part of the algorithm. Furthermore, it is essential to define lower bounds for the link travel times; otherwise, the most likely travel time is smaller than the free-flow travel time, or in extreme cases, even negative. The free-flow travel time is denoted as b_l and is the time needed to travel link l with the maximum allowed speed. It is calculated by dividing the link length by the maximum allowed speed. For example, despite the existence of some highways and areas with narrower streets, the speed limit in Manhattan is 25mph (NYC Taxi and Limousine Commission, 2019). It is suggested by Herring et al., 2010 to assume that the taxi drivers will travel at 40 to 50 mph to compute the minimum link travel time. However, in the case study presented in Section 2.5, we use 25mph as the free flow speed to calculate the free flow travel time. This constraint implies that path observations with an average speed greater than 25mph do not have a solution, and thus are removed from the path set.

The goal of finding the most likely travel times is also achieved by formulating a maximum likelihood function and finding its maximum. Still assuming that all link travel time distributions are Gaussian, the problem can be formulated as in problem (2.6), where $f\left(X_{l,t} \mid \mu_{l,t}, \sigma^2_{l,t}\right)$ denotes Gaussian probability density function for a given mean μ and a given variance σ^2 as a function of the link travel time $X_{l,t}$

$$\arg\max_{X} \prod_{l \in L_p} f\left(X_{l,t} \mid \mu_{l,t}, \sigma^2_{l,t}\right). \tag{2.6}$$

Again, to convert the product to a sum, the logarithm of the function is computed. Moreover, two constraints are added. The sum of the link travel

times lying on a path must be equal to the observed path travel time $Y_{p,t}$, and the link travel times $X_{l,t}$ must be larger than the free-flow travel time.

$$\begin{aligned} \arg\max_{X} & \sum_{l \in L_p} \ln\left(f\left(X_{l,t} \mid \mu_{l,t}, \sigma^2_{l,t}\right)\right) \\ & \text{s.t.} \\ & \sum_{l \in L_p} X_{l,t} = Y_{p,t} \\ & X_{l,t} \geq b_l, \ \forall l \in L_p. \end{aligned} \tag{2.7}$$

This problem needs to be solved for every observation $p \in P_t$, and this is done by the following method. First, the total expected path variance V and the difference between expected and observed path travel time Z need to be calculated

$$V = \sum_{l \in L_p} \sigma^2_{l,t}, \tag{2.8}$$

$$Z = Y_{p,t} - \sum_{l \in L_p} X_{l,t}. \tag{2.9}$$

As the next step, the expected travel time, adjusted by some proportion of Z, is allocated to each link. This proportion is computed by dividing the link variance by the total path variance

$$X_{l,t} = \mu_{l,t} + \frac{\sigma^2_{l,t}}{V} Z. \tag{2.10}$$

Links with high variance are the most likely source of discrepancies between observed and expected path travel time. The links with high variance get attributed to the largest part of Z. After this attribution, some links may violate the free flow constraint. These links are saved in the set J. After identifying the violating links and saving them in the set J, we calculate V and Z again. At this step, all the identified violating links saved in J ($l \in J$) have an expected travel time equal to the free-flow travel time, and these links do not contribute to the calculation of the total path variance V

$$V = \sum_{l \in L_p / J} \sigma^2_{l,t}, \tag{2.11}$$

$$Z = Y_{p,t} - \sum_{l \in L_p / J} X_{l,t} - \sum_{l \in J} b_l. \tag{2.12}$$

Then, the updated difference between the expected and observed travel time Z is attributed again with Equation (2.10). After this step, some links may still violate the constraint. Thus, J is updated, V and Z are recalculated, and Z is attributed to the links again. This procedure is repeated until the free-flow travel time constraint is met. On average, 1 to 5 iterations were necessary to meet the constraint in the use case at hand. Having solved the travel time allocation for all path observations $p \in P_t$, the output of the algorithm $X_{l,t}$ contains all the individual travel times allocated to the links $l \in L$ and time period t.

2.4 INTRODUCING SPATIAL CORRELATIONS

Considering the aforementioned theoretical backgrounds in Section 2.2 and the criteria of spatial correlation they all show in their works, we introduce our heuristic for both progressive and static correlations as follows:

The Travel time allocation method presented in 2.3 can be extended for correlated links if we assume that the travel time on these links is jointly normally distributed. Based on the multivariate central limit theorem ("The Multivariate Normal Distribution", 2002), the summation of all links' travel times is still normally distributed; therefore, this does not affect the maximum likelihood function formulation in the historic traffic model.

For each link in the set of L_p, we define the correlation between link $l_i \in L_p$ and $l_j \in L_p$ in path p by ρ^p_{ij} the Equations (2.8) and (2.10) will be updated as follows:

$$V = \sum_{l_i \in L_p} \sigma^2_{l_i,t} + 2 \sum_{l_i, l_j \in L_p, i \neq j} \sigma_{l_i,t} \sigma_{l_j,t} \rho^p_{ij,t} \tag{2.13}$$

$$X_{l_i,t} = \mu_{l_i,t} + \frac{\sigma^2_{l_i,t} + \sum_{l_i, l_j \in L_p, i \neq j} \sigma_{l_i,t} \sigma_{l_j,t} \rho^p_{ij,t}}{V} Z. \tag{2.14}$$

The correlation between the links can be considered both static and progressive. In the static version, we allocate the travel time in each iteration based on the same correlation coefficient defined at the beginning. In the progressive version, we update the correlation coefficient in each iteration based on the changes in parameters (in here, the mean value) in the last two iterations.

It is worth mentioning that the correlation coefficient here focuses on spatial correlation, and the temporal correlation is neglected in this study,

and we assume that the travel time estimation is independent between different time periods.

In the current work, the main contribution is to show the effect of considering spatial correlations to understand the model's performance regardless of considering temporal correlations. Also, since for every 15-minute time interval, we have an extensive amount of taxi trip data, it can provide us with enough input for that time interval reflecting the conditions propagated from the previous time interval (e.g., spillback). However, one can include the temporal correlation by incorporating the parameters about the travel time of each link from the previous interval to the next interval. If we include both correlations simultaneously, it is hard to understand the effects separately.

In the following, we explain each version in more detail:

- **Static Correlation**

  Defining a realistic spatial correlation matrix is a challenging task, and it is highly dependent on network characteristics (Sen et al., 1997). A basic rational approach for spatial correlation coefficient can follow the logic of the further you get from a link; the correlation coefficient will decrease accordingly (Tobler, 1970). Following this logic and the aforementioned background, the mathematical formulation of the spatial correlation should meet the following criterion: a) The correlation function should be descending by increasing the spatial distance b) The correlation coefficients should be near zero for very distant links. In our approach the static spatial correlation is calculated as follows: In a path with k links, the path p is a set of links: $L_p = \{l_1, l_2, l_3, ..., l_k\}$, $\rho^p_{ij,t}$ is the correlation coefficient between link l_i and l_j in the time interval t in path p, where $i, j \in \{1, k\}$ and $|i - j|$ is the rank order distance of l_i to l_j in the set of L_p

$$\rho^p_{ij,t} = \frac{1}{\alpha \cdot |i - j| + 1}, \forall l_i, l_j \in L_p, \tag{2.15}$$

$$\text{where } 0.1 \leq \alpha \leq 0.9.$$

  The α value defines how quickly the correlation between the links in a path can decrease by increasing the distance. The higher α value corresponds to the quicker reduction in the correlation coefficient between the links in the path by increasing distance.

  The correlation coefficient is calculated only on the basis of the paths, as the path observations are the only input in the proposed model. If two

paths have mutual links, the spatial correlation is calculated for each path separately, and the correlation coefficient for the mutual link is calculated in each path towards the other links in the path.

Remark 1: We note that the function in Equation (2.15) is only a candidate function and does not necessarily provide the best result among all the possible functions. One can find a near-optimal correlation function through hybrid approaches (Allström et al., 2016). However, the main focus of our work is to show how we can consider static spatial correlations and improve our results more realistically with a simple adjustment in the previously known parametric methods.

For example, the static correlation coefficient for the first and the middle link in a path is depicted in Figure 2.1. The profile definition in both diagrams in Figure 2.1 follows the same Function (2.15); the only difference is the starting link. We calculate the correlation coefficient between the first link and all other links in the path in the top figure. The bottom figure shows the correlation coefficient between the link in the middle of the path and all other links in the path, the links before the middle link and after the middle link. In the static version, the value of $\rho^{p}_{ij,t}$ remains the same through iterations for the calculation of Equation (2.13) and Equation (2.14).

- **Progressive Correlation**

  In the progressive version, we start by defining the correlation of the links in a path similar to Equation (2.15) in the first iteration ($n = 1$) and increase or decrease it based on the changes in the $\mu_{l_{i,t}}$, and $\mu_{l_{j,t}}$ in previous iterations. The iteration number is defined by n. Suppose $\Delta\mu_{l_{i,t,n}}$ and $\Delta\mu_{l_{j,t,n}}$ both are positive or negative ($\lambda_n > 0$), meaning that both link trends are following the same direction. Then, we increase the correlation coefficient $\rho^{p}_{ij,t}$ in the next iteration. If one is positive and the other negative ($\lambda_n < 0$), we decrease the correlation coefficient. We assume that the trend in the changes in the mean travel time of a link through iterations can reflect the correlation between the two links. This can be seen in the travel time distribution of the links and thus in the mean travel time changes in the iterative approach.

  The amount that the correlation coefficient is increased or decreased in iteration n follows the function introduced in (2.16). The mathematical formulation of the progressive approach needs to meet the following criterion: a) the function should gradually increase to an upper bound or gradually decrease to a lower bound, and b) The changing increment

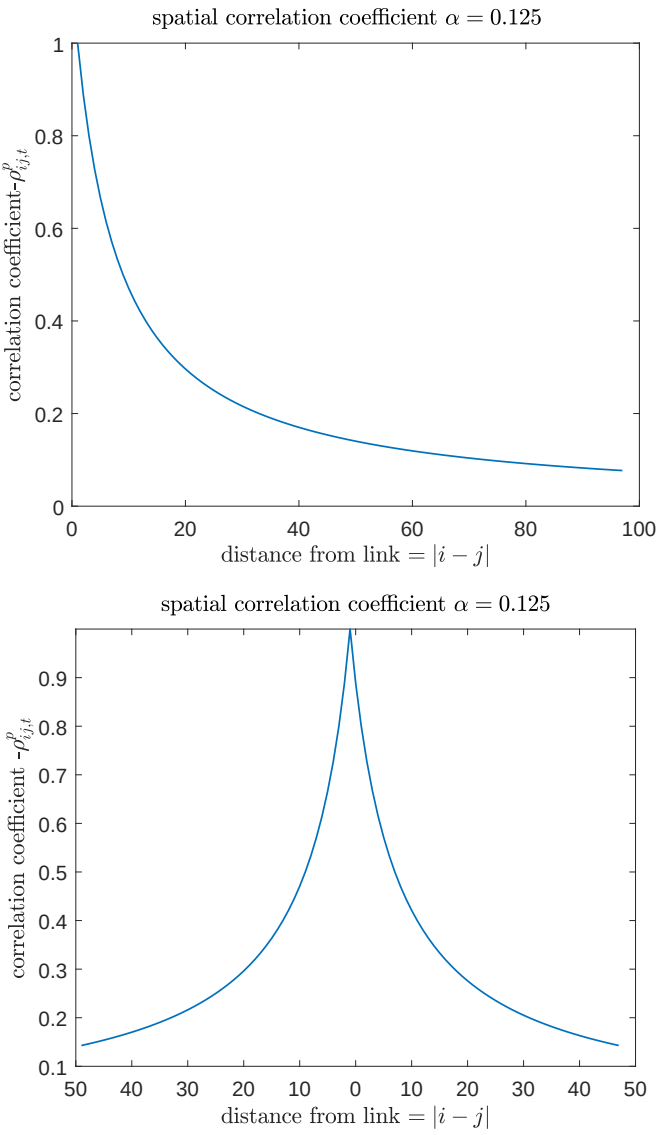


FIGURE 2.1: Static correlation coefficient example of a path with 98 links with $\alpha = 0.125$ (top: from the first link, bottom: from the middle link).

should be adjustable by defining a parameter. For example, in Equation

(2.16), we gradually increase the correlation coefficient up to the upper bound of $+0.8$, and similarly, we decrease it down to -0.8, that is the lower bound (Gajewski & Rilett, 2005).

$$\rho^{p}_{ij,t,n} = \rho^{p}_{ij,t,n-1} + C_{ij,t,n} \;\forall l_i, l_j \in L_p \tag{2.16}$$

$$\lambda_n = \frac{\Delta\mu_{l_{i,t,n}}}{\Delta\mu_{l_{i,t,n}}} = \frac{\mu_{l_{i,t,n}} - \mu_{l_{i,t,n-1}}}{\mu_{l_{j,t,n}} - \mu_{l_{i,t,n-1}}}$$

$$C_{ij,t,n} = \begin{cases} -a^{\beta} + a, a = |0.8 - \rho^{p}_{ij,t,n-1}|, \text{if } \lambda_n > 0 \\ b^{\beta} - b, b = |-0.8 - \rho^{p}_{ij,t,n-1}|, \text{if } \lambda_n < 0 \\ 0, \text{if } \rho^{p}_{ij,t,n-1} > 0.8 \text{ or } \rho^{p}_{ij,t,n-1} < -0.8 \end{cases}$$

$$\text{where } 0.01 \leq \beta \leq 0.09.$$

The β value corresponds to the increment that we increase or decrease the correlation coefficient between two links. The higher the β value, the faster we reach the upper/lower bounds. As an example, the progressive correlations for a link at the beginning of the path and a link in the middle of the path are depicted in Figure 2.2. In this figure, we present the changes in the correlation coefficient through iterations. Each line in Figure 2.2 is the correlation coefficient of the chosen link i to all the other links j in the path. For instance, in Figure 2.2 on top, we have the correlation coefficient of the first link ($i = 1$) of a path with 98 links to all the other $j = 1 : 98$ links. The X axis is $|i - j|$ and the Y axis is the correlation coefficient ρ^{p}_{ij} for each iteration. Here we presented only 20 iterations, each with a distinct color and line pattern, with the number of iterations and the line pattern in the graph's legend. The graph at the bottom presents the correlation coefficient ρ^{p}_{ij} of the link in the middle $i = 50$ to all other links $j = 1 : 98$ in the path p. As we see, the first iteration starts with the same values calculated for the static version and changes through iterations based on Equation (2.16). Negative correlations between the links can occur, for instance, due to having traffic signals in the path. If one link is highly congested due to a red signal, having a longer travel time, the others are empty and have free flow travel time. A negative correlation in our study can explain this situation. It means that an increase in travel time in the link i can strongly reduce the travel time in link j. We note that 2.16 may not provide us with the best mathematical formulation for the optimal performance indicator

using in the progressive approach. However, we show that the results improve by taking into account the changes in distribution function parameters through iterations for defining the correlation coefficient (see Table 2.3).

2.4.0.1 *Parameter Optimization*

Receiving $X_{l,t}$ from the travel time allocation step, optimizing the parameters is straightforward. Mean and variance are updated based on Equations (2.17) and (2.18), respectively. Note that $X_{l,t}(m)$ denotes the m_{th} observation of $X_{l,t}$. Reliable estimates are not possible for links with less than ten observations available. Thus, the parameters are not updated, but the initial ones are kept.

$$\mu_{l,t} = \frac{1}{|X_{l,t}|} \sum_{m=1}^{|X_{l,t}|} X_{l,t}(m), \tag{2.17}$$

$$\sigma^2_{l,t} = \frac{1}{|X_{l,t}|} \sum_{m=1}^{|X_{l,t}|} (X_{l,t}(m) - \mu_{l,t})^2. \tag{2.18}$$

To solve the chicken-and-egg problem entirely, initial parameters for all links $l \in L$ are still required. Herring, 2010 suggests that these should be chosen according to literature values, which are in keeping with the link characteristics (number of lanes, traffic lights, etc.). For this work, the initial parameters are based on assuming that all cabs had a constant velocity along their path. This allows allocating the travel times based on the length of the links (see Equation (2.19) below). D_l denotes the length of link l and D_{L_p} the sum of all link lengths lying on path p.

$$X_{l,t} = \frac{D_l}{D_{L_p}} Y_{p,t}. \tag{2.19}$$

The output of this initial travel time allocation is of the same type as $X_{l,t}$. The initial parameters are therefore calculated with Equations (2.17) and (2.18), having $X_{l,t}$ based on the constant velocity assumption as the input argument.

2.4.0.2 *Convergence*

With each iteration (going back and forth between travel time allocation and parameter optimization), the parameter values should become smaller

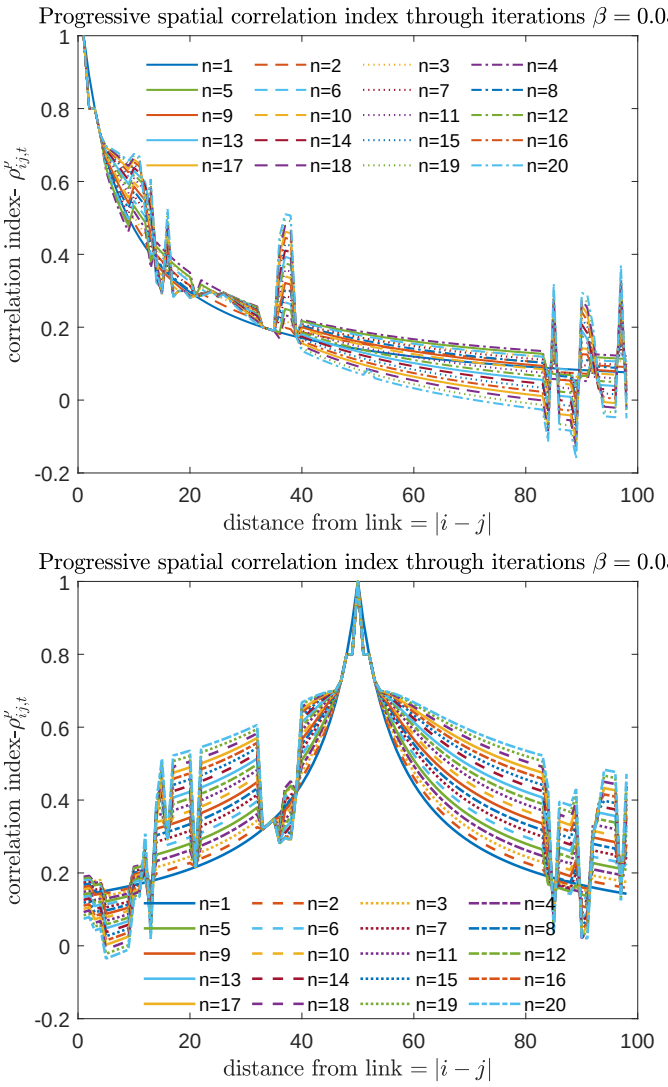


FIGURE 2.2: Progressive correlation coefficient example of a path with 98 links through 20 iterations $\beta = 0.05$ (top: from the first link, bottom: from the middle link, number on each line shows the iteration number).

until the parameter values no longer change significantly. This is called convergence. The parameters are the near-optimal solution Q_t for the

optimization problem (2.4) by reaching convergence. Herring, 2010 suggests that a global parameter $n_{\max}$ can define the criterion for convergence that stipulates the number of maximum iterations. In this work $n_{\max} = 100$ is set, which led to a reasonable convergence. In Table 2.2, the mean relative differences for the mean travel time values for all links in all time intervals in different models for the case study are presented in Section 2.5.

Alternatively, after each iteration, one could compute the absolute difference between the individual link parameters of the previous and the current iteration. These differences are then divided by the parameter values of the previous iteration, revealing the relative differences as well. We denote this difference as Δ_Q. The convergence criterion itself is defined as a maximum allowed relative difference of the parameters between two iterations that we call $\Delta_{Q,\max}$. For instance, an appropriate value for $\Delta_{Q,\max}$ is 0.01, meaning that convergence is reached as soon as none of the parameters change by more than one percent between two subsequent iterations. The downside of this type of convergence criterion is that a single iteration needs more computing time. However, this criterion is more general, and one can also avoid unnecessary iterations and therefore may save total computing time for the algorithm as a whole. For the second proposed convergence method, if we consider the relative difference in mean values for all links to be less than 0.01, which means 1% on average. With the presented values in Table 2.2 for the case study in Section 2.5, it is obvious that we need less than 100 iterations.

2.5 TRAVEL TIME ESTIMATION IN MANHATTAN: A CASE STUDY

In this section, the previously mentioned methodology is applied to the NYC taxi trip data set provided by the Taxi and Limousine Commission (TLC), available online at (NYC Taxi and Limousine Commission, 2019). In this case, the time periods of interest are quarter-hourly intervals from 7 am until 9 am on Tuesday the 1^{st} of February 2011. According to Grynbaum, 2010, traffic in Manhattan intensifies significantly between 7 am and 9 am and then remains relatively constant until 7 pm. The area of interest is limited to Manhattan; since it particularly suffers from congestion and has a high number of taxi trip observations available relative to its size (165737 on Tuesday the 1^{st} of February 2011 (NYC Taxi and Limousine Commission, 2019)).

The Manhattan network includes a grid road network consisting of 228 numbered streets running in the East-West direction and 11 avenues running

in the South-North direction. The network presented as a directed graph is generated using "Open Street Map", 2021, with the nodes representing intersections, and edges representing links. The weight of an edge represents the road distance between two intersections, and the direction of an edge represents the allowed driving direction. Also, the geographical coordinates of the nodes are known. However, other network information, such as the number of lanes, bus stops, and traffic lights is not considered.

The observed GPS coordinates of the starting and end points need to be assigned to a specific point in the graph. This can either be the points lying on an edge or a node. For simplicity, we chose the starting and ending point as the node that is closest to the observed GPS coordinate based on Euclidean distance. After this step, the GPS coordinates are no longer used. Instead, all the starting and ending points are now represented by node IDs corresponding to the graph. As explained in Section 2.3, this step is a source of error on two levels. On the one hand, GPS data is unavoidably noisy, and on the other hand, it is neglected that trips generally do not start and end at intersections. However, the consequences of the latter are not grave, since the average taxi trip observation from the NYC data set covered roughly 40 links (this number is based on the applied path inference method).

Time interval	Number of Observations	Number of links with more than 10 data points
7:00 – 7:15	1822	1781
7:15 – 7:30	1858	1987
7:30 – 7:45	2008	2011
7:45 – 8:00	2175	2222
8:00 – 8:15	2380	2321
8:15 – 8:30	2477	2390
8:30 – 8:45	2474	2497
8:45 – 9:00	2045	1976

TABLE 2.1: Number of observations for different time intervals.

The straightforward method explained in Section 2.3 is used for the path inference problem. By applying Yen, 1970's algorithm, up to 20-shortest paths are calculated to find the path with the least difference between the reported trip length and generated trip length. In addition, all trips violating the Inequality 3.5.1 are removed. The next step is to find out if the shortest

path assumption suffices. For this, we calculate the difference between the individual observed trip length and the shortest distance relative to the observed distance. The mean of this relative difference is 0.088, and the median is 0.052. Judging behalf of this, the accuracy of the shortest path assumption suffices. One could argue that multiple paths corresponding to an OD pair can have a very similar length but differ widely regarding the links they travel. A large number of path observations compensates for this.

After this step, the data are in the form of path observations. The number of path observations and the number of links with more than 10 data points are presented in Table 2.1. The time interval a path observation belongs to is defined by the pickup time.

In order to observe the effect of progressive spatial correlation modification, we present the results by comparing the outcomes of both static and dynamic correlated algorithms. Moreover, we present the results of the historic traffic model of Herring, 2010 in which the links' travel time are assumed to be independent and labeled as an uncorrelated model. The comparison of the mean travel times of individual links is understandable when they are normalized. This is achieved by dividing the individual mean link travel times $\mu_{l,t}$ by the link length. Hereby, we receive the travel time rates, which can be considered as the inverse of the mean velocity. Here, we use the unit seconds per meter. The travel time rate corresponding to the maximum allowed speed suggested by Herring, 2010 (25mph) is 0.0894 s/m. In Figure 2.3 and Figure 2.4, the normalized mean travel time rates are depicted relative to the free-flow travel time rate, where 1 is equal to the free-flow travel time rate, and 5 is five times the free flow travel time rate. In Figure 2.3, we show the distribution of the link travel times in each time interval by box plots. The top of the rectangle in the box plot indicates the third quartile (75%), the horizontal line near the middle of the rectangle indicates the median (50%), and the bottom of the rectangle indicates the first quartile (25%). In Figure 2.4, we present the normalized mean value of link travel time on each link on the Manhattan network. To highlight the links with particularly high travel time rates, the line widths are adjusted according to the mean link travel time rates.

Figure 2.3 supports the indication that the traffic overall becomes slower from 7 am to 9 am, which is in line with the earlier work conducted on the New York City taxi data set (Grynbaum, 2010). It also shows that the difference in travel time rates increases among the links; this can be judged from the widening of interquartile boxes over time. Comparing the results of static correlated and progressive correlated, we can observe that the

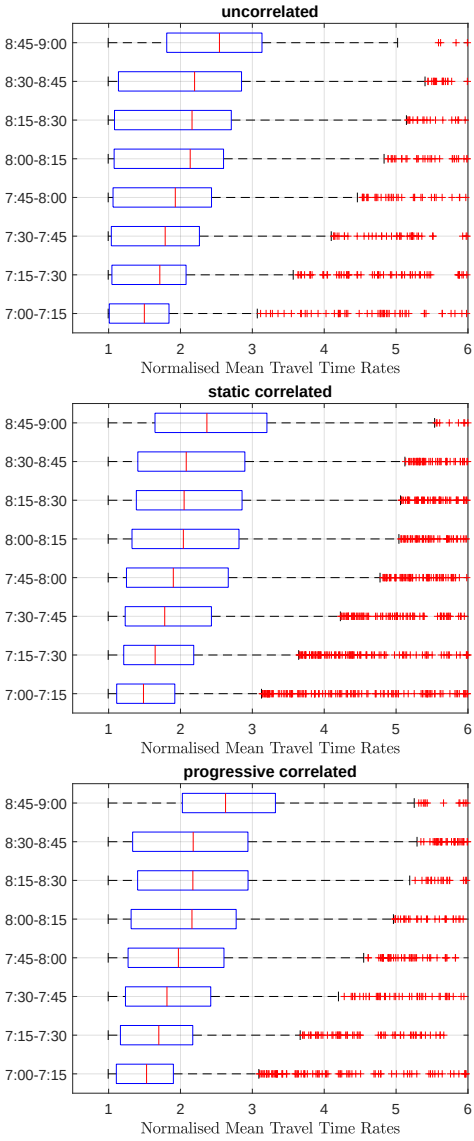


FIGURE 2.3: Normalized Mean Travel Time Rates (top: uncorrelated middle: static correlated, bottom: progressive correlated).

median value rates in progressive correlated box plots are slightly higher

than the static correlated ones. In the Appendix (A), similar results are presented for another weekday and a weekend.

In Figure 2.4, there is a clear tendency in the depicted time interval that streets converge toward much higher travel time rates than avenues. This confirms the empirically known fact that traffic on streets is slower than traffic on avenues (Bertsimas et al., 2019). These values are consistent with previous studies, which have found that the average traffic speed during the day in eastern Midtown is 6.3 mph (Zhan et al., 2013). This corresponds to the values 3 to 4 in Figure 2.4. Similar to Figure 2.3, the mean value rates depicted in the progressive correlated version are slightly higher than the static correlated one.

Moreover, in Figure 2.5, we show the results in the form of normalized relative differences. The relative differences are calculated based on the order of the models written in the title of each diagram. For instance, the relative difference progressive - static is calculated as follows:

$$\frac{\text{Normalized } \mu_{\text{progressive model}} - \text{Normalized } \mu_{\text{static model}}}{\text{Normalized } \mu_{\text{static model}}} \cdot 100\%$$

Figure 2.5 gives an instant overview of the changes in mean travel time for each link; however, the best comparison between the performance of the models is presented in Table 2.3, which is discussed later.

2.5.1 *Convergence analysis*

As explained in Section 2.4.0.2, the change in the parameter (mean and variance) values should become smaller up to a point where the parameter values will no longer change significantly through iterations. This is called convergence. Table 2.2 presents the mean relative differences for the mean travel time values associated with all links and all time intervals in different models for the case study. The result shows that all three models, after 100 iterations, have converged to an acceptable mean relative difference.

