TUM School of Engineering and Design
Technische Universität München

Microscopic modeling, optimization, and demand estimation for autonomous mobility-on-demand ridesplitting

Moeid Qurashi

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:
Prof. Dr.-Ing. Klaus Bogenberger

Prüfer*innen der Dissertation:
1. Prof. Dr. Constantinos Antoniou
2. Assoc. Prof. Dr. Hai Jiang,
   Tsinghua University, China

Die Dissertation wurde am 24.06.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 20.02.2023 angenommen.

Abstract

The two concepts of Autonomous mobility-on-demand (AMoD) and ridesplitting have emerged recently, showing invaluable market potential. While the driverless AMoD vehicles could be easily shared, they seem to be less sustainable without being combined with a service concept like ridesplitting, which targets high ridesharing occupancy through dynamic matching and detouring to provide more affordable travel with lesser resources and impacts. However, such dynamic service startups have had both successes and failures, posing uncertainties for all their stakeholders, including the operators, policymakers, and service users. The failures are partially attributed to ridesplitting's operational complexity, sensitivity to information accuracy, and lack of robust modeling and assessment methods specific to the service characteristics. Therefore, this dissertation contributes to developing tools, methods, and experiments specific to the ridesplitting service concept. On the one hand, the developed robust models can help significantly improve service planning, operational management, and impact assessment. On the other hand, the empirical findings help better our understanding of the factors related to user adoption, sustainable operations, and social and environmental benefits.

Ridesplitting, as a transport mode, differs in its supply and demand representation. The core of the service supply lies with the scheduling algorithm that dynamically optimizes the service upon the known information related to the network, service operations, and demand. Therefore, the information accuracy of these aspects directly affects the service's modeling and operational efficiency. Considering the importance, this research develops a ridesplitting modeling framework that integrates a dynamic and stochastic scheduling algorithm with a microscopic traffic model, allowing a realistic representation of traffic congestion and service operations, and incorporates stochastic modeling information in service optimization. Microscopic models also allow modeling vehicle driving behaviors, autonomous technologies for AMoD, and detailed performance assessment metrics like vehicle-level emissions in a dynamic traffic environment. Similarly, robust dynamic demand estimation methods are also proposed to estimate the network demand and help improve the accuracy of traffic congestion modeling and seed demand to estimate the service requests. The developed methods explicitly solve the practical implementation problems for applying Principal Component Analysis to large-scale network models and leverage its properties for faster and more efficient estimation.

Regarding the demand aspects, ridesplitting is on-demand and operationally differs from other services. Therefore, this research conducts a stated-preference experiment for ridesplitting that help identify the factors affecting user travel behavior, preferences, and service adoption. Then, two different ridesplitting demand modeling methods are also proposed to model aggregated and time-dependent service demand. These methods allow easier adaptability for service planning and modeling and specifically cater to supply-

demand equilibrium requirements and ridesplitting-specific characteristics. Moreover, a utility-based compensation pricing method is also proposed to reduce the uncertainty and inequity experienced by ridesplitting passengers in their trip level of service. The pricing method helps improve the service profitability or user adoption and can also be a smart subsidy alternative.

While the experiments conducted on the case studies of Munich city affirm the efficacy of the proposed methods. The developed ridesplitting simulation platform is further utilized to explore and quantify the service performance and impacts over ranges of multiple supply and demand variables. The empirical results further help to develop plots and regression models to better understand the relation of passenger serviceability, ridesharing occupancy, and social and environment benefits with themselves and the experimental variables. These findings significantly contribute to better understand the factors to achieve sustainable operation levels, and higher service benefits.

Zusammenfassung

Die beiden Konzepte "Autonome Mobilität auf Abruf" (AMoD) und "Fahrtenaufteilung" sind in jüngster Zeit aufgetaucht und zeigen ein unschätzbares Marktpotenzial. Während die fahrerlosen AMoD-Fahrzeuge leicht gemeinsam genutzt werden könnten, scheinen sie weniger nachhaltig zu sein, wenn sie nicht mit einem Dienstleistungskonzept wie Fahrtenaufteilung kombiniert werden, das durch dynamisches Matching und Umwege eine hohe Auslastung der Mitfahrgelegenheiten anstrebt, um erschwinglichere Fahrten mit geringeren Ressourcen und Auswirkungen zu ermöglichen. Solche dynamischen Dienste haben jedoch sowohl Erfolge als auch Misserfolge aufzuweisen, was für alle Beteiligten, einschließlich der Betreiber, politischen Entscheidungsträger und Nutzer, Unsicherheiten mit sich bringt. Die Misserfolge sind zum Teil auf die Komplexität des Betriebs von Fahrtenaufteilung, die Sensibilität für die Genauigkeit der Informationen und den Mangel an robusten Modellierungs- und Bewertungsmethoden zurückzuführen, die speziell auf die Merkmale des Dienstes zugeschnitten sind. Daher leistet diese Dissertation einen Beitrag zur Entwicklung von Instrumenten, Methoden und Experimenten, die speziell auf das Fahrtenaufteilung-Konzept zugeschnitten sind. Einerseits können die entwickelten robusten Modelle dazu beitragen, die Serviceplanung, das Betriebsmanagement und die Folgenabschätzung deutlich zu verbessern. Andererseits tragen die empirischen Ergebnisse zu einem besseren Verständnis der Faktoren bei, die mit der Nutzerakzeptanz, dem nachhaltigen Betrieb sowie dem sozialen und ökologischen Nutzen zusammenhängen.

Das Fahrtenaufteilung als Verkehrsart unterscheidet sich in der Darstellung von Angebot und Nachfrage. Das Kernstück des Dienstangebots ist der Planungsalgorithmus, der den Dienst auf der Grundlage der bekannten Informationen über das Netz, den Dienstbetrieb und die Nachfrage dynamisch optimiert. Daher wirkt sich die Informationsgenauigkeit dieser Aspekte direkt auf die Modellierung und die betriebliche Effizienz des Dienstes aus. In Anbetracht dieser Bedeutung wird in dieser Forschungsarbeit ein Rahmen für die Modellierung von Fahrtenaufteilung entwickelt, der einen dynamischen und stochastischen Planungsalgorithmus mit einem mikroskopischen Verkehrsmodell integriert, das eine realistische Darstellung von Verkehrsstaus und Betriebsabläufen ermöglicht und stochastische Modellierungsinformationen in die Serviceoptimierung einbezieht. Mikroskopische Modelle ermöglichen auch die Modellierung des Fahrverhaltens von Fahrzeugen, autonomer Technologien und detaillierter Leistungsbewertungsmetriken wie Emissionen auf Fahrzeugebene in einer dynamischen Verkehrsumgebung. Ebenso werden robuste Methoden zur dynamischen Nachfrageschätzung vorgeschlagen, um die Netznachfrage zu schätzen und die Genauigkeit der Verkehrsstaumodellierung und der Seed-Nachfrage zur Schätzung der Serviceanforderungen zu verbessern. Die entwickelten Methoden lösen explizit die praktischen Implementierungsprobleme bei der Anwendung

der Hauptkomponentenanalyse auf große Netzmodelle und nutzen deren Eigenschaften für eine schnellere und effizientere Schätzung.

Was die Nachfrageaspekte anbelangt, so ist Fahrtenaufteilung ein On-Demand-Service und unterscheidet sich in seiner Funktionsweise von anderen Services. Aus diesem Grund wird in dieser Forschung ein "stated-preference"-Experiment für Fahrtenaufteilung durchgeführt, das dazu beiträgt, die Faktoren zu identifizieren, die das Reiseverhalten der Nutzer, ihre Präferenzen und die Annahme des Dienstes beeinflussen. Anschließend werden zwei verschiedene Methoden zur Modellierung der Fahrtenaufteilung-Nachfrage vorgeschlagen, um die aggregierte und zeitabhängige Service-Nachfrage zu modellieren. Diese Methoden ermöglichen eine leichtere Anpassbarkeit für die Planung und Modellierung von Diensten und gehen speziell auf die Anforderungen des Gleichgewichts zwischen Angebot und Nachfrage sowie auf Fahrtenaufteilung-spezifische Merkmale ein. Darüber hinaus wird eine nutzwertbasierte Kompensationspreismethode vorgeschlagen, um die Ungewissheit und Ungerechtigkeit zu verringern, die Fahrgäste bei der Inanspruchnahme von Fahrtenaufteilung-Dienstleistungen erleben. Die Preismethode trägt dazu bei, die Rentabilität des Dienstes oder die Nutzerakzeptanz zu verbessern und kann auch eine intelligente Subventionsalternative darstellen.

Die Experimente, die an den Fallstudien der Stadt München durchgeführt wurden, bestätigen die Wirksamkeit der vorgeschlagenen Methoden. Die entwickelte Fahrtenaufteilung Simulationsplattform wird weiter genutzt, um die Leistung des Dienstes und die Auswirk- ungen über verschiedene Angebots- und Nachfragevariablen hinweg zu untersuchen und zu quantifizieren. Die empirischen Ergebnisse helfen bei der Entwicklung von Diagrammen und Regressionsmodellen, um die Beziehung zwischen der Servicequalität für die Fahrgäste, der Fahrgemeinschaft-Auslastung und den sozialen und ökologischen Vorteilen mit sich selbst und den experimentellen Variablen besser zu verstehen. Diese Ergebnisse tragen wesentlich zum besseren Verständnis der Faktoren bei, die einen nachhaltigen Betrieb und einen höheren Nutzen des Services ermöglichen.

Contents

List of Figures

List of Tables

1 Introduction

Contents

Autonomous mobility-on-demand (AMoD) and ridesplitting are emerging to be part of future urban mobility, receiving attention among academics, industry, and policymakers. Their sharing nature promise to reduce traffic and environmental imprints alongside discouraging car ownership and improving passenger affordability. However, being in early-stage realization, they still have lesser established practicability. Therefore, this dissertation focuses on developing suitable supply and demand modeling methods specific to AMoD ridesplitting.

This chapter introduces the context of this thesis, discusses the three main aspects of supply, demand, and model calibration in prospective of AMoD ridesplitting, and states the objectives, contributions, and the outline of this dissertation.

1.1 Background and context

1.1.1 Prospects of on-demand mobility

Since the mass motorization in the early twentieth century, the mobility infrastructure has remained car-dominated. The continuous domination made people tend to live and adore the flexibility provided by private transport/cars. However, the ever-growing population and its need for motorized traffic eventually pushed the infrastructure to its limits creating traffic congestion problems worldwide. Road transport also had a consistently highest share against all other transports to adversely affect the environment (accounting for 72% of all transport emissions pre-covid pandemic (EEA, 2022)). One of the primary factors behind this has been a continuous increase in car ownership. Therefore, transport planners soon realized that even the increase in infrastructure supply could not help mitigate these congestion problems and shifted their focus towards minimizing the impacts of motor traffic on urban living. Although policies such as better public transport development, tolls/taxes, and parking tickets had been used to discourage car ownership and usage. It could not stop cars from becoming a centric part of modern human lives because nothing yet could replace the flexibility offered by private vehicles. With the advancements in IT, communication, and vehicle automation, things are promised to change. New smart mobility concepts are emerging worldwide with prospects of more sustainable transport alternatives which can provide that on-demand flexible travel experience and shift transport choices from owned vehicles to a service used on-demand.

On-demand services, like taxis and paratransit, have long existed as a transport mode, with rider operations limited by reservations or on-spot presence (street taxis). However, the advancement in the information technology and communication industry has drastically changed on-demand transport around the globe, resulting in emergence of many on-demand service concepts within the last decade, i.e., car-sharing (car2Go, ZipCar, DriveNow), ride-hailing (Uber, Lyft, Careem), ride-pooling/ride-splitting (Uberpool, Lyft line, DiDi-Pool), and vanpooling/micro-transit (Jetty, Panda Bus, DiDi-Hitch, DiDi-Minibus). These service concepts fill the broad spectrum between fixed public transport and private cars (Currie and Fournier, 2020) and provide much better communication flexibility, safety (driver knowledge, ride–tracking), affordability, and on-door pickup options over conventional taxis. Uber and Didi are the two most popular transport network companies (TNCs) offering various ridesourcing services. Uber operates in 700 metropolitan areas worldwide with 15 million daily trips, and DiDi operates in 400 cities in China with 30 million rides per day (DMR, 2022a,b). Among different service types, ride-hailing has been the most common and popular service, offering private rides similar to taxis. However, from a policy perspective, they may effectively reduce car ownership, but waste human and vehicle resources and further induce traffic congestion and emissions due to empty vehicle hours spent or kilometers traveled. Therefore, ridesharing service types are also emerging and receiving attention among the academics, industry, and governments. There sharing nature also promises to reduce traffic and environmental imprints alongside discouraging car ownership and

improving passenger affordability. For instance, DiDi already has almost 19% of its average daily trips (i.e., 95,454 trips) served by ridesplitting (a ridesharing service type with dynamic matching), which results in saving around 58,000 Vehicle Kilometers Traveled (VKT), considering mode shifts from all active modes, including public transport (Chen et al., 2021).

Recent growth in vehicle automation has also given ride-sourcing much more popularity and interest from both research and industry (Narayanan et al., 2020; Fagnant and Kockelman, 2015). Autonomous vehicles (AVs) aroused the unification of all vehicle-based on-demand service concepts into a single concept of Autonomous Mobility on Demand (AMoD) or Shared Autonomous Vehicles (SAVs). Being driverless, AVs are anticipated to be shared, easily accessible (like private vehicles), and affordable (Fagnant and Kockelman, 2015), which could help reduce car–ownership and its related issues. Furthermore, connected AV technologies can also significantly improve the capacity of existing traffic infrastructure, e.g., by platooning (Shladover et al., 2012), coordinated intersection and ramp control, and traffic flow stabilization. The operational spectrum of AMoD services can also range between AMoD car-sharing and automated buses. Therefore, the AMoD systems can have a mix of different vehicle capacities (buses, vans, taxis), service stops flexibility (many-to-many/few/one), routing optimization (pre-route/static or in-route/dynamic), and demand assignment (reservation-based or on-demand/ real-time) (Hyland and Mahmassani, 2017; Narayanan et al., 2020).

Recognizing the potential in AMoD, the automotive industry is making huge investments in developing driverless cars to operate AMoD systems with new business models. However, its success and competitiveness highly depend on its operational actuality and impacts, which are still uncertain (Stocker and Shaheen, 2018; Firnkorn and Müller, 2012). Many studies investigate the possibility and impacts of using AMoD as an alternative to private vehicles (see chapter 2). Among these, AMoD car-sharing is the most studied (comparable to private car experience). However, as anticipated, most studies concluded that AMoD car–sharing has significantly high empty vehicle kilometers, resulting in 10 to 15% higher VKT than the use of private vehicles, deeming it unsustainable alone (Narayanan et al., 2020). Therefore, similar to the ridesourcing industry, the focus has been shifted towards adding ridesharing in the mix, where although fewer studies are available, they suggest that both conventional and AMoD ridesharing services are more sustainable (Narayanan et al., 2020).

The ridesharing service types include ride-pooling, ridesplitting, and microtransit (Shaheen and Cohen, 2019; Wang and Yang, 2019). Ridepooling is an extension of ridehailing (AMoD car-sharing), where riders having similar itineraries pool rides without experiencing much detours. In comparison, both ridesplitting and the agile form of microtransit (i.e., with flexible routing and scheduling, also named dynamic vanpooling (Li et al., 2019a)) serve riders with similar travel directions and dynamically matchable itineraries together through dynamic matching and detouring. These two types further differ by vehicle ownership, where dynamic vanpooling/microtransit specifically represents high occupancy ridesharing with vans primarily owned by service operators, whereas ridesplitting is more associated with ridesourcing services (Uberpool, DiDiExpress, DiDiHatch) representing different ridesharing occupancies with freelance drivers

having personal vehicles (Shaheen and Cohen, 2019; Wang and Yang, 2019). Since AMoD vehicles are driverless, both 'AMoD ridesplitting' and 'AMoD vanpooling' are justified; however, since ridesplitting is a more commonly used term in the literature, we also use the term 'AMoD ridesplitting' throughout the document to refer to high occupancy ridesharing with AV fleets (further details are present in section 2.1.1).

1.1.2 Needs for AMoD ridesplitting

AMoD ridesplitting has both less required operational resources and environmental impacts against AMoD car-sharing, while still providing similar flexibility. Therefore, they are perceived as a more sustainable future transport alternative. However, many of the current similar practical implementations have also failed to sustain. Currie and Fournier (2020) reported that, so far, over 70% of such modern era (2010-2019) ridesharing or microtransit service startups failed within three years, where more complex services with flexible service stops (many-to-many) were prone to higher failures, concluding that such systems are highly experimental and unreliable. One of the primary reasons for these failures is that operators must cater to service operational complexity and its sensitivity towards the involved dynamic and stochastic information for demand and supply (Wang and Yang, 2019). The agile microtransit operators charge much less than other ridesourcing services and price the trip per unit distance while aiming to establish sufficient ridesharing occupancy through detours for sustainable/profitable operations. Therefore to generate better value from the service, the operators require to:

1. have a competitive ride offer to attract a minimum critical demand that can trigger high occupancy ridesharing (which attribute to riders' perception towards service usage, trip travel times, and pricing)

2. optimize vehicle routing and request assignment to efficiently assign multiple requests to a vehicle without violating the riders' trip time preferences (by detours) under the given dynamic and stochastic information. The optimization must consider multiple objectives including, multiple riders' waiting time and detours, total travel distance, vehicles' occupancy, total served orders, service revenue and profit (Wang and Yang, 2019).

3. adequately model the dynamic and stochastic information of network state, service operational times, and demand to assist efficient optimization.

Since the AMoD ridesplitting operational complexity far exceeds other ridesourcing concepts due to dynamic matching and detouring. It requires more sophisticated tools to better plan, manage, or assess such a service. The foremost requirement is solving the service routing or demand assignment problem (referred as Dial-a-Ride problem - DARP), which directly affects the service efficiency and is therefore well focused in literature (Molenbruch et al., 2017; Ho et al., 2018). However, the optimization efficiency is highly sensitive to accuracy in demand and supply information, for which Ho et al. (2018) argues that any on-demand service operation is stochastic by nature (i.e.,

even pre-planned reservation trip has different expected versus actual service times), and the stochasticity exists in all aspects (i.e., service demand, operations, and network conditions). Therefore, efficient service modeling and optimization requires true representation of this stochasticity. Yet, most literature lack in robust service modeling methods and instead either focus on providing more robust service optimization solutions considering unrealistic congestion representation (Molenbruch et al., 2017; Ho et al., 2018) or model large–scale AMoD service scenarios with simplistic optimization (due to NP-hardness of the problem) (Levin et al., 2017; Narayanan et al., 2020), primarily due to the far exceeding modeling complexity and computational requirements for a comprehensive tool. Similarly, on-road service operations and vehicle (autonomous) driving behavior are seldom modeled. Note that although such a detailed modeling tool is complex and resource-hungry, simplistic models limit the accuracy of predicted benefits associated with AMoD services (mostly over–estimating (Levin et al., 2017)) and affect the assessment of ridesplitting business models.

Attracting minimum critical demand that triggers high occupancy ridesharing is crucial for sustainable ridesplitting operations. Therefore, ridesplitting demand modeling methods are also necessary for planning and assessment. Foremost is to understand how travelers perceive using ridesplitting services, i.e., the value of waiting, walking, and riding times, in the competition of other transport modes. For example, since the ridesplitting service serve riders with dynamic ride-matching and detouring, they experience uncertainty and inequity in their trip service, affecting their service perception and adaptability. Next come service demand estimation procedures that require evaluating equilibrium states between attracted demand and expected serviceability with any change in exogenous variables (e.g., service supply, pricing). Literature efforts generally employ market equilibrium methods to model static demand or market states for different on-demand services (Yang and Wong, 1998; Wong et al., 2001); however, no such methods exist that model the uniqueness of ridesplitting concept, i.e., dynamic ride-matching and detouring. For dynamic demand estimation, which can allow modeling of discrete and time-dependent rider requests, literature efforts generally employ agent-based frameworks (Basu et al., 2018; Horni et al., 2016) that already require an iterative simulation setup and therefore are feasible for evaluating service equilibrium states. However, such iterative methods are infeasible for service modeling tools that already model complex and computationally expensive DARP optimization with stochasticity modeling. It is also noteworthy that potential also exists in developing strategies that can help manage and optimize ridesplitting demand attraction (e.g., dynamic pricing).

Finally, model calibration is another crucial aspect that allows representation of realistic network and demand situations. Since AMoD ridesplitting is considerably sensitive to changes in traffic states and passenger demand (critical demand attraction), any bias present in modeling information can highly influence the service performance and impact assessment. Traffic model calibration is a well established research field and it focuses on calibrating model parameters to match the model outputs with observed network conditions. Among other model parameters, demand parameters are more crucial to estimate, because they are much more dynamic and influential, i.e., network demand changes more frequently, while wrong demand patterns can generate both biased network congestion

states and biased AMoD passenger demand. Demand estimation, known as Dynamic Demand Estimation Problem (DODE), searches for time-dependent OD demand matrices able to best fit measured traffic data. The main problem for DODE is that it is typically an unobservable and highly under–determined problem. Hence, most literature efforts struggle, computationally, to calibrate good quality DODE solutions, especially for large–scale networks. Since AMoD services are sensitive to any bias in information, their modeling and management frameworks would require frequent demand calibration iterations for dynamically updating demand and network states. Therefore, any enhancements, which can improve current DODE methods for their solution quality and computational efficiency are also desirable.

1.2 Modeling and optimizing AMoD ridesplitting services

Modeling and optimizing AMoD ridesplitting requires representing three main aspects: supply, demand, and model calibration. Since it utilizes road networks, its supply requires modeling traffic and vehicle driving characteristics. However, being dynamic and on-demand, it also needs modeling characteristics specific to its service concept, particularly the dynamic rider matching and vehicle routing based on service optimization algorithms. Furthermore, the AMoD demand modeling is also unique since the demand attraction is dynamic/stochastic, depending on the (dynamic) service attributes specific to each passenger. Similarly, the third aspect of model calibration that aims to estimate the model parameters is also crucial, especially DODE, for being more dynamic and influential on traffic congestion patterns and AMoD demand.

1.2.1 Supply modeling

On-demand service supply is modeled by three main aspects, i.e., fleet characteristics, network characteristics, and service optimization. Fleet characteristics mainly describe available passenger capacity (i.e., fleet size, vehicle capacity) and fleet vehicles' operational behavior (e.g., automation, driving behavior, boarding/alighting), while the road network representation includes network geometry and traffic/fleet propagation (e.g., by link speeds or microscopic traffic assignment). Similarly, service optimization is the continuous control of the fleet movement by the routing algorithm that optimizes fleet routes by dynamic riders' matching upon available demand and network information.

Service optimization

On-demand service optimization are generally termed as 'Dial-A-Ride Problem' (DARP) with two main aspects: operator's decision–making and known information (Ho et al., 2018). Operator's decision–making accounts for service optimization and includes vehicle routing, deployment, relocation, and demand requests assignment. Decisions are made either static (pre-defined) or dynamic (en-route), where dynamic DARPs relate to on-demand services, optimizing the vehicles during operation. The optimization is triggered either at specific time intervals, events (i.e., request arrivals, vehicle service stop),

or network conditions/incidents. Furthermore, dynamic DARPs also differ by considering optimization of vehicle routing and request assignment separately (e.g., AMoD car–sharing, ride–hailing) or combined (e.g., AMoD ridesplitting). Note that, dynamic DARP for AMoD ridesplitting is the most complex DARP. It finds vehicles and their optimum routes to assign new passengers while adding detours in in-service riders, under many constraints, e.g., time preferences of all riders, network routing and congestion.

The 'known information' aspect of DARP systems account for modeling certainty in information related to service demand, operations, and traffic network. Classified as either deterministic or stochastic DARPs, the difference among the two approaches is incorporating the stochasticity of future information (Hyland and Mahmassani, 2017), i.e., the dynamic-stochastic approaches cater to, e.g., future user requests and stochastic travel times, but dynamic-deterministic approaches mostly stick with real-time request arrivals for vehicle reassignment. As mentioned earlier, Ho et al. (2018) argued that all real-world DARP operations are stochastic and the stochasticity exists in all information aspects. Since AMoD ridesplitting is also highly sensitive to any change in information (considers multiple riders' time preferences as constraints), dynamic and stochastic DARPs fit better for it considering information stochasticity. Meanwhile, the service modeling should cater to replicate the stochasticity as in reality and integrate the DARP algorithm to directly use the stochastic/realistic information for formulating the optimization problems.

AMoD service modeling

AMoD ridesplitting requires modeling the vehicle attributes, driving behavior, service operations, and dynamic routing. Service vehicle attributes include fleet size, vehicle capacity, and vehicle type (dimensions, operational speeds), which vary by targeted ridesharing occupancy, e.g., taxis and vans. Similarly, the driving behavior modeling for AMoD includes replicating the autonomous driving interactions of service vehicles with the surrounding environment. This area of modeling autonomous driving is also relatively wide with different driving behavior models (e.g., ACC-CACC (Milanés and Shladover, 2014) and Wiedermann 99 (Sukennik et al., 2018)) and connectivity concepts (e.g., platooning with connected AVs, traffic signal coordination/prioritization), and is an active field of research. AVs can help increase the traffic capacity and stabilize traffic flows, and improve AMoD ridesplitting performance and operations by, e.g., platooning, signal/ramps coordination, high occupancy prioritization (Levin et al., 2017). Among the modeling literature, only a few studies (Alazzawi et al., 2018; Levin et al., 2017) focus on modeling and exploring autonomous driving behaviors and potential connected technologies for AMoD, mainly due to the lack of using microscopic traffic models.

Modeling AMoD service also requires replicating its operations, including boarding/alighting passengers at flexible origin/destination locations and idle waiting behavior. As with modeling autonomous behavior, very few modeling works model on-road service operations, i.e., only Alazzawi et al. (2018); Huang et al. (2021); Ronald et al. (2017) model on-demand services in a (link–level) microscopic traffic simulator SUMO (capturing interaction with network traffic), while all other studies (including MATSim (Bischoff

and Maciejewski, 2016; Hörl, 2017) and Simmobility (Basu et al., 2018)) simulate node-level operations with service times approximation. Such approximations lack both the service operation stochasticity and the effects of varying network and traffic characteristics, limiting the efficacy of service optimization and assessment. However, microscopic traffic models can better model the network and service operation stochasticity, but at the expense of significantly increased computational efforts that limit the scalability of service scenarios. Further, dynamic routing is another service characteristic, which needs modeling of consistent control on all service vehicles to update their routes during the simulations dynamically upon dynamic rider matching or network conditions. Note that, it requires integrating the service optimization along with the simulation to provide continuous communication of the service operations and traffic information to the optimization algorithm and receive optimized vehicle routing decisions.

Traffic congestion modeling

The road network and its traffic congestion patterns also define the supply of an AMoD system. They represent the (point–to–point) vehicle routing attributes, i.e., distance and travel times. The traffic congestion modeling mainly variates by two degrees of detail to represent network information. The first degree of detail is directly using travel time information, which is widely seen in literature, e.g., fixed travel time information is used by many DARP solution approaches (Molenbruch et al., 2017) and AMoD modeling methods (Chen et al., 2016; Fagnant and Kockelman, 2014; Zhang et al., 2015)). However, since the use of fixed travel times lacks any form of dynamic traffic information, some DARP researchers cater to it by considering time-dependent travel times (Xiang et al., 2008; Schilde et al., 2014; Li et al., 2019a) or even additionally future stochastic travel time information (Schilde et al., 2014; Li et al., 2019a). Dynamic Traffic Assignment (DTA) models are the second degree of detail to model network information. These models represent state of the art in modeling transport systems, as they provide a realistic representation of the congestion and a wide range of time-varying outputs, such as queue length, route costs, and travel times. They simulate traffic conditions by dynamically assigning traffic demand in time intervals and are distinguished into three levels that vary by the modeling detail of traffic flow and route choice, i.e., macroscopic, mesoscopic, and microscopic. Among these, microscopic models are the most detailed, modeling the traffic assignment with dynamic route choice and individual vehicle driving behaviors to generate detailed traffic dynamics throughout the network.

Most current modeling efforts that use DTA utilize the mesoscopic models (e.g., Bischoff and Maciejewski (2016); Basu et al. (2018); Fagnant and Kockelman (2018); Hörl (2017)) with the aim to run large traffic networks with high demands faster, mostly simulating node–level traffic flow without link based vehicular modeling and interactions (Ronald et al., 2017). Only a few efforts focus on modeling detailed network characteristics with microscopic models (Alam and Habib, 2018; Alazzawi et al., 2018) or cell transmission models (Levin et al., 2017). Similar to service behavior modeling, microscopic models are also best suited to model network characteristics for AMoD systems. They represent the detailed, dynamic, and stochastic traffic congestions to better repli-

cate real network conditions, especially useful for AMoD ridesplitting which is more sensitive to changes in network conditions.

1.2.2 Demand modeling

On-demand mode choice modeling is more complex than conventional transport modes, due to the stochastic service availability and uncertain trip utilities, requiring iterative supply-demand interaction (Liu et al., 2019). Therefore, most of the literature either use real-world datasets or assumes constant demand shares, apart from a few studies (e.g., Basu et al. (2018); Hörl et al. (2016); Liu et al. (2019)) that consider endogenous mode choice modeling with iterative supply-demand interactions. The supply-demand interaction is also modeled analytically using market equilibrium models, representing aggregated level demands or market states. Keeping demand estimation aside, estimation of user's preferences and total travel demand (discussed in section 1.2.3) is also necessary. Moreover, pricing is another crucial aspect since pricing strategies can help manage demand attraction for on-demand services (more crucial for ridesplitting to attain the required critical mass that triggers adequate ridesharing). Furthermore, in dynamic ride-matching and detouring, ridesplitting riders also experience uncertainty and inequity in their trip level of service, influencing service adaptability. Therefore, potential dynamic pricing strategies for compensation are also of need.

Passenger preferences

Ridesplitting differs in terms of operations and the sense it creates for riders. Especially the dynamic ride-matching and detouring for high occupancy ridesplitting, riders explicitly define the waiting and arrival time limits and expect dynamic detours, unique from all other ridesourcing services. Therefore, it is essential to scale how users perceive ridesplitting and understand the specific factors that influence its use by potential customers, especially considering that many of these services fail to become established (see, e.g., Bridj, Via, Kutsuplus, and others reported in Currie and Fournier (2020)). However, none of the existing research works on ridesharing user preferences includes high capacity ridesplitting (or dynamic vanpooling) as a discrete alternative in mode choice (and stated preferences) studies (Alonso-González et al., 2020; Frei et al., 2017; Kang et al., 2021; Lavieri and Bhat, 2019). Note that the user preferences for shared-ride services are said to be mainly affected by monetary cost (trip fare) and time cost (waiting and travel time) (de Ruijter et al., 2020; Qiu et al., 2018) and the preference coefficients are an essential input for, e.g., modeling the competition between ridesplitting and other transport modes, and developing and assessing demand management strategies like dynamic pricing.

Demand estimation

On-demand service supply has an intertwined relationship with attracted demand and requires equilibrium states estimation, in which the expected rider trip attributes should converge with the experienced trip attributes of the attracted demand. Note that, in

ridesplitting, the demand influence of on trip attributes is even more inclusive due to the possibility of ridesharing with dynamic ride-matching and detouring, where the trip waiting and detour times remain uncertain until being picked up and dropped off. Thus, modeling ridesplitting demand additionally requires the representation of dynamic trip utilities subject to service availability and utilization.

To model the equilibrium, one method is market equilibrium (ME) modeling that represent static or aggregated on-demand service markets. It analytically balance simultaneous supply-demand equations to represent aggregated market with much lower computational expense than using time-dependent models like DARP optimization models and traffic simulators. ME models allow the service providers to explore their operational strategies on a much wider range of demand and supply scenarios. Note that, since the current ridesplitting ridership is relatively low (Tu et al., 2021; Li et al., 2019c), a ridesplitting ME model can help transport management agencies and ridesourcing companies tailor the service, adapting to the market preference (Li et al., 2019c). However, among the few ridesourcing ME models in the literature (e.g., Bimpikis et al. (2019); Ke et al. (2020); Yang et al. (2002)), none caters to ridesplitting specific characteristics, which require network level representation that can reflect the network geometry and OD demand patterns across the network (Bimpikis et al., 2019).

The time-dependent modeling of ridesplitting also requires attaining supply-demand equilibrium states by iterative simulations (Liu et al., 2019). The mode choice decisions are case-specific for each rider due to the stochasticity (and uncertainty) in both service availability and potential trip utility. Therefore, literature efforts generally employ agent-based models, because they model individual agents with dynamic (activity-based) mode choice and require iterative simulations to damp changes in agent choices (departure, modes) to attain user equilibrium (Basu et al., 2018; Hörl et al., 2016). However, the disaggregated nature of these models also adds a lot of complexity and stochasticity (Wegener, 2011), especially for simultaneously calibrating all demand aspects (Moeckel et al., 2020), or causing significant variations in model outputs (Wegener, 2011). Since agent-based models lack in well-established methods, traffic models generally employ trip-based demand modeling which represents aggregated (zone-level) demand with Origin-Destination (OD) matrices. Being aggregated, OD matrices are relatively simpler to calibrate and conduct mode choice. However, suitable trip-based demand estimation methods are not available for ridesplitting which can allow easier adaptability in current traffic models. Further, running iterative equilibrium is rather infeasible for already complex and computationally expensive modeling tools like microscopic modelds with DARP optimization. Therefore, non-iterative or non-simulation based demand estimation methods can allow better use of complex service modeling tools for planning and assessment.

Pricing

Dynamic pricing strategies are helpful to manage demand attraction for on-demand services, especially for those with ridesharing, to attract critical demand that triggers enough ridesharing for profitable operations. Working with passenger preferences

(mainly affecting monetary and time costs), the operator can influence the adaptability of the service (Qiu et al., 2018; Guan et al., 2019a). At a minimum, unified pricing rates can be optimized given the fixed market exogenous variables, i.e., available service supply, network demand, and attributes from the competitive modes. However, in ridesharing literature, efforts are also available that consider both temporal elasticity and spatial heterogeneity of the demand to optimize demand attraction for specific objectives, e.g., maximizing profits (Sayarshad and Chow, 2015; Qian and Ukkusuri, 2017; Chen and Kockelman, 2016; Qiu et al., 2018). Note that, for ridesplitting, dynamic matching and detouring result in uncertainty and inequity among the riders for their trip level of service (LoS). Studies like Guan et al. (2019a,b) focused on applying cumulative prospect theory (CPT) to capture the influence of uncertainty on rider decision making and propose subsequent dynamic pricing methods. However, none of the available pricing strategies address the inequity problem common for ridesplitting. Therefore, practical compensation methods that can compensate individual trips based on their experienced LoS are desirable to promote both equity and certainty for using ridesplitting services.

1.2.3 Model calibration

DTA model calibration is a well-researched topic with literature focusing to increase scalability towards large-scale DTA models (Balakrishna, 2006; Antoniou et al., 2009, 2015). The calibration problem is extremely complex, due to high non–linearity and estimation of a large set of parameters (Marzano et al., 2009). Among others, mobility demand is an essential DTA input, as biased demand pattern will obviously lead to biased congestion patterns. However, the main problem for calibrating demand is that it is unobservable (Frederix et al., 2011) and practitioners usually turn to demand generation models to estimate it (McNally, 2007). Although these models provide an initial guess, the estimated OD matrix is at most an average demand approximation and daily demand patterns show substantial fluctuations for it (Balakrishna, 2006). Correcting these deviations is known as the Dynamic Demand Estimation Problem (DODE) that searches for time-dependent OD demand matrices able to best fit measured traffic data (Cascetta and Postorino, 2001). Calibrating other model parameters from supply and route choice models is much simpler because they are less in number and not as dynamic as the demand. Apart from its significance in model calibration (to correct time–varying network dynamics), DODE is also an important input for estimating AMoD service demand as discussed in 1.2.2.

The complexity of DODE is based on the amount of disaggregation (time intervals for estimation/prediction or ODs), network size and complexity, and available data. As DTA models are highly non-linear, the DODE complexity rapidly increases with the size of the transport network. This complexity mostly translates into three critical issues that need to be jointly considered:

- **Highly underdetermined system:** The number of variables to be calibrated usually far exceeds the number of traffic measures used to estimate them. This

results in an underdetermined system of equations with fewer equations than unknowns.

- **Non–linearity issues:** Congestion dynamics are non-linear by nature. Mapping OD pairs to traffic measurements is therefore also a non-linear process, with an additional layer of complexity, due to the stochastic nature of the demand and the route choice/traffic assignment model.
- **The curse of dimensionality:** As the model/network size increases, the number of estimation variables (OD pairs) also increase. For the increasing number of variables, the objective function becomes more complex and non–linear, a problem that makes the DODE both computationally demanding and highly inefficient.

Considering these DODE characteristics, many solutions exist in the literature (see chapter 2). However, most conventional algorithms, often fail to converge for large–scale networks, because their performance deteriorates rapidly with the increase of the problem scale and complexity. For example, 'Simultaneous Perturbation Stochastic Approximation' (SPSA) (Spall, 1998a) is one of the most popular algorithms for DTA model calibration (Balakrishna et al., 2007a), but with increase in network size, its gradient approximation gets highly sensitive against the definition of hyperparameters (expensive objective functions making trial-based setup infeasible) and the variation of OD magnitudes also increases. Therefore, to calibrate large-scale networks, focus shifted to increase the application scalability of conventional calibration approaches. The extensions either focus on reducing the problem dimensions or reducing problem non–linearity (adding structural/correlation information in the objective function). Within all such efforts, the application of PCA stood out for being significantly more efficient in reducing problem dimensions and non-linearity, and therefore, the application of Principal Component Analysis (PCA) has been widely adopted for many calibration approaches to do dimension reduction (Qurashi et al., 2022).

In PCA–based OD estimation, PCA leverages strong patterns and correlations (extracted from a series of historical estimates) to represent the problem with a few orthogonal/uncorrelated Principal Components (PCs) in a low dimensional space. It strongly relies on the presence and quality of the historical estimates, since it reduces the algorithm's search space to the variance present in the historical estimates. Therefore, the application and performance of PCA–based methods rely on the presence and quality/relevance of the historical data–set (relative to the target solution). Since conventional calibration techniques struggle to calibrate large–scale networks, these historical estimates are usually not available in practice, triggering a chicken and egg problem, where to use PCA–based models there is a need for historical estimates, which can only be obtained by calibrating the network. Similarly, although many PCA–based methods have been proposed, less focus has been given to explore the application properties of PCA for calibration large–scale DODE. For example, does the dimension reduction property of PCA not get directly affected by the problem dimensions, but by the variance present in historical estimates? (as mentioned by Qurashi et al. (2019)), or what is the effect of increasing the size or amount of variance present in historical estimates?,

or does PCA help ease the DODE formulation and setting up estimation algorithms like SPSA?. Answering these questions can help establish better implementation methods for PCA-based approaches.

1.3 Research scope and objectives

Ridesplitting services are emerging around the globe with both successes and failures, showing invaluable market potential and gaining attention from both academia and industry. However, despite the popularity, ridesplitting expansion is hindered due to the present stochasticity and uncertainty in the service for all stakeholders, including the operators, policymakers, and service users. Practitioners still require a deeper understanding of many service-related aspects, e.g., for policymakers: i) can the ridesharing nature help solve the urban congestion problems by reducing excess traffic volumes and emissions? ii) how would ridesplitting affect existing transportation systems? iii) is high occupancy ridesplitting fruitful, i.e., what are its costs versus benefits? iv) what are the prospects of autonomous (AMoD) ridesplitting?

Similarly, for operators planning and managing the service is rather complex. Therefore the current ridesplitting ridership is relatively low (Tu et al., 2021; Li et al., 2019c), and more than 70% similar startups failed to establish (Currie and Fournier, 2020). It requires both stronger models and efficient operational strategies specific to the service characteristics to better plan and manage. For service users, ridesplitting is unique from other ridesourcing services. Due to dynamic ride-matching and detouring, on the one hand, they perceive uncertainty and inequity, while on the other, they find a service most affordable among others with equal flexibility. For practitioners, it is also important to understand how users perceive this uniqueness.

Following these findings and the above discussions on different modeling aspects for AMoD ridesplitting, this research aims to advance the ridesplitting specific methods to improve its modeling, operations, and assessments. Specifically, the following research objectives are formulated:

- develop a comprehensive modeling method for AMoD ridesplitting that caters to modeling stochasticity in network and service operations and dynamic-stochastic DARP optimization;
- develop efficient demand estimation methods for both the network model and ridesplitting service to help improve service modeling and assessment;
- understand user preference towards ridesplitting and develop appropriate operational strategies that can help improve ridesplitting adaptability; and
- adopt a simulation-based experimental setup to check the efficacy of the developed methods, explore ridesplitting service performance and related effects, and understand the key relationships between service characteristics, benefits and related impacts.

1.4 Dissertation contributions

This thesis explores supply and demand modeling methods specific to ridesplitting services, bringing in the following scientific and practical contributions.

1. **Scientific contributions:**
    a) *Microscopic modeling and optimization of autonomous mobility-on-demand (AMoD) ridesplitting:* AMoD ridesplitting is the most dynamic among all types of on-demand ride services due to dynamic ride-matching and detouring. To operate the service requires extensive routing and demand assignment optimization, which is sensitive to the quality (and change) in any network, service, or demand information. Therefore, this research develops a comprehensive modeling framework to model AMoD ridesplitting in microscopic models with integration of dial-a-ride (DARP) optimization (presented in chapter 3). The developed tool allows detailed modeling of network dynamics, link-based service operations, and incorporation of the stochastic network, service, and demand information in service optimization, resulting to efficient modeling and optimization of AMoD ridesplitting operations.
    b) *Dynamic demand estimation using Principal component Analysis:* Principal Component Analysis (PCA) is establishing itself as the new state-of-the-art to tackle the dimensionality and non–linearity issues of calibrating large-scale traffic models. PCA application limits the optimization search space to lower dimensions defined by orthogonal Principal Components, evaluated upon a set of historical estimates. However, historical estimates are seldom available since conventional methods cannot estimate the problem. This thesis solves such practical implementation problems for PCA–based calibration techniques (in chapter 5). Specifically, a data–assimilation framework to propose multiple OD historical data–set generation methods is proposed, allowing the use of PC–based algorithms in case the historical data is irrelevant or unavailable. Furthermore, a simplified problem formulation is also proposed that leverages application properties of PCA for faster and more efficient calibration.
    c) *Evaluating user preference towards ridesplitting usage:* Since ridesplitting services have emerged relatively recently, user perception of the services is still not well-established. Therefore, this thesis conducts stated-preference experiments to identify the factors affecting user travel behavior in presence of high capacity ridesplitting as a transport mode. The experiments include hypothetical binary scenarios with an ordered choice between ridesplitting, private car, and public transportation. Variables, including in-vehicle travel time, total travel cost, and walking and waiting time or searching time for parking, vary across the choice scenarios. Meanwhile, an ordered probit model, a multinomial logit model, and two binary logit models are specified to understand user preferences and value toward the said trip attributes.

d) *Ridesplitting market equilibrium model:* Market equilibrium (ME) models model static or aggregated representation of on-demand service markets, allowing to model the intertwined supply-demand relationship of on-demand services analytically. This thesis develops a theoretic equilibrium model for ridesplitting markets that interprets sophisticated interactions between the service decision variables and the system's endogenous variables at the network level to cater to ridesplitting dynamic ride-matching and detouring characteristics. The proposed ME model models ridesplitting markets considering modal split among multiple transportation modes and can be employed to optimize or assess different operational strategies specific to ridesplitting.

2. **Practical contributions:**

a) *Trip-based demand estimation for AMoD ridesplitting:* With iterative supply-demand equilibrium requirement, agent-based (microscopic) demand modeling is more commonly used in the literature for on-demand services. However, agent-based modeling is still rather novel and not well-established, whereas trip-based demand modeling is a more conventional method for most traffic models. Moreover, iterative procedures are often impractical for already computationally expensive microscopic models (and integrated routing algorithms). Therefore, in chapter 4, this thesis proposes a simple and more practical trip-based demand modeling method for ridesplitting, exploiting its specific service characteristics (i.e., hard/explicit rider time constraints). The method helps remove the requirement of iterative simulations and allows much easier adaptability among most traffic simulators.

b) *Utility-based compensation pricing:* AMoD ridesplitting serves riders with ridesharing through dynamic matching and detouring. Therefore, the riders experience both uncertainty and inequity in their trip level of service (LoS). In chapter 4, this thesis proposes a utility-based compensation pricing method to address this issue. The pricing method compensates for the trip fares of riders with experienced trip utility lower than a threshold LoS. It reduces the standard deviation of trip utilities and adds more certainty and equity for the riders to choose the ridesplitting service.

c) *Implementation properties and guidelines for Principal component Analysis:* In chapter 5, the implementation properties of PCA and its combination with Simultaneous Perturbation Stochastic Approximation (SPSA) are explored by estimating dynamic demand for one of the largest case studies reported in the literature, the Munich metropolitan urban network. Multiple sensitivity analyses are employed to assess the toll and benefits of using PCA on estimation problems and SPSA setup. Using the results, simplified guidelines are established for practically implementing such PCA–methods on large–scale models.

d) *Simulation-based ridesplitting service exploration and impact assessment:* This thesis also explores the AMoD ridesplitting service performance and related

impacts. First, chapter 6 contributes by performing a detailed analysis to understand the 1) effects of microscopic service modeling, 2) ridesplitting benefits by varying vehicle types, and 3) factors and benefits behind high-occupancy ridesharing. Then chapter 7 contributes to analyzing AMoD ridesplitting under a more extensive experimental setup with varying service, demand, and traffic congestion scenarios. The impacts and relations of different supply and demand variables, such as fleet size, mode share, and passenger flexibility, are explored with serviceability, occupancy, and ridesplitting benefits. Similarly, other possible relationships, such as loss in riders' level of service versus gained ridesplitting benefits, are also modeled for deeper understanding of ridesplitting service impacts.

1.5 Thesis outline and list of publications

This dissertation is written in a monograph format, rearranging the work based on the following collection of publications:

[1] Lu, Q., Qurashi, M., & Antoniou, C., 2022. A ridesplitting market equilibrium model with utility-based compensation pricing. Transportation *(Under revision)*

[2] Qurashi, M., Cantelmo, G., Antoniou, C., 2022. Towards the AI in model calibration, In *AI in Intelligent Transportation Systems, in press.* CRC Press

[3] Qurashi, M., Jiang, H., & Antoniou, C. 2020. Modeling autonomous dynamic vanpooling services in sumo by integrating the dynamic routing scheduler. In Proceedings for SUMO User Conference, 2020.

[4] Qurashi, M., Jiang, H., & Antoniou, C., 2022. Microscopic modeling and optimization of autonomous mobility on–demand ridesplitting, *(Submitted)*

[5] Qurashi, M., Lu, Q., Cantelmo, G., Antoniou, C., 2022. Dynamic demand estimation on large scale networks using Principal Component Analysis: the case of non-existent or irrelevant historical estimates. Transportation Research Part C: Emerging Technologies, 136, 103504.

[6] Tsiamasiotis, K., Chaniotakis, E., Qurashi, M., Jiang, H., & Antoniou, C. (2021). Identifying and Quantifying Factors Determining Dynamic Vanpooling Use. Smart Cities, 4(4), 1243-1258.