2.5.2 *Comparing our results against other benchmarks*

In this section, we present the results of our exploration through available benchmark data and compare our results against them. For one of the benchmarks, we decided on travel time data provided by the Google direction API (Google Developers, 2020). Google historical data is used among

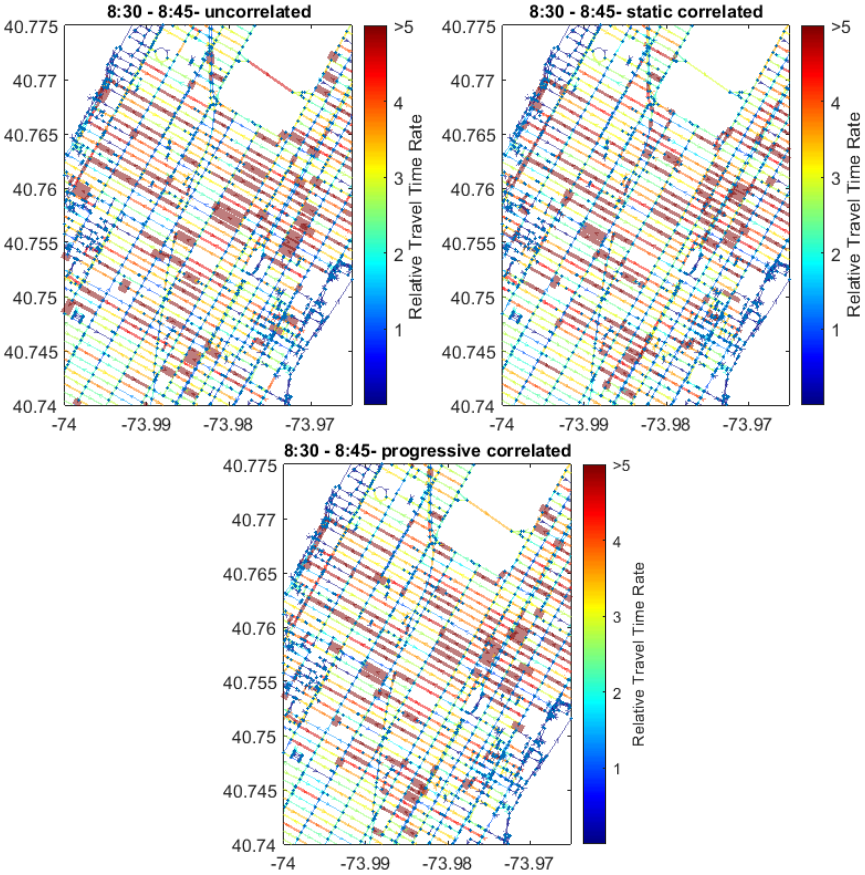


FIGURE 2.4: Normalized Mean Travel Time Rates on Manhattan network(top: uncorrelated, middle: static correlated, bottom: progressive correlated).

Model	mean relative difference % with 100 iterations
Uncorrelated (Herring's baseline model)	-0.01
Static Correlated	-0.02
Progressive Correlated	-0.02

TABLE 2.2: Convergence criteria results

other researchers as a comparison benchmark (Genser et al., 2022). Google historical travel time data is fetched through third-party website Outscraper, 2022 in which we could extract the instantaneous travel time from an origin to a destination exactly for the study time and date. The complete manual of how to extract historical data from Google is explained in Outscraper, 2022 for an interested reader. First, we tried to fetch all the travel times for all links in our network and produce travel times of the traveled paths by taxis reported by TLC by adding the travel time of the links (NYC Taxi and Limousine Commission, 2019). Since TLC does not report the exact path, we used the 20-shortest path calculated based on Yen, 1970's algorithm for each observation and chose the path with the lowest length difference from TLC's reported path length. In this approach, we realized there is a large discrepancy between the path travel time reported by TLC and the one we calculated by adding up the Google links travel times. Therefore, we extracted the exact path travel times from Google data with the same origin and destination reported by TLC. By this step, we tried to understand if the problem was raised by summing up the link travel times or not. Unfortunately, the same pattern was observed in the path travel time difference. Considering this problem, we could not directly consider Google data as a benchmark and tried to use their data in the following way.

We assume that the ratio of a link travel time to the path travel time is the only valuable data from Google we can benefit from. Since both the summation of link travel time and path travel time from Google is very different from the travel times reported by TLC, the only useful information is the proportion of the link travel time over the path travel time reported by Google. By obtaining all the link travel data and path travel time data from Google, we calculated the ratios for each link and path. By multiplying this ratio by the path travel time reported by TLC based on the following equation:

$$X_{l,\text{google benchmark}} = \frac{X_{l,\text{google}}}{Y_{p,\text{google}}} \times Y_{p,\text{TLC}}, \forall l \in L_p \tag{2.20}$$

$$X_{l,\text{google benchmark}} \sim N(\mu_{l,\text{google benchmark}}, \sigma^2_{l,\text{google benchmark}}), \quad \forall l \in L_p$$

we get the distribution of Google benchmark instantaneous travel times for each link. The mean of this distribution is considered as Google benchmark data for each link in our analysis.

Furthermore, to have another data set to compare our results, we use the baseline model proposed by Herring, 2010 and show the result against

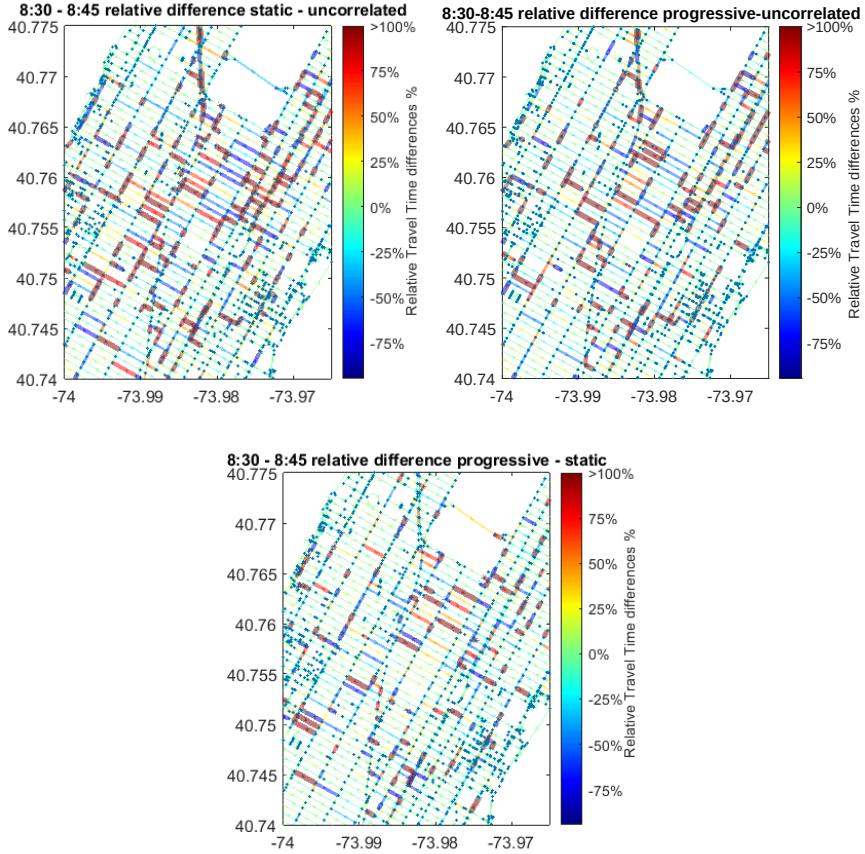


FIGURE 2.5: Normalized relative differences of mean travel times in different models on the Manhattan network.

this benchmark. In Figure 2.6, the histograms of normalized travel time rates are depicted for progressive correlated, static correlated, uncorrelated, and calculated Google benchmark as explained previously. Moreover, the comparison of RMSE is presented for all three methods in the following table. The metrics in Table 2.3 are calculated based on the following equations:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(\hat{x}_i - x_i)^2},$$

$$MPE = \frac{100\%}{n}\sum_{i=1}^{n}\left(\frac{x_i - \hat{x}_i}{x_i}\right),$$

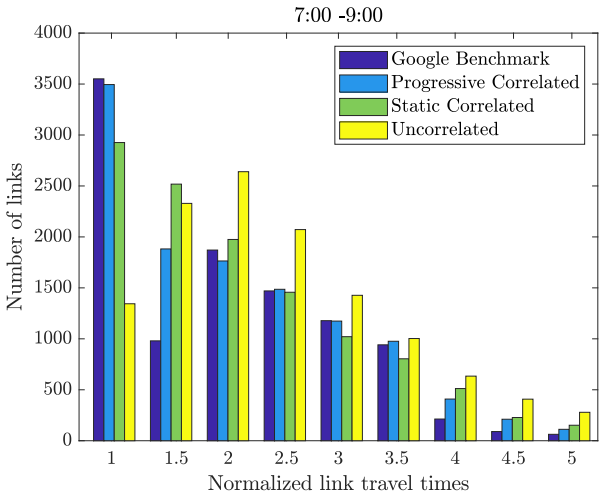


FIGURE 2.6: Normalized Mean Travel Time Rate Comparisons

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right|,$$

where x_i is the i^{th} path observed value for travel time reported by TLC in (NYC Taxi and Limousine Commission, 2019) and $\hat{x}_i$ is the estimated path observation achieved by summation of the link travel times in that path. Negative values of MPE mean that the estimated value is larger than the observed value.

Model	RMSE (sec)	MPE	MAPE
Uncorrelated(Herring's baseline model)	143.85	-5.47%	19.67%
Static Correlated	134.39	-4.41%	18.17%
progressive Correlated	127.73	-3.15%	16.71%

TABLE 2.3: Experimental results comparison between the proposed models and the baseline model.

The trend in Figure 2.6 shows that, in all normalized travel time rates, the progressive correlated model is closer to the Google benchmark data compared to the static correlated model results. However, the result in Figure 2.6 is very aggregated, and the comparison between the three models is best achieved by comparing the metrics in Table 2.3. In Table 2.3, we

observe that the progressive model values are showing the best result. Therefore, the progressive model can estimate the links' travel time more accurately than the other models.

2.6 CONCLUSIONS

This chapter proposes a methodology to estimate historical link travel times based on GPS OD data; historical means that the parameters uniquely belong to a past time period. Of course, such a process could be applied in a real-time setting or a hybrid model by combining historical estimates and real-time measurements. The proposed model infers the unknown path by the cabs with the simple assumption that the cabs always travel the shortest path based on the distance, and the difference between the observed and calculated path is reduced by calculating up to the 20-shortest path utilizing Yen, 1970's algorithm. The link travel times and their corresponding variances can then be estimated by formulating a maximum likelihood function. This optimization problem is computationally challenging but can be tackled by an iterative decomposition scheme suggested by Herring, 2010. In order to consider the spatial correlation, we have proposed a spatial correlation matrix for each sub-network and adopted the methodology for correlated links.

The model was applied to the Manhattan network for quarter-hourly time intervals from 7 am to 9 am on Tuesday, 1st of February 2011. The data used in this study were collected by the yellow New York City taxi cabs and are provided by NYC Taxi and Limousine Commission, 2019. The time of day had a significant effect on the means and variability of the travel times, with travel times gradually increasing on many links from 7 am to 9 am. The algorithm correctly detected a spatial pattern of streets having higher relative travel times than avenues in all time intervals. Furthermore, by comparing our results against other benchmarks, we show that the consideration of progressive correlation can improve the results, thus leading to a more accurate parametric travel time estimation approach. The proposed methodology can be applied to any GPS probe vehicle data set, for instance, synthetic data provided by (Batista et al., 2022; Flötteröd & Bierlaire, 2013) or real data set (NYC Taxi and Limousine Commission, 2019; Chicago Open Data, 2020), given that the data provide the origin, destination, and path travel time. Furthermore, the higher number of observations for a link travel time can increase the accuracy of the proposed methodology (Herring et al., 2010).

This study proposes a more accurate approach for estimating travel times that fully utilizes the partial information received from taxi data in cities as well as known or constructed (static or progressive) spatial correlations.

3

REAL-TIME RIDESHARING OPERATIONS FOR ON-DEMAND CAPACITATED SYSTEMS

The chapter is based on the following publications:

- Ghandeharioun, Z., and Kouvelas, A. (2023). "Real-time ridesharing operations for on-demand capacitated systems considering dynamic travel time information". *Transportation Research Part C: Emerging Technologies, 151, 104115.*
- Ghandeharioun, Z. and Kouvelas, A. (2020). "Online fleet management operations for on-demand capacitated ridesharing systems". Paper presented at the *99th Annual Meeting of the Transportation Research Board, TRB.*
- Ghandeharioun, Z. and Kouvelas, A. (2019)."Providing real-time operational solutions for the on-demand capacitated ride sharing problem". Paper presented at the *8th Symposium of the European Association for Research in Transportation, hEART.*
- Ghandeharioun, Z. and Kouvelas, A. (2019)."Providing real-time operational solutions for the on-demand capacitated ride sharing problem". Paper presented at the *7th meeting of the EURO Working Group on Vehicle Routing and Logistics optimization, VeRoLog.*

3.1 INTRODUCTION

As the urban population is growing (United-Nations, 2014), the need for more efficient transportation services motivates the authorities to implement new technologies in mobility services. In recent years the growth of smartphone technologies and inexpensive cellular communications have led to more individualized transport in urban areas; companies like Uber, Lyft, Via, Cruise, and Moia have risen that focus on developing demand-responsive services, known as Mobility-on-Demand (MoD). Furthermore, these companies have adjusted their services with sharing options considering the ridesharing potential and benefits. On the other hand, with the

parallel rising of automated driving technologies, semi- or fully-automated ridesharing services would be an attractive option in the near future. Satisfying customer needs cost-effectively has been the goal of many ridesharing systems (Zardini et al., 2021).

Ridesharing has improved urban mobility by providing reliable and reasonable on-demand services at any time. In the last decade, ridesharing has attracted a considerable share of demand, making up more than 50% of the rides provided by Lyft company in San Francisco and more than 30% of Lyft rides in New York City after one year of its introduction (Soper, 2015); moreover, the operational strategies that can be used to optimize on-demand ridesharing in the literature have shown that 20% of the current NYC Yellow taxi are sufficient to serve 98% of the demand (Alonso-Mora et al., 2017). Furthermore, Zhang et al., 2015 show that only 40% of the current fleet in Singapore is required to serve the personal mobility needs of the entire population. A better understanding of the complex ridesharing problem would allow for more effective system operation. The challenges arise from the fact that multiple stakeholders are involved with conflicting interests. If the objective of a private ridesharing company is to increase its revenue by offering more rides, this is in contradiction with the interests of governments (e.g., less traffic congestion and pollution). At the same time, a customer's objective is usually to travel from point A to B in the fastest and most inexpensive way.

The current work focuses on the operational assignment problem of on-demand ridesharing services. Our aim is to gain insights into the problem by modeling various stakeholders' objectives and designing policies that lead to efficient and mutually beneficial solutions (see 3.4.4). This is achieved through a real-time simulation framework that models the matching of supply and demand in a dynamic and deterministic manner, taking into account the tolerance times provided by the users. The optimization problem is solved using both heuristics and commercial solvers in a fast and efficient manner, allowing us to generate insights into the problem and identify areas for further improvement.

The key contributions of this work are:

- Development of a modular real-time simulation framework for the capacitated ridesharing problem.[1]
- Formulation of the ridesharing problem as a dynamic deterministic on-demand matching problem with tolerance times.

1 Capacitated ridesharing refers to a ride with a vehicle (autonomous or with a driver) accommodating up to 10 riders.

- Implementation of dynamic congestion by regularly updating link travel times during the simulation horizon.
- Solving the optimization problem in an online manner using both heuristics and commercial solvers.
- Modelling of stakeholders' multiple objectives and design of policies that lead to efficient and mutually beneficial solutions.

The present chapter is organized as follows: first, we briefly go through the related works and provide a review of different approaches for similar problems in Section 3.2. In the subsequent section, the studied problem is described in detail. The structure of the framework proposed by the current work is introduced in Section 3.4, and the optimization model is explained in detail (Section 3.4.4). The case study to evaluate the framework is presented in Section 3.5, with more insights about the simulation setup and simulation results. The chapter is then closed with the conclusions and final remarks in Section 3.6.

3.2 LITERATURE REVIEW

Optimization of on-demand ridesharing services has recently attracted a lot of research interest (Alonso-Mora et al., 2017; Simonetto et al., 2019; Bongiovanni et al., 2019). The general definition of a ridesharing assignment problem is how to optimally match the requests to vehicles and transfer people from an origin to a destination by reducing costs. Given that all the input data are available before determining the routes, the problem is classified as a static optimization problem (e.g., see Bongiovanni et al. 2019). On the other hand, in the dynamic version of the problem, some of the input data are communicated during the time horizon of the operational process (e.g., customer requests). Hence the solutions we are seeking are known as strategies to decide for real-time operations when a new request is received. Another classification presented in (Pillac et al., 2013; Hyland & Mahmassani, 2017) is based on whether the information received by request is certainly known (deterministic) or still undetermined and subject to changes (stochastic).

Regarding the classifications mentioned above, we focus on the dynamic deterministic ridesharing assignment problem in the current work. More specifically, we consider a ridesharing service with M shuttles. The operator faces unknown future requests with a pick-up and drop-off location. In addition, two different tolerance times are defined by the passenger within a request. One tolerance time is the waiting time for pick-up, which is the

time the passenger is willing to tolerate for being picked up, hereafter called D_i^p. The other tolerance time is the extra time the traveler is willing to accept induced by ridesharing, called D_i^d. Considering these two tolerance times, the problem can be formulated as a dynamic deterministic ridesharing problem with tolerance time.

Considering the literature, there has recently been an increase in investigating on-demand ridesharing services from different aspects. (Psaraftis et al., 2016; Zardini et al., 2021; Ho et al., 2018; Cordeau & Laporte, 2007; Hyland & Mahmassani, 2017). Here we review the most recent approaches that address dynamic deterministic on-demand ridesharing. The study in Ho et al., 2018 classifies the solutions into theoretical and experimental approaches. The theoretical solutions include a) an online algorithm, which has a proven competitiveness ratio versus its offline counterpart, or b) a methodology to compute a lower bound which is tighter compared to the previously introduced lower bounds (see, e.g., Waisanen et al. 2008; Yang et al. 2004). On the other hand, experimental approaches mainly develop simulation engines or other dynamic models. In these approaches, a new input (event) triggers the simulation engine or the model to make decisions in a short time. As stated in Ho et al. 2018, a passenger request is, in most cases, the simulation trigger for rescheduling the vehicles' routes (e.g., Berbeglia et al. 2012; Häll et al. 2015). Such approaches aim to serve the new request optimally. In most of the studied cases, there is a penalty when a request is rejected. The work in Ho et al. 2018 has recommended considering other triggers (events) for rescheduling as well, such as vehicle breakdowns and unexpected events, to have a more realistic representation of the dynamic deterministic ridesharing problem (see, e.g., Beaudry et al. 2010).

The operational tasks for ridesharing problems are categorized into four tasks: dispatching, routing, rebalancing, and ridesharing (Zardini et al., 2021). Most works have focused on solving four tasks together, Alonso-Mora et al., 2017 solved dispatching, routing, and ridesharing via an integer linear program and solved rebalancing through linear optimization. They provide a mathematical model for real-time high-capacity ridesharing that dynamically generates optimal routes concerning online demand and vehicle locations. The algorithm starts from a greedy assignment, makes a sharability network, and improves it via constrained optimization, quickly returning solutions and converging to the optimal assignment over time. Furthermore, the authors show that, with only 3,000 shared AVs instead of 13,000 registered taxis, 98% of the taxi rides in New York City, could be

served and that ridesharing leads to substantial additional benefits (e.g., less travel distance). Fielbaum et al., 2021, extended the work of Alonso-Mora et al., 2017 to assess the benefits of including walking sections to maximize system performance.

Sayarshad and Chow, 2017 focus on idle vehicle repositioning via queueing-based formulation; a Lagrangian Decomposition heuristic is developed. Using New York taxicab data, the proposed algorithm reduces the cost by up to 27% compared to the myopic case. Ma et al., 2019 provide a ridesharing strategy with a transit-oriented approach, in which they focus on dispatching and rebalancing in a bimodal network. Simonetto et al., 2019 decoupled ridesharing problem into two linear assignment sub-problems: 1) calculating costs for each vehicle by solving a single dial-a-ride problem, and 2) linear assignment of customers to the vehicle routes. Fagnant and Kockelman, 2018 investigate the fleet size and profitability optimization in different operation settings of shared autonomous vehicles using agent and network-based simulations. Pelzer et al., 2015 provide a ridesharing method by limiting the detour by dividing the network into distinct partitions to lower the algorithm's search space utilizing the ridesharing potential. Tsao et al., 2019 framed dispatching, routing, rebalancing, and ridesharing as network flow problems and formulated an Integer Linear Program (ILP) to optimize the costs and times of operations, implemented in real-time in a receding-horizon fashion that relies on forecasting for the demand.

One of the contributions of this work is that we consider the spatiotemporal dynamics of congestion in our approach. There are other studies in literature (Alonso-Mora et al., 2017; Ota et al., 2017; Simonetto et al., 2019) that have ignored this aspect; here, we include dynamic travel time estimates per link that are updated every 15 min; the same taxi dataset is utilized for these estimates, and the improved process of obtaining these is presented in Ghandeharioun and Kouvelas, 2022. Real-time traffic information in the simulator framework gives a realistic insight into the proposed approach's performance in a real-world scenario. However, estimating travel time based on traffic conditions in a network is a broad topic on its own, and many researchers are developing new methods under this subject, (e.g., Bertsimas et al., 2019; Zhan et al., 2013; Herring et al., 2010). Furthermore, when a network representation of the map is available, standard techniques for efficiently computing shortest paths can be used (Delling et al., 2009). The best methods can compute the shortest paths on networks with 70 million edges in less than a millisecond. Therefore, updating the shortest path with

new techniques presented in Delling et al., 2009 is not an issue for online operations.

As discussed in most review papers (Zardini et al., 2021; Ho et al., 2018; Hyland & Mahmassani, 2017), ridesharing has other aspects that attract researchers to study this problem. Gao et al., 2017 focus on infrastructure aspects of ridesharing systems and provide a multi-objective approach evaluating 10 metrics related to global efficiency, complexity, passenger, and platform incentives in settings designed to closely resemble reality in every aspect, focusing on vehicles of a capacity of two. Levin et al., 2017 provide a method based on the cell transmission model dynamic network loading simulator with a heuristic approach for real-time ridesharing and show how traffic congestion and travel patterns are affected by shared autonomous vehicles. Boesch et al., 2016 study the shared autonomous vehicles fleet size problem for the greater Zurich region, Switzerland, using a spatially and temporally highly detailed travel demand. They show that if waiting times of up to 10 minutes are accepted, a reduction of up to 90% of the total vehicle fleet can be possible even without active fleet management, like vehicle redistribution. Interested readers are referred to (Ho et al., 2018; Zardini et al., 2021; Hyland & Mahmassani, 2017).

3.3 PROBLEM DESCRIPTION

We aim to solve the ridesharing assignment problem in an online manner, as real-time requests arrive by the users in the operational center. Users are assumed to use an interface (e.g., smartphone app) to request a ride. They are also willing to share the vehicle with other passengers and provide two tolerance times for pick-up and drop-off.

We are considering a ridesharing service with M shuttles. The fleet is a set of shuttles with predetermined specifications. However, each shuttle may have unique characteristics, and the entire fleet may, in theory, be heterogeneous. Each shuttle $j \in \mathcal{M} = \{1, 2, \ldots, M\}$ is regarded as a separate object that is constantly moving and has the following parameters: capacity (C_j), current occupancy (P_j).

The operator is faced with the task of assigning unknown future requests with designated pick-up and drop-off locations to shuttles. These requests contain two key variables, defined by the passengers, which must be considered in the assignment process. The first variable, D_i^p, represents the maximum waiting time the passenger is willing to tolerate for pick-up. The second variable, D_i^d, represents the maximum extra time the passenger is

willing to accept due to the nature of ridesharing. By taking into account these two variables, the assignment problem can be framed as a dynamic deterministic ridesharing assignment problem with tolerance times.

The foundation of the road network is a directed graph. Roads and intersections, respectively, are represented by directed edges (E) and vertices (V). A graph $G(V, E)$ comprises the network. Moreover, in the current work we consider the spatiotemporal dynamics of congestion in our approach. Here, we include dynamic travel time estimates per link that are updated every 15 min; and update the shortest path queries for the further assignments of the arriving requests. Also, the arrival time and delay of the already matched requests are updated accordingly. Further explanation is provided in Section 3.5.1.

The operator receives every entity, including ride requests and the information associated with them, fleet data, and shortest path indices from the road network (updated travel times between the graph's vertices). The requests arrive in batches. To analyze the requests, we have fixed the time interval to be t, with a duration ranging between 5 and 30 seconds. Within each interval t, a number of requests are received by the operation center. It is important to note that while the time interval t remains constant, the number of requests received, represented by N, may vary from one interval to another. All the requests in a batch are matched to the shuttles in a parallel process. Each request is matched with the best candidate shuttle in a one-to-one approach.

The optimization algorithm helps the operator to find the best solution subject to predefined policy constraints that are established by the operator. Then based on the batch processing algorithm presented in Section 3.4.3, the process is repeated until all requests in the batch are either assigned to a matching shuttle or rejected.

The designed matching optimization algorithm needs to be fast enough (i.e., provide a response in some seconds) and consider different objective criteria the system's operator offers. Moreover, the main components of our framework are illustrated in Figure 3.1 and described in the next section in more detail.

3.4 METHODOLOGICAL FRAMEWORK

In this section, we present the developed simulation framework and its components in detail, together with the batching algorithm for processing many requests simultaneously. Moreover, we present the optimization

problem and different objective functions as representatives of different operational policies.

3.4.1 *Ridesharing operations simulation framework*

This section presents the developed simulation framework and the way it is utilized for online ridesharing operations. As the operator receives a new request, the framework decides in real-time (i.e., in some seconds) among all possible candidate shuttles to accommodate the passenger cost-efficiently. The cost is defined by the policy (e.g., delay minimization, sharing maximization) that the operator decides. In this process, the tolerance times provided by the users are the main constraints to choose a feasible shuttle. In the following, we introduce the simulation components depicted in the diagram in Figure 3.1.

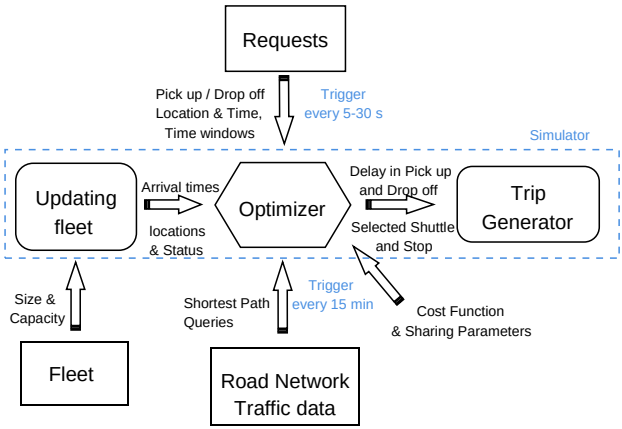


FIGURE 3.1: The shuttle ridesharing simulator diagram

3.4.2 *Simulation components*

The main components of our simulation framework are illustrated in Figure 3.1 and described here in more detail[2].

- *Requests*: The ride request $i \in \mathcal{N} = \{1, 2, \ldots, N\}$ consists of the following parameters, number of passengers in the request i, P_i, the time

2 A list of all notations is provided in Table C.1 in the Appendix C

Parameter	Description
i	request number $i \in \mathcal{N} = \{1, 2, \ldots, N\}$
N	total number of requests in a batch
P_i	number of passengers in the request i
T_i^p	time requested for pick-up (the time that the request arrived at the operator)
T_i^d	calculated drop-off time (if the request is served without any delay)
O_i	pick-up location (origin)
D_i	drop-off location (destination)
D_i^p	maximum waiting time that the commuters will tolerate as delay in pick-up
D_i^d	maximum in-car delay accepted by the passengers caused by ridesharing

TABLE 3.1: Ride Request Parameters

that the request arrived at the operator T_i^p, pick-up location or origin O_i, a tolerance time for the tolerable waiting time or delay in pick-up defined by the passenger: D_i^p, drop-off location or destination D_i, the tolerance time or additional time accepted by the passenger caused by sharing the ride with other requests, D_i^d. The operator also calculates the estimated earliest drop-off time if the request is served without any delay in pick-up or drop-off, T_i^d. The parameters are listed in Table 3.1.