The structure of the manuscript is shown in figure 1.1. To describe the outline, after the current chapter 1, chapter 2 develops an understanding of AMoD, discussing different service concepts and modeling requirements, and provides the state-of-the-art on topics like AMoD modeling and optimization and dynamic demand estimation. Then, chapter 3 develops the AMoD ridesplitting supply modeling methods by proposing a generic

AMoD modeling architecture, methods to integrate DARP optimization, ridesplitting-specific supply enhancement for microscopic modeling, and practical platform development. Similarly, chapter 4 addresses the demand modeling requirements specific to AMoD ridesplitting, showcasing ridesplitting-specific user preference survey and analysis, a simplified trip-based demand estimation method, ridesplitting market equilibrium, and a utility-based compensation pricing method. Further, chapter 5 focuses on improving the practicability of large-scale dynamic demand estimation methods. It proposes multiple historical data matrix generations methods for easier adaptability of Principal Component Analysis (PCA), simplified problem formulations to exploit its application properties, and a case study along with the sensitivity analysis to assess the efficacy of proposed methods and provide implementation guidelines. Later, chapter 6 provides a case study to assess the efficacy of microscopic AMoD ridesplitting platform and to explore the factors and benefits for varying ridesharing occupancies. It also includes a case study on modeling ridesplitting market equilibrium (ME) and utilizes it to explore the impacts of different service variables on market state and the benefits of the proposed utility-based compensation pricing method. Similarly, chapter 7 extends upon ridesplitting service exploration and assesses the impacts of multiple service-related variables on service performance and benefits under exogenous and endogenous demand scenarios. Finally, chapter 8 concludes upon the proposed methods and findings of the thesis and also mentions possible future research directions.

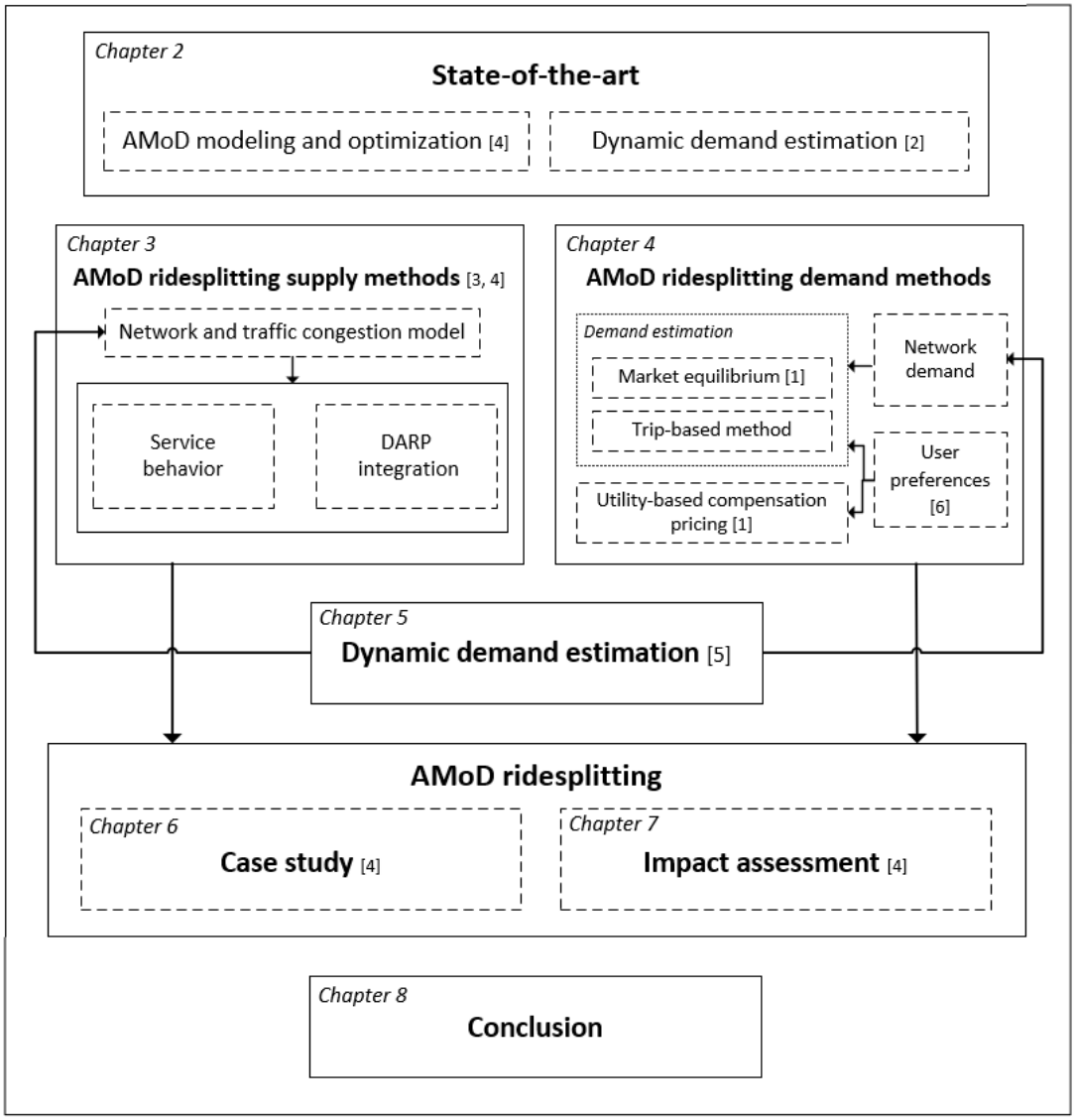


Figure 1.1: Thesis outline

2 State-of-the-art

Contents

This chapter provides a combined state-of-the-art on the overall scope of this thesis. First, it helps develop an understanding of AMoD, discussing different service concepts and modeling requirements. Then it further reviews the literature in detail, categorizing it specific to the identified requirements of modeling and optimizing AMoD ridesplitting. A parallel emphasis is also given to the topic of dynamic demand estimation, specifically its different problem formulations, estimation, and scalability methods.

The content of this chapter has been partially presented in the following works, while part of the content is unpublished to date:

Qurashi, M., Cantelmo, G., Antoniou, C., 2022. Towards the AI in model calibration, In *AI in Intelligent Transportation Systems, in press.* CRC Press

Qurashi, M., Jiang, H., & Antoniou, C., 2022. Microscopic modeling and optimization of autonomous mobility on–demand ridesplitting, *(Submitted)*

Lu, Q., Qurashi, M., & Antoniou, C., 2022. A ridesplitting market equilibrium model with utility-based compensation pricing,. Transportation *(Under revision)*

Tsiamasiotis, K., Chaniotakis, E., Qurashi, M., Jiang, H., & Antoniou, C. (2021). Identifying and Quantifying Factors Determining Dynamic Vanpooling Use. Smart Cities, 4(4), 1243-1258.

2.1 Understanding Autonomous Mobility on Demand

Smartphone-based real-time communication gave emergence to multiple ridesourcing concepts mainly to improve urban mobility convenience. Some concepts with ridesharing also served the need to resolve urban congestion problems and reduce the social and environmental imprints. These concepts are unified with vehicle automation growth, named Autonomous Mobility on Demand (AMoD). An AMoD system van behave similar to any on-demand service concept with driver-less vehicles, therefore different AMoD service types exist (alone or mixed), varying by service flexibility, costs, and user experience. The AMoD vehicles, either owned privately or by services operators, can be shared to provide on-demand mobility, significantly reducing car ownership, parking requirement, mobility cost, and environmental imprints. This section focuses on understanding AMoD systems by exploring the literature that helps describe the AMoD topology and characteristics. Meanwhile, we also explore the literature available on all different service types and use our review effort to enlist all possible components of modeling AMoD systems. The list also provides a taxonomy to classify different literature works and their contributions (used in the later sections).

2.1.1 Topology of AMoD / ridesourcing services

Since AMoD combines different ridesourcing service types operated on driverless vehicles, the topology of AMoD systems is somewhat similar to ridesourcing. Many notable review works exist in the literature that can help understand the topology of such systems. For example, Shaheen and Cohen (2019) provides the classification of on-demand ride services and compares different ridesharing concepts, including traditional carpooling/vanpooling, ridesplitting and microtransit. Similarly, Wang and Yang (2019) provides a general framework to describe the ridesourcing systems. It distinguishes different service types and reviews their literature from the perspective of supply, demand, pricing, platform operations, and impacts. From an AMoD perspective, Narayanan et al. (2020) gave a comprehensive review of relevant AMoD studies and their focus on various modeling components, identified impact, demand, and policies. Another notable effort is from Furuhata et al. (2013), which reviews the service types provided by different ridesourcing (ride-matching) agencies by classifying them under ride-matching variables (time, routing, location) and commute type (casual/dynamic, regular commute, long-distance). It also distinguishes the service operational characteristics as positional (commute locations and detouring), temporal and strategical elements. From the service optimization perspective, Hyland and Mahmassani (2017) defines the taxonomy to classify fleet management problems (with no classification on vehicle route flexibility, e.g., many-to-many or many-to-one type services). Similarly, Molenbruch et al. (2017) and Ho et al. (2018) focused on reviewing and classifying optimization problem variants and solution methods, also providing taxonomies related to real-time characteristics, service design, and solution methods.

The service concepts under ridesourcing or demand responsive transport vary by flexibility in vehicle routing, detouring, and rideshare capacity. The different service concepts are car-sharing, ride-hailing, ride-pooling, ridesplitting, and micro-transit. The car-sharing and ride-hailing services are most flexible, private, and expensive (commonly termed together as AMoD car-sharing for driverless AVs (Narayanan et al., 2020)). The trips are served with direct routes without sharing or detouring, acting close to the private car. Next comes ride-pooling services, which extend the ridehailing concept and serve pooled rides of riders matched together due to similar itineraries without experiencing much detours. Ridepooling of regular or reservation-based commutes (e.g., for work or airport pickups) long exists under services terms of carpooling and vanpooling (Shaheen and Cohen, 2019). Then, the third service type is ridesplitting which serves riders with similar travel directions and dynamically matchable itineraries together through dynamic match-

ing and detouring. Like ridehailing, ridesplitting is also administrated by ridesourcing companies (or TNCs) that operate on the ride-matching concept, i.e., freelance drivers with personal vehicles are matched with riders. Ridesourcing-based ridesplitting generally has lower vehicle occupancy due to the absence of optimization-based rider and vehicle assignment. The last service type is microtransit referring to high occupancy ridesharing. Microtransit service concepts can also vary further by their schedule and routing flexibility. The most agile form of microtransit is also named dynamic vanpooling (Li et al., 2019a) and acts similar to ridesplitting with the aim of higher ridesharing occupancy using operator-owned/controlled vehicles and optimization-based routing and assignment. Finally, the least flexible and cheaper (yet on-demand) on-demand ride service is the microtransit with fixed/pre-defined routes and flexible on-demand schedules. Such a type of microtransit acts similar to on-demand/crowd-sourced transit routes with/without reservations (e.g., Jetty).

2.1.2 Literature on different service concepts

As mentioned, AMoD combines multiple service concepts varying by flexibility of routing, ridesharing, and demand operation. The most prominent concept is AMoD car-sharing which is extensively studied in literature (Alam and Habib, 2018; Azevedo et al., 2016; Bischoff and Maciejewski, 2016; Boesch et al., 2016; Chen et al., 2016; Fagnant and Kockelman, 2014; Hyland and Mahmassani, 2018; Jäger et al., 2017; Marczuk et al., 2015; Moreno et al., 2018; Zhao and Kockelman, 2018). The literature also uses other terms for AMoD car-sharing concept, e.g., SAV, autonomous taxi, shared taxis, shared autonomous taxis, autonomous electric taxis, etc. Most of these present studies model on-demand service operations (Fagnant and Kockelman, 2018; Gurumurthy and Kockelman, 2018; Hyland and Mahmassani, 2018), while a few also focus on reservation-based demand operations (e.g., Levin (2017); Ma et al. (2017)) m. These efforts can also be further divided based on their focus, e.g., AMoD modeling (Alam and Habib, 2018; Azevedo et al., 2016; Bischoff and Maciejewski, 2016; Levin, 2017; Marczuk et al., 2015; Nahmias-Biran et al., 2019), service and impacts assessment (Bauer et al., 2018; Boesch et al., 2016; Bösch et al., 2018; Chen et al., 2016; Fagnant and Kockelman, 2014; Jäger et al., 2017; Moreno et al., 2018; Nahmias-Biran et al., 2021; Zhang et al., 2017; Zhao and Kockelman, 2018), and service assignment/optimization ((Ma et al., 2017; Hyland and Mahmassani, 2018)). The second prominent concept is of ridesharing, but it can be further categorized as ride-pooling and ridesplitting (en-route detours) (Gurumurthy and Kockelman, 2018). As mentioned before ride-pooling is an extension of the car-sharing service concept without detours (i.e., pooling of passengers with similar origin-destination). The prominent literature in ride-pooling includes (Bischoff et al., 2017; Fagnant and Kockelman, 2018; Gurumurthy and Kockelman, 2018; Hosni et al., 2014; Levin et al., 2017; Ma and Koutsopoulos, 2022; Nahmias-Biran et al., 2019; Santi et al., 2014; Zhang et al., 2015), among which most of the studies model and compare AMoD car-sharing with ride-pooling (Fagnant and Kockelman, 2018; Gurumurthy and Kockelman, 2018; Hörl, 2017; Levin et al., 2017; Nahmias-Biran et al., 2019), while Ma and Koutsopoulos (2022) evaluates the effects of user flexibility and advance requests on amount of ride-pooling and related benefits.

The literature on ridesplitting is also board and the studies can be grouped by their focus. The first group of studies which focus on modeling ridesplitting include (Alazzawi et al., 2018; Gurumurthy and Kockelman, 2018; Jäger et al., 2018; Lokhandwala and Cai, 2018; Martinez and Viegas, 2017; Stiglic et al., 2015, 2016; Vosooghi et al., 2019). These modeling studies vary by their service concept implementation. First prominent difference in by service vehicle capacity, e.g., Gurumurthy and Kockelman (2018); Lokhandwala and Cai (2018) model ridesplitting taxis with maximum occupancy of 4 passengers, Alazzawi et al. (2018); Martinez and Viegas (2017); Farhan and Chen (2018); Vosooghi et al. (2019) model minivans with vehicle occupancy upto 6

passengers, or Ronald et al. (2013) models vans with capacity of 10 passengers (named DRT). Then another difference is in flexibility of dynamic detouring, e.g., Vosooghi et al. (2019) consider 30% detour of the direct time, Martinez and Viegas (2017); Farhan and Chen (2018) models ride-pooling with small detours. Another group of dynamic ridesharing studies that focus on service optimization, i.e., Agatz et al. (2011); Aissat and Oulamara (2014); Alonso-Mora et al. (2017); Fielbaum et al. (2021); Li et al. (2019a); Stiglic et al. (2015); Simonetto et al. (2019); Tsao et al. (2019); Tafreshian et al. (2021). These studies also vary by their service concepts, e.g., Agatz et al. (2011) focus on ride-matching optimization, Agatz et al. (2011); Alonso-Mora et al. (2017); Li et al. (2019a); Tafreshian et al. (2021) focus on high-capacity ridesplitting named as dynamic DARP, and Aissat and Oulamara (2014); Fielbaum et al. (2021); Stiglic et al. (2015) focus on ridesplitting optimization with service location flexibility or walking/meeting point-based system.

2.1.3 Modeling components for AMoD systems

Modeling AMoD or any similar on-demand system is intrinsically complex due to flexibility, on-demand behavior, and many control alternatives, e.g., optimization and pricing. Therefore, it contains many different modeling components required to model the service concepts comprehensively. From literature, review efforts like Hyland and Mahmassani (2017); Narayanan et al. (2020) provide similar classifications. Narayanan et al. (2020) provides a set of 8 different modeling components, including demand, fleet, traffic assignment, vehicle assignment, vehicle redistribution, pricing, charging, and parking. Likewise, Hyland and Mahmassani (2017) also provides a similar and more detailed taxonomy for AV fleet management problems with categories from the perceptive of the service concept, fleet characteristics, information processing, and demand and network modeling. Since it is evident that modeling all components is difficult, most literature efforts focus on some of them while assuming simplifications for others to keep a manageable complexity (Narayanan et al., 2020). Thus, the literature can also be classified based on its focus on modeling components, a strategy we use in the sections below to classify literature. In this section, we enlist a set of components with the perspective of modeling and evaluating AMoD ridesplitting. Note that Hyland and Mahmassani (2018) which focuses on shared AV optimization, classified AMoD systems in three components, i.e., demand, fleet optimizer, and some representation of transport network. We also classify all components similarly in three main categories, i.e., supply, demand, and optimization. These components also act as the modeling requirements to model AMoD ridesplitting.

Optimization

- **DARP optimization** A Dial-A-Ride problem (DARP) solver, also referred to as scheduler, is the core part of the AMoD service operation, especially for ridesplitting. DARP algorithms are responsible for optimizing the vehicle routing and passenger request assignment. DARP for AMoD ridesplitting is also named vehicle routing problem with explicit time windows (VRPTW) (Hyland and Mahmassani, 2017; Tafreshian et al., 2021). Therefore, the optimization constraints include passengers' time preferences (e.g., time windows for waiting and destination arrival) of both new and in-service passengers along with vehicle operational attributes (e.g., vehicle capacity, service area, charging times for electric vehicles). Moreover, the optimization problem also includes network state information to optimize optimum vehicle routing. Note that AMoD services with the on-demand operation are dynamic DARPs, in which some types (i.e., stochastic DARP) also consider stochasticity in future information (details in section 2.2.2). The optimization targets could include, e.g., profit maximization, minimum cost, minimum wait, and maximum

vehicle occupancy. Note that the problem complexity and information sensitivity of dynamic DARP algorithms increase with higher ridesharing occupancy (Agatz et al., 2011; Alonso-Mora et al., 2017; Li et al., 2019a; Tafreshian et al., 2021). For further details, please refer to Molenbruch et al. (2017); Ho et al. (2018).

- **DARP information modeling:** Like any other optimization, DARP optimization performance depends on the type and quality of the problem information. The DARP information includes network state, service vehicles' status (routes, stops, and positioning), and service demand information. The information characteristics include its evolution (static or dynamic), quality (deterministic or stochastic), and availability (global or local) (Hyland and Mahmassani, 2017). These characteristics directly influence the efficiency of DARP optimization and require varying facilities to model them, e.g., a simulation-based dynamic service modeling requires an online feed from the simulation to provide updated information on the optimization problem to the DARP scheduler. Similarly, for stochastic DARP, information modeling also needs to cater to the stochasticity related to network dynamics, service operations, and demand.
- **Dynamic communication and control:** All on-demand services require consistent control of all service vehicles to do dynamic vehicles and request assignments. Although the modeling methods using simplified network representation (fixed travel times) do not require dedicated facilities, the microscopic traffic models need to simultaneously integrate the DARP scheduler to communicate the optimization problem information. The problem is formulated using variables like request attributes, vehicle state, vehicle routes, and network information. Whereas the solution resulting in control variables, i.e., service vehicles' routes and stops, require to be dynamically updated in the simulation using dynamic routing control of the service vehicles.

Supply

- **Network modeling:** The network modeling component models network state information and service vehicles' assignment. The network state information is directly used in dynamic DARP to find point-to-point travel times and optimum vehicle routes. Modeling network information can vary based on its degree of evolution, i.e., fixed average time travels, time-dependent travel times (Li et al., 2019a; Schilde et al., 2014), or using dynamic traffic assignment models (Levin et al., 2017; Narayanan et al., 2020). Modeling service vehicles' assignment also varies similarly. Although the use of travel time information is more common to evaluate service vehicle propagation, modeling dynamic traffic especially using microscopic traffic models, allows more detailed service interaction behavior with the environment and realistic service operation times. It also allows modeling of autonomous driving behavior and other possible improvements due to connected AV technologies (e.g., reservation-based intersection control, cooperative merging, covered by Levin et al. (2017)).
- **Service behavior modeling:** AMoD systems require modeling the service behavior component specific to the service concept and the network modeling resolution. It includes modeling the vehicle behavior and service operations. AMoD service vehicles need to replicate autonomous driving behavior with certain vehicle characteristics depending on the implementation of the service, e.g., vehicle capacity and driving behaviors generally vary between shared taxis and vans. Additionally, advanced connected AV technologies mentioned before are also part of modeling vehicle behavior and can be modeled in dynamic traffic models. Similarly, subject to the service concept, i.e., the flexibility in ridesharing and detouring, supporting facilities are required to model the service operations. AMoD

ridesplitting requires flexible service stop locations, boarding/alighting operations, and waiting for space for idle vehicles. Moreover, since ridesplitting is on-demand and shared, trips with varying attributes (i.e., origin, destination, cost, waiting, and travel times) are served together, resulting in a varying trip level of service among passengers. Therefore modeling discrete passengers is also necessary to further evaluate the overall system performance.

Demand

- **Passenger preference modeling:** Ridesplitting, with its characteristics of dynamic ride-matching and detouring, is unique from other ridesourcing services. The riders pre-define the extent of waiting and detour for their trips, which act as hard constraints for vehicle routing optimization and assignment. Therefore, traveling with ridesplitting should create a unique sense for the passengers that should be appropriately represented in their preference modeling for ridesplitting demand estimation. It can also include modeling different demand groups (i.e., market segmentation) with their respective preferences, suitable to represent the population heterogeneity.
- **Mode choice modeling:** A mode choice module is required to estimate the mode shares of AMoD ridesplitting. Being on-demand, AMoD demand estimation requires damping the intertwined relationship of the service supply and demand (Liu et al., 2019; Hörl et al., 2016; Basu et al., 2018). The mode choice is rider-specific due to stochasticity (and uncertainty) in both service availability and possible trip utilities for each trip. The equilibrium damps out the difference between the expected and experienced trip utilities for the set of attracted demand, given that all inputs are fixed. Note that the rider choice is scaled by the rider's preference towards the value of different trip attributes (cost and travel time).
- **Pricing:** Ridesplitting prolongs rider trip distance and travel time for service benefit (unlike ride-hailing or car-sharing). Therefore, trip pricing is set based on direct trip distance and passenger trip flexibility (with additional surcharges for factors like time of day and congestion). Although flexible pricing strategies for ridesharing are not much explored (Hyland and Mahmassani, 2017), trip price modeling and optimization based on the service cost, passenger preferences, and cost of competing for travel modes is also necessary to realistically model and evaluate AMoD ridesplitting. Further, dynamic pricing strategies (e.g., Bai et al. (2017); Gurumurthy et al. (2019); Guo et al. (2017); Guan et al. (2019a)) to exploit demand, and service-specific characteristics are also always desirable.

2.2 Modeling Autonomous Mobility on Demand systems

2.2.1 Supply modeling

The supply of an AMoD system is defined by the fleet characteristics (e.g., fleet size, vehicle capacity), the service concept (e.g., with or without ridesharing or detouring), and the service availability (effected by optimization and network assignment). In operational terms, modeling the AMoD supply attributes to represent the service characteristics (combination of fleet characteristics and service concept), the network congestion, and the vehicle routing (DARP) optimization. As mentioned before in section 2.1.3, modeling all AMoD components together is complex, and most literature efforts focus on detailed modeling of certain components while assuming simplifications for others (Narayanan et al., 2020). Therefore, this section reviews the

literature efforts by classifying them based on their focus on modeling different supply components.