- *Fleet*: The fleet represents a set of shuttles with predefined specifications. However, each shuttle can have different parameters, and the whole fleet, in principle, can be heterogeneous. We consider each shuttle $j \in \mathcal{M} = \{1, 2, \ldots, M\}$ as a distinct object always on the move and having the following parameters: capacity C_j, current occupancy (passengers that are already in shuttle j), P_j. Every shuttle has a chronologically sorted list of stops S_{K_j}. Each stop k_j contains the following information: stop location L_{k_j}, estimated arrival time to the stop A_{k_j}, and stop type. Stops can be either a pick-up or drop-off or a hot spot. A hot spot is a destination for a shuttle to increase the chance of serving more passengers. All stops in the stop list also have the corresponding request information related to the pick-up or drop-off stop. The parameters are listed in Table 3.2.

- *Road Network*: A directed graph represents the basis of the road network. Directed edges (E) and vertices (V) represent roads and intersections, respectively. The network constitutes a graph $G(V, E)$. If a road is a two-way street, two edges will be defined for that segment.

Parameter	Description
j	shuttle number $j \in \mathcal{M} = \{1, 2, \ldots, M\}$
M	total number of shuttles in the network
C_j	capacity of shuttle j
P_j	current occupancy of shuttle j (passengers that are already in shuttle j)
S_{K_j}	list of stops of shuttle j , $S_{K_j} = \{0_j, 1_j, 2_j, \ldots, K_j\}$
$K_j + 1$	the total number of stops in shuttle j
k_j	k_{th} stop in shuttle j, $k_j \in S_{K_j}$
L_{k_j}	location of stop k in shuttle j
A_{k_j}	estimated arrival time to stop k of shuttle j

TABLE 3.2: Shuttle Parameters

We denote with T_{IJ} the time required to travel between vertices I and J. We assume that each trip starts/ends at a vertex, and if a pick-up/drop-off location is in the middle of the edge, it is projected to the nearest vertex. Using this framework, we utilize real-time information about the traffic conditions as a weight for each edge. This can be updated to the network dynamically over time (in regular intervals, e.g., 15 minutes). The process of calculating the link weights based on real-time traffic information of the network is explained in Section 3.5.1.

- *Simulator*: The simulator receives all the entities: ride requests and corresponding information, fleet information, shortest path indices from the road network (updated travel times between the vertices in the graph), and traffic information. The simulator is triggered every time interval t, and multiple requests are received to perform the following process, 1) updating the status and location of the shuttles, 2) running the optimizer, and finding the best shuttle to accommodate the requested ride. 3) Generating the trip and assign to the selected shuttle. The algorithm behind the simulator is described in the simulation process presented in Section 3.4.3.

- *Optimizer*: The optimizer helps the simulator to decide on the optimum solution subject to predefined policy constraints that are set by the operator. The algorithm used in the optimizer is described in Section 3.4.4.

- *Cost Function*: Let $f(i, j)$ denote the cost function, i.e., the cost for a shuttle j to accommodate request i. In the present work, the cost function is the objective function calculated by the optimizer algorithm. Further details about different objective functions are provided in Section 3.4.4.
- *Trip Generator*: The outcome of the optimizer is delivered to the trip generator, and the request is assigned to the appropriate shuttle. Moreover, considering the changes that occurred with this new assignment, all the other trips in the shuttle are updated accordingly.

3.4.3 *Periodic batch processing of requests*

This section outlines the process carried out by the simulator for each batch of requests. The simulation algorithm is initiated when multiple requests are received within a time interval t, as outlined in Algorithm 1. At any given time interval t, we have a number of requests to be fulfilled (N), referred to as a batch, and M shuttles operating in the network.

The first step is to update the positions and states of the fleet based on the elapsed time since the previous trigger. Then, each request in the current batch is sent to the optimizer in a parallel process to determine the best matching shuttle, as determined by the optimization model outlined in Section 3.4.4. This process is performed concurrently for all requests in the batch. The result of each parallel optimization process is the optimal

Algorithm 1: Algorithm for processing batch of requests in the simulator

Input: Requests information, shuttles information
Output: Generated trips by assigning accepted requests to the shuttles.
Step 1: Run every request alone and optimize matching (parallel processing).
Step 2: Get the best solution for every request i and sort them based on the objective function value (according to the selected policy).
Step 3: Accept the first unique requests of the ordered list, assign them to shuttles, and remove them from $\mathcal{N}$. Reject all requests that are infeasible or have the same shuttle number as the accepted requests and remove them from $\mathcal{N}$.
Go to **Step 1** and repeat until $\mathcal{N} = \emptyset$

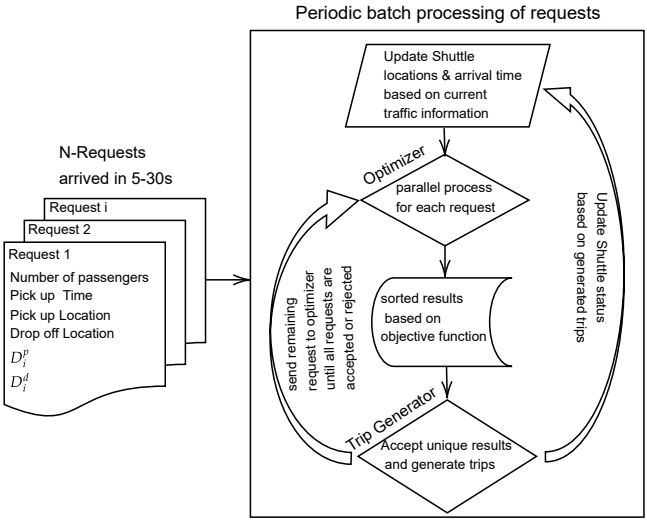


FIGURE 3.2: Batch processing diagram

matching of a request to a shuttle. These results are sorted based on the selected objective criterion (e.g., minimum delays, maximum acceptance rate).

Next, the requests are assigned to the corresponding matching shuttles based on the sorted list and removed from the batch. In the case where multiple requests are matched to the same shuttle, the one with the lowest value of the objective function is assigned, while the rest are removed to be processed again. During this step, all requests that were rejected by the optimizer as infeasible are also removed from the batch (as explained in the optimization model, Section 3.4.4). This three-step process is repeated until all requests in the batch have been either assigned to a matching shuttle or rejected. This process is illustrated in Figure 3.2.

3.4.4 *Optimization model*

This section outlines the key contribution of our work, the optimization model. The optimization model aims to solve the problem of assigning requests to shuttles in a way that meets the needs of both the operator and the passengers. Every request, denoted by i, is represented by three key variables: the pick-up desired time T_i^{p}, which is the time that the request

is received by the operator, the pick-up desired location O_i (origin), and the drop-off location D_i (destination). In addition to these variables, every request also specifies two tolerance times. The first tolerance time is the maximum delay tolerance for pick-up D_i^{p}, which represents the maximum waiting time that the passenger is willing to endure. The second tolerance time, D_i^{d}, represents the maximum in-car delay that the passenger is willing to accept. The optimization model, formulated as an integer program, decides whether to accept or reject each request and, if accepted, assigns it to a shuttle and schedules stops for pick-up and drop-off accordingly.

Every shuttle $j \in \mathcal{M} = \{1, 2, \ldots, M\}$ has capacity C_j and at time t carries P_j passengers. Every shuttle has a sorted stop list $S_{K_j} = \{0_j, 1_j, 2_j, \ldots, K_j\}$. Every stop k_j of shuttle j, where k_j denotes the index of the ordered sequence of stops, with $k_j \in \mathcal{S}_{K_j}$ corresponds to either a pick-up or a drop-off. The stop can also be a hot spot (not an actual customer stop but rather an operator-defined "dummy" stop) where the idle shuttles are led to be able to serve more customers. All stops in the stop list have a location L_{k_j} and an estimated arrival time to their location A_{k_j}. The variable 0_j denotes the current position of shuttle j in the network (i.e., the node of the graph) at time t. Obviously, if a shuttle is empty and cruising in the network, this implies that $P_j = 0$.

For every request i, we define the following binary decision variables

$$y_{k_j} = \begin{cases} 1 & \text{if the pick-up of request } i \\ & \text{is placed after stop } k_j \text{ of shuttle } j\,, \quad \forall j \in \mathcal{M},\ k_j \in \mathcal{S}_{K_j} \\ 0 & \text{else} \end{cases} \tag{3.1}$$

$$x_{k_j} = \begin{cases} 1 & \text{if the drop-off of request } i \\ & \text{is placed after stop } k_j \text{ of shuttle } j\,, \quad \forall j \in \mathcal{M},\ k_j \in \mathcal{S}_{K_j} \\ 0 & \text{else} \end{cases} \tag{3.2}$$

Remark 1: The subscript i for the variables y and x is not required as the optimization is solved for a single request at a time.

The problem at hand is a combinatorial optimization problem that will look for all possible configurations of matching the passengers to the shuttles and come up with the best solution.

Finally, all shortest path travel times T_{IJ} from any node I to any other node J of the network are pre-computed and saved in the memory; note, however, that they are updated in regular time intervals according to the prevailing traffic conditions.

3.4.5 *Constraints*

The optimization problem is formulated using the following constraints. To allow for flexibility in providing various objective functions that reflect different policies, all constraints are expressed as equalities with the use of variables called epsilon (e_1 to e_4). It should be noted that this conversion from inequalities to equalities does not impact the solution time. The epsilons in each constraint represent the deviation from the constraint's upper bound and can only take on positive values.

$$e_{1_j} \geq 0,\ e_{2_j} \geq 0,\ e_{3_j} \geq 0,\ e_{4_j} \geq 0, \quad \forall j \in \mathcal{M}$$

- Capacity constraint

$$\sum_{k_j \in \mathcal{S}_{K_j}} P_i \cdot y_{k_j} + e_{1_j} = C_j - P_j, \quad \forall j \in \mathcal{M} \tag{3.3}$$

  The constraint checks whether the number of passengers associated with request i, P_i, can fit the remaining capacity of shuttle j after stop k_j. Parameter P_i is multiplied by y_{k_j} (a variable 1 or 0) if only the shuttle j has the remaining capacity to accommodate the request i after stop k_j. The remaining capacity of shuttle j is equal to the total capacity of shuttle j, C_j, minus the current occupancy of shuttle j, P_j.

- No request can be placed in more than one shuttle (an infeasible solution is translated as a rejected request)

$$\sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} y_{k_j} = 1 \tag{3.4}$$

$$\sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} x_{k_j} = 1 \tag{3.5}$$

  Remark 2: Based on the constraint in Equation 3.4 and Equation 3.5 the algorithm is forced to put each request in one shuttle, if this is not possible, it means this constraint can not be met. Therefore, the

result of the optimization is an infeasible solution which is translated into a rejected request. Note that in a batch of N requests, we run the algorithm for each request separately, in a parallel process, so in the case of infeasibility only one request will be rejected, not all the batch.

- Pick-up and drop-off should happen in the same shuttle, but not necessarily after the same stop

$$\sum_{k_j \in \mathcal{S}_{K_j}} y_{k_j} - \sum_{k_j \in \mathcal{S}_{K_j}} x_{k_j} = 0, \quad \forall j \in \mathcal{M} \tag{3.6}$$

- Drop-off should take place after the pick-up

$$\sum_{k_j \in \mathcal{S}_{K_j}} k_j \cdot y_{k_j} \leq \sum_{k_j \in \mathcal{S}_{K_j}} k_j \cdot x_{k_j}, \quad \forall j \in \mathcal{M} \tag{3.7}$$

- Delay tolerance in pick-up location for all the accepted requests

$$\left(A_{k_j} + T_{IJ}\right) y_{k_j} + e_{2k_j} = T_i^{\mathrm{p}} + D_i^{\mathrm{p}}, \quad \forall j \in \mathcal{M},\ k_j \in \mathcal{S}_{K_j} \tag{3.8}$$

where $I = L_{k_j}$, $J = O_i$ and A_{k_j} is the estimated arrival time at stop k_j.

- Delay tolerance in drop-off location for all the accepted requests

$$\left(A_{k_j} + T_{IK} + T_{JI}\right) x_{k_j} + e_{3k_j} = T_i^{\mathrm{d}} + D_i^{\mathrm{d}}, \quad \forall j \in \mathcal{M},\ k_j \in \mathcal{S}_{K_j} \tag{3.9}$$

where $I = L_{k_j}$, $J = O_i$, and $K = D_i$.

- Delay tolerance in all pick-up and drop-off locations for the passengers moved after new insertion. In shuttle j, stop k is either a pick-up or a drop-off of a previously assigned request n, for that stop, we can calculate a remaining delay tolerance according to the request n initial delay tolerances and compare it to the delay caused by the new request i insertion as follow:

$$\begin{aligned}\sum_{k'=0}^{k'=k_j-1} (1 - x_{k'_j} y_{k'_j}) \left(\left(T_{IJ} + T_{JK} - T_{IK}\right) y_{k'_j} + \left(T_{IL} + T_{LK} - T_{IK}\right) x_{k'_j} \right) + \\ x_{k'_j} y_{k'_j} (T_{IJ} + T_{JL} + T_{LK} - T_{IK}) + e_{4k_j} = D_n^{p/d} - d_n^{p/d}, \\ \forall j \in \mathcal{M},\ k_j \in \mathcal{S}_{K_j} \backslash \{0_j\}\end{aligned} \tag{3.10}$$

where k' is a counter of stops from 1 to stop k and $I = L_{k'_j}$, $J = O_i$, $K = L_{k'_j+1}$, $L = D_i$, remaining delay tolerance at pick-up p or drop-off d for request n in the shuttle: $D_n^{p/d}$ (initial value) $-d_n^{p/d}$ (used value); the reader is referred to the Appendix (B) for a detailed explanation of constraint (3.10).

3.4.6 *Objective function*

The objective function of the problem can include different terms capturing e.g., passenger delays, penalties for rejection of service, quality of service (i.e., variance of delays among the passengers), maximization of the number of trips, etc. For now, we are focusing on the minimization of passenger delay which can be expressed as

$$\min_{y_{k_j},\, x_{k_j}} \quad \sum_{j\in\mathcal{M}} \sum_{k\in\mathcal{S}_{K_j}} \Big(A_{k_j} + T_{IJ}\Big)\, y_{k_j} + \sum_{j\in\mathcal{M}} \sum_{k\in\mathcal{S}_{K_j}} \Big(A_{k_j} + T_{IK} + T_{JI}\Big)\, x_{k_j} \quad (3.11)$$

where $I = L_{k_j}$, $J = O_i$, and $K = D_i$.

3.4.7 *Transformation to ILP and solver*

The presented optimization problem is essentially nonlinear because of the constraint in Equation 3.10.

However, it could be transformed into linear form (ILP) by considering a new variable $z_{k'_j} = x_{k'_j} \cdot y_{k'_j}$ [3] and defining the following constraints considering the binary characteristics of $x_{k'_j}$ and $y_{k'_j}$:

$$\begin{aligned} z_{k'_j} &\leq x_{k'_j}, \\ z_{k'_j} &\leq y_{k'_j}, \\ z_{k'_j} &\geq x_{k'_j} + y_{k'_j} - 1 \end{aligned}$$

See Glover, 1975 for more details. For solving the problem at hand we utilize the commercial solver Gurobi (Gurobi Optimization, LLC, 2021).

3 $x_{k'_j} \cdot y_{k'_j} \cdot y_{k'_j}$ is equal to $x_{k'_j} \cdot y_{k'_j}$ if $y_{k'_j} = [1, 0]$

Also, to reduce the solver's search space, we limit the number of shuttles in set $\mathcal{M}$ to the neighboring shuttles to the request. In this work in the presented case study, we consider the 30 closest neighboring shuttles to each request to solve the optimization problem.

3.4.8 *Exploring operational policies as objective functions*

In the current work, we have decided to investigate different operational policies. In order to benefit from the different e terms introduced in all the constraints, we have provided the following objective functions representing different policies based on operation preferences.

- Policy A: Maximizing level of sharing

$$\min_{y_{k_j},\, x_{k_j}} \quad \sum_{j \in \mathcal{M}} e_{1_j} \tag{3.12}$$

- Policy B: Maximizing operational reliability

$$\max_{y_{k_j},\, x_{k_j}} \quad \sum_{j \in \mathcal{M}} e_{1_j} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} e_{2k_j} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} e_{3k_j} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} e_{4k_j} \tag{3.13}$$

- Policy C: Minimizing delay by maximizing the distance to the upper bound

$$\max_{y_{k_j},\, x_{k_j}} \quad \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} e_{2k_j} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} e_{3k_j} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} e_{4k_j} \tag{3.14}$$

- Policy D: Maximizing acceptance rate by minimizing sharing

$$\max_{y_{k_j},\, x_{k_j}} \quad \sum_{j \in \mathcal{M}} e_{1_j} \tag{3.15}$$

- Policy E: Weighted combination of delay minimization and sharing maximization

$$\min_{y_{k_j},\, x_{k_j}} \quad \frac{\alpha}{\mathcal{M}} \sum_{j \in \mathcal{M}} \frac{e_{1_j}}{C_j - P_j} - \frac{1-\alpha}{3\mathcal{M} \cdot K_j} \left(\sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} \frac{e_{2k_j}}{T_i^{\mathrm{p}} + D_i^{\mathrm{p}}} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} \frac{e_{3k_j}}{T_i^{\mathrm{d}} + D_i^{\mathrm{d}}} + \sum_{j \in \mathcal{M}} \sum_{k_j \in \mathcal{S}_{K_j}} \frac{e_{4k_j}}{D_n^{p/d} - d_n^{p/d}} \right) \tag{3.16}$$

All aforementioned policies refer to different objective goals that the system operator could set in the optimization problem. This choice set corresponds to operational criteria in the sense of multi-criteria optimization. In this work, we are interested in exploring various operational solutions and evaluating their performance. Note that the proposed framework could also be utilized as a decision support system, i.e., operators could perform multiple scenarios on the fly, assess their performance and consider them when making real-time operational decisions. More precisely, in the results presented in the next section, there is a particular focus on objective function (3.11); many experiments have been investigated for this objective criterion. However, another part of our results focuses on assessing other criteria and explores the trade-offs among different operational policies.

The description of all studied policies in our experiments (and corresponding objective functions) is as follows: Policy A maximizes the sharing rate of the service (by minimizing spare capacity in all shuttles); Policy B maximizes the system's operational reliability (by maximizing the distance from binding for all operational constraints); the term "operational reliability" in policy B describes the algorithm's behavior in a more conservative objective function by summing the following terms: 1) having a higher acceptance rate by maximizing e_{1_j}, which might result in less sharing (opposite of policy A). 2) by having shorter delays in pick-up by maximizing e_{2_j}, which means less tolerance of the customers in using the service. 3) shorter drop-off delays by maximizing e_{3_j}, which means less tolerance in accepting sharing the rides. And 4) fewer delays in total for the already assigned requests by maximizing e_{4_j}. By maximizing all these terms, we can understand how reliable our algorithm is when customers have the lowest flexibility. Policy C refers to the same binding distance as Policy B

without including the capacity constraint e_{1_j} (the aim here is to look again at system reliability but at the same time incentivize ridesharing, i.e., we try to move away from private taxi service and aim at increasing sharing rate); Policy D is the exact opposite of Policy A, i.e., operators do not care about sharing rate and lean towards private taxi service (our conjecture is that this objective function could increase operator's acceptance rate and minimize rejection of trips – of course, this trend is not always obvious as derived solutions also depend on many other operational constraints); finally, Policy E refers to a weighted average of two conflicting criteria, namely, total delay minimization and ridesharing maximization; this objective function has been included to demonstrate the trade-offs that can arise among conflicting policies. One could create Pareto optimal solutions by creating various linear combinations of the aforementioned policies; this would most likely produce the most interesting solutions as they would be optimal in a multi-directional rather than mono-directional space.

3.4.9 *Properties of the algorithm*

Before presenting the results of our case study, we focus on elaborating on different aspects of our algorithm and comparing it to the existing well-known algorithms.

Our methodology differs from the recent work presented in Simonetto et al., 2019 in that we address the ridesharing problem through a combinatorial optimization approach, which simultaneously solves both the single vehicle Dial-a-Ride Problem (DARP) and the matching problem. In contrast, Simonetto et al., 2019 first calculates the cost of serving a request based on a single-vehicle DARP and an insertion heuristic, with a limit of four possible insertion position combinations. Our approach does not have this limitation. The second step in Simonetto et al., 2019 involves a linear assignment of requests, with the aim of minimizing cost among all relevant vehicles, utilizing the costs calculated in the vehicle logic module. The calculation of these costs (c_{ij} in Simonetto et al. 2019) in the vehicle logic module (Section 3.3 in Simonetto et al. 2019) involves a single-vehicle DARP to minimize route duration if there are three or fewer scheduled customers. For more than four customers, the insertion heuristic (Algorithm 1 in Simonetto et al. 2019) is applied, with possible insertion positions limited to four, and the result is further improved through the Local Neighborhood Search (Algorithm 2 in Simonetto et al. 2019). The application of destruction

and repair operators on the vehicle routes allows Algorithm 2 to explore a wider range of possible insertions, resulting in improved solutions.

In our methodology, the combinations of inserting new requests are accounted for within the optimization problem by incorporating constraints, specifically constraint (3.10). This allows us to define various cost functions, independent of the constraints. On the other hand, in the approach presented in Simonetto et al., 2019, the cost for the linear assignment is determined through the utilization of heuristics and takes into account only the detour time of inserting the request in different positions.

As noted in Simonetto et al., 2019, the computational demands of their methodology vary based on the number of scheduled customers per vehicle, as described in Section 3.3-Vehicle of Simonetto et al., 2019. Conversely, in our approach, the complexity of constraint 3.10, which evaluates the delay tolerance at all pickup and drop-off locations for passengers moved after insertion, is of order $O(M)$, where M represents the total number of shuttles. Additionally, our methodology is more straightforward for capacitated ridesharing systems with vehicle capacities greater than 4 and does not require different heuristics based on the number of scheduled customers. Ultimately, it is challenging to determine which of the two approaches is superior in various operational configurations.

Furthermore, in the famous work of Alonso-Mora et al., 2017, the method is based on multi-request, multi-vehicle assignment, which is very different from the method we have developed. Our method is solving the assignment of one request to the best candidate vehicle, and we parallelize our combinatorial matching algorithm for different requests in a batch of requests as explained in Section 3.4.3.

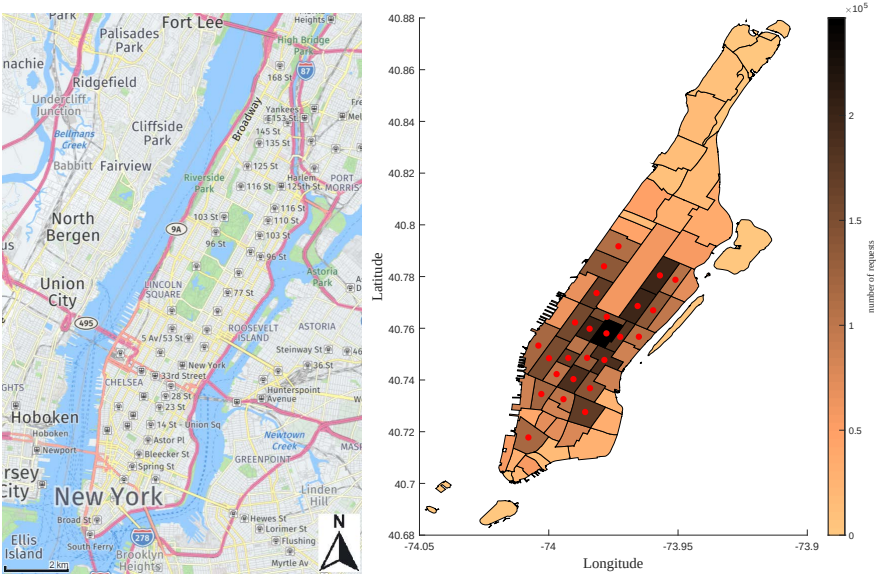
Considering hot spots in our work is a non-myopic technique to increase the efficiency of our algorithm. In most of the ridesharing algorithms, there are rebalancing methods in which cars that are not being used are sent to some zone where they might be needed more. The methods present in the literature differ mostly on how to measure the need for vehicles in each region. Some papers consider the current demand; in one of the methods introduced in Fielbaum et al., 2021, they modify the cost of each possible assignment between vehicles and set of requests, favoring those assignments that conduct the vehicle towards the most demanded zones and show that the vehicles-hour-traveled increases by about 10% and diminish the rejection rate to about 0.9 of its original value when no rebalancing method was used. In our work, we have implemented a similar technique as explained in Fielbaum et al., 2021.

3.5 CAPACITATED RIDESHARING IN MANHATTAN: A CASE STUDY

For the case study, we use the urban network of Manhattan in New York (see Figure 3.3(a)). The land area of Manhattan is around 59 km^2 and includes a grid road network consisting of 228 numbered streets running in the East-West direction (with ascending numbers as they move northward) and 11 avenues running in the South-North direction (with numbers ascending from east to west). There are 2,820 signalized intersections in Manhattan, and the speed limit is 25 mph. The digital model of the road network used in this case study consists of 4,092 nodes (intersections) and 9,453 edges (links). There are 58 taxi zones depicted in Figure 3.3(b), as reported by NYC Taxi and Limousine Commission, 2019. The color of each taxi zone represents the taxi trips demand, i.e., the darker the zone, the higher the demand for taxi trips, according to the data reported by TLC. The dispatching of shuttles in the case study is done randomly from taxi zones of the network, the more the historical demand of that zone, the higher proportion of the shuttles dispatched from that zone. The center of taxi zones with high demand, highlighted by red dots, are also defined as points of attraction (hot spots) for the shuttles in the current work, i.e., when shuttles idle in the network, they move towards these hot spots in order to increase the probability to accommodate more future requests; this is an assumption for this study to deal with idle shuttles. However, one could perform a completely new analysis on this interesting topic.

3.5.1 *Real-time traffic information*

In the current work, we adopt a straightforward approach based on the available data of our case study presented in Section 3.5. In this approach, we benefit from the New York taxi trip data reported by TLC (NYC Taxi and Limousine Commission, 2019). The reported trips contain information about the origin, the destination, and the travel time of each trip. However, the trip routes or taxi trajectories accommodating these trips are missing. In order to calculate the travel time for each link in the network, we implement the following process. First, by implementing the k-shortest path calculation for each set of origin and destination, we find a route for the given trip that minimizes the difference between the inferred and the observed path distance. Since the *k*-shortest path is a computationally expensive task,



(a) (b)

FIGURE 3.3: (a): Map of study area (source: https://wego.here.com). (b): Taxi zones in Manhattan as partitioned by New York City; color-bar depicts the demand from the utilized taxi dataset.

defining the k depends on the available resources for each study. After this step, the observations that violate the following inequality are removed.

$$0.5 \times \text{observed distance} < k\text{-shortest path distance} < 1.5 \times \text{observed distance}.$$

Second, the travel time reported for the trip is distributed among the links in the calculated shortest path based on the length of the link. With these two steps, we get many observations for each link in the network. It is worth mentioning that the proposed approach for calculating the travel times is only valid since there are hundreds of thousands of taxi travel time observations in Manhattan. Out of all these observations, we omit the ones violating the lower bound of travel time for a link, which is the free-flow travel time. Considering the speed limit in Manhattan of 25 mph reported in NYC Taxi and Limousine Commission, 2019, it can be assumed that the taxi drivers will travel up to 40 to 50 mph to calculate the minimum link travel time. Next, that link's travel time is reported by calculating the average of all the observations for a link in a defined time interval.

In the current work, we have implemented this process in the Manhattan network every 15 minutes. The average of all the observations for each link in 15 minutes over weekdays for one week is reported as the link travel time. For the links that the observations are not available (in the case of Manhattan, only a few links), the link travel time estimation is based on its neighboring links. The result of this process is the network graphs with updated travel time every 15 minutes used in the simulator. In Figure 3.4, the travel time of all links is normalized based on the link length and categorized into 5 different rates, from rate 1 depicted in Black representing free flow travel time to 5 illustrated in red as the highest travel time for a link, the graphs are presented quarterly for morning peak hours, as they are updated in the event based simulator. As seen in the network graphs in Figure 3.4, the travel time increases from 7:00 to 9:00 as we get closer to the morning peak hour.