2.2.1.1 Network modeling

Network modeling or traffic/vehicle assignment attributes to model the flow of service vehicles in the network or service area. It also assists the DARP optimization process, where the algorithm anticipates point-to-point travel times based on network state information. The present literature varies significantly from using fixed travel times to dynamic traffic simulators to represent the network. The first group of efforts that focus on developing dynamic DARP solutions considers mostly fixed travel times (see Molenbruch et al. (2017)), whereas only a few consider time-dependent travel times (e.g., Xiang et al. (2008); Schilde et al. (2014); Li et al. (2019a)) or additionally future stochastic travel time information (Schilde et al., 2014; Li et al., 2019a). Likewise, many efforts related to AMoD modeling literature also consider similar network assignment methods, e.g., assuming fixed travel times and factors to represent different network states (Jäger et al., 2018), using the average travel time of off-peak and peak hour (Chen et al., 2016; Fagnant and Kockelman, 2014; Zhang et al., 2015). Note that using travel time information with factors/weights helps represent different traffic patterns and simplify the network representation while using time-dependent travel times improves the degree of detail with more dynamic interval-based traffic representation. Again, these simplifications have been used by studies that primarily focused on developing DARP solutions (Molenbruch et al., 2017). Therefore they can neglect the lack of accuracy related to vehicle routing and operations. However, a few efforts that aim to do large-scale service assessments while considering simplified network representation significantly suffer in the accuracy of evaluated outcomes (Levin et al., 2017).

In literature, the next degree of detail for representing the network comes by the use of dynamic traffic assignment (DTA) simulations (Chiu et al., 2011). DTA has been widely adopted in literature to model AMoD systems (e.g., Alam and Habib (2018); Alazzawi et al. (2018); Azevedo et al. (2016); Basu et al. (2018); Bischoff and Maciejewski (2016); Boesch et al. (2016); Fagnant and Kockelman (2018); Hörl (2017); Lokhandwala and Cai (2018); Moreno et al. (2018); Vosooghi et al. (2019)). DTA models assign the time-dependent travel demand dynamically in individual trips with choices of their departure times and network routes. Although influenced by other factors, the choices are also affected by the dynamic network state (especially route choice). It results in realistic (stochastic) traffic congestion patterns changing dynamically. DTA models also differ by their degree of detail with three different resolutions (i.e., macro, meso, and microscopic). The most detailed and realistic traffic assignment is conducted by microscopic models, which model the movement of individual vehicles along with their driving behavior and interaction with the surrounding vehicles and network infrastructure. Similarly, another type is mesoscopic models that model detailed route choice decisions while depicting aggregated (macroscopic) properties of traffic flow (i.e., without modeling individual vehicle movements and their interactions). Mesoscopic models aim to simulate larger network and demand scenarios faster by sacrificing the accuracy in modeling detailed traffic dynamics (Chiu et al., 2011).

Among the DTA modeling studies, many different DTA simulation suites have been adopted. The foremost is MATSim (Horni et al., 2016), which many studies have adopted (e.g., Bischoff and Maciejewski (2016); Bischoff et al. (2017); Boesch et al. (2016); Hörl (2017); Moreno et al. (2018); Vosooghi et al. (2019)). However, MATSim is a demand-centric simulator that aims to run large traffic networks with high demands faster (Saidallah et al., 2016) (similar to mesoscopic models). It models only node-level traffic using queuing model method for representing link-based traffic flow without vehicular interactions (see Ronald et al. (2017) comparison of MATSim against microscopic models for DRT simulation). Then, another prominent simulation suite named

Simmobility (Adnan et al., 2016) has also been used by multiple literature works. These efforts model AMoD systems in both mesoscopic (Nahmias-Biran et al., 2019, 2021) and microscopic (Azevedo et al., 2016; Marczuk et al., 2015) resolutions. Azevedo et al. (2016) is the most notable effort proposing the framework for microsimulation of demand and supply of AMoD, employing the ST-level of Simmobility (i.e., MITSIM (Yang and Koutsopoulos, 1996)) to simulate vehicle movements at the microscopic granularity. However, it models only the AMoD car-sharing service. Further, other notable efforts include Alazzawi et al. (2018); Alam and Habib (2018), where Alam and Habib (2018) modeled SAV–based car-sharing system using network assignment through Vissim and Alazzawi et al. (2018) modeled AMoD ridesplitting service using a simplified ride-matching algorithm in SUMO (Lopez et al., 2018). Apart from the adoption of DTA models, another prominent effort is by Levin et al. (2017), who used the cell transmission model (CTM) to model the stochasticity of the traffic flow for modeling SAV operations. The modeling framework act as event-based, i.e., passenger requests or vehicle stop events are the output of the simulation.

Finally, it is noteworthy to mention that since AMoD car-sharing and ridepooling require less extensive service optimization and modeling all different AMoD components (e.g., microscopic traffic modeling and DARP optimization) becomes complex and restricted scalability, most studies mentioned above consider simplistic demand assignment methods (e.g., rule-based assignment (Narayanan et al., 2020)). Similarly, even the efforts modeling ridesplitting in microscopic models also consider simpler optimization methods, e.g., Alazzawi et al. (2018) models Robo-taxis (up to 6 passengers) using simplified ride-matching algorithm, Levin et al. (2017) models AMoD ridesplitting (up to 4 passenger capacity) with restrictive optimization, i.e., prioritizing assignment based on FCFS principle over the possibilities of dynamic matching.

2.2.1.2 Service modeling

AMoD service behavior attributes to the characteristics related to the service vehicles, fleet, and the service concept. Among these three groups, the service vehicle characteristics mainly include vehicle type and capacity, driving behavior, and level of automation; the fleet characteristics include fleet size, positioning, and other related attributes; the service concept related characteristics include flexibility of routing, detouring, and service locations (see section 2.1.2). From a modeling perspective, some of these mentioned characteristics are simple to model (i.e., vehicle type, capacity, fleet size, and positioning), while others require dedicated facilities and modeling methods depending on the modeling resolution and targeted complexity. Note that the complexity of modeling these characteristics depends on the network modeling method, where literature efforts that use simpler travel time information methods and mesoscopic models tend only to approximate their effect. While the use of microscopic traffic models allows detailed modeling of the service behavior, however, with the requirement of additional facilitates. Below, we discuss the literature on each of the modeling-intensive characteristics separately.

The first modeling-intensive characteristic is the dynamic routing and assignment, the primary attribute of all on-demand services. It requires continuous control on service vehicles to assign routes and passengers dynamically. Note that this attribute's complexity increases with the presence of dynamic detouring for ridesplitting services. It requires extended modeling methods to cater to more frequent optimization/decision instances (see section 2.2.2) and subsequent service control. Note that such dynamic vehicle rerouting is only modeling-intensive to the usage of some network modeling methods (covered in section 2.2.1.1), for example the use of microscopic traffic models require facilities for simultaneous communication and control setup with the simulation to cater to real-time service optimization (many examples are present in literature including Alam and Habib (2018); Alazzawi et al. (2018); Azevedo et al. (2016); Bischoff et al. (2017); Levin et al. (2017); Ronald et al. (2017)).

Another modeling-intensive characteristic relates to flexibility in service locations and modeling service operations. AMoD services tend to provide point-to-point on-demand service, where the service locations can be door-to-door (mostly car-sharing), meeting-points with walking, or fixed service points as in public transport. The meeting point-based services are conceptualized to reduce service detouring in case of ridesplitting (e.g., in Aissat and Oulamara (2014); Fielbaum et al. (2021); Martinez and Viegas (2017); Stiglic et al. (2015)), while fixed service points are used in fixed route on-demand DRT concepts. Note that the modeling-intensive aspect comes with modeling stop operations, which requires additional facilities to model or estimate on-road stop operations in case of flexible service locations. Furthermore, modeling on-road service operations require link-level network and service vehicle modeling to capture the boarding/alighting operations and interaction with the environment. However, almost all AMoD modeling literature efforts consider node-based network modeling and optimization (influenced by the graph and network theory considering nodes as entities (West et al., 2001)) and approximate the duration of service operations. While only very few efforts exist which model detail service operations using microscopic traffic models (Alam and Habib, 2018; Alazzawi et al., 2018; Azevedo et al., 2016; Ronald et al., 2017). Similarly, another service operation is modeling vehicle idle behavior, which also requires facilities in microscopic models, e.g., parking infrastructure, idle network circulating behavior, or arbitrary moving vehicle outside the network Lopez et al. (2018).

The third modeling-intensive characteristic is modeling vehicle driving behavior and automation technologies. Vehicle driving behaviors model detailed vehicle propagation through the network while interacting with the environment, e.g., traffic signals, neighboring vehicles, congestion. To model them requires microscopic traffic models with link-based traffic modeling, and therefore only the microscopic modeling literature does it. Note that no literature work is found that models the autonomous vehicle driving behaviors using AV–based car following and lane changing models (e.g., ACC-CACC (Milanés and Shladover, 2014) and Wiedermann 99 (Sukennik et al., 2018)). However, Levin et al. (2017) does mention the effects of connected AVs (CAVs) towards the increase in capacity and traffic flow stability and models reservation-based intersection control technology, where vehicles communicate wirelessly to reserve a space-time path through an intersection. It is noteworthy to mention that presence of AVs and CAVs with their developing technologies under both V2V and V2I communication methods does significantly change both the vehicle behaviors and impacts (see reviews from Do et al. (2019); Gora et al. (2020)). Therefore appropriate modeling and integration of AV models are also necessary to accurately model and evaluate AMoD systems.

2.2.2 Service optimization

Ridesourcing or AMoD service optimization varies significantly by problem complexity and solution efforts among different service concepts. It is generally refers as dial-a-ride problem (DARP) which is a generalization for vehicle routing problem with time windows (VRPTW). For AMoD car-sharing and ride-pooling, the vehicle assignment problem is less intensive and is referred as DARP with implicit/soft time constraints due to private rides or no ridesharing (Hyland and Mahmassani, 2018). Therefore, studies mostly consider just rule-based vehicle assignment methods (Narayanan et al., 2020) with a few exceptions which focus on system-level AV fleet optimization Ma et al. (2017); Hyland and Mahmassani (2018). On the other hand, since AMoD ridesplitting contains dynamic detouring, the vehicle assignment problem is much more complex and is referred as DARP with explicit/hard time constraints (i.e., for high occupancy ridesplitting). DARP optimization is a crucial component for such a service, significantly influencing its performance and operations. Therefore many literature efforts focus on developing stronger optimization models or DARP solutions (Ho et al., 2018), whereas the ridesplitting modeling

efforts still use some sort of simplified optimization formulations to cater to the assignment complexity. Note that, in ridesplitting optimization literature, approaches also vary depending upon ridesharing occupancy (taxi, minivans or vans), service location flexibility (e.g., meeting-point services) and detouring flexibility (see section 2.1.2 and 2.2.2.3).

2.2.2.1 DARP classification

DARP (Ho et al., 2018) or AV vehicle fleet management problem (Hyland and Mahmassani, 2017) is generally classified in context of both decision making and quality of information (Pillac et al., 2013). The decision making aspect relates with the evolution of information and classifies DARPs as static or dynamic, where in static DARPs information related to the travel requests and/or network state is considered known and the service routes are optimized before the start of the service, e.g., reservation-based SAV (Ma et al., 2017) or para-transit services, while in dynamic DARPs the evolution of information is considered time-dependent and the service decisions are dynamically optimized upon arrival of new information. Most AMoD services are inherently dynamic and are refers to dynamic DARP, where real-time and on-demand vehicle routing and request assignment is considered. The other context to classify DARPs relates with quality of information, i.e., either deterministic and stochastic. The difference among the two classes is considering the presence of uncertainty/stochasticity in available information. Although it is argued that all DARPs are stochastic in nature, where information related to travel times, service stop durations, and request arrivals never comes up exactly as known (Ho et al., 2018). In deterministic DARPs, the algorithm considers to know exact information related to these attributes while for stochastic DARPs the algorithm make decisions in context of imperfect information. Stochastic DARP solutions also consider exploiting the stochasticity in future information using probabilistic historical information for routing and repositioning (Hyland and Mahmassani, 2017).

The two mentioned classifications result in four different type of DARPs, among which the dynamic-stochastic DARP stands out for being the most realistic for AMoD services. A dynamic-stochastic DARP considers optimizing the service vehicles dynamically with arrival of information related to new passenger requests, delays in service operations and network information (incidents, etc) while also catering to the possible stochasticity towards future information. Note that, concerning the modeling of AMoD systems or evaluation of DARP algorithms, appropriate representation of stochasticity in above mentioned aspects is crucial to replicate the real-world conditions. Therefore, we discussed the importance of network and service modeling in microscopic resolution under section 2.2.1. Furthermore, integrating dynamic DARP optimization simultaneously with microscopic AMoD models is also another crucial aspect which allows to accurately model the optimization problem information dynamically for efficient DARP solutions (hence set as a requirement in section 2.1.3).

Apart from the above two classification categories, DARP optimization formulation has been further classified based on the service, demand and fleet related characteristics as well as the optimization targets (see Hyland and Mahmassani (2017)). A prominent class related to service concept is time window constraints, i.e., either explicit (hard constraint) or implicit (soft/suggested constraint). Explicit time windows are common for high capacity ridesplitting or microtransit. Due to the presence of dynamic detouring, passenger express explicit waiting and travel times and force the fleet operator to serve them within the specified time windows. Whereas implicit time windows are common for car-sharing concepts without hard time constraint since the delay mostly occurs by external factors like network congestion (also mentioned in Hyland and Mahmassani (2018)). Therefore DARP optimization is a requirement for ridesplitting oper-

ations, whereas in car-sharing or ride-pooling modeling rule-based optimization can work and is used prominently (discussed in section 2.2.2.3).

2.2.2.2 DARP-specific solution approaches

Ho et al. (2018); Molenbruch et al. (2017) are two prominent review efforts on DARP solutions, which classify the solution methods in two different groups due to the NP-hardness of the problem. The first group is the exact methods that provide the highest quality solutions but with significantly higher computational requirements. Therefore they are only available (or thus far possible) for static DARPs with small-sized instances. For dynamic DARPs, exact methods are said not to be able to provide timely solutions due to time constraints. The other group is the approximate method, i.e., heuristics or meta-heuristics methods, which tend to approximate the solutions with much shorter computational efforts and therefore are applied for both static and dynamic DARPs with all size instances. Given the limited application of exact methods, much research has focused on heuristic methods to improve their efficiency and effectiveness (Ho et al., 2018).

Ho et al. (2018) mentions that heuristic methods exhibit two main elements, i.e., the extent of diversification versus intensification. The diversification element defines the extent of search space for finding the solution (e.g., multiple neighborhood search, population methods), while the intensification element defines the extent of effort to improve the solution quality (e.g., single neighborhood search, optimization restarts). The prominent set of methods mentioned in the literature includes tabu search (TS), simulated annealing (SA), variable neighborhood search (VNS), genetic algorithms (GA), and hybrid methods. Note that by default, each type of method is inclined towards either being diversifying or intensifying, i.e., multiple neighborhood search methods like VNS are more diversifying methods, while tabu search (TS) following the principle of local search intensifying method. Therefore most researchers tend to compensate for such weakness among their proposed methods.

Concerning solution efforts relative to different DARP types, we only mention the literature related to dynamic DARP since AMoD services come under this category (for other types, please refer to Molenbruch et al. (2017); Ho et al. (2018)). Several solutions exist in the literature for dynamic DARP solutions, which are preliminarily differentiated by the information quality, i.e., deterministic or stochastic. To differentiate, Ho et al. (2018) mentioned that the dynamic and deterministic DARP methods generally only consider new requests as triggering events for optimization Berbeglia et al. (2012); Häll et al. (2015); Marković et al. (2015). Whereas the dynamic and stochastic DARP approaches vary by considering different types of information stochasticity, i.e., in-vehicle routing (Bent and Van Hentenryck, 2004), in request no show (Xiang et al., 2008), future user requests (Xiang et al., 2008; Schilde et al., 2011; Lowalekar et al., 2018; Tafreshian et al., 2021), stochastic travel times (Xiang et al., 2008; Schilde et al., 2014; Li and Chung, 2020), future user requests and stochastic travel times (Li et al., 2019a), and stochastic travel and service times (Zhang et al., 2022). Another distinction among dynamic DARP solutions can be made by their proposed method to cater to the stochastic and dynamic nature of DARP. For example, a common approach is the use of historical data for demand requests (Schilde et al., 2011; Lowalekar et al., 2018; Li et al., 2019a; Tafreshian et al., 2021; Wei et al., 2017). Then, another is to consider using or solving multiple scenarios of anticipated states for demand and/or network conditions (Bent and Van Hentenryck, 2004; Ghilas et al., 2016; Li et al., 2019a). Similarly, recently Tafreshian et al. (2021) proposed a two-phase (offline-online) data-driven method which reduces the burden on online or real-time routing by pooling near-optimal shuttle routes in the offline phase and proactively using them to assign shuttles.

2.2.2.3 AMoD-specific optimization methods

This section reviews the AMoD-specific literature efforts focusing on developing service optimization methods. As mentioned earlier, the DARP complexity varies among different service concepts (specifically for ridesplitting), which subsequently changes the optimization and computational intensity in the proposed solution methods. The AMoD car-sharing and ride-pooling are simpler DARP with implicit/soft time windows due to private rides or no ridesharing (Hyland and Mahmassani, 2018). Therefore, Narayanan et al. (2020) mentioned that most AMoD studies consider only rule-based vehicle assignment methods, e.g., assigning nearest vehicle to request. While only a few efforts focus on optimizing AMoD car-sharing service. Among these, Ma et al. (2017) proposed a linear programming model using a space-time network to create SAV chains for reservation-based car-sharing requests (static DARP). While, Hyland and Mahmassani (2018) compared six different vehicle assignment strategies for AMoD car-sharing that varying from only first-come-first-served (FCFS) strategy to idle and en-route pickup and drop-off strategy, concluding optimization-based strategies are more efficient in reducing both fleet miles and traveler waiting times. Similarly, Farhan and Chen (2018) proposed ride-pooling optimization (i.e., allowing only 20% detour time) by solving dynamic DARP of explicit time windows using Tabu search while explicitly considering the impacts of charging technology and infrastructure. Finally, much recently, Zhang et al. (2022) proposed a branch and price algorithm minimizing the AMoD car-sharing operational costs while considering the charging schedules as well as the stochasticity in travel and service operation times.

AMoD ridesplitting optimization is a more complex dynamic DARP with explicit time windows that constraint dynamic detouring. Many literature efforts develop optimization methods for AMoD ridesplitting; however, these approaches differ by their focus on, e.g., solving dynamic DARP for high-capacity ridesplitting (Alonso-Mora et al., 2017; Li et al., 2019a; Tafreshian et al., 2021), or optimizing walking/meeting points for ridesplitting Aissat and Oulamara (2014); Fielbaum et al. (2021); Stiglic et al. (2015). Among the dynamic DARP solution efforts, Alonso-Mora et al. (2017) developed a novel framework that does vehicle assignment using a reactive anytime optimal algorithm and does vehicle re-positioning using linear programming formulation. The authors employed the New York City taxi dataset to show the framework efficiency and ridesplitting effects. Similarly, Li et al. (2019a) developed a multiple scenario approach for proactively deploying vans, where each scenario solved by the Tabu search algorithm generates multiple potential decisions for van positions while catering to the stochasticity in future requests and traffic conditions. Much recently, Tafreshian et al. (2021) proposed a two-phase (offline-online) data-driven method for optimizing shuttles, where offline phase pools near-optimal shuttle routes for different historical demand scenarios and online phase proactively route shuttles based on the realized set of offline routes. Apart from the AMoD.specific literature, many of the dynamic DARP solutions mentioned in the previous section are applicable for AMoD ridesplitting optimization, like Schilde et al. (2011) which focus on optimizing partially dynamic para-transit system as dynamic and stochastic DARP using meta-heuristics (and similarly Schilde et al. (2014); Xiang et al. (2008)).

Among the literature efforts for optimizing meeting points flexibility for ridesplitting, Aissat and Oulamara (2014) and Stiglic et al. (2015) proposed meeting point-based ridesplitting to allow a reduction in driver detour and travel cost. Aissat and Oulamara (2014) compared both exact and heuristic approaches to optimize suitable meeting-point locations for a large real road network. In contrast, Stiglic et al. (2015) proposed a refined feasible match generation method that reduces the set of feasible driver-rider pairs required for evaluation by exploiting the problem structure and characteristics. Recently, Fielbaum et al. (2021) theoretically analyzes the benefits and optimization potential of PUDO (meeting) points for ridesplitting systems. The authors build groups-vehicles (GV) graphs to solve the request assignment problem as an integer linear

problem using heuristics. The approach is tested using the New York City taxi dataset. From the ride-matching perspective, Agatz et al. (2011) is another effort that developed the ride-matching optimization method by introducing a rolling horizon approach. The research compared greedy matching rules to more sophisticated optimization to show optimization benefits and the appearance of sustainable populations for ridesplitting even in low-density urban areas.

2.2.3 Demand modeling

2.2.3.1 Passenger preferences

The decision of individuals to choose shared ride services depends on the price discount, additional travel time and willingness to share the ride. Furthermore, in modeling travel choice behavior, Random Utility Models (RUM) is the most common demand method, where the user knows/considers mutually exclusive alternatives, each associated with its perceived utility and evaluated by the pre-trip choice probability. Although dynamic vanpooling has gained attention, studies related to it, dynamic ridesharing, flexible transit or micro transit are still scarce, e.g., the Frei et al. (2017) evaluated the potential demand of a flexible transit using stated preference experiments and identified the potential users of this mode of transportation. The stated-preference (SP) experiments included choices between public transportation, private car and a flexible transit mode and estimated value of time from the obtained parameters of the choice models (16.3$ per hour for the car mode and 21.1$ for the flexible transit). It was found that respondents who commute with public transportation and have a bike-share membership are more likely to choose a more flexible transit mode. In Alonso-González et al. (2020), user preferences towards pooled on-demand services for time–reliability–cost trade-offs is analyzed by stated preference experiments. They evaluated the value of time (VOT) and value of reliability (VOR) of different trip stages, reporting in-vehicle VOT (7.88–10.80 €/hr), waiting VOT (9.27–16.50 €/hr), waiting to in-vehicle VOT ratio (between 1–1.5) and VOR/VOT ratio for both the waiting and in-vehicle stage to be around 0.5. In Kang et al. (2021), a joint revealed-stated preference model for choice between pooled and private ride-hailing is developed from a 2019 survey of Austin, Texas residents, reporting value of travel time and willing to pay extra to not pool as $27.80, $19.40 and $10.70 per hour and $0.62, $1.70 and $1.32 per hour for commute, shopping, and leisure travels. It was also found that women and older adults have a lower propensity to do pooled rides while individuals who are employed, highly educated and live-in high-density areas have a high propensity. Regarding preferences towards social aspects, Sarriera et al. (2017) reported that user attraction is the ease and speed of service compared to walking and public transport, safety is an important concern (women preferring same sex passengers), social interactions are relevant but not as much as traditional factors of time and cost and that negative experience is more deterrent than a positive social interaction experience. In Lavieri and Bhat (2019), it is also reported that a stranger presence is less critical than added detour time towards using a shared service. In Alonso-González et al. (2021), it is found out that less than one third of respondents have strong preferences against sharing their rides.

Further, to understand the effect of individual's demographics, Ko et al. (2021) stated that gender, car ownership and education can significantly affect the preference to use shared mobility services. Specifically, considering their perceived value of time, Atasoy et al. (2011) stated that attributes referred to attitudes influence the travel behavior of individuals. Based on a qualitative survey, revealed preferences (RP) survey and stated preference (SP) survey, they concluded that middle-aged users with high income and an active social life have a higher value of time. Then, considering the future of on-demand mobility with autonomous vehicles, Steck et al. (2018) investigated the effect of autonomous vehicles on the value of travel time savings (VTTS) and mode choice for commuting trips, comparing privately owned autonomous vehicles

and shared autonomous vehicles in stated preference experiments. The study indicated that the attributes of the alternatives such as in-vehicle travel time and cost significantly affect the mode choice and the sociodemographic characteristics play an important role in the modal split. In Kolarova et al. (2018), the value of time for autonomous vehicles using revealed and stated preference methods is estimated. Specifically, stated preference (SP) experiments were conducted, comparing autonomous vehicles, public transportation and private cars. The results show that in-vehicle travel time and travel cost play an important role on the mode decision. On the contrary, gender and age do not influence the preference of the users on the adoption of autonomous vehicles. Also, Krueger et al. (2016) examined the travel behavior of the respondents from the introduction of shared autonomous vehicles (SAV) by stated preference experiments comparing SAVs (with or without ridesharing) and public transportation. The model showed that waiting time plays a significant role in the choice of SAVs and young users are more likely to use the share service.