In our work, we update the arrival time of each vehicle at each stop, including the already picked-up passengers, when the travel times of links

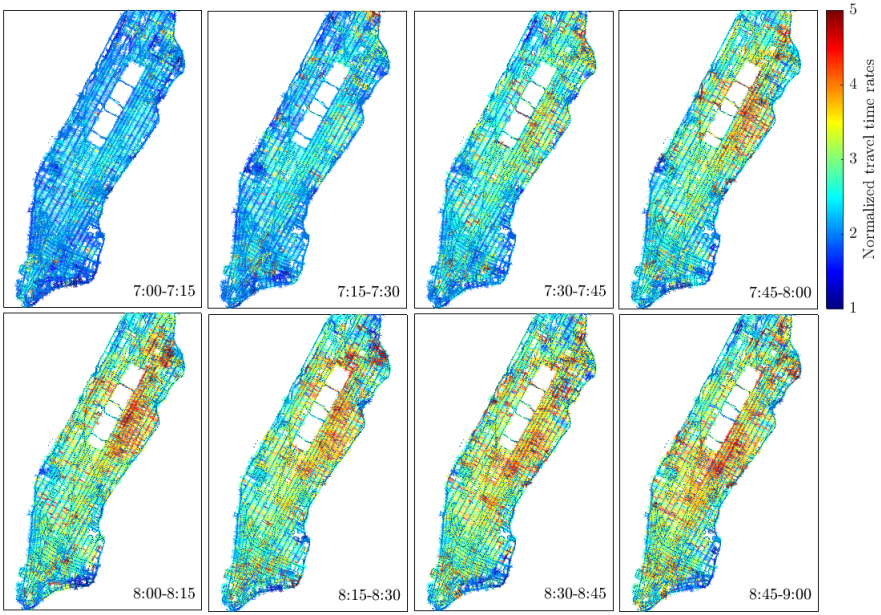


FIGURE 3.4: Normalized travel time rates in Manhattan network: from rate 1 (Black color) that denotes free flow travel time to rate 5 (red color) that depicts the highest estimated travel time rate.

are updated. This is a proposed solution in Hörl and Zwick, 2022. In our simulation setup, since the tolerance of delay, while sharing a ride D_i^d is set to 15 or 20 minutes, the amount of trips that violate this tolerance time due to travel time changes in the network is very limited. Moreover, the frequency of travel time updates which in our work is a quarter-hourly does not cause a significant difference in the trip travel times since the trips are usually very short and the travel time changes are not very large. Of course, the method of updating the travel time in the ridesharing algorithm can be adjusted in more efficient ways as proposed by Hörl and Zwick, 2022. Still, this adjustment is out of the scope of our current work and is a good direction for future work.

3.5.2 *Numerical simulations set-up*

We evaluate the performance of the proposed method using real data from a randomly selected sample day of NYC trip data available online at NYC Taxi and Limousine Commission, 2019. The dataset contains the following information: latitude and longitude for pick-ups and drop-offs accommodated by over 13,000 active taxis, number of passengers, and pick-up and drop-off times. We have only considered all the trips starting and ending in Manhattan. We have also cleaned and filtered the data that was not correctly reported (e.g., an unrealistically short trip). For the current work, the input data considers the peak hour trips between 7:00 a.m. and 9:00 a.m. on the first of February 2011. At this time of day, 18854 trips are reported by TLC. We consider the complete road network of Manhattan, depicted in Figure 3.5 with dynamic link travel times updated quarterly based on the method explained in Section 3.5.1. In our simulation, the fleet is initialized in the beginning by assigning random locations to all shuttles and moving them towards the predefined hotspots based on historical trip demand distribution (see Figure 3.3); then, they continuously circulate in the network and receive commands from our optimization routine in order to optimally accommodate the real (taxi) requests as extracted from the dataset at hand. Requests are collected and batched during a tolerance time – we have selected 15 seconds here, but this is a parameter of the designed framework. Batch requests are sent to the optimizer for processing, and after the solution is returned, they are assigned to the vehicles; the complete process flow is explained in Algorithm 1.

Our simulation experiments are divided into two parts. First, in Section 3.5.3, we focus on a specific objective function (equation (3.11)) and demon-

strate the obtained results and insights. We have investigated different sets of parameters for the simulation module for these scenarios. Parametrization of this part of the framework can affect the system's performance and provide insights into the quality of the obtained solutions. The computational efficiency of the simulation engine is not so crucial in this part (i.e., simulation speed); the vital part is the computation time needed to solve the optimization problem and to obtain solutions for batch requests in a reasonable time for real-time operations. Technological constraints, such as communication delays, simulation engine, mobile app performance, etc., are not considered in the current study.

This first part of the results has provided inspiration and motivation for the second stage (Section 3.5.4), where all simulation engine's parameters are fixed based on conclusions drawn from Section 3.5.3. In the second part, we focus on exploiting different objective functions and assessing

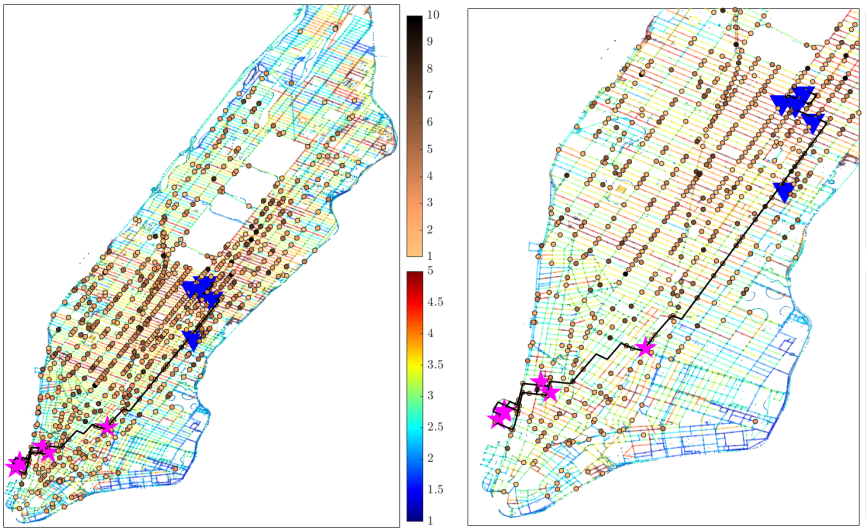


FIGURE 3.5: (a): A network snapshot with 1000 vehicles at their current locations. The color bar on the top indicates the vehicle's occupancy, ranging from 1 to 10. The real-time link travel time rate is illustrated by the bottom color bar, rating from 1 (Black, free flow travel time) to 5 (red, congested travel time). (b): Close view of a scheduled path with five pick-up locations (pink stars) in southwest Manhattan and drop-off locations (inverted Black triangles) in middle Manhattan.

their performance. They represent different policies, as described in Section 3.4.4, and, based again on the solution of the optimization problem (only objective function differs), explore the trade-offs among various operational policies.

3.5.3 *Exploring the simulation framework's parametrization space*

The summary of simulated scenarios and different parameters based on fixed objective function presented in equation (3.11) is provided in Table 3.3. For a better understanding of the table, it is necessary to define some of the parameters:

- **Service rate** represents the acceptance rate of the scenario.
- **Share rate** is defined as the percentage of trips shared at least with another request.
- **Idle time** represents the percentage of time shuttles do not serve any passenger in the entire scenario horizon.

By altering three parameters, namely, (a) fleet size, (b) a combination of t_{delay} and t_{extra}, and (c) the number of neighboring vehicles, we have produced six different scenarios, which are presented in Table 3.3. Moreover, for a more detailed analysis of presented scenarios, we provide the relative frequency distributions of trips based on duration, distance, t_{delay}, and t_{extra}, respectively, in Figure 3.6. By studying these simulation results, we can observe that by increasing sharing, the duration of trips gets longer

Scenarios					
Label	Vehicles	Capacity	t_{delay} [min]	t_{extra} [min]	Neighbors
S1	1000	10	5	15	30
S2	1000	10	5	15	50
S3	1000	10	10	20	30
S4	2000	10	5	15	30
S5	2000	10	5	15	50
S6	2000	10	10	20	30

(a)

TABLE 3.3: Different scenarios (a) parameters and (b) results.

	Outcome						
Scenario	Service Rate %	Mean Waiting [min]	Mean In-Car Delay [min]	Mean Distance [km]	Mean Travel [min]	Share Rate %	Idle time Rate %
S1	80.93	2.17	8.58	4.73	20.29	97.7	4.85
S2	83.69	2.28	8.39	4.69	20.18	97.7	4.69
S3	81.05	3.44	11.15	5.28	22.68	97.9	5.09
S4	97.18	1.45	6.32	3.68	15.66	72.1	17.50
S5	97.23	1.40	6.36	3.67	15.48	70.9	17.20
S6	98.05	1.58	9.37	4.14	17.65	83.0	16.86

(b)

compared to the reported dataset by TLC, which is quite reasonable; when passengers are willing to share their trips, they should also accept a compromise in their trip time. Furthermore, as also expected, in the scenarios with smaller fleet size or larger t_{delay} and t_{extra}, the trips are longer. Trade-offs among these key problem variables could be studied to provide further policy determination insights. Moreover, in Figure 3.6 (c), the percentage of trips regarding trip distance is depicted. The area under each scenario line represents the total km traveled in each scenario, which can serve as a measure for choosing a scenario considering the operational policies.

Figure 3.7 presents the occupancy of vehicles for different scenarios (capacity is 10 in all experiments) and clearly demonstrates the sharing trend. It is interesting that more trips happen with 1–6 passengers and there are much fewer for 7–10; presumably, there are operational constraints that do not allow for full utilization of spare capacity. Obviously, in most studied scenarios, the binding constraints that provide the best solutions are the assumed windows for delay tolerance (t_{delay} and t_{extra}); by increasing these, one could achieve higher capacity utilization. Finally, Figure 3.8 presents the capacity distribution for all vehicles over time, for the 6 aforementioned scenarios. Note that in all presented figures, TLC corresponds to the reported trips as provided by NYC Taxi and Limousine Commission, 2019.

Comparing S1 and S3 we can conclude that an increase in the tolerance times for pick-up and drop-off, t_{delay}, t_{extra}, respectively, can contribute to a slight increase in the service rate. A better understanding of the effect of this parameter can be provided in Figure 3.7, where the relative frequency of trips with higher vehicle occupancy has increased in S3 when compared to S1. It shows that longer tolerance time for waiting time and in-car delay contribute to an increase in the number of shared trips. The same conclusion

can be drawn by comparing S4 and S6. Moreover, we can conclude that in scenarios with a 2000 fleet size, the impact of larger tolerance times is more apparent on the service rate, in comparison to the impact of increasing the search area.

Our experiments show that implementing high-capacity ridesharing with a vehicle fleet smaller than 15% of the current number of active taxis in Manhattan, can accommodate 80% to 98% of the requests, with different mean waiting times and in-car delays. By altering the fleet size from 1000 to 2000, we notice a significant increase in service rate, from 80% in S1 to 97% in S4. Consequently, the average waiting time and in-car delay are decreased. By looking at Figure 3.6 we can see that in all scenarios with the fleet size 2000, the trips are shorter.

In the period of high demand, high vehicle occupancy is observed; in Figure 3.7 we present the distribution of generated trips regarding the occupancy over simulation time. Lower fleet size and longer waiting/in-car delays increase the possibility of ridesharing. In scenario S3 we observe that for a fleet of 1000, more than 20% of the trips have 5 to 10 passengers. Furthermore, in Figure 3.8 we observe that over 30% of the fleet in S1, S2, and S3 has an occupancy of over 4 passengers in the peak time of demand. Furthermore, by comparing the share rates, we notice that the increase in fleet size contributes to fewer shared trips, i.e., from 97% in S1 to 72% in S4.

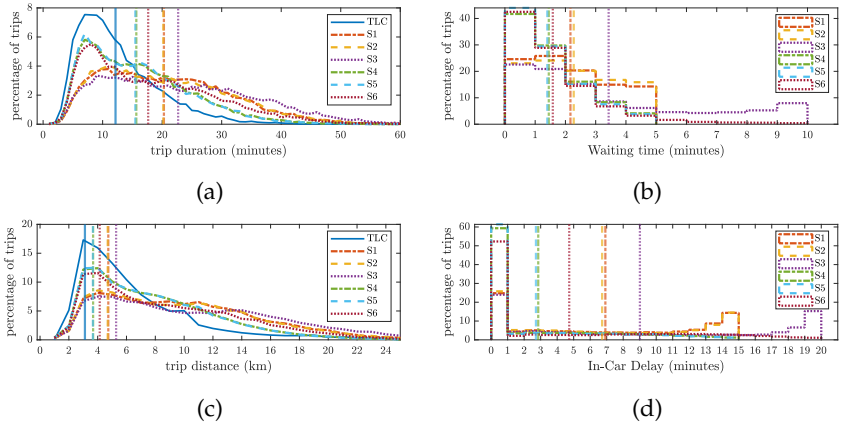


FIGURE 3.6: Distribution of generated trips regarding (a) duration (min) (b) waiting time (min) (c) distance (km) (d) in-car delay (min) in different scenarios. Mean values are illustrated as vertical lines.

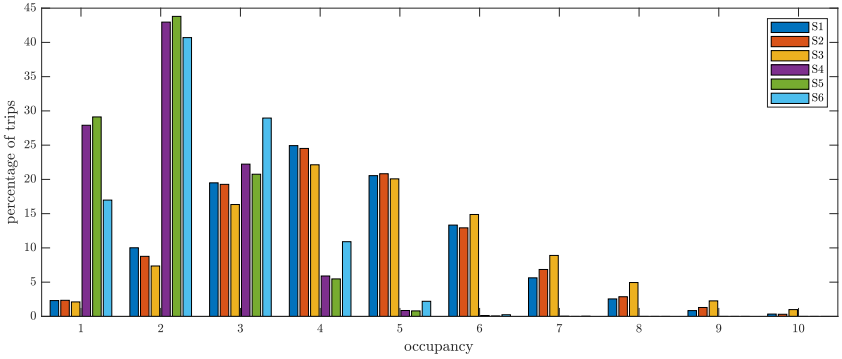


FIGURE 3.7: Distribution of generated trips regarding occupancy in different scenarios.

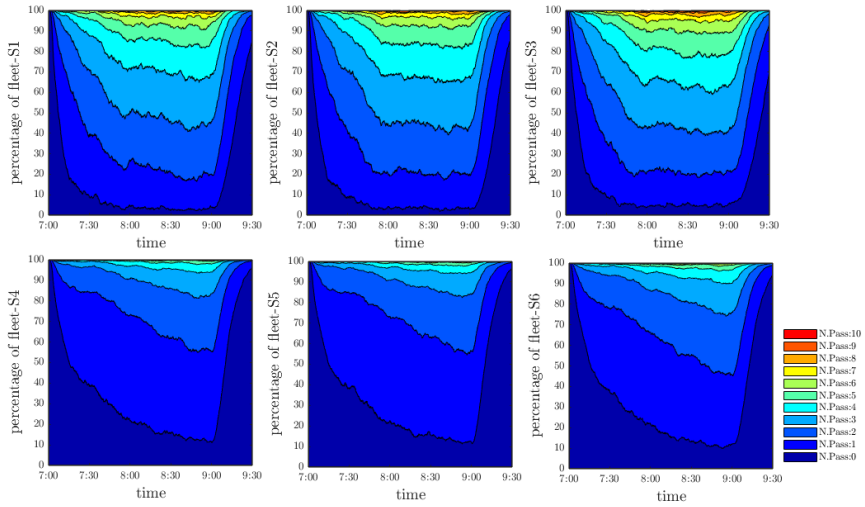


FIGURE 3.8: Occupancy rates over time for all studied scenarios.

Finally, the distribution of trips regarding the occupancy rates, presented in Figure 3.7, shows that the frequency of trips with an occupancy higher than 5 is almost equal to zero for scenarios with 2000 vehicles fleet size. In addition, the idle time rates of the fleet are also increased.

3.5.4 *Exploring the trade-offs among different operational policies*

In this section, we focus on the obtained results based on the different policies introduced in Section 3.4.4; note that simulation parameters for all policies are the same as scenario S1 in the previous section; i.e., the number of vehicles is set to 1000, t_{delay} and t_{extra} are considered 5 and 15 minutes, respectively, and 30 neighboring vehicles are considered when searching for the best possible shuttle to accommodate a trip.

Policy	Service Rate %	Mean Waiting [min]	Mean In-Car Delay [min]	Mean Distance [km]	Mean Travel [min]	Share Rate %	Idle time Rate %
A	65.45	2.69	9.20	5.26	22.20	98.5	19.42
B	67.33	2.57	8.42	4.86	20.85	97.0	14.67
C	78.12	2.12	6.92	4.31	18.73	94.1	4.54
D	63.13	2.61	9.46	5.24	22.40	97.9	17.07
E	73.55	2.45	7.85	4.81	20.65	97.4	6.59
Z	80.93	2.17	8.58	4.73	20.20	97.7	4.85

TABLE 3.4: Different policies results

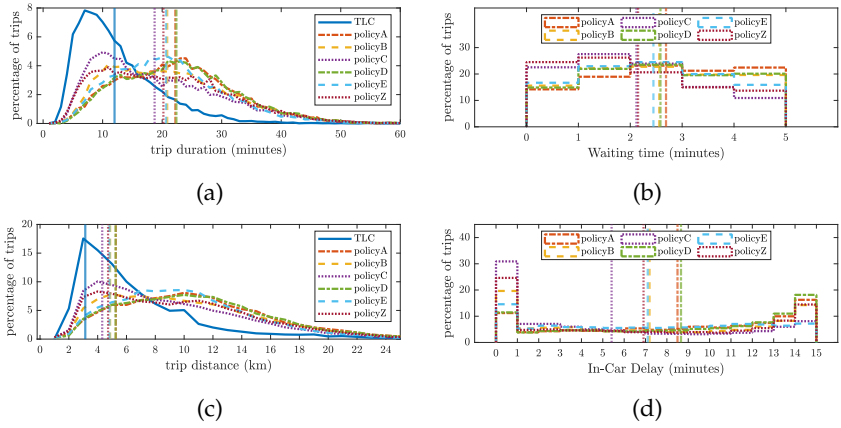


FIGURE 3.9: Distribution of generated trips regarding (a) duration (min) (b) waiting time (min) (c) distance (km) (d) in-car delay (min) in different policies. Mean values are illustrated as vertical lines.

The summary of results for different policies is presented in Table 3.4. Policy Z is based on the objective function presented in equation (3.11); minor differences between the results of policy Z and scenario S1 stem from the random seed used in our experiments, which is primarily used for initializing the shuttles at the beginning of the simulation and allows for some degree of stochasticity. Moreover, similar to the previous section, we provide for these experiments the relative frequency distributions of simulated trips based on duration, distance, and tolerance times t_{delay} and t_{extra} in Figure 3.9. Additionally, Figure 3.9 presents the proportion of trips categorized by their distance in percentage. The total distance traveled in each scenario can be determined by calculating the area under the respective line, providing a metric for selecting a scenario based on operational policies. Accordingly, Figure 3.10 presents the distributions of occupancy for all studied scenarios and how this is affected by the different operational policies. It is clear that different objectives induced by the service provider in the optimization process can shift the operational level of service (and costs accordingly). When we compare the distribution of Figure 3.10 to the one of Figure 3.7, it becomes apparent that the assumed policies of this section have increased the number of shared trips. For exactly the same requests, and passengers' willingness to share, the nominal capacity of the fleet is utilized better, leading to solutions that explore the system's capacity

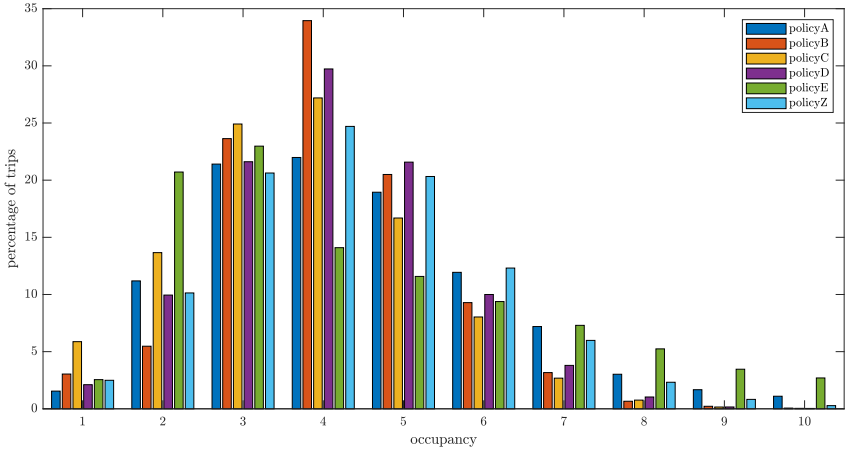


FIGURE 3.10: Distribution of generated trips regarding occupancy in different policies.

space in an improved way. This is also illustrated in Figure 3.11, where fleet capacities are depicted over the simulation horizon; the differences with Figure 3.8 from the previous section are again considerable. One could attempt to investigate solutions with dynamic fleet sizes (i.e., solving the dynamic dispatching problem) together with optimization of operations. It is obvious that there is more flexibility for optimizing real-time operations when both problems are coupled together.

Finally, when we compare the different objective functions studied in this section, some interesting conclusions can be drawn. Among all policies, the ones focusing on minimizing total delay, namely policies C and Z, have achieved the highest service rates and consequently lowest idle time rates. On the other hand, policies B and E, which have conflicting objectives (system reliability and multiple weighted objectives, respectively) appear to decrease service rates and increase idle times; nevertheless, sharing rates always remain significantly high. To conclude, the service operator could define different (policy-oriented) trade-offs among a variety of objectives, and then solve the online optimal operations problem.

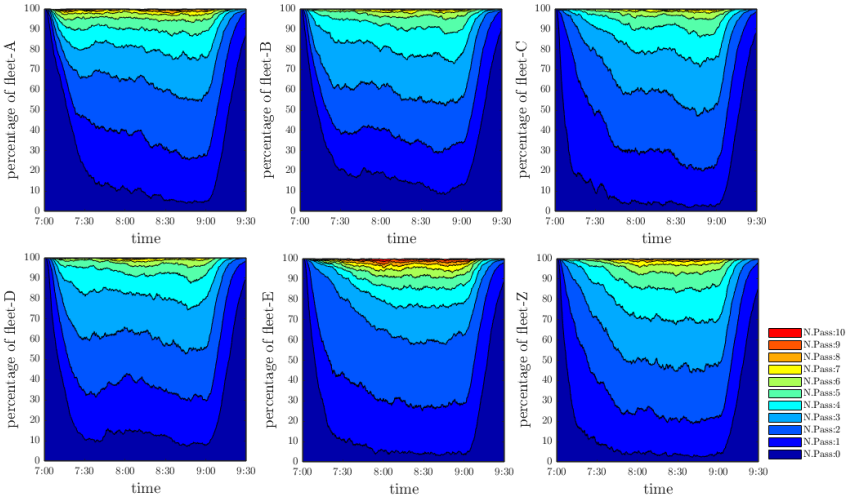


FIGURE 3.11: Percentage of fleet regarding occupancy in different policies.

Label	Vehicles	Capacity	t_{delay} [min]	t_{extra} [min]	Neighboring Vehicles	Objective function	Mean CPU Time [s]
S1/Policy Z	1000	10	15	15	30	Z	0.19
S2	1000	10	15	15	50	Z	0.31
S3	1000	10	20	20	30	Z	0.14
S4	2000	10	15	15	30	Z	0.23
S5	2000	10	15	15	50	Z	0.23
S6	2000	10	20	20	30	Z	0.15
Policy A	1000	10	15	15	30	A	0.26
Policy B	1000	10	15	15	30	B	0.18
Policy C	1000	10	15	15	30	C	0.21
Policy D	1000	10	15	15	30	D	0.23
Policy E	1000	10	15	15	30	E	0.21

TABLE 3.5: Mean computational time for one request in a 16 core 3.2GHz PC.

3.5.5 *Computational complexity*

One of the contributions of this work is to manage to solve the online optimization problem in a reasonable time (i.e., some seconds), making it feasible and applicable for real-time operations. In that respect, we have proposed two approximations of the original full optimization problem, (a) we consider only a spatial subset of the fleet by defining a radius from the request coordinates and considering only the closest shuttles, and (b) we have applied a sequential search over the possible shuttles (instead of considering all combinations simultaneously) to accommodate each request; this reduces the search complexity from exponential to linear. The resulting integer optimization problem can be solved in a few seconds, depending also on hardware specifications and coding architecture.

In the presented simulation experiments, the number of neighboring vehicles can be considered as implementing a (spatial) heuristic to specify the search area for the solver. As we see in Table 3.3 by increasing the number of neighboring vehicles from 30 in S1 to 50 in S2 there is a slight increase in service rate. However, it is worth mentioning that this increase in service rate, achieved by searching a larger number of vehicles to accommodate a trip, requires higher computational time for the solver; some statistics about this trade-off can be seen in Table 3.5. The last column of the table presents the average computational time for one iteration of our algorithm for all studied scenarios; this average value has been computed

by all the replications that have generated the results of previous sections. Note also, that the impact of the search area (or the number of shuttles) is more significant in smaller fleet sizes, and this is related to the running time of the simulator utilized here. Finally, by comparing S4 and S5 the service rate has not increased significantly by increasing the search area.

3.6 CONCLUSIONS

In this chapter, we have introduced a simulator with real-time performance assessment for the method for assigning passenger trip requests to a fleet of vehicles. The modular simulation framework provides a flexible and scalable solution to experiment with a different set of parameters for the various inputs and system parameters to achieve the desired goals. For instance, making heterogeneous combinations of fleet capacities based on demand at different times of the day in the network, or defining flexible tolerance times for each of the requests as desired by passengers. On top of that, by altering the operator's objective function we can build insights to model stakeholders' multiple objectives. The core of this work is the formulation of an online ridesharing problem to match the system's demand and supply; this is modeled as a dynamic deterministic ridesharing problem with tolerance times; moreover, we have explored the ability to handle different combinations of inputs simultaneously.

We have assessed the performance of our method against a real data set and quantified experimentally the trade-offs between fleet size, total travel time, distance, waiting time, and passengers' delay. Furthermore, by implementing dynamic link travel times, we have replicated a more realistic situation about the changing congestion patterns in the network. Moreover, by experimenting with different objective functions, we show the effect of different operational policies on the performance of the provided service. We show that 97% of taxi rides in Manhattan, which are now provided by nearly 13,000 taxis, could be accommodated by just 2,000 taxis with a capacity of up to 5 passengers each, with an average pick-up waiting time of fewer than 2 minutes and an average in-car delay of fewer than 7 minutes; this is derived by using the assumptions and assessment presented in scenario S4. These results demonstrate a significant potential for improving the traffic situation in the network by having fewer cruising taxis.

In times of high demand, high vehicle occupancy is observed, and lower fleet size and longer waiting time/in-car delay increase the possibility of ridesharing. High-capacity ridesharing provides an accepted service

rate of 80% to 98%. However, the fleet size plays an important role in reducing the distance traveled by each vehicle. Finally, our findings show that capacitated ridesharing services could significantly enhance urban transportation systems, especially when optimized in real-time based on dynamic data and feedback. It should be noted that system parameters such as fleet size and pre-specified tolerance times, as well as well-defined service requirements depending on demand patterns, have a crucial impact on the quality of the service. Future work could deal with some of these important aspects.

4

SHORT-TERM PASSENGER DEMAND PREDICTION FOR ON-DEMAND TRANSPORTATION

The chapter is based on the following publication:

- Ghandeharioun, Z., Zendehdel Nobari, P., & Wu, W., (2023). "Exploring deep learning approaches for short-term passenger demand prediction". *Data Science for Transportation, 5, 19.*

4.1 INTRODUCTION

On-demand transportation services can now be provided in a unique and popular way due to the rise of online apps and platforms for car-hailing services. Nearly every transportation service advertises that they are committed to attracting more customers. Hailing a taxi on the street is getting less popular as people would rather use taxi service platforms such as Uber, DiDi, and Lyft, which are gaining popularity at a rapid rate, to request a pickup on their smartphones. This is because these platforms offer more customization options.

Several of the companies that provide on-demand transportation are losing money due to an imbalance between supply and demand. The cause of the imbalance between supply and demand is, on the one hand, long wait times for passengers who are located in an area where there are no available taxis, and, on the other hand, idle taxis and drivers who are lingering around to find customers. Both of these factors contribute to the problem. This leads to a loss of income not only for taxi firms but also for individual drivers, as well as a waste of time for passengers who are waiting.

The forecasting of short-term taxi demand is an essential component of the solution to the problem of an imbalance between supply and demand for taxi companies. If taxi businesses were to have accurate demand predictions in advance, they would be able to pre-allocate taxi fleets from areas of oversupply to regions of excess demand in order to fulfill the demand of passengers and enhance the performance of their services. Both transportation operators and passengers would benefit from proactive decision

assistance provided by predictive data analytics. Therefore the subject of demand forecasting has constantly been under study by many researchers over the last several decades, and a large number of solutions have been proposed. These solutions range from model-based time series analysis through machine learning approaches and model-free deep learning models.

The conventional method of analyzing time series makes use of statistical models, which may provide accurate projections of future values based on the recurrence of historical patterns. The most well-known ones are, Bayesian Forecasting, the autoregressive integrated moving average (ARIMA) model and the Kalman filter (Guo et al., 2014). These techniques have been used in a wide variety of more complex statistical models, including applications in the prediction of traffic flow (Vlahogianni et al., 2004; Williams & Hoel, 2003); for example. These methods are dependent on the particular mathematical assumption about the input data, which limits the application of these methods. Moreover, the mathematical assumptions constrain coping with the complex properties of the data collected from a wide variety of sources.

Data-driven methods try to tackle the same problem of imbalance between demand and supply by considering the different data sources. These data can be categorized by three aspects, called dependencies according to Zhang et al., 2017.

Temporal dependencies: passenger demand has a significant periodicity (for example, it is predicted to be high during morning and evening peaks and low during sleeping hours), and short-term demand is reliant on the trend of the closest previous demand.

Spatial dependencies: passenger demand in one zone was endogenously reliant on all zonal variables in the network (Yang et al., 2010). An enhanced model that can capture local spatial dependencies is needed since adjacent factors affect more than distant variables.

Exogenous dependencies: weather conditions, points of interest (POI), and travel time rates may strongly affect short-term passenger demand. Exogenous variables have temporal and spatial interdependence.

In this work, we tackle the prediction of the demand for on-demand services primarily by deep learning approaches, which are among the most recent methods applied in the literature (see, Ke et al., 2021; Vlahogianni et al., 2004). In particular, we conduct a cross-comparison between models and datasets. We investigate the effect of temporal aggregation and consideration of spatial and exogenous dependencies. The contribution of our work is as follows:

- This chapter studies short-term demand prediction for on-demand services.
- The short-term demand prediction model studies different data aggregation levels of the input data and shows how this affects the prediction results.
- The current study includes independent and dependent temporal and spatiotemporal variables, considering the characteristics of demand prediction.
- In the current work, we provide a representation of time, in the form of vector embedding, to automatize the feature engineering process and model time better.
- The method presented in this chapter is compared with classical machine learning methods to show the performance of the algorithm.

The rest of the chapter is organized as follows. In Section 4.2, we review the most relevant literature on the short-term demand prediction for on-demand services in terms of motivation and methods. In the subsequent section, we state the problem at hand, and in Section 4.4 we explain all the models studied in the current work. The datasets and their features are presented in Section 4.5. Finally, the results and conclusions are provided in Section 4.6 and Section 4.7 respectively.

4.2 LITERATURE REVIEW

We investigate two aspects in the literature that underpin our work: (1) Motivations for accurate short-term demand prediction for on-demand services (2) Machine learning methods for short-term ride-hailing demand prediction.

4.2.1 *Motivations for accurate short-term demand prediction for on-demand services*

The placement of idle cars to predict future demand and operating states is crucial for the operation of on-demand services like taxis, dynamic ridesharing, or vehicle sharing (Sayarshad & Chow, 2017). Demand prediction for optimizing the operation of on-demand mobility systems is studied in different approaches (Zardini et al., 2021). Fleet operators rely on estimates

of upcoming user requests to efficiently place empty vehicles and manage their fleets of vehicles (Dandl et al., 2019).

Yang et al., 2002; Yang et al., 2010; Yang and Yang, 2011 developed a meeting function to define the search frictions between drivers of unoccupied taxis and waiting passengers in light of the fact that they cannot be matched concurrently in a certain zone. The meeting function made it clear that the density of waiting passengers and available taxis in a given zone at a given time determined the meeting rate, indicating that the waiting time for passengers, the searching time for drivers, and the arrival rate of passengers (demand) were all endogenously correlated. When the arrival rate of empty cabs perfectly matched the arrival rate of waiting passengers, the equilibrium condition was attained. The exogenous factors, such as the number of taxis in the fleet and the fare per trip, had an impact on this equilibrium state as well as the endogenous variables. The taxi operator may use the on-demand service platform to coordinate supply and demand, therefore influencing the equilibrium state by controlling the entrance of taxis and setting the taxi price structure, such as non-linear pricing (Yang et al., 2010). However, researchers discovered that when there was an excess of empty taxis or waiting passengers in that location, a regional disequilibrium would arise (Moreira-Matias et al., 2013). Due to this imbalance, resources may not meet supply and demand, resulting in poor taxi usage in certain areas and low taxi availability in others. The taxi operator must thus prioritize developing a short-term passenger demand forecasting model that can be used to perform effective taxi dispatching and expedite route finding to reach equilibrium across metropolitan regions (Zhang et al., 2017).

Additionally, when demand is unknown, drivers frequently behave extremely differently. For instance, if parking is available, drivers might simply wait in one spot. Alternatively, one could wander the streets. Due to increasing traffic congestion and the diverting of passengers from public transportation, they are likely to have a negative impact on the environment (Zhang & Zhang, 2018). However, some drivers go in advance to possible passenger pickup sites based on past knowledge of demand patterns. Improved operations using algorithms targeted at enhancing empty vehicle routing and repositioning for both taxi and ride-sourcing systems is a vast area of research (Yu et al., 2019; Chen et al., 2021).

There are many other applications for more precise forecasting. A more precise dynamic surge pricing setting is made possible by knowing which areas will likely have higher demand in the upcoming timestep (Iglesias

et al., 2017; Chen et al., 2016). When demand is strong in a particular region, and at a particular time, ride-hailing businesses may use dynamic pricing for a variety of reasons, such as weather-related events, special occasions, and so forth.

Previous studies have shown that the ride-hailing compensation model is problematic for drivers because there is a risk that they won't have a job due to demand uncertainty, which results in lost wages for drivers as they wait for new passengers and are unsure of where demand may be for high-yielding rides that could potentially provide them with a source of income (Chen et al., 2021; Li et al., 2019; Zoepf et al., 2018).

Surge pricing has a detrimental impact on passenger demand as well (Chen et al., 2015). The issues that can arise in on-demand services include cancellation by the passenger due to a long wait before being assigned to a vehicle, cancellation by the passenger due to a longer-than-expected pick-up time after being assigned a vehicle, and passenger reordering and rebooking after cancellation, despite the fact that passenger behavior in on-demand services has not been as thoroughly studied as driver behavior (Wang & Yang, 2019; Chen et al., 2021).

4.2.2 *Machine learning methods for short-term ride-hailing demand prediction*

Short-term prediction of transport demand is a topic, which attracts many researchers. Vlahogianni et al., 2004, in their study of the literature on short-term traffic forecasting, noted that due to the rapid advancements in data accessibility and computing capacity, researchers were switching from traditional statistical models to neural network-based methodologies. Deep learning in particular has been widely used in transportation state prediction (Ke et al., 2021). And new opportunities arises with advancements in deep learning techniques to address short-term transport behavior. Deep learning often involves the training of convolutional neural networks (CNNs) which are capable of capturing high-order spatial-temporal correlations in transportation prediction problems. There is a broad range of problems in the domain of transportation, which are similar to short-term passenger demand forecasting. Researchers have used CNNs for a variety of prediction tasks, such as speed evaluation (Ma et al., 2015), bike usage prediction (Zhang et al., 2016), and demand-supply prediction for ride-hailing services (Ke et al., 2017). To analyze time series data, a number of deep learning models have been proposed, and they have demonstrated cutting-edge performance in practical applications. For instance, Huang

et al., 2014 introduced a multi-task learning structure to perform road traffic flow prediction and provided a deep belief network to detect the spatiotemporal properties. Cheng et al., 2016 proposed a DL-based approach to forecasting day-to-day travel demand variations in a large-scale traffic network. Similarly, Lv et al., 2015 used a stacked autoencoder model based on the traffic prediction method. Ma et al., 2015 extended the deep learning theory for the large-scale traffic network analysis, and predicted the evolution of traffic congestion with the help of taxi GPS data. Recurrent neural networks (RNNs) and their extensions such as long short-term memory (LSTM) are well fit for processing time series data streams. Xu et al., 2018 applied LSTM to predict taxi demand in New York City. Some researchers integrated RNNs with CNNs to make full use of spatial-temporal information to forecast short-term ride-hailing demand (Ke et al., 2017). Tan et al., 2016 studied different pre-training approaches of DNN for traffic prediction.

Extensions to the integrated deep learning algorithms have been made, drawing on but not limited to the CNN and RNN mechanisms. To forecast the demand for taxis, Li et al., 2019 created a contextualized spatial-temporal network that includes local spatial context, temporal evolution context, and global correlation context. For the purpose of forecasting demand for ride-hailing services, Geng et al., 2019 put out a spatial-temporal MCG (STMCG) model that makes use of non-euclidean correlations. Zhou et al., 2018 created an attention-based deep neural network to estimate multi-step passenger demand for bikes and cabs based on an encoding-decoding structure between CNNs and ConvLSTMs. An LSTM model is used to simulate the temporal features as well as other traffic-related data in Yang et al., 2019's hybrid deep learning architecture.

4.3 RESEARCH PROBLEM

The goal of the short-term demand forecast is to predict the number of on-demand taxi ride requests that will be needed at some point in the future in a certain unit area by taking into account the requests that have been placed in the past. In order to forecast the demand for on-demand services, in addition to using data from past demand, we also consider temporal and spatiotemporal features. Short-term passenger demand depends on additional explanatory factors in addition to its own spatiotemporal characteristics (some with spatiotemporal properties and some only with temporal properties). Generally, the problem can be seen as a time series

forecasting problem. There are several time-series prediction algorithms. While many studies focus solely on the location and time of the demand, we aimed to incorporate all publicly available features in our methodology to explore their impact on the prediction. To the best of our knowledge, no other studies have examined such a wide range of features and their influence on the prediction.

The inclusion of all these features necessitates working at a certain level of data aggregation provided by the publicly available feature data source. In contrast to earlier approaches which mostly use temporal networks, we use raw counts of ride-hailing pickups along with a variety of temporal and spatial features (such as socioeconomic variables, spatial heterogeneity, weather, point of interest, etc.) that are used in a multivariate architecture for the effective short-time demand prediction of ride-hailing services. Moreover, we propose a representation of time in the form of vector embedding so that the feature engineering process may be automated and time can be modeled more accurately. The vector embedding of time is then combined with machine learning methods. Considering these reasons, we selected methods that could accommodate all the data and features we intended to work with. In the following section, we explain the studied models in more detail.

4.4 MODELLING

Most short-term demand forecast systems in transportation have traditionally relied on running models using univariate trip count data obtained by loop detection, GPS, and other sources (Vlahogianni et al., 2004). The trip count is a continuous and time-dependent variable that constitutes time-series data. Predicting time-series data falls under the regression class of machine learning algorithms.

We devised five models that are currently widely used in the literature for the prediction task: Random Forest (RF), Long Short-term Memory (LSTM), Convolutional Neural Network (CNN), LSTM-CNN autoencoder, and Deep CNN. Furthermore, we combined the vectorization of time (Time2Vec Kazemi et al., 2019) as a layer to stack it with other models and try its power in our case study. In the following, we briefly explain each model.

Random Forest Regressor

Random Forest is an ensemble learning approach, which can undertake classification and regression tasks. A random forest is an ensemble of unrelated decision trees. These multiple decision trees are averaged to build a more robust model with a better generalization performance and less susceptibility to overfitting (Breiman, 2001). It is implemented with sklearn.ensemble.RandomForestRegressor(). Within the random forest regressor, four hyperparameters are tuned: Max depth, which is the maximal length of a path from the decision tree root to the leaf; Max features, which is the maximum number of features that are examined for the splitting of each node within the decision tree; Min samples split, which is the minimum samples limit that is imposed to stop the further splitting of nodes; N estimators, which is the number of trees in the forest. The tuning process is conducted by means of an exhaustive grid search. Finally, model RF-Model 1 is the model with 200 trees; RF-Model 2 is the model with 50 trees; and RF-Model 3 is the tuned forest. The grid search space of the tuned model for the different datasets is presented in Table 4.1. The hyperparameter space was adjusted according to the number of data points within each dataset.

Dataset	**Dataset A**	**Dataset B**	**Dataset C**
Number of fits	750	2500	1080
max-depth	[3,4,5,6,7]	[7,10,12,15,18]	[7,10,12,15,18]
max-features	[3,4,5,6,7]	[3,4,5,6,7]	[6,15,21,28]
min-samples-split	[3,6,12,18,24]	[4,12,24,96]	[20,387,1574]
n-estimators	[50,70,100,200]	[30,50,70,100,200]	[20,30,50,70,100,200]

TABLE 4.1: Grid search space for RF-Model 3 hyperparameter tuning across different datasets

One of the major pros of the random forest model is that it is an interpretable transparent model. Due to this model's attribute, a relative feature importance analysis can be carried out to determine which features have a more pronounced effect on the prediction outcome. Furthermore, it is also possible to extract a random tree from the ensemble and to examine the splitting and the residual errors after each split according to a feature. In any decision tree, the upper levels of the tree usually comprise splits based on more important features. The deeper we run down the tree levels, the more the data's variance is covered by more and more splits of less

important features. The feature importance analysis of our dataset will be presented in Section 4.5.

Long Short-Term Memory Network (LSTM)

Long Short-term Memory Network is a special type of Recurrent Neural Network. An LSTM unit consists of the cell, input, output, and forget gates. The model is implemented with Tensorflow.keras.layers.LSTM(). The relevant hyperparameter is the number of LSTM units. The model takes transformed timestep window tensors as input. The different timestep window sizes for all datasets are shown in Table 4.2. Six different architectures are designed for the task. The schematic diagrams of the architectures are shown in Figure 4.1.

Dataset	Timestep window size	Interpretation	Input shape
A (Temporal 1 hr)	3	3 hours	(3,8)
B (Temporal 15 min)	2	30 mins	(2,9)
C (Spatio-Temporal 15 min)	8[a]	2 hours	(8,34)
C (Deep Architecture)	96[b]	6 hours	(96,34)

TABLE 4.2: Timestep window setting for data transformation

a see Figure 4.15
b see subsection 4.4 Deep CNN

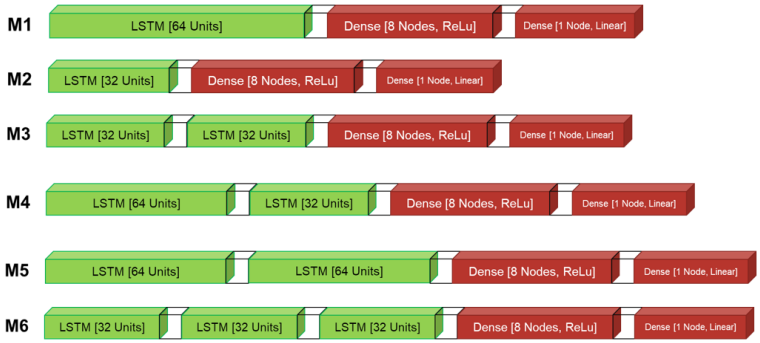


FIGURE 4.1: Different LSTM architectures with various hyperparameter settings

Models 1 and 2 share the same architecture. In the first layer, successive LSTM units learn the demand patterns which are then passed to a neural layer with 8 nodes which interprets the features further and then feeds it to the output layer with one node for the single output variable. Models 3,4 and 5 share the same two-layer stacked LSTM architecture. The difference between M3, M4, and M5 is the hyperparameter setting. Finally, there is M6 which has three stacked LSTM layers and is the deepest of all architectures.

Time2Vec For a variety of challenges involving sequence modeling, recurrent neural networks (RNNs) have shown outstanding results. The majority of RNN models assume that inputs are synchronous and do not include time as a characteristic. Since time is recognized as a crucial component, it is often included as yet another input dimension. In actual application, RNNs often fall short of adequately using time as a feature. Many researchers create hand-crafted time features tailored to their particular problems and input those characteristics into the RNN to aid in better use of time. But, hand-crafting features can be costly and demands subject-matter knowledge.

In the current work, we implement Time2Vec as explained in Kazemi et al., 2019 and adopt their solution in developing Neural Network with the LSTM model explained above. Their goal is to provide a vector embedding for the representation of time. Mathematically, the implementation of Time2Vec is as follows:

$$t2v\,(\tau)[i] = \begin{cases} \omega_i \tau + \phi_i, & \text{if } i = 0. \\ \mathcal{F}(\omega_i \tau + \phi_i), & \text{if } 1 \leq i \leq k. \end{cases} \tag{4.1}$$

Where k is the Time2Vec dimension, τ is a raw time series, $\mathcal{F}$ is a periodic activation function, ω, and ϕ are a set of learnable parameters. In order to enable a chosen algorithm to detect periodic behaviors in data, we set $\mathcal{F}$ to be a sin function. In addition, the linear term captures non-periodic patterns in the input that rely on time while also representing the passage of time. Several architectures may simply apply this vector representation of time due to its simplicity. In this instance, by changing a simple Keras dense layer, we attempt to translate this idea into a Neural Network structure. This custom layer's output consists of the user-specified hidden dimension ($1 \leq i \leq k$), which comprises the network's learned sinusoids and a linear representation of the input ($i = 0$). With this instrument in our possession, all we have to do is to stack it with more layers to test its effectiveness in

our case study. We stack the Time2Vec layer on the best-performing LSTM from the architectures shown in Figure 4.1 and show the results in Section 4.6.

Convolutional Neural Network (CNN)

CNN is an artificial neural network that is modeled after the visual cortex. In a convolutional layer, the algorithm carries out a mathematical operation called convolution on a rolling kernel across the input space. Convolutions include different filters which extract feature maps from the input space. CNN is implemented with Tensorflow.keras.layers.Conv1D(). Three relevant hyperparameters are considered: input shape (units take input values in time step format), filters (number of filters in the convolutional layer), and kernel size (length of the one-dimensional convolutional window). The timestep window setting is the same as in LSTM models (see Table 4.2). Two models with different numbers of filters are designed (See Figure 4.2). In this architecture, the CNN layer extracts feature maps from the

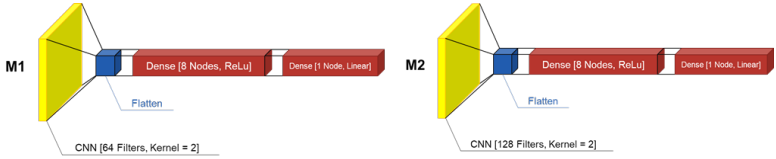


FIGURE 4.2: CNN architecture with various hyperparameter settings

transformed input tensor, and then a flattening layer feeds the outputs of the CNN layer to a fully connected layer to articulate the features. Finally, same as before, one node outputs the target variable.

LSTM-CNN Autoencoder

This model is a hybrid of CNN and LSTM models, using a CNN layer as an encoder and an LSTM layer as a decoder. Implementing methods include four-layer functions from Tensorflow.keras.layers, including LSTM(), Conv1D(), and Maxpooling1D(). Max pooling is used to reduce the dimensionality of the feature map so higher-order patterns can be extracted. There are five relevant hyperparameters: input shape (units take input values in tuples of time step format), filters (number of filters in the convolutional layer), kernel size (length of the 1D convolutional window), pool size

(size of the 1D max pooling window), and units (number of LSTM units). The same time step windows in Table 4.2 were used. The Four different architectures are shown in Figure 4.3 .

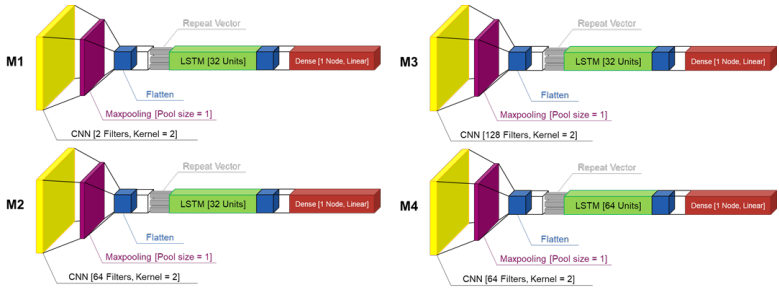


FIGURE 4.3: LSTM-CNN autoencoder architecture with various hyperparameter settings

The models share the same architecture, and their differences lie in their varying hyperparameter settings. The repeat vector creates LSTM unit readable input tensors from flattened data.

Deep CNN

In deep CNN architectures, convolutional layers are stacked using max pooling layers. This is realized by four functions from Tensorflow.keras.layers: Conv1D(), MaxPooling1D(), BatchNormalization(), and Dropout(). Compared with CNN, there are two additional hyperparameters. Pool size is the size of the 1D max pooling window, while dropout rate is the dropping rate of units during training. Table 4.2 shows the time step window configuration, which is 96 instances. The window size is set large enough to enable a multi-layer stacked architecture. Successive convolutional and max pooling layers with kernel sizes and pool sizes larger than 1 rapidly reduce the input shape dimensionality, which is why a larger window size is warranted. As a result of these stacked layers, the model should theoretically be able to extract more far-reaching patterns. Figure 4.4 shows the six different models. Models M1 and M2 share the same architecture and have different hyperparameter settings. They include three convolutional layers followed by three max pooling layers. The resulting feature maps are flattened and are connected to a one-node output layer. Model 2 has more convolutional layers compared to Model 1. Model 3 has another fully connected layer with 32 nodes which is close to the independent feature count

of 28. Model 4 uses an additional layer of batch normalization following each convolutional layer which standardizes the layer inputs. Models 5 and 6 make use of a 20% dropout in the second and second to third max-pooling layers, respectively. Dropout acts like a mask that randomly nullifies the contribution of some neurons randomly during training, creating a large number of neural networks with different architectures in parallel. This is an effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all types.

FIGURE 4.4: Different deep CNN architectures with various hyperparameter settings

4.5 DATA

In this project, we build a multivariate time-series dataset using a variety of variables provided by transport operations and travel behavior study fields to be useful to demand prediction. These criteria include temporal, meteorological, socioeconomic, and demographic factors. As a result, the data is grouped into three categories and comes from four separate sources.

First, the trip data includes green taxi trip data. The data were collected from the NYC Taxi and Limousine Commission, 2019 and are available to the public. Furthermore, in order to capture interannual demand patterns for green taxis we use 2 years of green taxi pickup data from 2017 to 2019. This decision stems from two reasons: firstly, we decided to set the study

period cutoff date before 2020 to avoid the impact of COVID-19 and its lockdowns on travel behavior; secondly, from 2017 onwards the trip records pickup and drop-off locations are formatted according to the NYC TLC's own taxi zones (see Figure 4.5) as opposed to coordinates in previous records. The coordinates formatted trip records are more tedious to work with and are not as precise due to GPS logging errors as shown in Figure 4.6, where a considerable number of pickup records fall outside the study area. Taxi zones formatted data is more suitable for integrating spatial features and is less prone to errors.

The green taxi data has detailed information about each trip. The most important and relevant variables in these datasets are the location and time of pickup and drop-off since they are the keys to spatial data augmentation and temporal features such as day, hour, weekday, and the like can be extracted from them. The location variable is the code of the taxi zone where pick-up or drop-off happens and the time variable is when respectively the trip starts and ends in minutes.

Second, the socio-demographic features are gathered from the U.S. Census Bureau, 2023. These features are arranged into five different types of features: population, housing unit, social characteristics, economic characteristics, and household characteristics. All features and their respective descriptions are shown in Table 4.4.

Third, climate features are also included. They are provided by the National Oceanic and Atmospheric Administration, 2023. The New York Central Park daily weather data is used to describe meteorological and inclement weather impacts. The dataset has not only information on temperature and precipitation, but also data on heating degree days, cooling degree days, snowfall, and snow depth.[1]

Finally, point-of-interest data is obtained from NYC Open Data from City of New York Department of Information Technology & Telecommunications, 2023 and categorized in six categories explained in Table 4.4

1 Degree day is a quantitative index demonstrated to reflect demand for energy to heat or cool houses and businesses. This index is derived from daily temperature observations at nearly 200 major weather stations in the contiguous United States. Heating degree days are summations of negative differences between the mean daily temperature and the 65°F base; cooling degree days are summations of positive differences from the same base (National Oceanic and Atmospheric Administration, 2023).

FIGURE 4.5: Taxi zones in New York City according to NYC Taxi and Limousine Commission, 2019

4.5.1 *Data aggregation and exploratory data analysis*

To carry out the predictions, the trip data are aggregated in both temporal and spatial bins. The rationale behind this aggregation is two-fold: firstly, ride-hailing service providers usually look for demand forecasts in a given region for the next 15 minutes or 1 hour to dispatch vehicles, and secondly, aggregation results in fewer data instances and less model training time as a result. In order to account for these needs and limitations, both temporal aggregation and spatial aggregation are employed. To investigate the effects of temporal aggregation on prediction accuracy, we apply two temporal aggregation schemes: a 15-minute aggregation and a 1-hour aggregation scheme.

The trip counts resulting from different temporal aggregation schemes are shown in Figure 4.7. In the first week of 2017, clear patterns of daily traffic peaks are discernible in both 15-minute and 1-hour aggregation. A morning peak and an evening peak can be observed nearly every day. In the first month of 2017, a regular pattern across weeks is shown in both 1-hour and 15-minute aggregated data in the middle row of Figure 4.7. Finally, the trend for 2017 shows that the demand for green taxis is higher in the

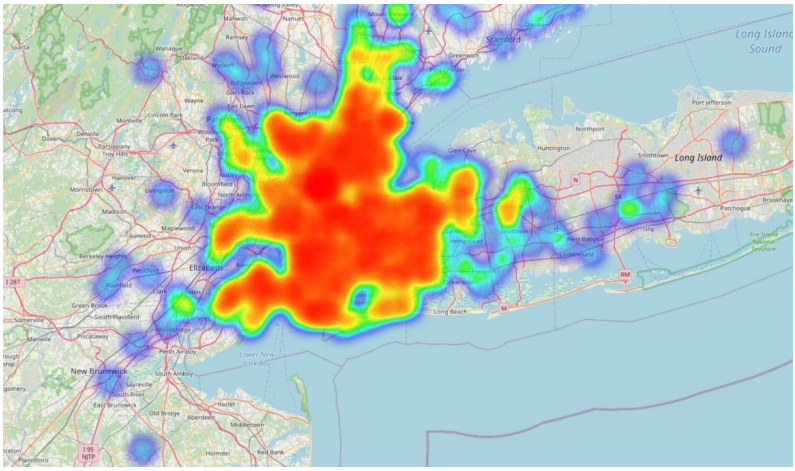


FIGURE 4.6: The spatial distribution of pickup coordinates from green taxi trip records

colder months and decreases with warmer weather, which further indicates the seasonality of green taxi demand and underlines the importance of including climate features for prediction purposes.

Spatial aggregation aims to further augment data with spatial features (socio-demographic and climate features). The trip data and the spatial features have different spatial resolutions: trip data is at the taxi zone level, while the spatial features are at the borough level. Eventually, all the trip data are aggregated to the borough level so they can be joined with spatial features. There are two major reasons why taxi zone level aggregation is not pursued in this project. On the one hand, the available socio-demographic data is not at the taxi zone level. If the taxi zone is set as the target spatial aggregation level, the spatial features will have to be broken down and distributed among taxi zones, and a distribution method must be estimated, which may, in turn, introduce bias to the dataset. On the other hand, a finer aggregation level results in a much larger dataset, raising computational limits. Compared with the 5 boroughs, there are 263 taxi zones in New York City, and a spatial aggregation at the taxi zone level, and considering all the features will lead to an increase in training compute for an LSTM model with 8 input neurons and 100 epochs by approximately 10 orders of magnitude (see Sevilla et al., 2022 for estimating training compute of deep learning models).