2.2.3.2 Market equilibrium

Cairns and Liston-Heyes (1996) developed an equilibrium model for the taxi market to understand the competition in the industry. It found that the unregulated industry does not satisfy the conditions of competition, and the existence of equilibrium depends on the regulation of price, entry, and intensity of use of licensed taxis. Besides, it also presented the models of monopoly, the social optimum and the second-best in the taxi industry. However, it did not consider the spatial difference in demand patterns. An initial attempt to model the taxi market at a network level considering the OD demand pattern was in Yang and Wong (1998). They then constantly improved this model in a series of works by further incorporating demand elasticity and congestion effect (Wong et al., 2001), exploring the impacts of regulatory restraints on the equilibrium (Yang et al., 2002). Furthermore, the improved equilibrium model was also applied to investigate the performance of nonlinear fare structures on the perceived profitability in Yang et al. (2010). Moreover, Li et al. (2019b) applied a queuing theoretic equilibrium model to assess the impact of regulations of transportation network companies (TNCs) on market supply and demand.

In some sense, due to their implicit resemblance, the exemplary works on the taxi ME can shed light on the research of equilibrium in the ridesharing markets. Adopting the modeling framework of previous works on the taxi industry, Ke et al. (2020) presented an equilibrium model for ridesharing markets and elucidated the complex relationships between system endogenous variables and decision variables (trip fare, vehicle fleet size, and allowable detour time). It proved that the monopoly optimum, first-best and second-best social optimum are always in the normal regime rather than the wild goose chase (WGC) regime[1]. However, they restricted the problem in the situation with two passengers sharing a trip at most. In addition, the market was modeled at an aggregate level without considering the network structure and OD demand patterns.

Contrary to Ke et al. (2020), where the service provider is the service operator, Bimpikis et al. (2019) formulated the equilibrium state for a matching agency. It pointed out that only when the demand pattern[2] across the network is balanced the benefit of applying spatial price discrimination can be observed. Leveraging the spatial pricing method can facilitate the pattern

[1] The wild goose chase regime is an inefficient equilibrium where vehicles take substantial time to pick up riders. We refer the interested readers to Ke et al. (2020) for the details of its definition and the corresponding analysis

[2] Demand pattern of a network is defined as a combination of a demand vector for zones and a weighted adjacency matrix. And it is said to be balanced if, at each zone, the potential demand for rides weakly exceeds the available drivers in the same zone after completing rides.

of the served demand becomes more balanced. The result of numerical experiments implied that the total profit and consumers' surplus are maximized at the equilibrium with the optimal pricing policy when the demand pattern of network is balanced.

Although the models developed in the aforementioned works perform well, there is still room for improvement. None of them consider passenger preference in the presence of multiple transport modes. The value of time (or willingness to pay) of passengers is the only factor to be considered in their modeling framework regardless of the service attributes of other transport options. This will result in an inaccurate demand estimation when the service attributes of ridesharing become incomparable with that of the other transport modes. Besides, there is no work established a network-based equilibrium model for ridesharing markets that can capture the travel demand patterns and network characteristics of the specific market.

2.2.3.3 Pricing

Demand estimation is a main focus of all dynamic pricing strategies for ridesharing services. Some aim to capture the temporal elasticity of demand to provide optimal solutions for a specific objective (e.g., profit maximization) (Sayarshad and Chow, 2015; Qian and Ukkusuri, 2017). Some try to improve the reliability of the proposed solution by considering the spatial heterogeneity of demand over the network (Chen and Kockelman, 2016; Guo et al., 2017; Qiu et al., 2018; Bimpikis et al., 2019). Furthermore, the users' heterogeneity, which is represented by passenger preference/behavioral models, is also an aspect that has been heavily researched in the literature (Chen and Kockelman, 2016; Qiu et al., 2018; Guan et al., 2019a).

Sayarshad and Chow (2015) proposed a non-myopic pricing method for the non-myopic dynamic dial-a-ride problem to maximize social welfare under the assumption of elastic demand. It pointed out that ignoring the elasticity of demand can result in an overestimation of the improvement in LoS with non-myopic considerations. Motivated by the demand elasticity among a day, Qian and Ukkusuri (2017) developed a time-of-day pricing scheme to maximize the profit for taxi service, where price multipliers are used to dynamically alter trip cost. It concluded that a strict pricing scheme should consider both temporal heterogeneity and spatial heterogeneity in demand, supply and traffic condition, together with additional consideration of users heterogeneity in price elasticity.

The Multinomial Logit (MNL) model was applied to estimate the mode share of shared autonomous electric vehicle (SAEV) in an agent-based framework in Chen and Kockelman (2016). It investigated the trade-offs between the revenue and mode share of SAEV under different pricing schemes including distance-based pricing, origin-based pricing, destination-based pricing, and combination pricing strategy. Guo et al. (2017) provided an elaborated demand analysis and dynamic pricing analysis of the ride-on-demand service provided by Shenzhou Ucar in Beijing, China. They adjusted the trip price dynamically by applying appropriate pricing multipliers for different regions based on the demand characteristics in both spatial and temporal dimensions. Knowing the passenger preference, demand distribution, and traffic information of the network, Qiu et al. (2018) proposed a dynamic programming framework to solve the profit maximization problem for a monopolistic private shared mobility-on-demand service (SMoDS) operator. And the MNL model was used to model the passenger preference and was integrated into the price optimization model at the request level.

With the consideration of demand difference over the network, Bimpikis et al. (2019) established an infinite-horizon, discrete-time model for ridesharing services, and explored the impact of the demand pattern on platform's prices, profits, and the induced consumer surplus. Furthermore, considering the uncertainty of travel time and waiting time in SMoDSs, Guan et al. (2019a) applied the Cumulative Prospect Theory (CPT) to capture the subjective decision mak-

ing of passengers under uncertainty. A dynamic pricing strategy was proposed on the passenger behavioral model based on CPT, which incorporates a dynamic routing algorithm proposed in Guan et al. (2019b) and thus can provide a complete solution to SMoDSs.

Note that, as a common phenomenon in the ridesharing services, however, the inequity existed among individual trips has not been considered in the existing literature. Thus, it is desirable to have a practical method to improve the LoS and equity of ridesharing services on the basis of the pricing method commonly adopted by the operators currently, e.g., distance-based unified pricing.

2.3 Dynamic OD estimation (DODE)

Dynamic Traffic Assignment (DTA) models contain three major sets of parameters i.e., supply, mobility demand and behavioral (route choice, departure time choice parameters). Supply and route choice parameters can differ among different transport modeling resolution (i.e., supply among macroscopic/mesoscopic and microscopic, and route choice among macroscopic and mesoscopic/microscopic), while demand parameters are consistent among all three resolutions (apart from aggregated macroscopic models). Demand models include Origin–Destination (OD) matrices and travel behavior model parameters (i.e., departure time choice). In a OD demand matrix each cell represents the number of trips traveling from a certain origin to a certain destination, while time–dependent OD matrices are basically a set of OD matrices for subsequent time intervals resulting from the combination of static OD demand and departure time choice model (both are same for all modeling resolutions). As discussed in chapter 1, mobility demand is unobservable and state of the art traffic measurement systems only measure the effect of the demand on the network, hence, conventional demand generation models are used which estimate the average demand. The daily demand patterns contain substantial fluctuations which are corrected by using time specific traffic measurements and this problem is referred as Dynamic OD Estimation (DODE).

Calibration of DTA models can be distinguished in two types depending on their modeling sensitivity temporally, i.e., 'off–line calibration' and 'on–line calibration'. Off–line DTA model calibration (i.e., by using database of historical information to represent average network conditions) is always needed. Then, these calibrated parameters can be further calibrated on–line for real–time applications concerned with system performance on a given day (Antoniou et al., 2009). Similarly, the estimation problem formulation can also be sub–divided into two major types: 1) Optimization based formulation 2) State space formulation. In an optimization–based formulation, the calibration problem finds the optimum set of model parameters that can minimize the error between simulation outputs and observed measurements. Such a problem formulation is solved using suitable optimization techniques e.g., gradient–based path searching algorithms (Gradient Descent, Finite–Difference methods) or Random search heuristics algorithm (Genetic Algorithm, Swarm Search). State space formulations setup calibration problem as a rolling horizon of states having two operations of state–to–state transition and state estimation/correction. In DTA model calibration, state space formulations are majorly used as an on–line approach for DODE with the aim to calibrate time–dependent ODs by setting a rolling horizon i.e. each time interval OD is a single state. Kalman filter have been used widely for DODE as state space formulation with different variants addressing non–linearity and targeting e.g. optimal solutions, data flexibility, time constraints etc.

Since DTA models are highly non-linear, the complexity of their calibration problem increases rapidly with the size of the transport network, limiting the application of conventional calibration approaches. For calibrating large–scale networks, extensions of existing calibration approaches are developed using dimension reduction techniques e.g., Principal Component Analysis (PCA)

(Djukic et al., 2012), which can help reduce the number of estimation parameters significantly with or without using historical estimates. The dimension reduction can help with both the problems of dimensionality and under–determination and reduce non–linearity by clustering orthogonal variables. The limitations of such dimension reduction techniques include in the assumption of either searching solution within the search space available in historical estimates (PCA) or pre–specifying model correlations e.g., constant trip generation–distribution correlation (quasi dynamic) (Cascetta et al., 2013). Apart from dimension reduction techniques other gradient–based algorithm extensions (e.g., W-SPSA (Antoniou et al., 2015), c-SPSA (Tympakianaki et al., 2015)) and response surface–based techniques e.g., metamodels (Zhang et al., 2017; Osorio, 2019a,b) have also been proposed as an alternative solution to approximate the calibration of large scale DTA models.

Following up, this section is further divided into three subsections, where the first subsection discusses the DODE problem formulation in detail describing all different problem formulations found in literature. Then, the next subsection focuses on all different groups of estimation methods used for DODE (including both optimization–based and state space solutions). Then, the last subsection covers the literature on dimension reduction techniques and their usage with different calibration approaches to help extend their application scalability.

2.3.1 DODE problem formulation

This section focuses majorly on the optimization–based formulation (while the state space formulation will be discussed in section 2.3.2.4 alongside its solution approaches). The optimization–based formulation requires specification of the objective function (also known as goal function), its variables (i.e., elements of the OD demand matrix and model parameters to be estimated), and its constraints related to feasibility and routing conditions. The aim of the estimation problem is to find the set of model parameters that minimizes the distances with respect to both the traffic measurements and the historical (prior) values of the model (Balakrishna, 2006).

$$\underset{\beta,x}{Minimise}\ z(\boldsymbol{y^{obs}}, \boldsymbol{y^{sim}}, \boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{x^p}, \boldsymbol{\beta^p}) \tag{2.1}$$

subject to:

$$\begin{aligned} \boldsymbol{y^{sim}} &= f(\boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{G}) \\ \boldsymbol{l_x} &\leq \boldsymbol{x} \leq \boldsymbol{u_x} \\ \boldsymbol{l_\beta} &\leq \boldsymbol{\beta} \leq \boldsymbol{u_\beta} \end{aligned}$$

Where:

- $\boldsymbol{y^{sim}}/\boldsymbol{y^{obs}}$ are the vectors of simulated and observed traffic measurements on the links.
- $\boldsymbol{x}/\boldsymbol{x^p}$ are the vectors of estimated/prior values of the demand.
- $\boldsymbol{\beta}/\boldsymbol{\beta^p}$ are the estimated/prior values of the model parameters (i.e., supply and behavioral parameters).
- $\boldsymbol{l_v}/\boldsymbol{u_v}$ are the lower and upper bound values of variable v.
- $\boldsymbol{G}$ is the transport network
- $z(\bullet)$ is an estimator used to measure the goodness of fit between observed/prior and estimated/simulated values.
- $f(\bullet)$ is the non–linear function mapping x and β to y^{sim}

Equation 2.1 depicts the minimization problem formulation, where the simulated traffic data $\boldsymbol{y^{sim}}$ are explicitly modeled through a (non–linear) function $f(\bullet)$. The optimization constraints for $\boldsymbol{x}$ and $\boldsymbol{\beta}$ help bound the search space and impose non–negativity in certain model parameters e.g., OD flows, link capacities etc. Further, the time–dependent representation of the model can be given as:

$$\boldsymbol{y_h^{sim}} = f(\boldsymbol{x_1}, ...\boldsymbol{x_h}; \boldsymbol{\beta_1}, ..., \boldsymbol{\beta_h}; \boldsymbol{G_1}, ..., \boldsymbol{G_h}) \tag{2.2}$$

Where, $\boldsymbol{y_h^{sim}}$ are outputs detected in time interval h and are a function of all OD flows $\boldsymbol{x}$, model parameters $\boldsymbol{\beta}$ and road network $\boldsymbol{G}$ till interval h.

OD estimation problem

In DTA model calibration, OD demand estimation problem typically dominates the calibration process due to its characteristics of being more dynamic, unobservable, underdetermined and dimensionally explosive (increase rapidly with network size and modeling duration). Traditional OD flows estimation techniques assume the function $f(\bullet)$ to be linear as given in equation 2.3.

$$\boldsymbol{y^{sim}} = A\boldsymbol{x} + \ \varepsilon \tag{2.3}$$

Where A is assignment matrix representing the assignment of OD flows on network links and ε is the observation error. In theory, the OD estimation problem can be considered as the reverse of the assignment problem – e.g., given the observed traffic data $\boldsymbol{y}$ calculate the most likely set of demand parameters $\boldsymbol{x}$. However, this formulation returns an infinite number of possible solutions as the problem is highly indeterminate (i.e., fewer equations than unknown). For Dynamic OD Estimate (DODE), time dependent mapping of OD is required and hence the simulated traffic measurements $\boldsymbol{y_h^{sim}}$ for time interval h is a function of model parameters from current interval h and all prior intervals. The assignment matrix–based function (equation 2.3) can be reconstructed with time–dependent linear approximations of assignment matrices A_h^p.

$$\boldsymbol{y_h} = \sum_{p=h-p'}^{h} A_h^p \boldsymbol{x_p} \tag{2.4}$$

The problem with this type of linear assignment function is the assumption that the link flows on a certain link during a time interval h can only be changed by changing one of the OD flows passing on that link in time interval h. As this is clearly not the case, using equation 2.4 to approximate the (non–linear) function $f(\bullet)$ makes the problem even more complex and ill predictable. Specifically, non–linearity influences the problem in two ways. First, this linear correlation does not consider the congested or uncongested state of the network. Second, the model does not consider spatial and temporal correlations between ODs. As congestion can propagate from one link to the other, we need to consider the correlations between ODs and network elements. Three main options are usually adopted to capture these non–linear relationships. One is to use different types of measures (such as speeds) to explicitly include information about the congested state of the link. A second option is to use variance and covariance matrices to explicitly map the correlation between variables. A third option is to use advance optimization techniques that include the hessian matrix – such as the Gauss Newton method, as the hessian matrix entails calculating 2nd order derivatives that can capture this correlation. Also, equation 2.4 restricts the estimation by use of traffic counts only, since the relationship of OD flow with other traffic measurements is non–linear and not possible to map use this method. Hence, the DTA model with time–dependent interaction between OD flows, model parameters, and the road network is required (as in equation 2.5).

DTA–based DODE estimation

To create a problem formulation for DODE, equation 2.1 can be reconstructed with time dependent variables having separate functions z_1, z_2, and z_3 measuring distance between OD flows, model parameters, and traffic measurements (equation 2.5).

$$\underset{\boldsymbol{x_h},\boldsymbol{\beta_h}}{minimize}[z_1(\boldsymbol{y_h^{obs}},\boldsymbol{y_h^{sim}});z_2(\boldsymbol{x_h},\boldsymbol{x_h^a});z3(\boldsymbol{\beta_h},\boldsymbol{\beta_h^a})] \tag{2.5}$$

Equation 2.5 represent the most general formulation for the DODE. However, the behavioral parameters $\boldsymbol{\beta}$ change less frequently than the demand values $\boldsymbol{x}$. Hence, for DODE the problem formulation can be redefined as in equation 2.6 keeping model parameters $\boldsymbol{\beta}$ constant within the estimation period.

$$\underset{\boldsymbol{x_h},\boldsymbol{\beta}}{minimize}[z_1(\boldsymbol{y_h^{obs}},\boldsymbol{y_h^{sim}});z_2(\boldsymbol{x_h},\boldsymbol{x_h^a})] \tag{2.6}$$

subject to:

$$\boldsymbol{y_h^{sim}} = f(\boldsymbol{x_1},...,\boldsymbol{x_h};\boldsymbol{\beta};\boldsymbol{G_1},...,\boldsymbol{G_h})$$

Where z_1 and z_2 are two goodness of fits that measure the distance between the OD flows and traffic measurements separately and the objective is to minimize the combined error. Equation 2.6 estimates the most likely set of OD flows given a set of observed traffic counts $\boldsymbol{y^{obs}}$ and a prior (or historical) demand $\boldsymbol{x^a}$. Simply stated, among all the infinite solutions, the model will return the optimal demand $\boldsymbol{x*}$ that is close to its prior estimates $\boldsymbol{x^a}$ such that the error in the traffic counts ($\boldsymbol{y^{obs}}$ and $\boldsymbol{y^{sim}}$) is also minimized. Additionally, differently from equation 2.1, equation 2.5 and 2.6 allows for different estimators $z_1(\bullet)$ and $z_2(\bullet)$ for different types of data. This is particularly important when dealing with different data sources (such as speeds and counts) which might have a different impact on the objective function if treated jointly.

Utility–based problem formulation

Apart from the above–mentioned problem formulations for DODE, Cantelmo et al. (2018) also proposed a Utility based estimation approach to solve DODE problem by estimating activity–based departure time choice models. Previous formulations assume that the set of behavioral parameters $\boldsymbol{\beta}$ mostly includes routing strategies and supply–parameters. Departure time choice models also use a set of behavioral parameters $\boldsymbol{\beta_T}$ to model the decision to travel during a certain time interval h. We can then rewrite the function $f(\bullet)$ as a function of both $\boldsymbol{\beta}$ and $\boldsymbol{\beta_T}$:

$$\boldsymbol{y_h^{sim}} = f(\boldsymbol{x_1}(\boldsymbol{\beta_T}),...,\boldsymbol{x_h}(\boldsymbol{\beta_T});\boldsymbol{\beta};\boldsymbol{\beta_T};\boldsymbol{G_1},...,\boldsymbol{G_h}) \tag{2.7}$$

Equation 2.7 shows that the time–dependent demand variable $\boldsymbol{x_h}$ depends directly on the variable $\boldsymbol{\beta_T}$, which is intuitive as $\boldsymbol{\beta_T}$ dictates in which time interval the user decides to travel. The relationship between $\boldsymbol{x_h}$ and $\boldsymbol{\beta_T}$ can be described as:

$$\boldsymbol{x_h} = P_h(\boldsymbol{\beta_T})\boldsymbol{x} \tag{2.8}$$

Where $P_h(\boldsymbol{\beta_T})$ represent the probability to depart during time interval h and is usually calculated by discrete choice models and utility maximization theory (therefore the term – utility–based framework). $\boldsymbol{x} = \sum_h \boldsymbol{x_h}$ represents the static (i.e., non–time dependent) value of the mobility demand as defined in equation 2.1. By combining equation 2.9 and equation 2.8, equation 2.6 can be rewritten as:

$$\underset{\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\beta_T}}{minimize}[z_1(\boldsymbol{y_h^{obs}}, \boldsymbol{y_h^{sim}}); z_2(\boldsymbol{x_h}, \boldsymbol{x_h^a}); z_3(\boldsymbol{\beta}, \boldsymbol{\beta^a})] \tag{2.9}$$

subject to:

$$\boldsymbol{y_h^{sim}} = f(\boldsymbol{x}; \boldsymbol{\beta}; \boldsymbol{\beta_T}; \boldsymbol{G_1}, ..., \boldsymbol{G_h})$$

The main advantage of equation 2.9 is that there is no need to model the demand as a time–dependent variable, as the correlation between $\boldsymbol{x}$ and $\boldsymbol{x_h}$ is explicitly modeled through $\boldsymbol{\beta_T}$ (equation 2.8). This also means that number of variables to be calibrated decreases drastically. Additionally, time–dependent demand flows are now temporally correlated which, together with the lower number of variables, help creating a more robust optimization framework. Unfortunately, this comes with a significant cost in terms of computational time, as solving equation 2.9 requires many DTA simulations.

Sequential versus simultaneous estimation

The DODE problem formulation can be stated using two different calibration methods i.e., simultaneous estimation and sequential estimation (Cascetta, 2009). Simultaneous estimation means to calibrate the OD demand for all the time intervals together as a single problem formulation (equation 2.10).

$$\boldsymbol{x_1^*}, ..., \boldsymbol{x_h^*} = \underset{\boldsymbol{x_1} \geq 0, ..., \boldsymbol{x_h} \geq 0}{argmin} [z_1(\boldsymbol{x_1}, ..., \boldsymbol{x_h}; \boldsymbol{x_1^a}, ..., \boldsymbol{x_h^a}) + z_2(\boldsymbol{y_1^{obs}}, ..., \boldsymbol{y_h^{obs}}; \boldsymbol{y_1^{sim}}, ..., \boldsymbol{y_h^{sim}})] \tag{2.10}$$

Sequential estimation means to calibrate the OD demand sequentially for each time interval i.e., for a given time interval h, the formulation should be given as:

$$\boldsymbol{x_h^*} = \underset{\boldsymbol{x_h} \geq 0}{argmin}[z_1(\boldsymbol{x_h}; \boldsymbol{x_h^a}) + \ z_2(\boldsymbol{y_h^{obs}}; f(\boldsymbol{x_h}/\boldsymbol{x_1^*}, ..., \boldsymbol{x_{h-1}^*}))] \tag{2.11}$$

where $\boldsymbol{x_h^*}$ is the OD demand estimated for time interval h and $\boldsymbol{y_h^{sim}}$ is considered correlated to the demand of current time interval $\boldsymbol{x_h}$ and estimated demand of all previous time intervals $(\boldsymbol{x_1^*}, ..., \boldsymbol{x_{h-1}^*})$. The advantage of using sequential estimation approach is to reduce the number of estimation variables and correlations, reducing the computational complexity. Also, the demand estimated in one interval can be used as initial estimates for a subsequent interval (Cascetta, 2009).

2.3.2 Estimation methods

2.3.2.1 Gradient based algorithms

Gradient based algorithms are iterative algorithms which minimize a given problem formulation with search directions defined by gradients evaluated at each iteration. A standard form of gradient based algorithm can be given as:

$$x_{k+1} = x_k - a_k {g'}_k(x_k) \tag{2.12}$$

Where, x_{k+1} is the estimated variable minimized based on the evaluated gradient g'_k and a predefined step size a_k. Also, k defines the iteration number. A classic first order gradient–based algorithm e.g., Gradient Descent (Ruder, 2016) requires differentiable function to find their

analytical gradients and their application on DTA problem formulations would need second order derivatives to be able to capture the involved non–linearity. Hence, such algorithms that rely on exact knowledge of analytical gradient are not suitable for DTA model calibration. Another form of gradient based algorithms is the Stochastic Approximation (SA), which considers stochastic approximations of the gradient. Finite Difference Stochastic Approximation (FDSA) is one of the first type of first order SA algorithms having stochastic gradient approximations that partially considers the presence of non–linearity (Kiefer and Wolfowitz, 1952). Equation 2.13 gives the mathematical form of FDSA's gradient evaluation.

$$\boldsymbol{g'_k}(\boldsymbol{x_k}) = \begin{bmatrix} \frac{z(\boldsymbol{x_k}+c_k\boldsymbol{\xi_1})-z(\boldsymbol{x_k}-c_k\boldsymbol{\xi_1})}{2c_k} \\ \cdot \\ \cdot \\ \frac{z(\boldsymbol{x_k}+c_k\boldsymbol{\xi_n})-z(\boldsymbol{x_k}-c_k\boldsymbol{\xi_n})}{2c_k} \end{bmatrix} \tag{2.13}$$

Where, $\boldsymbol{\xi_i}$ is a n–dimensional vector with 1 at i^{th} place and else 0, while n is the size of the decision variable and c_k is the perturbation size. Gradients for each element i requires two objective function evaluations $z(\bullet)$, hence, $2n+1$ evaluations are required within one iteration (Spall, 2003). Although, FDSA caters for non–linearities present in DTA model calibration, the number of evaluations required for each iteration makes it infeasible (computationally expensive) especially on medium to large scale networks.