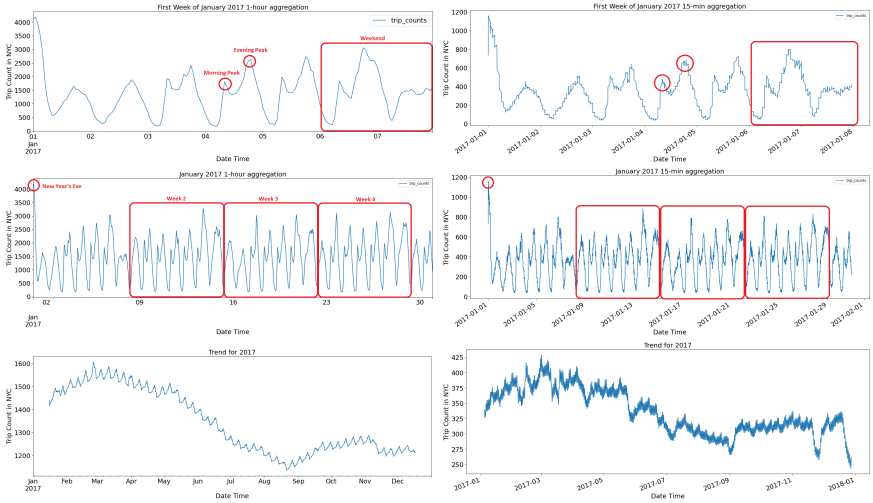


FIGURE 4.7: Effect of temporal aggregation on trip count data (left: 1 hr aggregation, right: 15 min aggregation, top row: trip count for the first week of January 2017, middle row: trip count for January 2017, bottom row: trip count for 2017)

As mentioned before, apart from univariate trip records, socio-demographic, and climate features are also included. Figure 4.8 shows the spatial distribution of two example socio-demographic features: employment rate and mean commute time. One can observe the lower employment rates in less affluent boroughs such as Staten Island and the Bronx. Lower employment rates can result in fewer commute trips and fewer trips in general. Conversely, mean commute times in less affluent boroughs are much higher than in more affluent boroughs like Manhattan. This suggests that more residents from the Bronx and Staten Island go to work in other, more affluent boroughs where there are more jobs and opportunities.

Furthermore, Figure 4.9 shows two example climate features: daily minimum temperature and precipitation. The seasonality of both features is what we come to expect from climate data.

4.5.1.1 *Dataset design*

For the purpose of investigating the effect of temporal aggregation and the inclusion of spatial features on green taxi demand prediction, three datasets were designed for cross-comparison. As seen in Figure 4.10 and

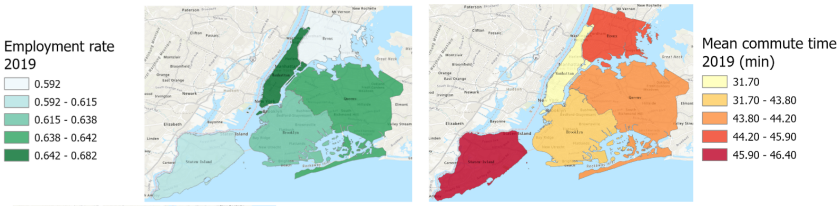


FIGURE 4.8: Spatial distribution of employment rate (left) and mean commute time (right) in New York City in 2019 (NYC Taxi and Limousine Commission, 2019)

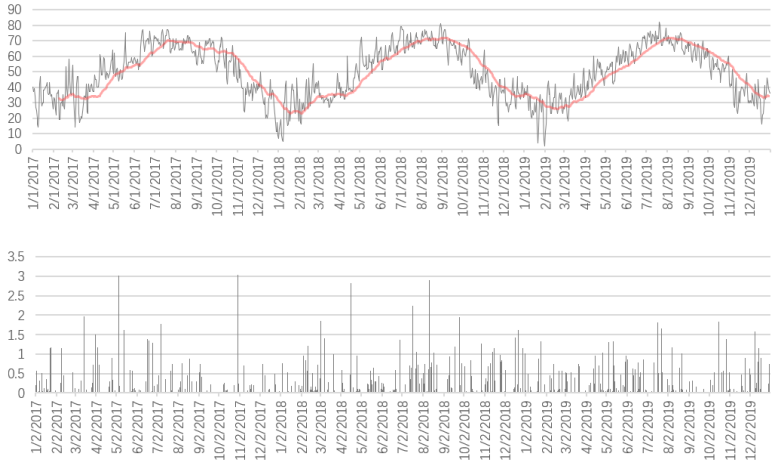


FIGURE 4.9: Daily minimum temperature in Fahrenheit (top) and precipitation in inches (bottom)

Table 4.3, dataset A is comprised of trip records and temporal features, and the data is temporally aggregated in one-hour intervals. Dataset B is designed to have the same trip records and similar temporal features, however, its data is temporally aggregated in 15-minute intervals allowing for more fine-grained predictions in time. Finally, spatial features and climate features are integrated into dataset B to create dataset C, which is a spatiotemporal dataset. It is spatially aggregated at the borough level and temporally aggregated in 15 minutes.

FIGURE 4.10: The three different datasets, dataset A and B, are temporal, and dataset C is a spatiotemporal dataset

Dataset	Temporal aggregation	Spatial features
Dataset A	1 hour	Location and time
Dataset B	15 minutes	Location and time
Dataset C	15 minutes	Location, time, Climate, POI and socio-demographic

TABLE 4.3: Dataset design

4.5.1.2 *Data preparation*

The datasets are cleaned to remove any NA (Not Available) instances. Moreover, the data is scaled to avoid model bias towards features with larger values. To this end, the min-max scaler is employed to scale and standardize continuous data. The intention is to preserve the feature distribution while standardization, which is why data normalization was not used. After scaling, all the features are compressed to fit from 0 to 1, apart from the categorical data coded with one-hot coding. These categorical features include weekend and borough, which inform about whether the trip occurred on the weekend and in which borough the trip started.

4.5.1.3 *Feature selection*

Feature selection is an important step that aims to reduce the computational cost of modeling and improve performance, by reducing the number of input variable dimensionality (Vlahogianni et al., 2004). The objective here is to filter out redundant features that are linearly correlated to other features and are, therefore, not independent variables that add more information to the model. To achieve this, a Pearson correlation matrix is produced for dataset B and dataset C (dataset A has the same features as dataset B), illustrating the degree of linear collinearity between the two features. Collinearity of more than 0.5 indicates that the two features are correlated

and a collinearity of 1 indicates a perfect correlation. Collinearity of -1 indicates a perfect inverse correlation and is also equally unwanted. Figure 4.11 shows the correlation matrices for datasets B and C.

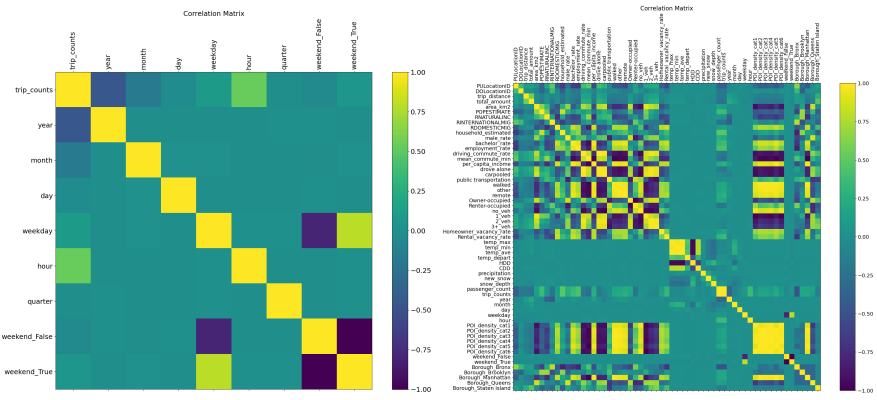


FIGURE 4.11: : Correlation matrices for dataset B with 9 features (left) and dataset C with 60 features (right)

Upon closer examination, it becomes clear that most socio-economic and climate features are correlated and should be filtered. The opted feature selection method is a procedural pairwise collinearity check that progressively filters out the redundant features with the least correlation to the target variable, in this case, trip counts. This procedure is depicted in Figure 4.12 as a flowchart. Initially, we set a correlation threshold of 0.7. In other words, pairs of features with a correlation of more than 0.7 or less than -0.7 are selected for further inspection. After this step, the feature with the least correlation to the target variable is filtered and eliminated. As a result of feature selection, the number of features drops from 60 down to 34 (see Figure 4.12).

It should be noted that adding one more feature was left out not directly due to this filtering procedure but to other reasons. This feature is passenger count with a linear correlation of 0.99 in relation to the target variable trip counts. This would suggest that most green taxi trips have only one passenger on board in New York City, which renders the problem of counting the number of trips within a given time interval almost indistinguishable from that of counting the number of passengers per trip in the same time window. This would mean that including the passenger count feature is tantamount to a tautological endeavor since one would essentially try to predict trip counts with trip counts. Furthermore, the one-hot-coded

spatial borough features are also excluded from the filtration process to allow for the observation of the regional differences in ride-hailing demand contribution.

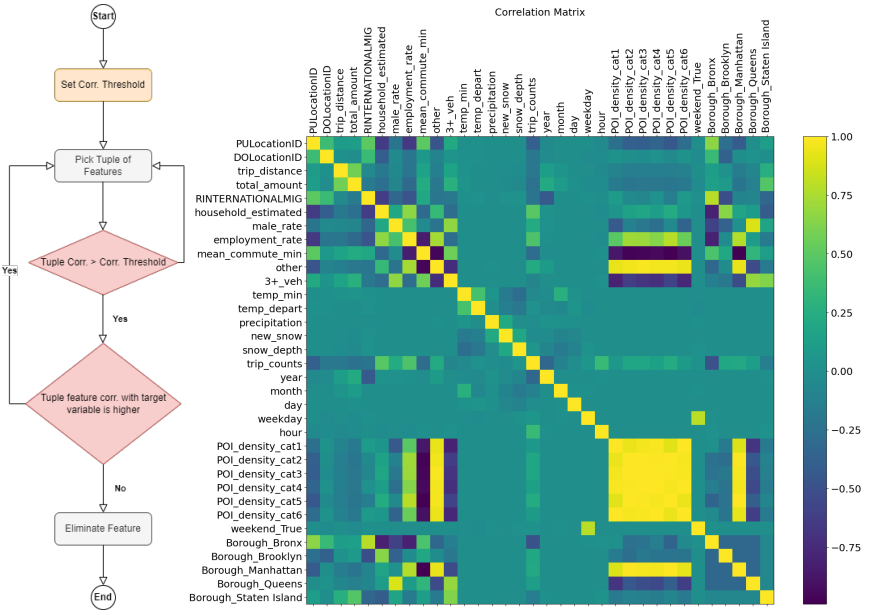


FIGURE 4.12: : Feature selection procedure flow chart (left), filtered dataset C with 34 features (right)

Feature importance: as mentioned in Section 4.4 in **Random Forest Regressor**, one of the advantages of the Random Forest model is to be interpretable and transparent, this attribute can provide a feature importance analysis to identify which features can significantly influence the prediction.

The feature importance analysis based on Random Forest (4.13) on dataset C shows that the most important variables are hours of the day, number of households in the departing borough, and employment rate in the departing borough. Moreover, the result suggests that few trips are from Staten Island, whereas the Bronx has the largest share of trip counts and therefore contributes more to the prediction outcome.

4.5.1.4 *Data transformation and splitting*

In preparing for the final data manipulation step, the data is sorted according to each incident's timestamp. Consequently, the data is split into

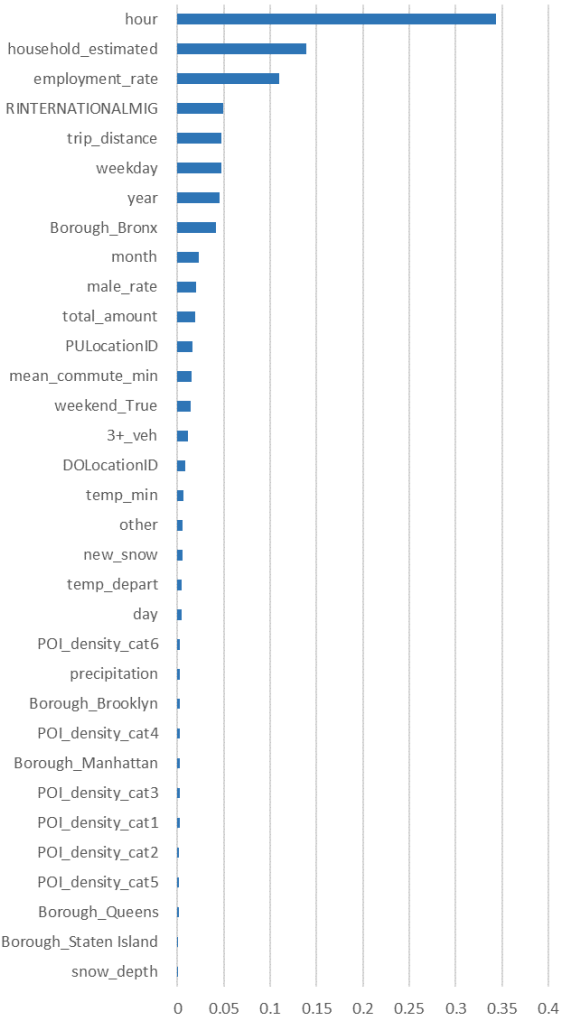


FIGURE 4.13: Random Forest feature importance analysis of dataset C.

training, testing, and validation sets in an 80-10-10% split as a general rule of thumb for machine learning, according to Figure 4.14.

Importantly, since we're dealing with time-dependent data, it is essential that the splitting follows a chronological order so that earlier instances in time are used to predict instances in the future. The training data spans

Independent Variable Name	Variable Description	Variable Type
LocationID	TLC Taxi Zone in which the taximeter was engaged	int
LocationID	TLC Taxi Zone in which the taximeter was disengaged	int
trip_distance	The elapsed trip distance in miles.	float
total_amount	The total amount charged to passengers. Does not include cash tips.	float
RINTERNATIONALMIG	Net international migration rate	float
household_estimated	Annual estimates of the number of housing units	float
male_rate	Male rate among 15-year-olds and over	float
employment_rate	In-labor-force rate among 16-year-olds and over	float
mean_commute_min	Mean travel time to work in minutes	float
other	Mode share of means of transport other than driving, PT and walking	float
3 $\pm$ veh	Households with more than 3 vehicles	float
temp_min	Lowest temperature of the day (F)	float
temp_depart	Daily temperature departure from normal (F)	float
precipitation	Rain intensity (inch)	float
new_snow	Snowfall (inch)	float
snow_depth	Depth of snow (inch)	float
year	The year of trip	int
month	The month of trip	int
day	The day of trip	int
weekday	The day of the week in which the trip took place	int
hour	The hour of trip	int
weekend_true	Whether the trip took place on a weekend	int
POI_Density_Cat1	Residential	float
POI_Density_Cat2	Transportation facility, Government Facility	float
POI_Density_Cat3	Cultural, Recreational, Religious Facility	float
POI_Density_Cat4	Social and Health facility	float
POI_Density_Cat5	Commercial POIs	float
POI_Density_Cat6	Education facility	float
Borough_Bronx	Whether the trip took place in the Bronx	int
Borough_Brooklyn	Whether the trip took place in Brooklyn	int
Borough_Manhattan	Whether the trip took place in Manhattan	int
Borough_Queens	Whether the trip took place in Queens	int
Bourough_Staten_Island	Whether the trip took place in Staten Island	int
Dependant Variable Name	**Variable Description**	**Variable Type**
trip_counts	Number of trips counts within 15 minutes (1 hr in case of dataset A)	float

TABLE 4.4: List of all selected features in dataset C

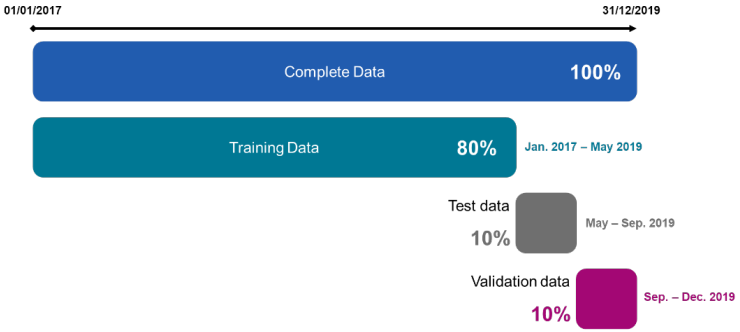


FIGURE 4.14: Dataset Splitting

29 months from January 2017 to May 2019 and the testing and validation data each span 5 and 4 months from May 2019 to September 2019 and from September 2019 to December 2019 respectively. The models fit on the training sets and are optimized with the help of the validation data. Ultimately, the predictive performance of the fitted model on the held-out testing set is assessed. As a further step, some of the machine learning models presented in Section 4.4, require the data to be transformed into a tensor with a certain shape. These models train on rolling timestep windows of a given size and extract certain patterns and features in the timestep window to predict the value immediately after the timestep window. Figure 4.15 shows such a rolling timestep window mechanism with a time-step window size of 8.

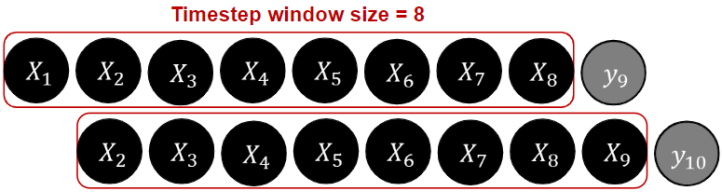


FIGURE 4.15: Data shape transformation and rolling timestep window

In this example, the models are trained on a time window of using 8 instances of independent feature vectors X_1 to X_8 to extract intrinsic patterns and features within the dataset in order to predict the target variable instance number 9 (y_9). In the next iteration, the time window is

rolled over the one-time unit, and independent feature vectors X_2 to X_9 are used to predict target variable instance y_{10} and so forth.

To train the models, we use 100 epochs and a learning rate of 0.0001. An epoch is a forward and backward pass of the dataset through the neural network, which is used to change the weights of the hidden layers, otherwise known as optimization. The learning rate is a tuning hyperparameter that determines the step size of the optimization iteration. The weights of the models are tuned through the minimization of the loss function, which in our case, is the mean squared error since this is a regression problem. Moreover, we use Adam as the optimizing algorithm because it is a stochastic algorithm that is more computationally efficient than gradient descent (Kingma & Ba, 2017). Each model's testing and training root mean square error (RMSE) is used as a comparison metric since it can be interpreted as the accuracy of prediction in terms of the number of trip counts per unit of time.

4.6 RESULTS

The results for all the models and architectures on the testing dataset are reported in Table 4.5. Moreover, the training times for dataset C for all the models are also reported. Root mean square error (RMSE) is the reported metric to compare the models defined as:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(\hat{x}_i - x_i)^2},$$

where x_i is the i^{th} Actual value and $\hat{x}_i$ predicted value of the demand, respectively, and n is the size of the test set.

Our experiment platform is a server with 8 CPU cores (Intel(R) Xeon(R) CPU @ 3.20 GHz), 256 GB RAM, with Python 3.9.16.

One of the main findings is that including spatial features can improve model performance. Compared with the best-performing model of Dataset B (LSTM Model 4), the best one for Dataset C (LSTM Model 4) has a 54.5% lower RMSE. Furthermore, all other models are improved by introducing spatial and climate features (see Table 4.5). Among all the models in our explorations, the best-performing models in each architecture are depicted in Figure 4.16 for all three datasets.

Results in Table 4.5 show that two-layer stacked LSTM architectures (M3-M5) performed better on average across the different datasets, and their performance got better with more data points. Model 4 of LSTM

Model	Architecture	Dataset A Testing	Dataset B Testing	Dataset C Testing	Training time (min)
LSTM	Model 1	69.18	18.77	7.62	215
	Model 2	96.07	19.59	9.38	190
	Model 3	95.02	17.49	7.74	226
	Model 4	90.91	16.31	7.42	217
	Model 5	86.25	18.28	8.13	224
	Model 6	84.37	16.49	8.57	463
LSTM+Time2Vec	Model 4+Time2Vec	77.27	13.86	6.80	220
CNN	Model 1	76.12	18.81	8.52	42
	Model 2	124.73	18.75	8.55	53
Random Forest	Model 1	198.23	51.87	16.43	15
	Model 2	198.00	52.00	16.43	11
	Model 3	236.01	57.37	16.06	327
LSTM-CNN Autoencoder	Model 1	99.74	20.23	8.94	60
	Model 2	146.16	20.83	7.64	94
	Model 3	128.64	17.09	7.60	310
	Model 4	89.58	19.66	8.28	135
Deep CNN	Model 1	-	-	18.41	1500
	Model 2	-	-	19.97	2040
	Model 3	-	-	18.40	1992
	Model 4	-	-	18.37	1450
	Model 5	-	-	17.58	1413
	Model 6	-	-	17.48	1202

TABLE 4.5: Summary of RMSE of prediction models on green taxi test dataset and the training time for each model

architecture with two layers of stacked LSTM learns the demand pattern and then passes it to a neural layer with eight nodes, which makes the interpretation. The last layer of one node produces the single output variable. The difference between model 4, model 3, and model 5 results stems from different hyperparameter settings. Model 1 and Model 2 of LSTM architecture have lower prediction accuracy on average. Interestingly, Model 6 does not outperform two-layer stacked architectures even though it is the deepest LSTM architecture. Furthermore, by stacking Time2Vec embedding with LSTM Model 4 we achieve lower RMSEs. This shows that the vector embedding of time series helps the algorithm to detect periodic behaviors in data in a more precise way.

The results for Random Forest show that RF-Model 1 and RF-Model 2 show signs of overfitting since training performance are orders of magnitude better than testing performance. In other words, the model is biased towards the training set, and it mimics its patterns to perfection, but as soon

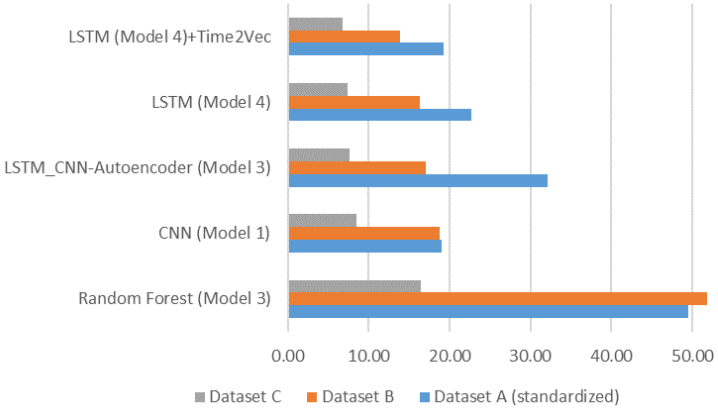


FIGURE 4.16: RMSE comparison between best-performing architectures for green NYC taxi data

as it is faced with out-of-set data, it cannot generalize the learned patterns enough to be able to make decent predictions. Moreover, the performance curve of testing indicates that the models underestimate the demand at evening peaks. To address the overfitting issue, the tuned RF-Model 3 is designed, which is the best-performing Random forest model out of 750 fits for dataset A. Although overfitting is mitigated, the model is not performing very well.

CNN models are much lighter than LSTM models and therefore run faster through training epochs. CNN Models have slightly lower but comparable accuracy levels compared to LSTM architectures. Interestingly, increasing the number of filters from 64 to 128 resulted in a worse accuracy across all datasets, indicating that more complicated feature maps do not necessarily output more accurate results. The LSTM-CNN Autoencoder architecture performs better than simple CNN, but it runs slower. However, it is more computationally efficient than LSTM architectures. In deep CNN architecture, Model 2, with more convolutional layers, results in better model performance compared to Model 1. It is interesting to observe that increasing the convolutional filters and thereby increasing the number of feature maps only has a constructive effect in deeper models, which might suggest that increasing filters in lighter models leads to overfitting and lower performance. The performance of Model 3 with another fully connected layer with 32 nodes which is close to the independent feature count of 28, dropped compared to Models 1 and 2. Model 4, with an additional layer

of batch normalization following each convolutional layer, results in faster models (fewer epochs are required for training). All things constant, model 4 indeed trained faster compared to model 3. Models 5 and 6 make use of a 20% dropout in the second and second to third max-pooling layers, respectively, which contributes to better results compared to other deep CNN models.

Model	Architecture	Dataset A (standardized)	Dataset B	Performance improvement
		Testing	Testing	
LSTM	Model 4	22.72	16.31	28.21%
CNN	Model 1	19.03	18.81	1.15%
Random Forest	Model 1	49.55	51.87	-4.47%
LSTM-CNN Autoencoder	Model 3	32.16	17.09	46.85%

TABLE 4.6: Standardized comparison between 1-hour aggregation (dataset A) and 15-minute aggregation (dataset B)

The different temporal aggregation has different impacts on model performance. By comparing the results of dataset A and dataset B shown in Table 4.5, RMSEs for dataset B are approximately 75% lower than A. This is because the RMSE for dataset A indicates the error within one hour, while in dataset B this error is for 15 minutes. Therefore, a comparison should be based on standardized values as presented in Table 4.6. The results indicate that a finer temporal aggregation improves models involving LSTM layers, as well as LSTM-CNN Autoencoder architectures, while CNN and random forest are not sensitive to the change of aggregation level. Consequently, the aggregation approach should be carefully selected for LSTM modeling.

4.7 CONCLUSIONS

In this work, we explored the task of predicting the demand for on-demand services with different deep-learning approaches. By conducting a cross-comparison between different architectures, we showed how different layers can affect the prediction performance. By vector embedding the time series, we improved the results. Moreover, by examining the temporal aggregation levels, we showed how different architectures are affected by the level of aggregation.

Improved datasets and modeling approaches should be included to develop the work further. The explored models can still be improved. Currently, all models except the random forest models are tuned manually.

To further improve the performance, the models need to be fine-tuned through robust grid searches. Moreover, spatial features at higher resolution levels can realize prediction at the taxi zone level, which is more helpful to the taxi service providers. Then, temporal features like real-time traffic load, congestion sites, and accidents can also be included. Finally, the models can be applied to other transportation modes like private cars and buses. To acquire these data, extensive data collection and fusion are needed. On the other hand, advanced models, and architectures such as customized CNN layers and multi-step architectures can be employed to improve model performance further.

It is worth mentioning that many studies on this subject focus solely on the location and time of demand as the only input to their algorithms. In our methodology, we aimed to incorporate all publicly available features to explore their impact on the prediction. A comprehensive list of all the features we have included is provided in Table 4.4 of our manuscript. To the best of our knowledge, no other studies have examined such a wide range of features and their influence on the prediction. The inclusion of all these features necessitates working at a certain level of data aggregation provided by the publicly available feature data source. Consequently, obtaining fine-grained feature data for smaller zones was not feasible for us. Considering these reasons, we selected methods that could accommodate all the data and features we intended to work with. In order to compare our results with other works, we refer to the work by Chen et al., 2021. The results provided in Table 2 and Table 5 of their manuscript (Chen et al., 2021) show that the RMSE values for their predictions are worse than our results, and moreover, the number of features we have considered is greater.

5

CONCLUSIONS AND OUTLOOK

This chapter offers a brief overview, summarizing the three main themes discussed in this thesis. The focus lies on their significant contributions and implications in optimizing on-demand transportation within urban environments. Additionally, the chapter highlights the limitations of the research and offers insights into future prospects.

5.1 SUMMARY OF FINDINGS AND INSIGHTS

Link travel time estimation

The first publication presented in Chapter 2 introduces a methodology for estimating historical link travel times using sparse GPS probe data. It allocates travel time data to different links traveled between GPS observations, incorporating spatial correlations between network links. The key contribution is demonstrating how progressive spatial correlations can be considered to improve results more realistically with a simple adjustment to existing parametric methods. The main findings of this chapter are as follows:

- By incorporating spatial correlations into our methodology, we observed significant enhancements in results compared to prior approaches. This highlights the importance of accounting for spatial relationships between links within a network to achieve more accurate estimations of travel times.
- Our study demonstrates that taking into account progressive correlation leads to improved outcomes. This nuanced consideration of how correlations evolve over time allows for a more refined estimation process resulting in better predictions of travel times.
- We found that employing a more precise correlation matrix is crucial for optimizing the performance of our algorithm. Fine-tuning the correlation matrix enhances the accuracy of our estimations, ensuring

that the algorithm effectively captures the complex interplay between spatial factors influencing travel times.

- The methodology effectively captures congestion patterns, aligning well with established methods for travel time estimation. This capability ensures that our estimations remain consistent with past observations and understanding of traffic dynamics, further validating the reliability of our approach.
- Our proposed methodology exhibits versatility by applying to a wide range of GPS probe vehicle datasets, whether they are synthetic or real-world. This adaptability underscores the robustness and generalizability of our approach across different data sources and urban environments.