Simultaneous Perturbation Stochastic Approximation (SPSA)

SPSA (Spall, 1998b) is another stochastic approximation algorithm used traditionally for large scale non–linear problems having expensive objective function evaluations. Compared to other gradient based algorithms like Gradient Descent and FDSA, SPSA has the advantage for requiring only two objective function evaluations. Equation 2.14 gives the mathematical formulation of SPSA's gradient evaluation.

$$\boldsymbol{g'_k}(\boldsymbol{x_k}) = \frac{z(\boldsymbol{x_k}+c_k\boldsymbol{\Delta_k})-z(\boldsymbol{x_k}-c_k\boldsymbol{\Delta_k})}{2c_k}\,[\Delta_1,\Delta_2,\ldots,\Delta_n]^T \tag{2.14}$$

Where, n is the size of the decision vector $\boldsymbol{x_k}$ and $\boldsymbol{\Delta}$ is a n–dimensional vector generated randomly from a ± 1 Bernoulli distribution with zero mean. SPSA perturbs the whole decision vector $\boldsymbol{x_k}$ twice, simultaneously (with $\boldsymbol{\Delta_k} \times c_k$) to evaluate a noisy gradient at each iteration. Although, SPSA gradient approximation is much more random and noisier than FDSA and hence would require a greater number of iterations to converge the error, the overall number of objective function evaluation are far less (Spall, 1998b). SPSA algorithm is rather simple and computationally efficient algorithm but being stochastic its performance greater depends on the definition of its gain sequence parameters at initialization.

Variants of SPSA

In the context of DTA model calibration, SPSA is a widely adopted algorithm and was first proposed by Balakrishna (2006). Along being simple and computationally efficient it allows the use of any traffic measurement due to its generic problem formulation. Balakrishna et al. (2007b) used multiple gradient replications and the scaling of gain sequence parameters to cater for different magnitude of OD pairs and avoid high gradient values. Then, Cipriani et al. (2011) proposed asymmetric differencing for gradient approximation to reduce the required function evaluations in expense of added bias. Spall (2000) proposed adaptive SPSA based on Hessian estimates to

automatically scale the estimation variables with adaptive scaling and shifting of gain sequence parameters. These modifications in SPSA were mostly aimed on different gradient approximation and gain sequence scaling methods to improve convergence or required computation efforts but even along them, SPSA's performance is limited to small networks with lesser non–linearity and number of calibration parameters. Due to its random stochastic behavior, SPSA application on larger networks requires further improvements such as by adding network correlation information to reducing problem non–linearity or decompose the problem in sub–problem/clusters to cater for dimensionality issues. Cantelmo et al. (2014a) proposed the Adaptive SPSA (A–SPSA), an adaptation of the second order SPSA proposed by Spall (2000) that specifically targets the calibration problem. The second order SPSA exploits the Hessian matrix to capture correlations between variables, using the following equation to update the current solution.

$$x_{k+1} = x_k - w_k^{-1} g'_k(x_k) \tag{2.15}$$

Where w_k^{-1} is a vector with dimension $[n \times 1]$, with n the number of decision variables. In the original version proposed by Spall (2000), w_k^{-1} is a function of the Hessian matrix. However, as the calculation of the Hessian matrix is computationally unfeasible in the case of OD estimation, the A–SPSA proposed by Cantelmo et al. (2014a) uses the assignment matrix A_h^p to approximate the Hessian matrix. The main advantage is therefore that the computational complexity of the A–SPSA is approximately the same as the one of the SPSA. Similarly, Lu et al. (2015) proposed weighted SPSA (W–SPSA) to incorporate network information through adding a weight matrix. This weight matrix w includes spatial (network topology, traffic conditions and driver's route choice behavior) and temporal (departure time choice and travel times information) correlation information between calibration model parameters and traffic measurements. Equation 2.16 depicts the definition of w, where $h = 1, 2, \ldots, H$ are time intervals, $p = 1, 2, \ldots, P$ are model parameters and $m = 1, 2, \ldots, M$ are traffic measurements.

$$W = \begin{bmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,M} & \cdots & W_{1,M\times H} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,M} & \cdots & W_{2,\ M\times H} \\ \vdots & \vdots & & \vdots & & \vdots \\ W_{P,1} & W_{P,2} & \cdots & W_{P,M} & \cdots & W_{P,M\times H} \\ \vdots & \vdots & & \vdots & & \vdots \\ W_{P\times H,1} & W_{P\times H,2} & \cdots & W_{P\times\ H,M} & \cdots & W_{P\times H,M\times H} \end{bmatrix} \tag{2.16}$$

The correlation weight matrix helps in improving the gradient approximation by reducing the noise generated by uncorrelated measurements (see equation 2.17), especially for sparsely correlated large–scale networks with many time intervals. Antoniou et al. (2015) demonstrated the practical implications of W–SPSA by exploring multiple techniques for evaluating an effective weight matrix. The W–SPSA and A–SPSA are conceptually similar with a difference that the A–SPSA requires the knowledge of the assignment matrix, as this is used during each iteration of the model to calculate a new set of weights. The W–SPSA, on the other hand, can be used with any traffic simulator as it does not require to calculate the assignment matrix.

$$g'_k(x_k) = \frac{y(x_k + c_k\Delta_k) - y(x_k - c_k\Delta_k)}{2c_k\Delta_k} W \tag{2.17}$$

Tympakianaki et al. (2015) proposed cluster–wise SPSA (c–SPSA), which subdivides the set of estimation model parameters (OD flows) into homogenous clusters c and approximate each cluster's gradient separately to reduce gradient bias. The idea of clustering acts as a hybrid of

FDSA and SPSA, increasing the number of objective function evaluations from 2 (traditional SPSA) to $2 \times c$ (FDSA requires $2 \times n$, means if $c \rightarrow n$, c–SPSA is same as FDSA). Equation 2.18 gives the mathematical gradient evaluation for c–SPSA with three different OD clusters L (low), M (medium), H (high).

$$g'_k(x_k) = \begin{bmatrix} \frac{y^+_{kL} - y^-_{kL}}{2c_{kL}\Delta_{kL}} \\ \frac{y^+_{kM} - y^-_{kM}}{2c_{kM}\Delta_{kM}} \\ \frac{y^+_{kH} - y^-_{kH}}{2c_{kH}\Delta_{kH}} \end{bmatrix} \tag{2.18}$$

For clustering techniques, Tympakianaki et al. (2015) proposed the magnitude–based clustering using k–means clustering algorithm which improves the gradient bias by only reducing problem dimensions. Then, Tympakianaki et al. (2018) proposed another clustering technique based on spatial interactions of OD pairs computed by network travel times. This clustering technique additionally helps to reduce the non–linearity of the estimation problem by clustering non–correlated variables together as a cluster. As much as the variables within a cluster are uncorrelated, the lesser will they have non–linearity present and robust will the SPSA perform.

Recently, Qurashi et al. (2019) proposed PC–SPSA, combining Principal Component Analysis (PCA) with SPSA to reduce the problem dimensions into a much lower dimensional space of their Principal Components (PCs). PC–SPSA replaces the direct estimation of model parameters with estimation of their orthogonal PCs, reducing the number of estimation parameters manyfold, restricting the SPSA search space within the variance present in its historical estimates and minimizing the presence of non–linearity due to presence of orthogonality in PCs. Further details about PC–SPSA and its comparison with other SPSA variants are described in section 2.3.3.

2.3.2.2 Random search algorithms

Random search algorithms use probabilistic methods to randomly update decision variables towards the optimum solution. Being gradient free, these algorithms are useful for optimizing problems which are not continuous and differentiable. Different random search algorithms especially Genetic Algorithm (GA) (Holland et al., 1992) has been applied for solving DTA model calibration problem with optimization–based formulation. A standard genetic algorithm works on evolutionary search and natural selection concept. A population of chromosomes (solutions) is evolved iteratively based on three genetic operations i.e., selection, mutation, and crossover. A random starting population is generated within the predefined search space and their objective function is evaluated. Chromosomes which seem more fitted are selected and crossed over in pairs to start a new generation of solutions which are further mutated (perturbed) to increase randomness.

For DTA model calibration GA have been proposed frequently in literature e.g., Kim et al. (2005), and Chiappone et al. (2016) used GA for supply model calibration; and Stathopoulos and Tsekeris (2004), Kattan and Abdulhai (2006), Vaze et al. (2009) applied GA for demand calibration. GA is naturally suited for integer variables and differ fundamentally from other optimization methods that can perform better for continuous problems formulations. It discretizes the search range of all calibration (continuous) variables creating numerous possibilities, especially not suitable for DODE (due to randomness i.e., exploring all possibilities instead of starting from a previous estimate as in gradient–based algorithms). Also, the longer run times and the need to specify many hyper parameters critical to the algorithm's performance (i.e., search space definition, selection method, and crossover and mutation probabilities) makes GA implementation computationally very expensive (Henderson and Fu, 2004).

Different variants of GA have also been proposed recently to cater for its limitations e.g., Omrani and Kattan (2013) implemented Distributed GA (DGA) to do DTA model calibration, the evaluation of each population is distributed/parallelized on multiple parallel computing sources. Cobos et al. (2016) used a combination of Non–dominated Sorting Genetic Algorithm–II (NSGA–II) (for global search) and Simulated Annealing (SA) (for local search) named as NSGA–II–SA. Also, Zhu et al. (2021) used Island GA (IGA) for calibrating link speeds, IGA uses isolated islands of population maintaining genetic diversity due to independent evolution and evolves much faster than standard GA. Although, different variants help improve the computational performance of GA, still due to its characteristics, GA is mostly suitable only in cases where other calibration algorithms cannot be considered for DTA model calibration.

2.3.2.3 Response surface methods

Response surface methods (RSM) aim to learn/replicate the behavior of underlying DTA simulations by fitting a response surface (as called metamodel or surrogate model). The surface is an analytical representation defined using a differentiable functional form useful to leverage derivative–based optimization techniques. Response surface is fitted based on the objective function evaluations capturing the simulation's input–output relationship. These points of objective function evaluations can either be the results from optimization iterations or by additional sampling strategies (depending on the RSM) (Zhang et al., 2017). RSM algorithms differ for being either regression or interpolation based, nature of functional forms used (e.g., polynomial, Kriging, neural networks), and sampling and response surface update strategy. The literature under RSM is rather vast but two of its types i.e., Metamodel–based optimization and Bayesian optimization have been used frequently to solve DTA model calibration problem with the idea of calibrating large scale model with lesser computational effort.

Metamodel methods are indirect gradient methods which optimize the problem using gradient of deterministic functions (metamodel). The simulation based objective function f (equation 2.2) is replaced by its analytical approximation m_k (as in equation 2.19). Zhang et al. (2017) gives a mathematical form for metamodel based optimization formulation as:

$$\underset{\theta \in \Theta}{Min} m_k(\theta; \beta_k) \tag{2.19}$$

Where function m_k is the metamodel i.e., a parametric function, with its iteration–specific parameter vector as β_k (fitted often based on simulation observation). A metamodel based optimization technique consists of two major step which are revised iteratively until a good convergence is achieved. A metamodel is created based on the available sample simulation observations (step 1), then the optimization is performed on the analytical function to get the trail point for least error value (step 2). The trial point is evaluated by simulation, resulting availability of new simulation observations. New simulation observations are used to improve metamodel fitting/accuracy (step 1) followed by its optimization to get a better trail point (step 2). A comprehension review of metamodels is given in Pisano (2010).

Zhang et al. (2017) proposed a metamodel based DTA model calibration technique specifically for large–scale traffic simulators. The proposed analytical metamodel consist of both functional (general purpose) and physical metamodels (problem specific functions), where physical metamodels are used to capture the structural information. Later, Osorio (2019a) also proposed a similar metamodel based technique with addition of a sampling strategy within each iteration for better model fitting (strategically choosing observation points) for converging to the optimum solution in lesser number of iterations. Also, Osorio (2019b) proposed another metamodel based calibration technique for offline calibration problems which use a single metamodel for whole network instead of individual link based meta models. The use of single metamodel can further

improve the scalability and computational efficiency of the metamodel calibration techniques. Apart for these, Cheng et al. (2019) also proposed a surrogate–based simulation optimization technique for calibrating route choice parameters, where the kriging metamodel is adopted to surrogate the optimization function of the calibration process.

Like metamodels, Bayesian optimization methods have also been used in literature. A Bayesian optimization method builds response surface using techniques such as Kriging/Gaussian process regression. For DTA models, Flötteröd et al. (2011) first, used Bayesian inference for DTA demand calibration which incorporate analyst's prior knowledge. Much recently, Schultz and Sokolov (2018) proposed a Gaussian Process Bayesian framework optimizing macroscopic traffic simulators which later is also applied by Sha et al. (2020) on microscopic traffic simulation using acquisition functions to determine more promising values for future evaluation.

2.3.2.4 State space approaches

State space formulations setup the calibration problem as a rolling horizon of states having two operations of state–to–state transition and state estimation/correction. In DTA model calibration, state space formulations are majorly used as an on–line approach for DODE with the aim to calibrate time–dependent ODs by setting a rolling horizon i.e. each time interval OD is a single state. The formulation is composed of three main elements: 1) the state vector, i.e., the set of variables that can uniquely describe the evolution of the system, 2) the transition equation which models the evolution of the system over time and, 3) the measurement equation which maps the available traffic data to the state vector. These state space formulations are solved using Kalman Filter (KF) and the most prominent state–space model formulation is done in terms of traffic state deviations (Ashok and Ben-Akiva, 2000). The use of deviations instead of OD flows allows the model to consider the historical structure of the demand matrix during the estimation.

The transition equation obtains estimates of state vector at each step through the autoregressive process, representing the "expected" evolution of the system based on historical information. While the measurement equation measures the influence of the additional available information – i.e., the traffic counts – on the system. The Kalman Filtering approach assumes two Gaussian distributions at each step, one for the predicted state (transition equation) and the other for the observed data (measurement equation). For each time interval update, a weighing matrix so–called 'Kalman Gain' is estimated using the state vector covariance matrix and the assignment matrix. Then, by combining the Kalman Gain with the prediction from the transition equation, the most likely values of the state vector are estimated according to both the measurement and transition equations.

Variants of Kalman filter

The original Kalman filter theory applies to linear systems. However, the DODE problem is non–linear by nature and hence several solutions for non–linear models have been proposed in the literature (Antoniou et al., 2007a). Extended Kalman Filter (EKF) is the most straightforward extension of Kalman filter, which involves using a first order Taylor series expansion to locally linearize the function around the current value of the estimate. Since the in–direct measurement equation does not (in general) have an analytical expression, analytical derivation is not possible and numerical methods are needed (which gets computationally expensive). Moreover, first order expansion is not sufficient to capture highly non–linear systems. Limiting Extended Kalman Filter (L-EKF) (Antoniou, 2004) is an approximated version of the EKF focusing on reducing the computational effort. Standard EKF calculates the kalman gain on–line i.e., computationally

demanding requiring linearization of the measurement equation. In the L–EKF, the kalman gain is calculated off–line based on historical data.

Similarly, the Unscented Kalman Filter (UKF) (Antoniou, 2004) is an alternative filter that can be used to model highly non–linear systems. While in EKF, the state distribution is approximated by a random variable which is then analytically propagated through the first order linearization of the non–linear system (inaccurate for highly non–linear functions). UKF uses a deterministic sampling approach (Unscented Transformation, UT) to overcome this issue and represent the state distribution using a (small) number of deterministically selected sample points (sigma points). These points can capture the posterior mean and covariance accurately to the second order (Taylor series expansion) for any non–linearity.

Moreover, Local Ensemble Transformed Kalman Filter (LETKF) applied by Carrese et al. (2017) is based on the family of Ensemble Kalman filters (EnKF). EnKF chooses an ensemble of initial conditions around the current estimate and propagates each ensemble member based on a non–linear model. Thus, the uncertainty of the estimation is propagated from one time interval to the next and the ensemble is used to parameterize the distribution of the state variables. LETKF is a specific type of Kalman Filter that does not requires to explicitly calculate the Kalman Gain and hence no calculation of derivatives. LETKF avoids the linearization of the dependency between OD flows and observed measurements, but implicitly captures this correlation through a traffic simulator rather than through an analytic formula. The main issue with the LETKF is the definition of the initial ensemble, LETKF requires k function evaluations, where k is the number of ensembles and for highly non–linear systems, a large set of ensembles is needed to ensure that the set of conditions captured by the ensemble is statistically representative.

2.3.3 Dimension reduction

For DODE, problem dimensionality and non–linearity is a major issue limiting most algorithm's scalability to smaller networks. The increase in network size mainly increases the size of the OD matrix x and its correlation with the number of plausible network routes/links (size of assignment matrix A) resulting the traffic measurements y (equation 2.3). OD matrices are high dimensional multivariate structures, and their estimation complexity is measured mainly by three factors: 1) OD matrix size 2) Complexity of model (assignment matrix etc.) 3) Number of traffic measurements. Increase in any of these factors add more burden on calibration algorithms both computationally and methodologically. For example, the performance of SPSA algorithm generally deteriorates (slower and less convergence) as the network size increase. And, similarly, for Kalman filter algorithms, where the computational complexity is typically in the order of $O(n)^3$ (where n is the OD pairs), the increase in network size dramatically increase the required computational effort.

From all the previously stated calibration algorithms, it is rather clear that any form of problem dimension reduction can help improving the application scalability of DTA model calibration. For example, c-SPSA (Tympakianaki et al., 2015), which is a variant of SPSA helps in catering the dimension increase by clustering of the estimation variables, where each cluster is estimated separately (ideally the cluster's should have minimum correlation among themselves i.e., each cluster's ODs correlate minimum with other cluster's assignment matrices, to reduce maximum amount of model complexity/non-linearity). Although, the clustering does help improving the scalability methodologically, it still somewhat requires similar computational effort (i.e., SPSA effort $\times$ clusters) limiting direct computational advantage which may be otherwise achieved with other dimension reduction techniques. In literature, the most widely used dimension reduction technique is Principal Component Analysis (PCA) which reduces the problem dimensions by leveraging on the patterns present within historical data. Apart from PCA based dimension

reduction, Quasi dynamic approaches are also proposed which does assumption-based dimension reduction (e.g., assuming the correlation between trip generation-distribution) without any requirement of historical data.

2.3.3.1 PCA–based approaches

Principal Component Analysis (PCA) is proposed by Pearson (1901) and Hotelling (1933) to describe the variation in a multivariate dataset with a set of uncorrelated variables. Djukic et al. (2012) proposed the application of PCA for OD demand estimation to significantly reduce the problem dimensions and required computational efforts. A set of historical OD estimates is decomposed into orthogonal Principal Components (PCs) explaining the present spatial and temporal variations. Later, the set of estimation variable or OD matrix is reduced to a set of PC scores in a lower dimension space defined by these orthogonal PCs. The application of PCA helps in both reducing the number of estimation variables and making them uncorrelated (orthogonal) to help cater for both dimensionality and model non-linearity/complexity. The detailed procedure of applying PCA on DODE problem is discussed in chapter 4.

Since the application of PCA on DODE by Djukic et al. (2012), several different approaches are developed combining conventional algorithms with PCA's dimension reduction. Starting with Djukic et al. (2012), who considered the dimensionally reduced PC scores as the fixed structure of OD demand instead of the OD matrix and proposed to update them online through the given state space model solved with colored noise Kalman filter algorithm accounting for temporal correlated measurement noise. Then, Prakash et al. (2017) proposed PC-GLS, estimating the PC scores using traditional GLS approach. Similarly, Prakash et al. (2018) proposed PC-EKF, solving the PC scores–based state–space formulation using Constrained Extended Kalman Filter which can incorporate non-linear relationships between parameters and traffic measurements. Much recently, Qurashi et al. (2019) proposed PC-SPSA, estimating the PC scores–based DODE formulation using SPSA. Lastly, Castiglione et al. (2021) proposed PC-LETKF, estimating OD demand PC scores with Local Transformed Ensemble Kalman Filter (LETKF). PCA's application with LETKF shows better results than Ensemble Kalman Filter (EnKF) and LETKF, especially when calibrating with small ensembles.

PCA's application with SPSA is significant, considering SPSA is arguably the most popular assignment matrix–free method due to its ability to deal with non–linear and stochastic systems, a generalized problem formulation, and ease of implementation. The application of PCA enhancing SPSA application limits on both larger DTA models and possibly for online calibration. PC-SPSA outperforms most of the other SPSA variants (under certain conditions) due to added characteristics by PCA dimension reduction. The notable variants (i.e., c-SPSA, W-SPSA, A-SPSA, etc.) either aim to reduce dimension burden or non-linearity on the gradient estimation by clustering (c-SPSA) or adding correlation information (W-SPSA, A-SPSA). PC-SPSA, using PC-scores does all together, i.e.:

- dimensions are reduced manifolds (e.g., Qurashi et al. (2019) showed a 40 times reduction 3249 OD pairs to 80 PC scores)
- all PCs are orthogonal to each other having no correlation among them
- information about OD pairs temporal and spatial correlation is captured in PC-directions

2.3.3.2 Quasi dynamic approaches

Quasi-dynamic approaches aim to reduce the dimension of estimation variables (OD flows) based on the assumption that a correlation between trip generation and trip distribution exists. The

advantage of quasi-dynamic estimation approaches is that it does not require any extra information (i.e., historical OD information) to reduce the problem dimensions while the limitation is that the estimation accuracy strongly relies on how realistic is the underlying assumption. While in PCA–based methods, the dimensionality reduction is data driven, in the Quasi-Dynamic (QD) it is assumption–based. Cascetta et al. (2013) is the first one to propose a quasi-dynamic based approach for off-line estimation of OD flows named 'QD-GLS' (Quasi-Dynamic Generalized Least Squares). The QD assumption assumes that changes in the trip generation (i.e. the amount of trips traveling from one origin) are more frequent that changes in the trip distribution (i.e. the amount of trips traveling from an origin to a destination). In addition of directly estimating OD flows by quasi-dynamic formulation, Cascetta et al. (2013) also showed that use of quasi-dynamic estimates instead of simultaneous estimates as historical OD flows significantly improve the quality of the estimation seed matrix. Later, Marzano et al. (2018) developed the similar quasi-dynamic based framework for online OD estimation under state space formulation using EKF (QD-EKF). QD-EKF extended the quasi-dynamic application for online estimation. Also, Cantelmo et al. (2014b) developed a similar quasi-dynamic framework named as the two-step approach, where the first step is to do quasi-dynamic estimation for getting reasonable generation values using FDSA and the second step estimating OD shares (removing the assumption) using SPSA. The main difference between these methods is how the Quasi-dynamic assumption is modeled. In Cantelmo et al. (2014b), the author proposes a two-step approach that iteratively correct trip-generation and trip-distributions. First, a strict QD assumption, where trip distribution is assumed to be constant, is adopted to drastically reduce the number of variables in the system and perform a broad exploration of the solution space. Then, in the second step, the quasi-dynamic assumption is removed, hence the conventional DODE problem is used to locally find the best solution. In Marzano et al. (2018), the relationship between trip-generation and distribution is explicitly modeled as a parameter. As a stronger assumption implies a stronger reduction in the dimension, the number of variables is set in order to balance known and unknowns.

3 Microscopic modeling and optimization of AMoD ridesplitting

Contents

With dynamic matching and detouring, ridesplitting requires extensive routing and demand assignment optimization to satisfy time constraints from multiple sharing passengers. Therefore, it is highly sensitive to the accuracy of representing dynamically changing information on network, demand, and service operations. This chapter introduces a novel framework for modeling AMoD ridesplitting systems in microscopic traffic model with simultaneously integrated DARP (Dial-a-Ride problem) algorithm. The platform allows detailed modeling of network dynamics, link-based service operations, and stochastic information modeling for DARP optimization, resulting in efficient modeling and optimization of AMoD ridesplitting operations.

The content of this chapter has been partially presented in the following works, while part of the content is unpublished to date:

Qurashi, M., Jiang, H., & Antoniou, C., 2022. Microscopic modeling and optimization of autonomous mobility on–demand ridesplitting, *(Submitted)*

Qurashi, M., Jiang, H., & Antoniou, C. 2020. Modeling autonomous dynamic vanpooling services in sumo by integrating the dynamic routing scheduler. In *Proceedings for SUMO User Conference, 2020.*

3.1 Introduction

3.1.1 Background and context

Although ride-hailing / taxi services have been more common, dynamic ridesplitting promises to be a more sustainable future mode with less economical and environmental imprints from daily travels. However, the practical implementations for on-demand ridesharing services have been prone to failure (Currie and Fournier, 2020) and require more robust models for both better optimization and assessment. Note that AMoD ridesplitting is more sensitive to changes in service operations, demand, and network conditions because the service vehicles tend to serve multiple passengers of different origin-destination and departure times with shared rides by adding detours in in-service passenger trips; therefore requires accurately predict trip travel times. The performance of such ridesharing systems depends on routing optimization (called 'Dial-A-Ride Problem' - DARP) because of the need to satisfy time constraints from multiple sharing passengers.

AMoD ridesplitting optimization refers to dynamic DARP, which optimizes the service vehicles during operation. The optimization is triggered at specific time intervals, events (i.e., request arrivals, vehicle service stop), or network conditions/incidents. Note that these DARP algorithms optimize the service upon the available/modeled information, and as already discussed in chapter 1, Ho et al. (2018) argued that all real-world DARP operations are stochastic and the stochasticity exists in all information aspects. Hence, the dynamic and stochastic DARPs (Xiang et al., 2008; Schilde et al., 2011, 2014; Li et al., 2019a) are deemed more suitable, incorporating the stochasticity of future information in DARP optimization (e.g., future user requests and stochastic travel times). To cater to the presence of stochasticity, another crucial aspect is modeling the service and network information through realistic/stochastic environments to better replicate real-world conditions. Since the dynamic DARPs solve routing problems dynamically, they use the available modeled information. Hence, an appropriate representation of the stochastic environment and operations is highly favorable for improving service optimization and having more reasonable service assessments.

To model the stochasticity, microscopic traffic models are highly suitable, providing realistic and dynamic traffic assignment with the interaction of detailed supply and demand models to better replicate dynamic traffic congestion. However, modeling AMoD ridesplitting with microscopic models is rather complex. For instance, the core of the service lies in the scheduling (DARP) algorithm, which needs to be integrated within the traffic model, simultaneously optimizing the service vehicles' routes and dynamic request assignments. Then, the service fleet characteristics that include vehicle driving behavior, dynamic rerouting, flexible stoppage, and idle waiting should be modeled during the simulation. Given the complexity and novelty of such systems, literature shows only a few major efforts to model AMoD ridesplitting systems. For example, Maciejewski et al. (2017), who developed a DRT simulation extension for MATSim (Horni et al., 2016) to model dynamic DARP, and Basu et al. (2018), who modeled on-demand SAVs (AMoD) in MT-level of Simmobility (Adnan et al., 2016). However, these efforts are demand-centric and model the traffic on mesoscopic resolution, compensating for traffic congestion's (sensitive/stochastic) modeling. Similarly, they use simplistic DARP optimization methods with deterministic information and simple heuristic methods.

Efforts focusing on modeling detailed traffic congestions also exist, e.g., Alam and Habib (2018); Alazzawi et al. (2018); Levin et al. (2017); Huang et al. (2021), but these efforts also limit the scope of AMoD ridesplitting by either limiting vehicle capacities or detours along with the employment of simple DARP optimization methods. Similarly, only a few literature efforts focus on modeling autonomous driving behaviors and their connected technologies for AMoD (i.e., Levin et al. (2017); Alazzawi et al. (2018)) or model link-based service operations (e.g.,

Alam and Habib (2018); Alazzawi et al. (2018)). Meanwhile, the literature also shows similar scarcity in solution efforts for dynamic and stochastic DARP optimization (see Ho et al. (2018)). The only prominent works include Schilde et al. (2011, 2014); Li et al. (2019a); Tafreshian et al. (2021), which considered either stochastic requests or stochastic time-dependent travel times or both, respectively. This is, in fact, due to the optimization complexity and time constraint problem of solving the ridesplitting DARPs, where even the application of available dynamic DARP algorithms is limited to small or medium-size service scenarios.

3.1.2 Research contributions

Given the potential, complexity, and lack of modeling efforts for the AMoD ridesplitting (especially at microscopic resolution), this research contributes to developing AMoD ridesplitting modeling methods in microscopic traffic models integrating dynamic DARP algorithms. We propose a generic methodological framework applicable to most state–of–the–art microscopic traffic simulators. Then, we propose the subsequent service and demand modeling enhancements necessary to model the service behavior and integrate simultaneous dynamic DARP optimization. The framework can model link–level service operations and integrates real-time stochastic information from the simulation in the service optimization. To demonstrate the application, we implement the architecture in the open–source traffic simulator SUMO using python and TraCI simulation interface. The dynamic and stochastic DARP algorithm proposed by Li et al. (2019a) is integrated for the ridesplitting service optimization. The research included in this chapter brings in the following contributions:

1. **AMoD modeling framework:** This research proposes a comprehensive AMoD modeling framework that can be adopted by most state–of–the–art microscopic simulators. The framework contains a bi–level architecture in which the upper level focuses on defining AMoD scenarios (service definition and demand estimation), while the lower level defines the online interaction of the routing algorithm with the simulation modules. The framework adoption is convenient since it uses conventional trip–based demand modeling for AMoD demand estimation (covered in chapter 4) and microscopic driving behavior models for service vehicles. The use of microscopic traffic simulations also allows extensive modeling of many key performance indicators (KPIs) (related to e.g., traffic efficiency and emissions.) useful for modeling service impacts.

2. **Microscopic modeling of AMoD operations:** This research contributes to model AMoD operations in microscopic traffic simulations. The microscopic resolution allows the simulation of discrete vehicles on the network to model detailed traffic dynamics. Thus, AMoD vehicles operate in the dynamic traffic environment, driving with autonomous behavior models and conducting naturalistic stop operations on network links (boarding/alighting operations, interacting with neighboring vehicles, etc.). Moreover, discrete person trips are modeled for service requests (also allowing multimodal travel modeling). Note that, such detailed service modeling integrated with the routing optimization allows realistic replication of the ridesplitting service behavior. Moreover, the simulation–based modeling also enhances the routing optimization efficiency with more realistic information.

3. **Integrated dynamic and stochastic DARP algorithm:** AMoD ridesplitting is a complex DARP because it contains multiple constraints (i.e., time preferences of multiple shared rides, network/route finding), hence requiring efficient optimization to sustain adequate ridesharing. While current modeling methods only use simplistic optimization techniques which can allow large–scale scenario modeling in this research, we integrate a

dynamic and stochastic DARP algorithm developed by Li et al. (2019a). The DARP algorithm optimizes the service simultaneously along the simulation using a scenario–based approach. It also caters to stochastic demand information and time–dependent travel times. Note that, such extensive optimization although restricting the scale of service scenarios, is necessary for modeling and optimizing the stochasticity in ridesplitting systems.

4. **Simulation–based stochastic information modeling for service optimization:** Efficiency of any routing optimization algorithm can be as good as the accuracy of modeling the problem information. Since it is argued that all DARPs are stochastic in nature (Ho et al., 2018), in this research, we model simulation–based stochastic information for service optimization. Microscopic modeling of the network traffic and service operations allows modeling the present stochasticity adequately, while the simultaneous integration of the DARP algorithm allows modeling simulation–based problem information (i.e., online communication of AMoD vehicles status at each event, periodic update of network information). Moreover, the stochasticity present in various aspects of service demand (i.e., arrival times, time preference, and travel locations) is also modeled in the framework (covered in chapter 4).

3.1.3 Outline

The rest of the chapter is structured as follows. Section 3.2 describes a generic modeling architecture proposed for AMoD ridesplitting systems and section 3.3 describes the methodology used in the integrated DARP optimization algorithm alongside the integration method. Then, section 3.4 provides details on the proposed modeling enhancements to model the AMoD ridesplitting service. While section 3.5 describes the platform implementation in SUMO following the proposed modeling architecture. It describes the operational details for developing the architecture components, methods to model AMoD service microscopically, and the code implementation structure. Finally, section 3.6 concludes by describing the overall contributions of the research chapter alongside its future implications and possible research directions.

3.2 Generic modeling architecture for AMoD

In this section, a generic modeling architecture is proposed to model AMoD ridesplitting system. The architecture diagram is shown in figure 3.1 and it outlines the overall modeling method to model on–demand ridesplitting services with integration of dynamic DARP algorithm. The architecture can be divided into two levels, where the upper level gives the flowchart for service scenario definitions and management. It aims to set up AMoD simulation scenarios by contributing the scenario variables (i.e., the simulation network, time–dependent OD demand, AMoD fleet attributes, and AMoD attracted demand) into the lower level architecture named as *'Simulation executor'*. The simulation executor contains the simulation workflow of executing an AMoD scenario i.e., defining the online interactions between the DARP algorithm (named *'Scheduler'*), supply/simulation module, and service demand module. Note that, a scenario is referred to be a single simulation run given the AMoD demand and service attributes under certain network conditions.

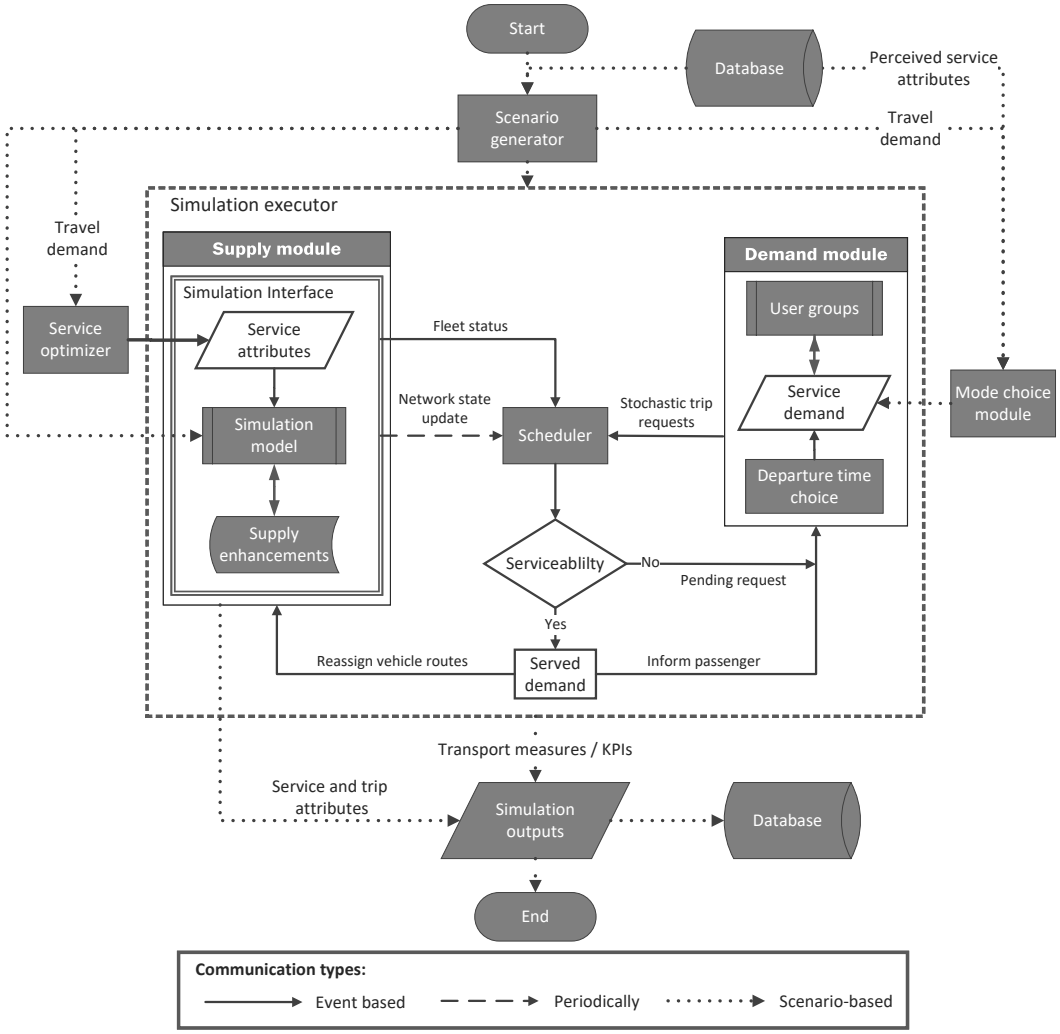


Figure 3.1: Generic architecture to simulate ridesplitting services

Scenario manager

The scenario management architecture consists of five major components i.e. scenario generator, simulation executor, service optimizer, mode choice module, and database. The scenario generator module generates a simulation scenario (using the database) to represent a certain time frame of the real traffic network. It sets up all different traffic simulation settings e.g., the model network, time–dependent travel demand, simulation times, etc. Then, the service optimizer component defines the AMoD fleet attributes (i.e., fleet size, vehicle capacities, and deployment) given the scenario travel demand or time–of–day settings. Similarly, the mode choice module generates the overall AMoD demand attracted based on the available network travel demand and perceived service attributes (discussed in chapter 4). Finally, the simulation executor is responsible to simulate the generated scenario in the microscopic traffic environment and provide the resulting simulation outputs (i.e., service and trip attributes, different transport measures/KPIs) that are saved within the database.

Simulation executor

The architecture under the simulation executor contains three major components i.e. the supply module, the demand module, and the scheduler. All three components interact online - simultaneously during the simulation to serve stochastic request arrivals by dynamically routing service vehicles given the request preference constraints and current network conditions. The supply module consists of the simulation model (i.e., the network model and travel demand interacting to replicate the dynamic traffic conditions) along with the supply enhancements for modeling service behavior and the service attributes (discussed in section 3.4.1). The demand module consists of the AMoD demand (resulting from the mode choice module) and its interaction with pre–defined user groups (which generate varying passenger trip preferences) and the departure choice model to generate stochastic trip requests. The demand module is discussed in more detail under chapter 4. The third and core component of the architecture is the DARP scheduler responsible to optimize the incoming dynamic routing problems formulated based on the detected status of fleet vehicles, time constraints of in–service passengers, future stochastic information, and the information related to the available serviceable requests. The AMoD fleet routes and request assignments are updated dynamically upon solving the said DARP problems. Description related to DARP algorithm and its integration is discussed in section 3.3.

Interaction levels

In the proposed AMoD modeling framework, all different modules interact at three different levels. These levels are defined based on the communication purpose and frequency between all different components.

- **Scenario–based interaction:** This interaction mainly occurs among the components of the scenario management level. The communication is used to define the scenario variables for simulation, feeding them in the simulation executor and then communicating the generated simulation outputs and updating the database. As said by the name, this interaction occurs only once per scenario evaluation.
- **Event–based interaction:** It is the dynamic interaction that occurs within the simulation executor module. It contains the communication for AMoD fleet status, trip request preferences, in–service trips' attributes, and optimized fleet routes. The interaction is triggered by an event (i.e., request arrival or vehicle stop) but instead can also be periodic for other AMoD services, depending on their implementation settings, modeling scale, and scheduler optimization capacity.
- **Periodical network updates:** This interaction exists depending only depending upon the type of platform implementation (i.e., route choice is part of the simulation or scheduler). It communicates the updates on the network traffic state from the microscopic traffic simulator to the scheduler.

3.3 DARP based optimization

Service optimization of the AMoD ridesplitting corresponds to dynamic DARP and acts as the core part of the service, controlling vehicle routing and request assignment. Among the several solution approaches for dynamic DARP, only a few consider optimization based on stochastic information (Ho et al., 2018). In this research, we use the solution approach developed by (Li et al., 2019a). It considers both the stochasticity of future requests and route–finding based on time–dependent travel times. The integration of the DARP algorithm in the microscopic traffic

environment also provides the opportunity to model simulation–based stochasticity in network, demand, and service operations. Below, we provide a brief description of the scheduling algorithm followed by its integration method in a microscopic traffic simulation.

3.3.1 Scheduling algorithm

As per Li et al. (2019a), the solution algorithm models the scheduling problem as a DARP. $R = \{r_1, r_2, \cdots\}$ is the set of requests, which is being updated during the operation to include newly received requests. Each request r_i has its pickup node r_i^p, delivery node r_i^d, status r_i^s which can be new, rejected, waiting for pickup, picked, or delivered, route r_i^v which is the index of the van picking this request. Each request r_i has a pickup time window $[e_{r_i^p}, l_{r_i^p}]$ and a delivery time window $[e_{r_i^d}, l_{r_i^d}]$. The pickup time window is used to ensure passengers' expected pickup time and the delivery time window is used to limit maximum detour. For the pickup time window, $e_{r_i^p}$ is set as passenger's expected pickup time, and $l_{r_i^p} = e_{r_i^p} + u_w$, where u_w represents the maximum allowed waiting time. For the delivery time window, $e_{r_i^d} = e_{r_i^p}$, and $l_{r_i^d} = l_{r_i^p} + u_d DTT(r_i^p, r_i^d)$, where u_d represents the maximum allowed detour ratio and $DTT(r_i^p, r_i^d)$ is the direct travel time between the pickup and the delivery node under average travel speed.

Scenarios are used to represent the stochastic information about future requests and traffic conditions. $S^r(t) = \{s_1^r(t), s_2^r(t), \cdots\}$ is the set of request scenarios we use at time t. $S^s(t) = \{s_1^s(t), s_2^s(t), \cdots\}$ is the set of travel speed scenarios we use at time t. $S(t) = \{s_1(t), s_2(t), \cdots\}$, where $s_k(t) = \langle s_k^r(t), s_k^s(t)\rangle$, is the set of scenarios we use at time t, which combines request scenarios and travel speed scenarios. The goal is to optimize the operating profit and the user experience. The operating profit is calculated as the operating cost minus the service revenue. The user experience includes the waiting time and the detour. In the implementation, the objective function is a linear combination of the cost, the revenue, the waiting, and the detour.

The scheduling procedure uses scenario–based approaches to decide whether to accept each new request and design schedules for each van. For each given scenario, it needs to solve a deterministic problem and this is done by a tabu search algorithm. The scheduling procedure gets the following inputs: (1) the set of vans V which includes the position of each van; (2) the set of requests R which includes newly received requests and accepted requests with their status, such as, whether the request is picked or not, which van the request in on. The scheduling procedure first decides whether to accept each new request, then decides the routes of each van.

In the first step, to decide whether to accept each request, the brief idea is to compare the expected objective function value when accepting the request with the one when rejecting the request. To achieve this goal, an evaluation procedure is developed as shown in Figure 3.2. In this evaluation procedure, it needs to input the current state of vans and requests, $state(t)$. The evaluation procedure estimates the average objective function value of the current state. It loops through each scenario $s_k(t) \in S(t)$. With a given scenario $s_k(t)$, the stochastic problem becomes a deterministic problem. Tabu search is used to solve the deterministic DARP under the given state and scenario, which gives an optimal objective function value $obj_k(t)$ under each scenario. The average value of these objective function values under different scenarios is used to represent the expected objective function value of the current state. With this evaluation procedure, it first marks the request as rejected and uses the evaluation procedure to evaluate the average objective function value under a given state, denoted as $obj_{rejected}$. Then it marks the request as accepted and inserts it into a random route, runs the evaluation procedure again, and gets $obj_{accepted}$. If $obj_{rejected} > obj_{accepted}$ we reject the request, otherwise we accept it.

In the second step, to decide the route of each van, a scenario–based search is developed to generate and evaluate potential decisions. The main idea is demonstrated in Figure 3.3. Similarly, the $state(t)$ represents the current state of the system. In each iteration of the loop, a scenario

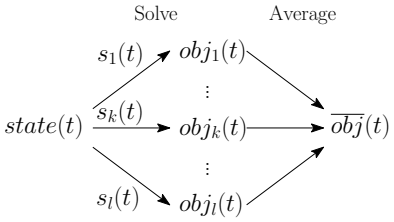


Figure 3.2: Illustration of the evaluation procedure.

$s_k(t) \in S(t)$ is selected. Tabu search is used to find the optimal decision for the given state and scenario. For each scenario $s_k(t)$ it can get an optimal decision, denoted as $decision_k(t)$, which is called a candidate decision. Because $s_k(t)$ includes potential future requests, in some candidate decisions, the vans may be dispatched to future requests if this can lead to a better solution. After generating candidate decisions, it needs to evaluate these decisions and choose a final decision. For each $decision_k(t)$, it first updates the state according to the decision. By doing this, it gets a new state $state_k(t)$ which represents the consequence of executing the decision. Then the evaluation procedure is used to loop through scenarios again to get an expected objective function value $\overline{obj}_k(t)$ of the candidate decision. Finally, the candidate decision with the best–expected objective function value is chosen as the final decision.

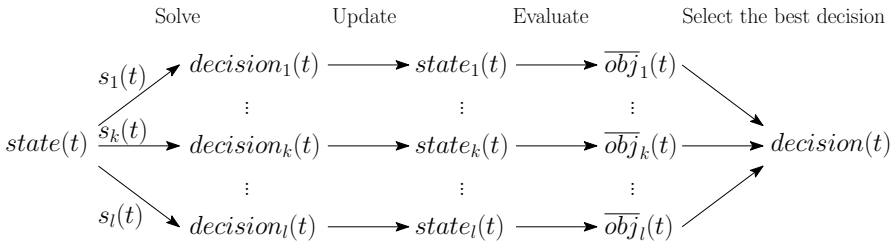


Figure 3.3: Illustration of the scenario–based search.

3.3.2 Scheduler integration

Modeling AMoD ridesplitting in a traffic simulation environment requires integration of the DARP algorithm to simultaneously optimize the vehicle routing. Each event (triggered either by request) is solved as a separate optimization problem in real–time. The problem information is fed from the simulation to the scheduler to optimize and generate new vehicle routes. The integration allows the opportunity to model the DARP information aspect using microscopic simulation catering for the stochasticity in traffic congestion and service operations, while the algorithm is only responsible to optimize the service. Hence, the combination allows doing efficient optimization along with realistic AMoD simulations.

To integrate the above–mentioned scheduler inside the microscopic traffic simulation, we define an interfacing module (figure 3.4) with different interaction protocols. The simulation configuration is loaded as a server to use commands from the simulator's API for different operations e.g., to read/write simulation attributes, calculate simulation steps, call scheduler optimization, etc. The interfacing module takes in the scenario inputs (as in figure 3.1), runs the simulation, and outputs person trips and service vehicle attributes.