Real-time ridesharing operations for on-demand capacitated systems

In the second publication in Chapter 3, a modular real-time simulation framework is developed to tackle the complexities of the capacitated ridesharing problem. This framework allows for flexible simulations, accommodating various scenarios and system configurations. The problem is formulated as a dynamic deterministic on-demand matching problem, with tolerance times improving the matching process. Dynamic congestion is implemented, updating travel times regularly to reflect evolving traffic conditions. To solve the optimization problem online, a mix of heuristic algorithms and commercial solvers is used. The main findings of this chapter are:

- Formulation of an online ridesharing problem as a dynamic deterministic model with tolerance times, allowing for the handling of different combinations of inputs simultaneously.
- Assessment of method performance against real data, demonstrating trade-offs between fleet size, total travel time, distance, waiting time, and passengers' delay.
- Implementation of dynamic link travel times to replicate realistic congestion patterns and evaluate different operational policies' effects on service performance.
- Demonstration that a significant reduction in taxi fleet size is feasible while maintaining service quality, with the potential to alleviate traffic congestion.

- Observation of increased vehicle occupancy during times of high demand, highlighting the potential for ridesharing.

Short-term passenger demand forecasting for on-demand transportation

The third publication presented in Chapter 4 focuses on developing accurate forecasting models specifically designed for short-term demand prediction, with an emphasis on deep learning approaches. The main findings are as follows:

- Our research highlights the correlation between data granularity and prediction accuracy, demonstrating that increased granularity leads to improved predictive performance. However, it's worth noting the influence of the machine learning architecture in this relationship.
- Integration of spatiotemporal features enhances prediction results, albeit future endeavors should focus on acquiring spatiotemporal data with greater precision to refine predictive models further.
- Through the utilization of vector embedding for time representation, our approach automates feature engineering, enhancing prediction accuracy by effectively capturing temporal patterns without requiring manual intervention.

5.2 LIMITATIONS

The proposed methodology in Chapter 2 for estimating historical link travel times based on GPS on-demand data presents several advantages, but there are also some limitations to consider. These limitations include:

- Assumption of shortest path: The model assumes that cabs always travel the shortest path based on distance. However, in real-world scenarios, drivers may not always choose the shortest path due to factors like traffic congestion, road closures, or driver preferences. This assumption may lead to inaccuracies in estimating travel times.
- Spatial correlation assumptions: The study proposes a spatial correlation matrix for each sub-network to consider spatial correlation. However, the effectiveness of these assumptions may vary depending on the specific road network and traffic patterns. It's important to validate and calibrate these assumptions for different locations and datasets to ensure accurate results.

- Data availability and quality: The study relies on GPS probe vehicle data collected by New York City taxi cabs. The accuracy and reliability of the data can influence the results. It's essential to ensure the quality of the data and consider potential biases or limitations associated with the data collection process.

- Lack of external validation: While the study mentions comparing the results against other benchmarks, it doesn't provide detailed information on the specific benchmarks used or the extent of the comparison. External validation against independent datasets or established models would strengthen the reliability and accuracy of the proposed methodology.

The simulation framework presented in Chapter 3 highlights the advantages and potential of the introduced method for assigning passenger trip requests to a fleet of vehicles, there are several limitations to consider:

- Sensitivity to parameters and inputs: The methodology mentions the ability to experiment with different parameters and inputs. However, the sensitivity of the method's performance to these parameters and inputs is not thoroughly explored. It is important to understand how sensitive the results are to changes in parameter values, input data, and whether the method consistently performs well across a range of parameter settings and input.

- Generalizability: The conclusions are based on a specific dataset and scenario (e.g., taxi rides in Manhattan). It is important to consider the generalizability of the findings to different cities, regions, or transportation systems with varying characteristics. The performance of the method may vary depending on the specific context.

- External factors: The study focuses primarily on the performance of the method itself and the impact of various parameters. However, there may be external factors, such as regulatory policies, infrastructure limitations, or user behavior, which can significantly influence the outcomes of implementing the method in a real-world setting. These external factors are not explicitly addressed in the study.

- Scalability: While the modular simulation framework claims to provide a flexible and scalable solution, the extent to which it can handle larger and more complex scenarios is not discussed. The performance

and computational requirements of the method may differ when applied to larger networks or with increased demand, and this scalability aspect needs to be further explored.

In the work presented in Chapter 4, the task of predicting the demand for on-demand services using various deep-learning approaches is explored. A cross-comparison between different architectures is conducted and the influence of different layers on prediction performance is demonstrated. By vector embedding the time series, improved results are achieved. However, despite these achievements, the work has certain limitations that need to be addressed:

- Spatial features at higher resolution: The study suggests that including spatial features at higher resolution levels, such as the taxi zone level, would be more beneficial for taxi service providers. However, the work does not address how to acquire and integrate these spatial features. The challenge lies in extensive data collection and fusion, which would require additional efforts.
- Temporal features: The method mentions the potential inclusion of real-time traffic load, congestion sites, and accidents as temporal features. Integrating such information into the models could improve their ability to capture dynamic changes in demand patterns. However, it does not provide specific details on how to obtain and incorporate these temporal features.
- Better fine-tuning: The models, except for the random forest models, are manually tuned. Fine-tuning the models through robust grid searches can potentially enhance their performance. Automated hyperparameter optimization techniques could be employed to systematically explore the hyperparameter space and identify optimal configurations.

Addressing these limitations would require additional research, data collection efforts, and methodological advancements.

5.3 FUTURE RESEARCH DIRECTIONS

The methodology explained in Chapter 2 presents several promising directions for future applications. Firstly, the proposed methodology can be extended to enable real-time travel time estimation by combining historical

estimates with real-time measurements. This advancement would provide up-to-date and accurate travel time information, leading to improved traffic management, routing optimization, and real-time navigation systems. Additionally, the methodology's versatility allows for its application to various datasets, including synthetic data or real-world datasets from different cities and transportation authorities. This flexibility opens up opportunities for widespread adoption and implementation. Another notable outlook is the consideration of spatial correlation, which can be further explored and refined to enhance the accuracy of travel time estimation. This aspect is crucial for understanding traffic patterns, congestion levels, and overall road network performance, making it valuable for traffic management and urban planning initiatives. Ultimately, this work's outcomes contribute to the potential for more precise travel time estimation, efficient traffic management systems, and informed decision-making in urban transportation planning.

The future direction for the work presented in Chapter 3 involves several key areas of research and development. Firstly, there is a need to further optimize and customize ridesharing services by fine-tuning parameters such as fleet capacities and tolerance times based on demand patterns. By leveraging the modular simulation framework, future studies can focus on achieving more efficient and tailored ridesharing experiences that meet the desired goals of both passengers and stakeholders. Additionally, considering the perspectives and objectives of various stakeholders is crucial. This can involve modeling the objectives of passengers, drivers, and transportation authorities to identify trade-offs and design policies that account for multiple objectives simultaneously. Furthermore, integrating reassignment strategies into the ridesharing algorithms can significantly improve their efficiency. Exploring the potential benefits of reassignment and developing algorithms that dynamically adjust the assignment of trip requests to vehicles can be a promising direction for future work.

Another important future direction is the optimization of capacity utilization and reduction of traffic congestion. The findings highlight the potential to reduce fleet size while maintaining or improving service quality, which can contribute to alleviating congestion in urban areas. Further research can focus on incentivizing higher vehicle occupancy, promoting ridesharing during peak hours, and integrating ridesharing with public transportation networks. Understanding user preferences and acceptance is another critical aspect. Future studies can investigate factors such as pricing models, waiting times, in-car delays, and overall convenience to enhance user expe-

rience and acceptance levels. By considering user needs and preferences, researchers can design and operate ridesharing services that align with user expectations. Additionally, it is essential to analyze system parameters and demand patterns. Research can explore different demand patterns and their effects on ridesharing system performance. This analysis can involve studying data from diverse cities or regions to gain insights into demand variations and develop adaptive strategies that account for specific urban characteristics and transportation requirements. By considering these areas, future work can contribute to the development of more efficient, user-friendly, and sustainable ridesharing systems.

There are several areas of improvement that will shape the future direction of the work presented in Chapter 4. Researchers can focus on gathering improved datasets and exploring novel modeling approaches to enhance the prediction of on-demand service demand. By acquiring comprehensive and diverse data, and investigating alternative deep learning architectures or advanced modeling techniques, the accuracy and effectiveness of predictions can be enhanced. Additionally, fine-tuning the models through robust grid searches can improve performance by optimizing the models' hyperparameters. Integrating spatial features at higher resolution levels, such as geographical attributes and neighborhood characteristics, would refine the models to better capture spatial dependencies and improve the accuracy of demand forecasting. Furthermore, incorporating real-time traffic data and other temporal features into the models would enable them to account for dynamic factors that influence on-demand service demand. Finally, employing advanced models and architectures, such as customized CNN layers and multi-step architectures, shows promise in capturing long-term dependencies and further improving prediction performance. These advancements have the potential to enhance the accuracy and effectiveness of predicting on-demand service demand.

Ultimately, the combined findings from Chapters 2, 3, and 4 highlight promising avenues for optimizing shared on-demand urban transportation. Integrating methodologies like demand forecasting in both fleet dispatching and idle vehicle replacement of ride-sharing platforms presents opportunities for designing more efficient shared on-demand mobility services.

5.4 OUTLOOK

Further investigation is required to understand the effects of various models on the mobility ecosystem. Factors such as elastic demand, endogenous

congestion, operational constraints, and ride-sharing have significant implications and need to be explored in-depth. Elastic demand, for example, refers to the phenomenon where the demand for mobility services changes based on factors such as pricing or availability. Understanding how this demand elasticity impacts the overall system and its efficiency is crucial for optimizing the provision of mobility services. Similarly, endogenous congestion, which arises from the interactions among travelers, needs to be studied to develop strategies that alleviate congestion and improve traffic flow. Additionally, operational constraints and the potential benefits and challenges of integrating ride-sharing services into existing transportation systems should be thoroughly investigated. These areas of research will pave the way for more efficient and sustainable mobility solutions.

In the context of designing mobility solutions, it is essential to adopt a co-design approach that considers the system they enable. Co-design emphasizes the need to involve multiple stakeholders, including policymakers, urban planners, transportation providers, and the public, in the design process. By actively involving these stakeholders, the resulting solutions can better address their diverse needs and priorities. Co-design also fosters collaboration and enables the identification of potential conflicts or trade-offs that may arise in the mobility ecosystem. Furthermore, it encourages the integration of innovative technologies and novel ideas into the design process, leading to more comprehensive and inclusive solutions. Ultimately, co-designing mobility solutions ensure that they are tailored to the specific context and requirements of the system, promoting their effectiveness and long-term sustainability.

Lastly, it is crucial to address concerns related to fairness, privacy, and trust in the deployment of on-demand shared mobility services. Fairness among customers should be a paramount consideration to prevent discrimination and ensure equal access to transportation resources. Additionally, protecting privacy and establishing trust in handling private data are essential for maintaining user confidence in these services. Robust tools and frameworks are needed to rigorously reason about the interactions among stakeholders and address these concerns. By proactively addressing fairness, privacy, and trust, the deployment of on-demand shared mobility can foster public acceptance and contribute to a more equitable and secure mobility ecosystem.

A

APPENDIX A: RESULTS FOR THE CASE STUDY 2.5

In this appendix, we present the results of all the proposed models in the second chapter of this thesis 2.3 for another day of the week (Wednesday 02-02-2011) in Figure A.1 and a Weekend day (Saturday 05-02-2011) in Figure A.2. In addition, the experimental result comparison between the proposed models is presented in Table A.1 for Wednesday 02-02-2011 and in Table A.2 for Saturday 05-02-2011. We can conclude that, the progressive model has the best performance comparing to the other models.

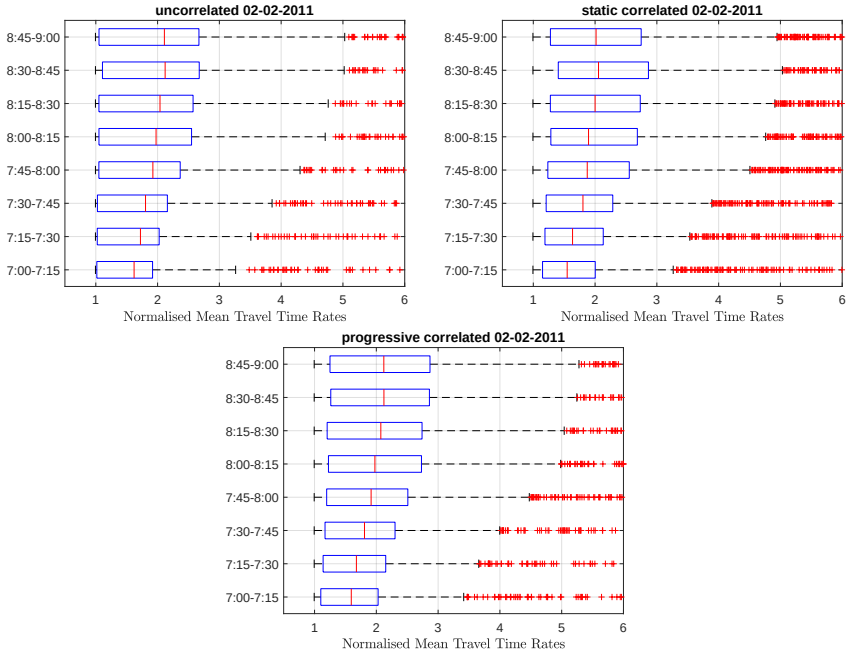


FIGURE A.1: Normalized Mean Travel Time Rates for all models for Wednesday 02-02-2011

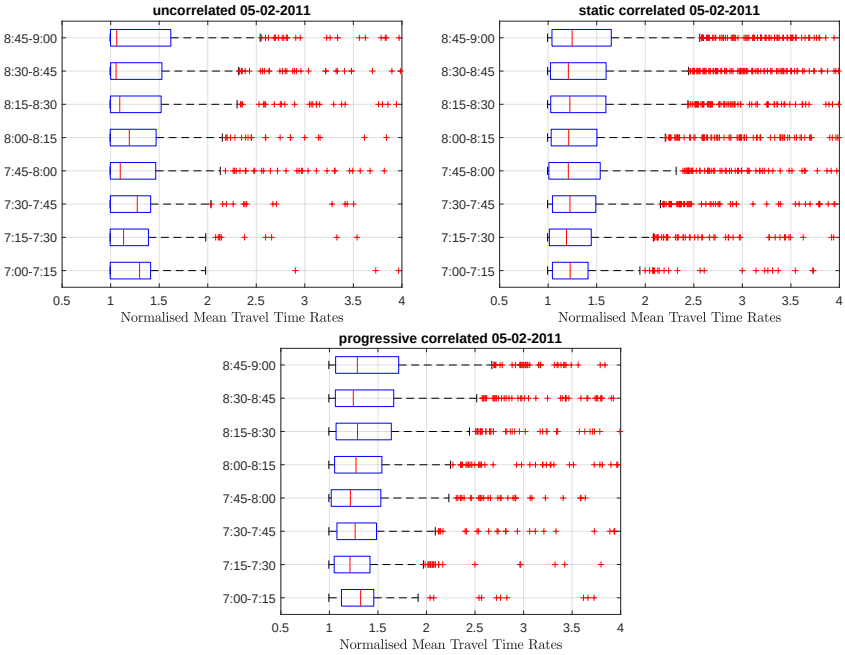


FIGURE A.2: Normalized Mean Travel Time Rates for all the models for Saturday 05-02-2011

Model	RMSE (sec)	MPE	MAPE
Uncorrelated(Herring's baseline model)	144.72	-5.83%	20.89%
Static Correlated	137.98	-5.42%	19.85%
Progressive Correlated	127.73	-4.15%	18.08%

TABLE A.1: Experimental results comparison between the proposed models and the baseline model for Wednesday 02-02-2011.

Model	RMSE (sec)	MPE	MAPE
Uncorrelated(Herring's baseline model)	76.25	-5.20 %	15.77%
Static Correlated	72.00	-4.49 %	15.30%
Progressive Correlated	70.17	-3.68%	14.32%

TABLE A.2: Experimental results comparison between the proposed models and the baseline model for Saturday 05-02-2011.

B

APPENDIX B: SUPPLEMENTAL EXPLANATION OF CONSTRAINT 3.10

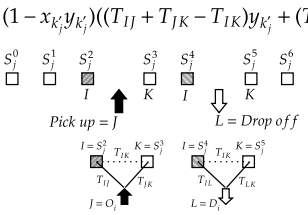
Constraint (3.10) in our optimization problem is quite complex, as we have decided to write it in a compact form and include all possible combinations that can arise. This makes the problem more efficient time-wise as it helps the solver to reduce the search space; however, it is quite complex for the reader to grasp the physical meaning of this constraint and what it represents for the acquired solution. As mentioned in the main document, constraint reads

$$\sum_{k'=0}^{k'=k_j-1} (1 - x_{k'_j y_{k'_j}}) \left(\left(T_{IJ} + T_{JK} - T_{IK}\right) y_{k'_j} + \left(T_{IL} + T_{LK} - T_{IK}\right) x_{k'_j} \right) +$$

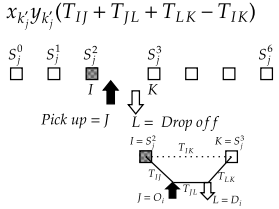
$$x_{k'_j y_{k'_j}} (T_{IJ} + T_{JL} + T_{LK} - T_{IK}) + e_{4k_j} = D_n^{p/d} - d_n^{p/d}, \quad \forall j \in \mathcal{M},\ k_j \in \mathcal{S}_{K_j} \backslash \{S_{0_j}\}$$

where k' is a counter of stops from 0 (first) to stop k_j (last) and $I = S_{k'_j}$, $J = O_i$, $K = S_{k'_j+1}$, $L = D_i$. Here we try to show how the constraint works with the help of graphics depicted in Figure B.1. If shuttle j has six stops in its stop list (squares in Figure B.1).

The First part of the constraint checks if the pick-up of the new request and the drop-off are assigned after different stops in the shuttle (see Figure B.1 (a)), what would be the extra travel time caused by this assignment. $(T_{IJ} + T_{JK} - T_{IK})$ is the added travel time for picking up the new request, and $(T_{IL} + T_{LK} - T_{IK})$ corresponds to drop-off detour for serving this request. The constraint checks if the already assigned customers in the shuttle can tolerate the added travel time caused by this assignment. Remaining delay tolerance at pick-up p or drop-off d for request n in the shuttle is equal to $D_n^{p/d}$ (initial value) $-d_n^{p/d}$ (used value). The term $(1 - x_{k'_j y_{k'_j}})$ in this part omits the values for which the pick-up and drop-off are happening after the same stop since the extra travel time is calculated differently. In the second part, the case that pick-up and drop-off of the new request are assigned after the same stop is considered in Figure B.1 (b), and the extra travel time $(T_{IJ} + T_{JL} + T_{LK} - T_{IK})$ is depicted more clearly.



(a) Pick-up and drop-off of new request happens after different stops



(b) Pick-up and drop-off of new request happens after the same stops

FIGURE B.1: Graphical explanation of constraint 10

Furthermore, we provide a detailed arithmetic example to illustrate the necessity and functionality of constraint (3.10).

Let us now consider a specific example for a shuttle j that has 5 stops in its current plan (to follow). This constraint would then decompose to (for all)

- planned stop $k_j = 0$ or current location; k' from 0 to -1: that is the reason we need to exclude stop 0, i.e., $\forall \mathcal{S}_{K_j} \backslash \{S_{0_j}\}$
- planned stop $k_j = 1$; k' from 0 to 0:[1]

$$(1 - x_{0_j} y_{0_j}) \left(\left(T_{0p} + T_{p1} - T_{01} \right) y_{0_j} + \left(T_{0d} + T_{d1} - T_{01} \right) \cdot x_{0_j} \right) + x_{0_j} y_{0_j} (T_{0p} + T_{pd} + T_{d1} - T_{01}) + e_{4_1 j} = D^{p/d}_{\text{R:S:1}} - d^{p/d}_{\text{R:S:1}} \quad \text{(B.1)}$$

1 Note that R:S:k denotes that this request is matched with stop k in all equations hereafter.

- planned stop $k_j = 2$; k' from 0 to 1:

$$\begin{aligned}(1 - x_{0_j} y_{0_j})((T_{0p} + T_{p1} - T_{01}) y_{0_j} + (T_{0d} + T_{d1} - T_{01}) x_{0_j}) + \\ x_{0_j} y_{0_j} (T_{0p} + T_{pd} + T_{d1} - T_{01}) + \\ (1 - x_{1_j} y_{1_j})((T_{1p} + T_{p2} - T_{12}) y_{1_j} + (T_{1d} + T_{d2} - T_{12}) x_{1_j}) + \\ x_{1_j} y_{1_j} (T_{1p} + T_{pd} + T_{d2} - T_{12}) + e_{4_2 j} = D^{p/d}_{\text{R:S:2}} - d^{p/d}_{\text{R:S:2}}\end{aligned} \quad \text{(B.2)}$$

- planned stop $k_j = 3$; k' from 0 to 2:

$$\begin{aligned}(1 - x_{0_j} y_{0_j})((T_{0p} + T_{p1} - T_{01}) y_{0_j} + (T_{0d} + T_{d1} - T_{01}) x_{0_j}) + \\ x_{0_j} y_{0_j} (T_{0p} + T_{pd} + T_{d1} - T_{01}) + \\ (1 - x_{1_j} y_{1_j})((T_{1p} + T_{p2} - T_{12}) y_{1_j} + (T_{1d} + T_{d2} - T_{12}) x_{1_j}) + \\ x_{1_j} y_{1_j} (T_{1p} + T_{pd} + T_{d2} - T_{12}) + \\ (1 - x_{2_j} y_{2_j})((T_{2p} + T_{p3} - T_{23}) y_{2_j} + (T_{2d} + T_{d3} - T_{23}) x_{2_j}) + \\ x_{2_j} y_{2_j} (T_{2p} + T_{pd} + T_{d3} - T_{23}) + e_{4_3 j} = D^{p/d}_{\text{R:S:3}} - d^{p/d}_{\text{R:S:3}}\end{aligned} \quad \text{(B.3)}$$

- planned stop $k_j = 4$; k' from 0 to 3:

$$\begin{aligned}(1 - x_{0_j} y_{0_j})((T_{0p} + T_{p1} - T_{01}) y_{0_j} + (T_{0d} + T_{d1} - T_{01}) x_{0_j}) + \\ x_{0_j} y_{0_j} (T_{0p} + T_{pd} + T_{d1} - T_{01}) + \\ (1 - x_{1_j} y_{1_j})((T_{1p} + T_{p2} - T_{12}) y_{1_j} + (T_{1d} + T_{d2} - T_{12}) x_{1_j}) + \\ x_{1_j} y_{1_j} (T_{1p} + T_{pd} + T_{d2} - T_{12}) + \\ (1 - x_{2_j} y_{2_j})((T_{2p} + T_{p3} - T_{23}) y_{2_j} + (T_{2d} + T_{d3} - T_{23}) x_{2_j}) + \\ x_{2_j} y_{2_j} (T_{2p} + T_{pd} + T_{d3} - T_{23}) + \\ (1 - x_{3_j} y_{3_j})((T_{3p} + T_{p4} - T_{34}) y_{3_j} + (T_{3d} + T_{d4} - T_{34}) x_{3_j}) + \\ x_{3_j} y_{3_j} (T_{3p} + T_{pd} + T_{d4} - T_{34}) + e_{4_4 j} = D^{p/d}_{\text{R:S:4}} - d^{p/d}_{\text{R:S:4}}\end{aligned} \quad \text{(B.4)}$$

- planned stop $k_j = 5$; k' from 0 to 4:

$$\begin{aligned}(1 - x_{0_j} y_{0_j})((T_{0p} + T_{p1} - T_{01}) y_{0_j} + (T_{0d} + T_{d1} - T_{01}) x_{0_j}) + \\ x_{0_j} y_{0_j} (T_{0p} + T_{pd} + T_{d1} - T_{01}) + \\ (1 - x_{1_j} y_{1_j})((T_{1p} + T_{p2} - T_{12}) y_{1_j} + (T_{1d} + T_{d2} - T_{12}) x_{1_j}) + \\ x_{1_j} y_{1_j} (T_{1p} + T_{pd} + T_{d2} - T_{12}) + \\ (1 - x_{2_j} y_{2_j})((T_{2p} + T_{p3} - T_{23}) y_{2_j} + (T_{2d} + T_{d3} - T_{23}) x_{2_j}) + \\ x_{2_j} y_{2_j} (T_{2p} + T_{pd} + T_{d3} - T_{23}) + \\ (1 - x_{3_j} y_{3_j})((T_{3p} + T_{p4} - T_{34}) y_{3_j} + (T_{3d} + T_{d4} - T_{34}) x_{3_j}) + \\ x_{3_j} y_{3_j} (T_{3p} + T_{pd} + T_{d4} - T_{34}) + \\ (1 - x_{4_j} y_{4_j})((T_{4p} + T_{p5} - T_{45}) y_{4_j} + (T_{4d} + T_{d5} - T_{45}) x_{4_j}) + \\ x_{4_j} y_{4_j} (T_{4p} + T_{pd} + T_{d5} - T_{45}) + e_{4_5 j} = D^{p/d}_{\text{R:S:5}} - d^{p/d}_{\text{R:S:5}} \end{aligned} \tag{B.5}$$

For instance, if the optimization results in picking up the request after planned stop $k_j = 0$ and dropping it off after planned stop $k_j = 2$, this would result in decision variables $y_{0j} = 1$, $x_{2j} = 1$. This constraint would then give

- planned stop $k_j = 1$; k' from 0 to 0:

$$(T_{0p} + T_{p1} - T_{01}) y_{0_j} + e_{4_1 j} = D^{p/d}_{\text{R:S:1}} - d^{p/d}_{\text{R:S:1}} \tag{B.6}$$

- planned stop $k_j = 2$; k' from 0 to 1:

$$(T_{0p} + T_{p1} - T_{01}) y_{0_j} + e_{4_2 j} = D^{p/d}_{\text{R:S:2}} - d^{p/d}_{\text{R:S:2}} \tag{B.7}$$

- planned stop $k_j = 3$; k' from 0 to 2:

$$(T_{0p} + T_{p1} - T_{01}) y_{0_j} + (T_{2d} + T_{d3} - T_{23}) x_{2_j} + e_{4_3 j} = D^{p/d}_{\text{R:S:3}} - d^{p/d}_{\text{R:S:3}} \tag{B.8}$$

- planned stop $k_j = 4$; k' from 0 to 3:

$$(T_{0p} + T_{p1} - T_{01}) y_{0_j} + (T_{2d} + T_{d3} - T_{23}) x_{2_j} + e_{4_4 j} = D^{p/d}_{\text{R:S:4}} - d^{p/d}_{\text{R:S:4}} \tag{B.9}$$

- planned stop $k_j = 5$; k' from 0 to 4:

$$(T_{0p} + T_{p1} - T_{01})y_{0_j} + (T_{2d} + T_{d3} - T_{23}) \cdot x_{2_j} + e_{4_5 j} = D^{p/d}_{\text{R:S:5}} - d^{p/d}_{\text{R:S:5}} \tag{B.10}$$

In another case, if the optimization result is that the pick-up and drop-off happen after planned stop $k_j = 2$, this would result in decision variables $y_{2j} = 1$, $x_{2j} = 1$. This constraint would then give

- planned stop $k_j = 1$; k' from 0 to 0:

$$e_{4_1 j} = D^{p/d}_{\text{R:S:1}} - d^{p/d}_{\text{R:S:1}} \tag{B.11}$$

- planned stop $k_j = 2$; k' from 0 to 1:

$$e_{4_2 j} = D^{p/d}_{\text{R:S:2}} - d^{p/d}_{\text{R:S:2}} \tag{B.12}$$

- planned stop $k_j = 3$; k' from 0 to 2:

$$x_{2_j} y_{2_j} (T_{2p} + T_{pd} + T_{d3} - T_{23}) + e_{4_3 j} = D^{p/d}_{\text{R:S:3}} - d^{p/d}_{\text{R:S:3}} \tag{B.13}$$

- planned stop $k_j = 4$; k' from 0 to 3:

$$x_{2_j} y_{2_j} (T_{2p} + T_{pd} + T_{d3} - T_{23}) + e_{4_4 j} = D^{p/d}_{\text{R:S:4}} - d^{p/d}_{\text{R:S:4}} \tag{B.14}$$

- planned stop $k_j = 5$; k' from 0 to 4:

$$x_{2_j} y_{2_j} (T_{2p} + T_{pd} + T_{d3} - T_{23}) + e_{4_5 j} = D^{p/d}_{\text{R:S:5}} - d^{p/d}_{\text{R:S:5}} \tag{B.15}$$

C

APPENDIX C: NOTATION TABLE FOR CHAPTER 3

Parameter	Description
i	request number
n	assigned request number in a shuttle
N	total number of requests , $i \in \mathcal{N} = \{1, 2, \ldots, N\}$
P_i	number of passengers in the request i
T_i^p	time requested for pick-up of request i
O_i	pick-up location (origin)
D_i	drop-off location (destination)
D_i^p	maximum waiting time that the commuters will tolerate as delay in pick-up (initial value)
D_i^d	maximum in-car delay accepted by the passengers caused by ridesharing (initial value)
T_i^d	calculated drop-off time of request i (if the request is served without any delay)
d_i^p	calculated waiting time at pick-up after assignment
d_i^d	calculated in car delay caused by ridesharing after assignment
t	current time
y_{k_j}	Variable defining if the pick-up of request i will be placed after stop k in shuttle j
x_{k_j}	Variable defining if the drop-off of request i will be placed after stop k in shuttle j
j	shuttle number
M	total number of shuttles $j \in \mathcal{M} = \{1, 2, \ldots, M\}$
C_j	capacity of shuttle j
P_j	current occupancy of shuttle j
S_{K_j}	list of stops of shuttle j ,$S_{K_j} = \{0_j, 1_j, 2_j, \ldots, K_j\}$
$K_j + 1$	the total number of stops in shuttle j
k_j	k_{th} stop in shuttle j, $k_j \in S_{K_j}$
L_{k_j}	location of stop k in shuttle j
A_{k_j}	estimated arrival time to stop k in shuttle j
0_j	current position of shuttle j in the network at time t and, $A_{0_j} = t$.
E	directed edges in the network graph
V	vertices in the network graph
$G(V, E)$	network graph consisting of vertices and directed edges
T_{IJ}	travel time from vertex I to vertex J in the network graph
$f(i, j)$	function defining the cost of serving request i with shuttle j
e	epsilon values in each constraint, distance to the upper bound of the constraint
α	coefficient for defining the percentage of each part in an objective function

TABLE C.1: Notation Table

BIBLIOGRAPHY

Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2010). *Sustainable Passenger Transportation: Dynamic Ride-Sharing* (ERIM Report Series Research in Management ERS-2010-010-LIS). Erasmus Research Institute of Management (ERIM), ERIM is the joint research institute of the Rotterdam School of Management, Erasmus University and the Erasmus School of Economics (ESE) at Erasmus University Rotterdam.

Allström, A., Ekström, J., Gundlegård, D., Ringdahl, R., Rydergren, C., Bayen, A. M., & Patire, A. D. (2016). Hybrid approach for short-term traffic state and travel time prediction on highways. *Transportation Research Record*, *2554*(1), 60.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, *114*(3), 462.

Amey, A. M. (2010). *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.

Amirkiaee, S. Y., & Evangelopoulos, N. (2018). Why do people rideshare? an experimental study. *Transportation Research Part F: Traffic Psychology and Behaviour*, *55*, 9.

Batista, S. F. A., Cantelmo, G., Menéndez, M., & Antoniou, C. (2022). A gaussian sampling heuristic estimation model for developing synthetic trip sets. *Computer-Aided Civil and Infrastructure Engineering*, *37*(1), 93.

Beaudry, A., Laporte, G., Melo, T., & Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR spectrum*, *32*(1), 77.

Berbeglia, G., Cordeau, J.-F., & Laporte, G. (2012). A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*, *24*(3), 343.

Bertsimas, D., Delarue, A., Jaillet, P., & Martin, S. (2019). Travel time estimation in the age of big data. *Operations Research*, *67*(2), 498.

Boesch, P. M., Ciari, F., & Axhausen, K. W. (2016). Autonomous vehicle fleet sizes required to serve different levels of demand. *Transportation Research Record*, *2542*, 111.

Bongiovanni, C., Kaspi, M., & Geroliminis, N. (2019). The electric autonomous dial-a-ride problem. *Transportation Research Part B: Methodological*, *122*, 436.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5.

Burris, M. W., & Winn, J. R. (2006). Slugging in houston—casual carpool passenger characteristics. *Journal of Public Transportation 9*, *5*(2.).

Carey, W. P. (2016). Do ridesharing services affect traffic congestion ? an empirical study of uber entry.

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. (2000). *Advanced public transportation systems: The state of the art update 2000*. Tech.

Chan, K., Lam, W., & Tam, M. (2009). Real-time estimation of arterial travel times with spatial travel time covariance relationships. *Transportation Research Record*, *2121*, 102.

Chen, B. Y., Lam, W., & Li, Q. (2016). Efficient solution algorithm for finding spatially-dependent reliable shortest path in road networks. *Journal of advanced transportation*, *50*, 1413.

Chen, L., Mislove, A., & Wilson, C. (2015). Peeking beneath the hood of uber. *Proceedings of the 2015 Internet Measurement Conference*.

Chen, L., Thakuriah, P. (, & Ampountolas, K. (2021). Short-term prediction of demand for ride-hailing services: A deep learning approach. *Journal of Big Data Analytics in Transportation*, *3*(2), 175.

Chen, M., & Chien, S. I. J. (2001). Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based. *Transportation Research Record*, *1768*(1), 157.

Cheng, Q., Liu, Y., Wei, W., & Liu, Z. (2016). Analysis and forecasting of the day-to-day travel demand variations for large- scale transportation networks: A deep learning approach.

Chester, M., Pincetl, S., Elizabeth, Z., Eisenstein, W., & Matute, J. (2013). Infrastructure and automobile shifts: Positioning transit to reduce life-cycle environmental impacts for urban sustainability goals. *Environmental Research Letters*, *8*(1).

Chicago Open Data. (2020). Taxi trips reported to the city of chicago.

City of New York Department of Information Technology & Telecommunications. (2023). Point of interest (commonplace).

Code of Virginia. (2015). "ridesharing arrangement" defined.

Coifman, B. (2002). Estimating travel times and vehicle trajectories on freeways using dual loop detectors. *Transportation Research Part A: Policy and Practice*, *36*(4), 351.

Cordeau, J.-F., & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of operations research*, *153*(1), 29.

Dailey, D., Loseff, D., & Meyers, D. (1999). Seattle smart traveler: Dynamic ridematching on the world wide web. *Transportation Research Part C: Emerging Technologies*, *7*(1), 17.

Dandl, F., Hyland, M., Bogenberger, K., & Mahmassani, H. S. (2019). Evaluating the impact of spatio-temporal demand forecast aggregation on the operational performance of shared autonomous mobility fleets. *Transportation*, *46*(6), 1975.

Delling, D., Sanders, P., Schultes, D., & Wagner, D. (2009). Engineering route planning algorithms. In J. Lerner, D. Wagner, & K. A. Zweig (Eds.), *Algorithmics of large and complex networks: Design, analysis, and simulation* (pp. 117–139). Springer Berlin Heidelberg.

Eisele, W., & Rilett, L. (2002). Estimating corridor traveltime mean, variance, correlation with intelligent transporta-tion systems link travel time data. *In Proceedings of the Transportation Research Board 81st Annual Meeting, Washing-ton, D.C.*

El Esawey, M., & Sayed, T. (2011). Travel time estimation in urban networks using limited probes data. *Canadian Journal of Civil Engineering*, *38*(3), 305.

Ercan, T., Onat, N. C., & Tatari, O. (2016). Investigating carbon footprint reduction potential of public transportation in united states: A system dynamics approach. *Journal of Cleaner Production*, *133*, 1260.

Fagnant, D. J., & Kockelman, K. M. (2018). Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas. *Transportation*, *45*, 143.

Fielbaum, A., Kronmuller, M., & Alonso-Mora, J. (2021). Anticipatory routing methods for an on-demand ridepooling mobility system. *Transportation*.

Flötteröd, G., & Bierlaire, M. (2013). Metropolis–hastings sampling of paths. *Transportation Research Part B: Methodological*, *48*, 53.

Fu, L., & Rilett, L. (1998). Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B: Methodological*, *32*(7), 499.

Fulton, L., Mason, J., & Meroux, D. (2017). *Three revolutions in urban transportation*. UC Davis Institute for Transportation & Development Policy.

Gajewski, B., & Rilett, L. (2005). Estimating link travel time correlation: An application of bayesian smoothing splines. *Journal of Transportation and Statistics*, *7*(2-3), 53.

Gao, J., Wang, Y., Tang, H., Yin, Z., Ni, L., & Shen, Y. (2017). An efficient dynamic ridesharing algorithm. *2017 IEEE International Conference on Computer and Information Technology (CIT)*, 320.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., & Liu, Y. (2019). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 3656.

Genser, A., Hautle, N., Makridis, M., & Kouvelas, A. (2022). An experimental urban case study with various data sources and a model for traffic estimation. *Sensors*, *22*(1).

Ghandeharioun, Z., & Kouvelas, A. (2022). Link travel time estimation for arterial networks based on sparse gps data and considering progressive correlations. *IEEE Open Journal of Intelligent Transportation Systems*, *3*, 679.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, *22*(4), 455.

Golob, T. F., & Regan, A. C. (2001). Impacts of information technology on personal travel and commercial vehicle operations: Research challenges and opportunities. *Transportation Research Part C: Emerging Technologies*, *9*(2), 87.

Google Developers. (2020). The directions api overview.

Greenblatt, J. B., & Saxena, S. (2015). Autonomous taxis could greatly reduce greenhouse-gas emissions of us light-duty vehicles. *Nature Climate Change*, *5*(9), 860.

Grynbaum, M. M. (2010). Gridlock may not be constant, but slow going is here to stay. *The New York Times*, 1 Section A.

Guo, J., Huang, W., & Williams, B. M. (2014). Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, *43*, 50.

Gurobi Optimization, LLC. (2021). Gurobi Optimizer Reference Manual.

Hall, R. W. (1986). The fastest path through a network with random time-dependent travel times. *Transportation Science*, *20*(3), 182.

Häll, C. H., Lundgren, J. T., & Voß, S. (2015). Evaluating the performance of a dial-a-ride service using simulation. *Public Transport*, *7*(2), 139.

Haselkorn, M., Spyridakis, J., Goble, B., & Michalak, S. (1994). 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*.

Herring, R., Hofleitner, A., Abbeel, P., & Bayen, A. (2010). Estimating arterial traffic conditions using sparse probe data. *13th International IEEE Conference on Intelligent Transportation Systems*, 929.

Herring, R. J. (2010). *Real-time traffic modeling and estimation with streaming probe data using machine learning* (Doctoral dissertation). Industrial Engineering and Operations Research, University of California, Berkeley.

Ho, S. C., Szeto, W., Kuo, Y.-H., Leung, J. M., Petering, M., & Tou, T. W. (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, *111*, 395.

Hofleitner, A., Herring, R., & Bayen, A. (2012). Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning. *Transportation Research Part B: Methodological*, *46*(9), 1097.

Hörl, S., & Zwick, F. (2022). Traffic Uncertainty in On-Demand High-Capacity Ride-Pooling. *101st Annual Meeting of the Transportation Research Board (TRB)*.

Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, *15*, 2191.

Hunter, T., Herring, R., Abbeel, P., & Bayen, A. (2009). Path and travel time inference from gps probe vehicle data. *NIPS Analyzing Networks and Learning with Graphs*, *12*(1), 2.

Hyland, M., & Mahmassani, H. S. (2020). Operational benefits and challenges of shared-ride automated mobility-on-demand services. *Transportation Research Part A: Policy and Practice*, *134*, 251.

Hyland, M., & Mahmassani, H. (2017). Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility. *Transportation Research Record*, *2653*, 26.

Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., & Pavone, M. (2017). Data-driven model predictive control of autonomous mobility-on-

demand systems. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1.

Javidi, H., Simon, D., Zhu, L., & Wang, Y. (2021). A multi-objective optimization framework for online ridesharing systems. *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 252.

Jenelius, E., & Koutsopoulos, H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological*, *53*, 64.

Johnson, D., Ercolani, M., & Mackie, P. (2017). Econometric analysis of the link between public transport accessibility and employment. *Transport Policy*, *60*, 1.

Julagasigorn, P., Banomyong, R., Grant, D. B., & Varadejsatitwong, P. (2021). What encourages people to carpool? a conceptual framework of carpooling psychological factors and research propositions. *Transportation Research Interdisciplinary Perspectives*, *12*, 100493.

Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., & Brubaker, M. (2019). Time2vec: Learning a vector representation of time.

Ke, J., Feng, S., Zhu, Z., Yang, H., & Ye, J. (2021). Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. *Transportation Research Part C: Emerging Technologies*, *127*, 103063.

Ke, J., Zheng, H., Yang, H., & Chen, X. ((2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, *85*, 591.

Kelly, K. L. (2007). Casual carpooling-enhanced. *Journal of Public Transportation 10*, *4*(6.).

Khreis, H., Warsow, K. M., Verlinghieri, E., Guzman, A., Pellecuer, L., Ferreira, A., Jones, I., Heinen, E., Rojas-Rueda, D., Mueller, N., Schepers, P., Lucas, K., & Nieuwenhuijsen, M. (2016). The health impacts of traffic-related exposures in urban areas: Understanding real effects, underlying driving forces and co-producing future directions. *Journal of Transport & Health*, *3*(3), 249.

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

Künzli, N., Kaiser, R., Medina, S., Studnicka, M., Chanel, O., Filliger, P., Herry, M., Horak, F., Puybonnieux-Texier, V., Quénel, P., Schneider, J., Seethaler, R., Vergnaud, J.-C., & Sommer, H. (2000). Public-health

impact of outdoor and traffic-related air pollution: A european assessment. *The Lancet*, *356*(9232), 795.

Leduc, G., et al. (2008). Road traffic data: Collection methods and applications. *Working Papers on Energy, Transport and Climate Change*, *1*(55), 1.

Levin, M. W., Kockelman, K. M., Boyles, S. D., & Li, T. (2017). A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. *Computers, Environment and Urban Systems*, *64*, 373.

Li, R., & Rose, G. (2011). Incorporating uncertainty into short-term travel time predictions. *Transportation Research Part C: Emerging Technologies*, *19*(6), 1006.

Li, S., Tavafoghi, H., Poolla, K., & Varaiya, P. P. (2019). Regulating tncs: Should uber and lyft set their own rules? *Transportation Research Part B: Methodological*.

Litman, T. (2012). *Evaluating public transportation health benefits*. Victoria Transport Policy Institute Victoria, BC, Canada.

Liu, H. X., & Ma, W. (2009). A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials. *Transportation Research Part C: Emerging Technologies*, *17*(1), 11.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865.

Ma, T.-Y., Rasulkhani, S., Chow, J. Y., & Klein, S. (2019). A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. *Transportation Research Part E: Logistics and Transportation Review*, *128*, 417.

Ma, X., Yu, H., Wang, Y., & Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PLOS ONE*, *10*(3), 1.

Ma, Z., Koutsopoulos, H. N., Ferreira, L., & Mesbah, M. (2017). Estimation of trip travel time distribution using a generalized markov chain approach. *Transportation Research Part C: Emerging Technologies*, *74*, 1.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013). Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1393.

The multivariate normal distribution. (2002). In *Methods of multivariate analysis* (pp. 82–111). John Wiley & Sons, Ltd.

National Oceanic and Atmospheric Administration. (2023). National oceanic and atmospheric administration.

Nielsen, O. A. (September 1997). On the distributions of the stochastic components in sue (stochastic user equilibrium) traffic assignment models. *Transportation planning methods Volume 11. Proceedings of seminar F held at PTRC European Transport Forum, Brunel University, England, 1-5 p. 77-93*, *Volume P415*.

NYC Taxi and Limousine Commission. (2019). NYC TLC Trip Record Data.

Open street map. (2021).

Ota, M., Vo, H., Silva, C., & Freire, J. (2017). Stars: Simulating taxi ride sharing at scale. *IEEE Transactions on Big Data*, *3*(3), 349.

Outscraper. (2022). Google maps traffic extractor.

Park, D., & Rilett, L. R. (1998). Forecasting multiple-period freeway link travel times using modular neural networks. *Transportation Research Record*, *1617*(1), 163.

Pelzer, D., Xiao, J., Zehe, D., Lees, M. H., Knoll, A. C., & Aydt, H. (2015). A partition-based match making algorithm for dynamic ridesharing. *Trans. Intell. Transport. Syst.*, *16*(5), 2587.

Peng, B., Du, H., Ma, S., Fan, Y., & Broadstock, D. C. (2015). Urban passenger transport energy saving and emission reduction potential: A case study for tianjin, china. *Energy Conversion and Management*, *102*, 4.

Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, *225*(1), 1.

Profillidis, V., & Botzoris, G. (2018). *Modeling of transport demand analyzing, calculating, and forecasting transport demand*. Elsevier.

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, *67*(1), 3.

Rachtan, P., Huang, H., & Gao, S. (2013). Spatiotemporal link speed correlations: Empirical study. *Transportation Research Record*, *2390*(1), 34.

Rahmani, M., Jenelius, E., & Koutsopoulos, H. (2015). Non-parametric estimation of route travel time distributions from low-frequency floating car data. *Transportation Research Part C: Emerging Technologies*, *58B*, 343.

Rakha, H., EL-Shawarby, I., Arafeh, M., & Dion, F. (2006). Estimating path travel-time reliability. *2006 IEEE Intelligent Transportation Systems Conference*, 236.

Ramezani, M., & Geroliminis, N. (2012). On the estimation of arterial route travel time distribution with markov chains. *Transportation Research Part B: Methodological*, *46*(10), 1576.

Rilett, L., & Park, D. (2001). Direct forecasting of freeway corridor travel times using spectral basis neural networks, 140.

Sayarshad, H. R., & Chow, J. Y. J. (2017). Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. *Transportation Research Part E-logistics and Transportation Review*, *106*, 60.

Sen, A., Thakuriah, P., Zhu, X., & Karr, A. F. (1997). Frequency of probe vehicle reports and variances of link travel time estimates. *Journal of Transportation Engineering, ASCE*, *123*, 290?297.

Sevilla, J., Heim, L., Hobbhahn, M., Besiroglu, T., Ho, A., & Villalobos, P. (2022). Estimating training compute of deep learning models.

Sherali, H. D., Desai, J., & Rakha, H. (2006). A discrete optimization approach for locating automatic vehicle identification readers for the provision of roadway travel times. *Transportation Research Part B: Methodological*, *40*(10), 857.

Siddiqi, Z., & Buliung, R. (2013). Dynamic ridesharing and information and communications technology: Past, present and future prospects. *Transportation Planning and Technology*, *36*(6), 479.

Simonetto, A., Monteil, J., & Gambella, C. (2019). Real-time city-scale ridesharing via linear assignment problems. *Transportation Research Part C: Emerging Technologies*, *101*, 208.

Soper, T. (2015). "lyft's carpooling service now makes up 50% of rides in san francisco; 30% in nyc". *"GeekWire"*.

Sperling, D. (2018). *Three revolutions: Steering automated, shared, and electric vehicles to a better future*. Island Press.

Srinivasan, K. K., & Raghavender, P. N. (2006). Impact of mobile phones on travel: Empirical analysis of activity chaining, ridesharing, and virtual shopping. *Transportation Research Record*, *1977*(1), 258.

Steg, L., & Gifford, R. (2005). Sustainable transportation and quality of life. *Journal of Transport Geography*, *13*(1), 59.

Tan, H., Xuan, X., Wu, Y., Zhong, Z., & Ran, B. (2016). A comparison of traffic flow prediction methods based on dbn. In *Cictp 2016* (pp. 273–283).

Tang, K., Chen, S., Liu, Z., & Khattak, A. J. (2018). A tensor-based bayesian probabilistic model for citywide personalized travel time estimation. *Transportation Research Part C: Emerging Technologies*, *90*, 260.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, *46*, 234.

Tsao, M., Milojevic, D., Ruch, C., Salazar, M., Frazzoli, E., & Pavone, M. Model predictive control of ride-sharing autonomous mobility-on-demand systems. In: *2019-May*. 2019, 6665.

Tyndall, J. (2017). Waiting for the r train: Public transportation and employment. *Urban Studies*, *54*(2), 520.

United Nations. (2018). Un dep. econ. soc. aff. 2021. 68% of the world population projected to live in urban areas by 2050, says un.

United-Nations. (2014). *World urbanization prospects*. United Nations.

U.S. Census Bureau. (2023). United States Census Bureau.

Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. *Transport Reviews*, *24*(5), 533.

Waisanen, H. A., Shah, D., & Dahleh, M. A. (2008). A dynamic pickup and delivery problem in mobile networks under information constraints. *IEEE Transactions on Automatic Control*, *53*(6), 1419.

Wang, H., & Yang, H. (2019). Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological*, *129*, 122.

Wang, H., Tang, X., Kuo, Y.-H., Kifer, D., & Li, Z. (2019). A simple baseline for travel time estimation using large-scale trip data. *ACM Trans. Intell. Syst. Technol.*, *10*(2).

Wang, Y., Winter, S., & Tomko, M. (2018). Collaborative activity-based ridesharing. *Journal of Transport Geography*, *72*(100), 131.

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, *129*(6), 664.

Wu, C.-H., Ho, J.-M., & Lee, D. (2004). Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, *5*(4), 276.

Xu, J., Rahmatizadeh, R., Bölöni, L., & Turgut, D. (2018). Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *19*(8), 2572.

Yang, H., Leung, C. W., Wong, S., & Bell, M. G. (2010). Equilibria of bilateral taxi–customer searching and meeting on networks. *Transportation Research Part B: Methodological*, *44*(8), 1067.

Yang, H., Wong, S., & Wong, K. (2002). Demand–supply equilibrium of taxi services in a network under competition and regulation. *Transportation Research Part B: Methodological*, *36*(9), 799.

Yang, H., & Yang, T. (2011). Equilibrium properties of taxi markets with search frictions. *Transportation Research Part B: Methodological*, *45*(4), 696.

Yang, J., Jaillet, P., & Mahmassani, H. (2004). Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, *38*(2), 135.

Yang, S., Ma, W., Pi, X., & Qian, S. (2019). A deep learning approach to real-time parking occupancy prediction in spatio-temporal networks incorporating multiple spatio-temporal data sources.

Yen, J. Y. (1970). An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*, *27*, 526.

Yeon, J., Elefteriadou, L., & Lawphongpanich, S. (2008). Travel time estimation on a freeway using discrete time markov chains. *Transportation Research Part B: Methodological*, *42*(4), 325.

Yu, X., Gao, S., Hu, X., & Park, H. (2019). A Markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological*, *121*(100), 114.

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., & Huang, Y. (2010). T-drive: Driving directions based on taxi trajectories. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 99.

Zardini, G., Lanzetti, N., Pavone, M., & Frazzoli, E. (2021). Analysis and control of autonomous mobility-on-demand systems: A review. *ArXiv*, *abs/2106.14827*.

Zeng, W., Miwa, T., Wakita, Y., & Morikawa, T. (2015). Application of lagrangian relaxation approach to α-reliable path finding in stochastic networks with correlated link travel times. *Transportation Research Part C: Emerging Technologies*, *56*, 309.

Zhan, X., Hasan, S., Ukkusuri, S. V., & Kamga, C. (2013). Urban link travel time estimation using large-scale taxi data with partial information. *Transportation Research Part C: Emerging Technologies*, *33*, 37.

Zhang, D., He, T., Lin, S., Munir, S., & Stankovic, J. (2017). Taxi-passenger-demand modeling based on big data from a roving sensor network. *IEEE Trans. Big Data*, *3*(3), 362.

Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016). Dnn-based prediction model for spatio-temporal data. *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

Zhang, R., Spieser, K., Frazzoli, E., & Pavone, M. (2015). Models, algorithms, and evaluation for autonomous mobility-on-demand systems. *2015 American Control Conference (ACC)*, 2573.

Zhang, Y., & Zhang, Y. (2018). Exploring the relationship between ridesharing and public transit use in the united states. *International Journal of Environmental Research and Public Health*, *15*(8).

Zheng, F., & Van Zuylen, H. (2013). Urban link travel time estimation based on sparse probe vehicle data. *Transportation Research Part C: Emerging Technologies*, *31*, 145.

Zhou, X., Shen, Y., Zhu, Y., & Huang, L. (2018). Predicting multi-step citywide passenger demands using attention-based neural networks. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*.

Zoepf, S., Chen, S., Adu, P., & Pozo, G. (2018). The economics of ride hailing: Driver revenue, expenses and taxes.

CURRICULUM VITAE

PERSONAL DATA

Name	Zahra Ghandeharioun
Date of Birth	21 March 1987
Place of Birth	Tehran, Iran
Citizen of	Iran, Germany

EDUCATION

2018 – 2024	Doctoral student *Swiss Federal Institute of Technology (ETH) Zurich* Institute of Transport Planning and Systems
2011 – 2014	Master of Science in Transportation Systems *Technical University of Munich* Department of Civil, Geo and Environmental Engineering, Munich, Germany
2005 – 2010	Bachelor of Science in Civil Engineering *K.N.Toosi University of Technology* Faculty of Civil Engineering, Tehran, Iran

PROFESSIONAL EXPERIENCE

2018 – Present	Research assistant *Swiss Federal Institute of Technology (ETH) Zurich* Zurich, Switzerland
2015 – 2018	Engineering consultant *BMW Group, Altran Germany S.A.S. & Co. KG,* Munich, Germany
2014 – 2015	Traffic engineer *Gevas Humberg and Partner,* Munich, Germany

2013 – 2014 Intern
BMW Group, Department of Research and Technology,
Munich, Germany

2013 – 2014 Student research assistant
Chair of Traffic Engineering, Technical University of Munich
Munich, Germany

PUBLICATIONS

Articles in peer-reviewed journals:

Ghandeharioun, Z. & Kouvelas, A. (2022). Link travel time estimation for arterial networks based on sparse GPS data and considering progressive correlations. *IEEE Open Journal of Intelligent Transportation Systems*, *3*, 679.

Ghandeharioun, Z. & Kouvelas, A. (2023). Real-time ridesharing operations for on-demand capacitated systems considering dynamic travel time information. *Transportation Research Part C: Emerging Technologies*, *151*, 104115.

Ghandeharioun, Z., Nobari, P. Z., & Wu, W. (2023). Exploring deep learning approaches for short-term passenger demand prediction. *Data Science for Transportation*, *5*, 19.

Conference contributions:

Ghandeharioun, Z., Rau, M., & Kouvelas, A. Travel time estimation for urban arterials based on origin-destination data and spatial correlations. In: *101th Annual Meeting Online, pp. TRBAM-22-01549, Washington, DC: Transport Research Board*. The National Academies of Sciences, Engineering, and Medicine. 2022.

Ghandeharioun, Z. & Kouvelas, A. Link travel time estimation with spatial correlations based on OD data. In: *21st Swiss Transport Research Conference (STRC 2021), Ascona, Switzerland Ascona: STRC, September 12–14*, STRC. 2021.

Ghandeharioun, Z. & Kouvelas, A. Online fleet management operations for on-demand capacitated ridesharing systems. In: *99th Annual Meeting of the Transportation Research Board (TRB 2020)*. The National Academies of Sciences, Engineering, and Medicine. 2020.

Rahimi, M., **Ghandeharioun, Z.**, Kouvelas, A., & Corman, F. Multi-modal management actions for public transport disruptions: An agent-based simulation. In: *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Heraklion, Greece, 2021, pp. 1-6*. 2021.

Ghandeharioun, Z., Bigi, F., Corman, F., & Kouvelas, A. Integrated multimodal network management: An agent based approach. In: *20th Swiss Transport Research Conference (STRC 2020) (virtual), Ascona, Switzerland Ascona: STRC, May 13-14, 2020.* STRC. 2020.

Ghandeharioun, Z. & Kouvelas, A. Providing real-time operational solutions for the on-demand capacitated ride sharing problem. In: *8th Symposium of the European Association for Research in Transportation (hEART 2019), Budapest, Hungary, September 4-6, 2019.* European Association for Research in Transportation. 2019.

Ghandeharioun, Z. & Kouvelas, A. Providing real-time operational solutions for the on-demand capacitated ride sharing problem. In: *Book of Abstracts: Workshop of the EURO Working Group on Vehicle Routing and Logistics optimization (VeRoLog), pp. 17-17,* VeRoLog. 2019.

Ghandeharioun, Z. & Kouvelas, A. Online fleet management for on-demand capacitated ride sharing problems. In: *19th Swiss Transport Research Conference (STRC 2019), Ascona, Switzerland Ascona: STRC, May 15-17, 2019.* STRC. 2019.