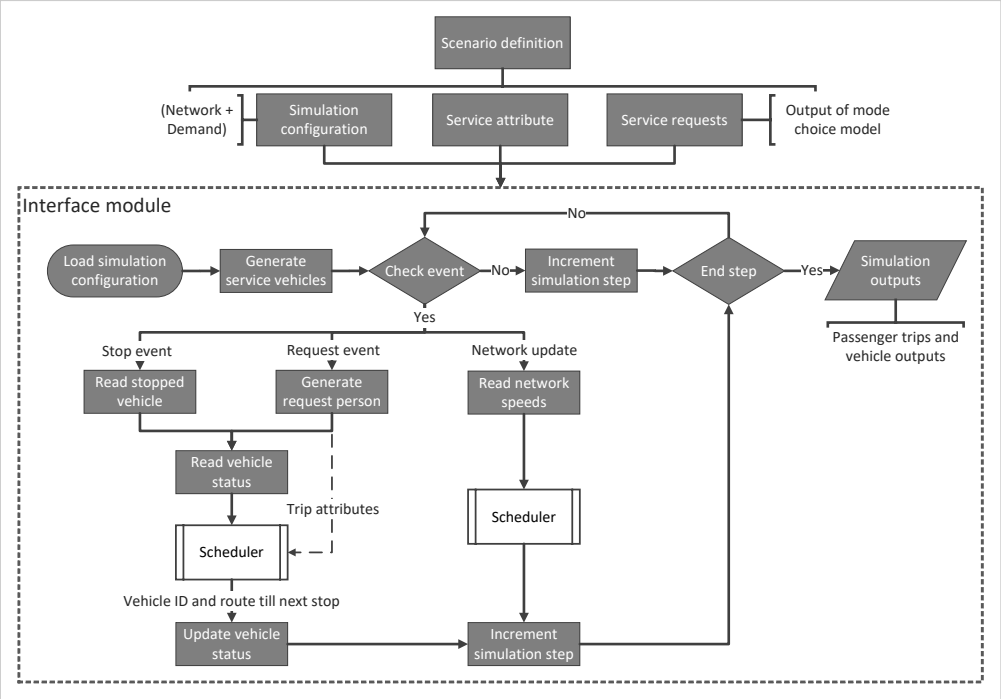


Figure 3.4: Simulation interfacing module

The simulation runs incrementally upon each simulation step to detect the occurrence of an event. There are three main types of events defined to trigger the interaction (figure 3.4), i.e., request event, stop the event, and network update. Network update is triggered periodically calling the simulation to read the network state attributes (e.g., link speeds) and update the routing scheduler (assuming route finding is part of the scheduler). The other two event protocols are triggered by the simulation upon either a request arrival or a vehicle stop at a service location. The scheduler is called for both events with all service vehicle status (i.e. network positions, routes, and occupancy) as input. For request events, the subsequent request information (i.e. pick up and drop off locations and time window) is also communicated. Scheduler, then, solves the DARP optimization problem and returns the vehicle ID along its new service route which is appended accordingly into the simulation environment.

3.4 Modeling AMoD ridesplitting service

AMoD ridesplitting serves on–demand passenger requests with different origin–destination (OD) locations through shared rides and flexible operations. Hence to model the service in microscopic simulation, additional facilities are required to replicate the service vehicle characteristics and the operational behavior. Similarly, the attracted demand (mode choice) of the service is also dynamic due to uncertainty/stochasticity in service availability and experienced trip utilities (solved by recursive supply–demand interaction in literature). Demand attraction is also influenced by trip pricing, which although is based on direct trip distance, could also depend on individual passenger trip preference (detour and waiting flexibility) through dynamic pricing

or utility–based compensation. Below, we discuss the proposed enhancements for modeling the service in the microscopic simulation environment.

3.4.1 Service behavior

Microscopic traffic simulators simulate discrete vehicles on network links to model detail traffic dynamics (e.g., SUMO (Lopez et al., 2018)). It allows modeling of the AMoD service operations in a realistic traffic environment. The AMoD vehicles operate in a dynamic traffic environment, driving with autonomous behavior models and conducting service stop operations with the interaction of neighboring vehicles, infrastructure, and vehicle driving capabilities. Meanwhile, individual passenger trips can also be modeled, where individual persons wait, board, and alight at flexible service locations. The different service behavioral components are modeled as per the following:

- **Autonomous service vehicle:** An AMoD service vehicle drives autonomously with a certain vehicle type and capacity to allow ridesplitting. For modeling the service vehicle, new vehicle class is defined with these attributes:
  - ***Driving behavior:*** In microscopic models, autonomous driving behavior is modeled using state–of–the–art car following models (it varies upon the availability among different traffic simulators). For example, ACC car following model (Milanés and Shladover, 2014) can be used which requires the definition of four different control modes for speed, gap, gap–closing and collision avoidance, or similarly Wiedermann99 car following model (Sukennik et al., 2018) can be used requiring the definition of a larger set of 10 different control parameters. Note that, the estimation or calibration of autonomous driving model parameters is still an active research area due to limited data or field experiments for AVs driving. Still, the literature contains many AVs modeling studies and the driving model parameters can be taken for them (e.g., Sukennik et al. (2018) provides guidelines for using the Wiedermann99 model).
  - ***Vehicle type:*** Defining the vehicle type for AMoD vehicle includes its classification as 'passenger' vehicle (vehicle movement rights as a regular vehicle), capacity varying between 6 to 12 passengers, suitable vehicle dimensions, operational speeds, less agile acceleration/deceleration patterns, and adequate dwell time at service stops for passenger boarding/alighting operations.
- **Flexible service operations:** To model on–demand operations with flexible service locations, AMoD should board/alight the passengers flexibly at almost all locations on the network. Microscopic models allow modeling these operations upon network links. The stop operations can be carried either as lane stops (on–road stops hindering traffic flow) or parking stops depending on road type and available infrastructure. The stoppage purpose is only for passenger boarding/alighting and the dwell time is defined in vehicle class attributes. Otherwise, idle vehicles which don't have any in–service passengers wait as stopped at parking or off-road facilities. Note that, all previous modeling efforts in literature, model the service operations on network nodes (i.e., unrealistic, since regardless of the link lengths, passengers board and alight on network junction locations only), even all DARP solutions including the integrated DARP algorithm by Li et al. (2019a) model and optimize the service based on node–level operations (requiring transformations during platform implementation as discussed in section 3.5). Hence, this research, by modeling the AMoD service in a microscopic model represents the most detailed and realistic link–based service operation.

- **Service locations:** AMoD can have a wide operational spectrum, especially with dynamic matching, the ridesplitting service concept can be optimized to improve its serviceability e.g., reduce detour distances by meeting–point based service (Stiglic et al., 2015) and increase serviceability for having minimum (critical) demand served for the ridesharing. Similarly, modeling link–based operations in microscopic simulators could have certain requirements due to modeling of individual vehicle driving behaviors e.g., braking distance for a service stop. For these reasons, a set of service locations are also defined using a suitable method i.e., either by optimization for the meeting point–based service, filtering non–serviceable links, etc. Note that, ideally AMoD ridesplitting service should be modeled with the ability to board/alight the passengers at almost all locations on the network.
- **Dynamic vehicle's control:** AMoD ridesplitting is modeled with the integration of a dynamic DARP scheduler that solves vehicle routing and passenger assignment problems online, optimizing the vehicle routes dynamically. This optimization procedure or simulator–scheduler interaction is triggered by events (i.e., passenger request arrival or vehicle stop, see section 3.2), and to model this AMoD dynamic behavior, the simulation is run as a controlled server continuously detecting the occurrence of an event at each time step. Upon detection, the simulation stops for dynamic optimization, and required communication is conducted first for optimization problem formulation and later for updated vehicle routes as the solution. These routes are updated for corresponding service vehicles in the simulation to replicate the said dynamic vehicle control. Meanwhile, to cater for new information (i.e., traffic congestion or passenger request) existing waiting passengers are also reassigned to other service vehicles, while for the vehicles not serving any passengers, the DARP scheduler also calls for relocation routes or to retain idle behavior.
- **Passenger trip variables:** Since in ridesplitting, passengers with different origin–destination and depart times are served by shared rides, it results in varying trip attributes for all individual passengers. Hence, to model AMoD ridesplitting service performance, discrete persons are generated for each request at their corresponding depart time having the ability to do multi–modal travel (mainly walking and riding, but can include other modes like public transport). The passenger trip variables include waiting times, walking time, travel time, and trip cost are estimated for each person during the simulation while being served by a subsequent service vehicle. Furthermore, passenger demographical variables can also be attached for modeling impacts with market segmentation.

3.4.2 AMoD demand modeling

The mode choice modeling for on–demand mobility is dynamic due to uncertainty/stochasticity in service availability and experienced trip utilities. Literature efforts for modeling AMoD (and DRT) estimate the service demand by recursive supply–demand interaction which aims to attain equilibrium between the estimated demand and its experienced service attributes. The equilibrium is required to be re–evaluated with any change in supply or demand attributes. Note that, the recursive simulations method is feasible for microscopic demand modeling or agent–based modeling frameworks because they anyway require them to do agent scoring based simulation (Horni et al., 2016; Basu et al., 2018). But, most state–of–the–art traffic simulators model the traffic demand with the aggregated trip–based demand modeling method and don't require recursive simulation for traffic modeling, hence for them the on–demand mode choice modeling by recursive supply–demand interaction is computationally expensive and infeasible.

To overcome this limitation, different possible demand estimation methods can be explored. For example, using the conventional mode choice method using the Multinomial Logit (MNL) model (since integrated dynamic DARP scheduler takes into account the passenger trip time

preferences as its optimization constraints), or market equilibrium approach which has been adopted in literature for modeling demand estimation for on–demand services. Moreover, trip pricing that is modeled as a function of distance for AMoD ridesplitting can be proposed to be dynamic based on flexibility in passengers' preference or their served trip utility (which variate among all passengers). All these demand modeling aspects are covered in detail under chapter 4. Note that, this means that the description related to the components of mode choice and demand module (from figure 3.1) are covered in chapter 4.

3.5 Platform development and implementation

3.5.1 Platform implementation in SUMO

To implement the proposed modeling architecture and AMoD modeling enhancements, an open–source traffic simulator SUMO (Lopez et al., 2018) is used with python. SUMO is a microscopic traffic simulation suite capable of modeling individual vehicle driving and detailed traffic assignment, hence allowing realistic modeling of AMoD service and its operations. The platform implementation follows the structure of the proposed generic architecture (figure 3.1) and consists of two levels i.e., scenario management and simulation executor. The scenario management modules define the simulation setup (simulation network, time–of–day, OD demand), the service setup, and attracted service demand. While the simulation executor module simulates the scenario, first, by loading the scheduler and scenario variables, then generating stochastic service requests, and finally executing the simulation (where the service vehicles are simultaneously optimized by the integrated DARP scheduler). Outputs including passenger trip attributes, vehicle attributes, and network–related KPIs are eventually generated upon completion of the scenario simulation.

SUMO

SUMO is a non–commercial open–source simulation tool widely adopted for research. It can do microscopic traffic assignment by modeling the driving behavior of discrete vehicles along with their interaction with each other and the infrastructure. The driving behavior is modeled based on the state–of–the–art car following and lane changing models, including the capabilities of modeling autonomous driving (Shladover et al., 2012). Microscopic traffic assignment in SUMO allows modeling detailed driving of AMoD vehicles and link–level realistic operations under dynamic traffic environment. Moreover, SUMO also allows modeling intermodal transport systems including private vehicles, public transport, and pedestrians. Individual pedestrian modeling allows modeling discrete passenger trips for whom the level of service (LOS) can variate significantly. It also allows modeling multimodal use cases of AMoD e.g., last–mile service, etc. Furthermore, SUMO also provides many different supporting tools, e.g., route finding, emission modeling, visualization, etc to assist comprehensive modeling and simulation–based impact assessment for AMoD services.

The most important support tool for modeling AMoD that is available in SUMO is named 'Traffic Control Interface' TraCI (Wegener et al., 2008). It provides online access to a running traffic simulation, allowing value retrieval of simulated objects and change of their behavior. It uses a TCP–based server to give access inside the simulation and multiple object groups e.g., vehicles, routes, traffic lights, links, etc can be accessed online using TraCI. The object group list is increasing with continuous development and also includes taxi operations (although it is not used in the current implementation). The current platform implementation uses TraCI API for four main object groups, i.e., route, vehicle, lane, person objects.

Scenario management

Scenario management architecture contains four main components i.e., traffic simulation, DARP optimizer, mode choice, and simulation executor. The traffic simulation component is represented by the conventional modeling of the traffic network in SUMO which requires the network (net.xml) and the time–dependent traffic demand in form of trips (trip.xml). The service optimizer component defines the fleet attributes (i.e., fleet size, vehicle type, network positioning, driving model, and vehicle capacity) and service flexibility attributes (e.g., service locations either door–to–door or meeting point–based). All vehicle attributes are defined in an additional file (add.xml) which is loaded along with the simulation in SUMO, while fleet size, positioning, and service flexibility attributes are defined in the python coded framework before the start of simulation execution. The mode choice module follows the methodology discussed in chapter 4 to generate a (trip–based) AMoD OD demand and is also set in python coded framework (which also converts the AMoD OD matrix into stochastic passenger requests). The implementation of the mode choice module is also discussed all its methodology. Finally, the simulation executor module (describe later in detail) executes the microscopic simulation having an online AMoD ridesplitting service running with DARP scheduler optimization.

Simulation executor in SUMO

The simulation executor module is the core component of the modeling platform and acts as a simulation controller written in python. Figure 3.5 represent the module workflow before and during the SUMO traffic simulation. The module first loads its two main components i.e., the simulation model and the DARP scheduler. The simulation model is loaded using SUMO's TraCI server, halting at its first simulation step, where then it is further run and controlled by TraCI API. The DARP scheduler proposed by Li et al. (2019a) is written in C++ and should be integrated online with the python framework for better computational efficiency by retaining its information memory i.e., network state, vehicle information, etc. The is done by converting the DARP scheduler as a dynamic link library (DLL). DLL is then directly loaded in the python framework and the scheduler functions are directly from python using `ctypes` library. Next, to initialize the traffic simulation, four tasks are conducted:

- AMoD vehicle objects are generated with certain (optimized) initial positioning and setup of service flexibility (stop locations).
- Setting up of Traffic simulation, i.e., running simulation warm–up and generating AMoD vehicles.
- Setting up the DARP scheduler, i.e., loading the network state and AMoD vehicle characteristics (fleet size, capacity, and positioning) in the scheduler memory
- Stochastic requests are generated using AMoD OD demand (details in chapter 4)

After finishing all the initialization steps, the simulation starts to run iteratively over the simulation steps during which events are detected and the simulation is paused for each event. Events include departing of new request(s) and vehicle stop at the service location. Upon event detection first, the TraCI server is called to get service vehicle status, and then, the scheduler optimization function is called with the problem information. The optimization function returns vehicle routes of all service vehicles (including new and current) and passenger assignment information. Note that, the status of vehicle route positioning is also indexed based on the latest vehicle routes communicated by the scheduler. Finally, the route stops, and passenger assignment information is updated in the simulation using TraCI API. For having dynamic rerouting operations, the scheduler only sends information till the next vehicle service stop and this is created as an event for later to again trigger the interaction. Also, the passenger list served by

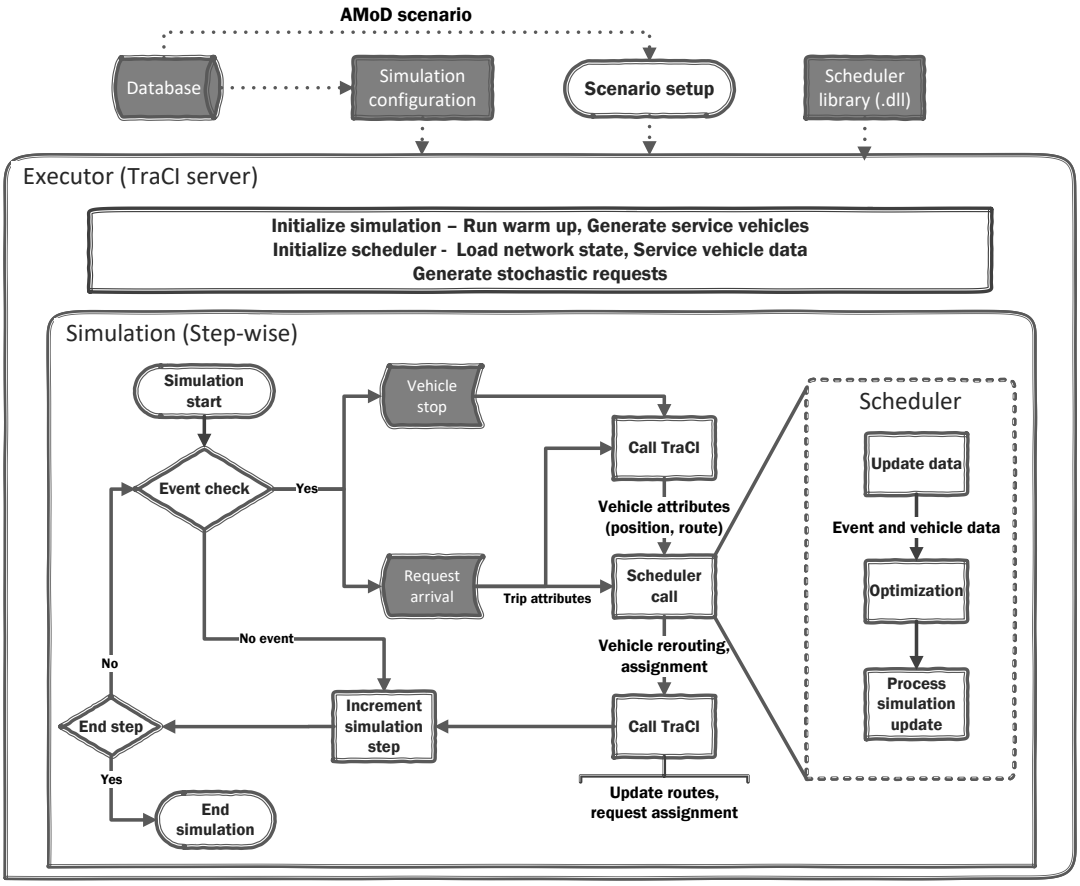


Figure 3.5: SUMO implementation for DARP integration

each vehicle and their corresponding pick–up and drop–off stops are also logged in python code to update and validate stopping and passenger service information after each interaction.

3.5.2 AMoD modeling in SUMO

AMoD vehicles

To model AMoD vehicles in SUMO, a vehicle type is defined in an additional file (add.xml) which is loaded along with the SUMO simulation configuration (*.sumocfg) in the TraCI server. The first parameter defined is the vehicle class as the default 'passenger' vehicle (denoting normal passenger cars and corresponding rights of lanes usage). Then, to model the AMoD autonomous driving behavior, *'Wiedermann99'* is set as the car–following model. The model also requires defining 10 control parameters, which are also set as parameters based on the guidelines given in Sukennik et al. (2018)). Further, the other vehicle parameters included in the setup are vehicle length, person capacity, and maximum speed. Apart from these parameters, a minimum boarding and alighting duration is set as a parameter for vehicle python objects and is used during the event procedure.

```
1    <vType id="AV" vClass="passenger" carFollowModel="W99"
2    CC1="0.9" CC2="0" CC3="-8" CC4="-0.10" CC5="0.10"
3    CC6="0" CC7="0.10" CC8="3.5" CC9="1.5"
4    length="7" personCapacity="8" maxSpeed="22.22"
5    type="ElectricVehicle" emissionClass="Energy/unknown"/>
```

Listing 3.1: AMoD vehicle definition in XML format

Link–based modeling through node–based optimization

The state–of–the–art in graph or network theory considers nodes as entities and edges as information links between them (West et al., 2001). Therefore, the standard approach is to do node–level modeling/optimization of networks. Following the standard, all literature efforts for modeling and optimizing the AMoD services, model node–based service operations due to similar practical reasons (i.e., standard and convenient network modeling, vehicle routing, etc.). The dynamic DARP algorithm (Li et al., 2019a) integrated with the platform also does node–level network modeling to do service optimization. On the other hand, microscopic traffic models differentiate from this standard. It focuses on representing detailed traffic network characteristics where nodes are traffic junctions and links represent actual road segments. It models link–based traffic and vehicle routes which also helps to model realistic service operations (e.g., roadside boarding/alighting), and vehicle driving behaviors and interactions. Hence, in order to use the DARP schedulers with microscopic models, appropriate node–to–link modeling transformation methods are required. In microscopic models, link–based driving behavior of individual vehicles also struggles with operational issues for modeling service stops (e.g., enough link lengths for breaking distance) and therefore require filtering un–serviceable links or changes in service vehicle driving.

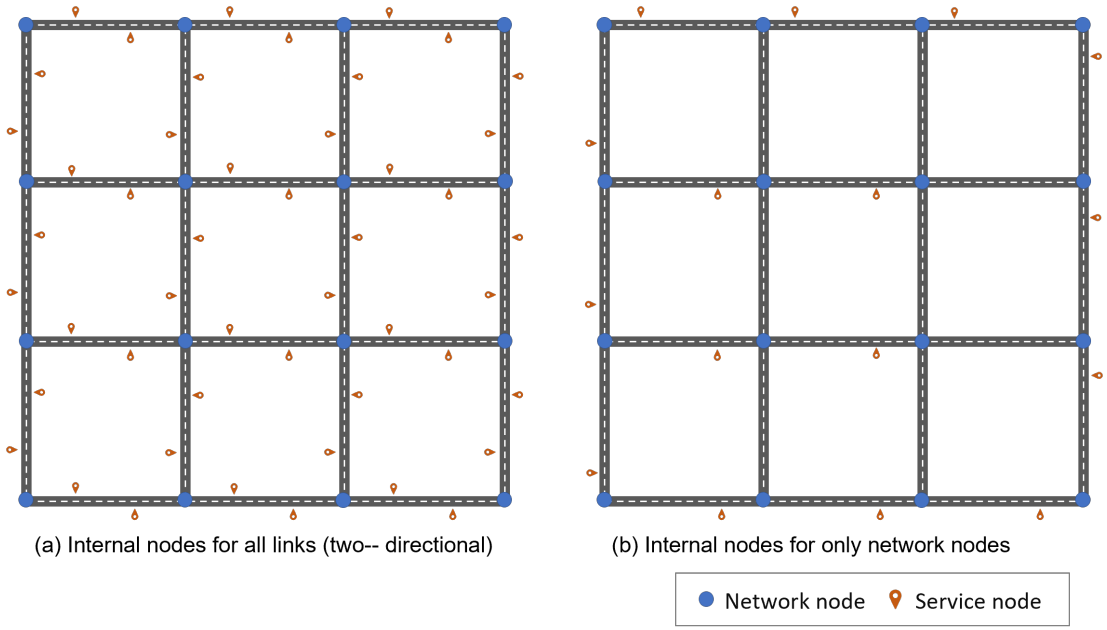


Figure 3.6: Internal nodes for DARP scheduler

To integrate the dynamic DARP scheduler, a node–to–link modeling transformation method is used in the platform implementation. The idea is to add virtual internal nodes on all service links of the network. These internal nodes are only present in the network used by the DARP scheduler and stop service is only allowed on them (where actual network nodes only act as routing nodes). Figure 3.6 visualizes the idea for using internal service nodes. Note that, the addition of internal nodes does add routing and memory burden in the scheduler, and hence for operational reasons lesser amount of virtual nodes can be added to still cover all areas, e.g., figure 3.6 (a) shows the virtual service nodes to represent maximum service flexibility at all links, while figure 3.6 (b) shows an optimized allocation of service nodes to cover each network node. The method also requires to use of node to link translation dictionaries during communications between the scheduler and SUMO because the scheduler takes in node–based information (vehicle positions, etc) and also provide routing information with node ids including internal nodes, while SUMO takes in route information with link ids.

Service location optimization and use cases

Figure 3.6 shows the use of virtual service nodes for DARP optimization. On one hand, the idea of using lesser service nodes (figure 3.6) can be used for operational reasons i.e., to represent the average demand locations of each traffic zone (similar idea to that of TAZ connectors used for generating demand for the trip–based demand modeling (de Dios Ortúzar and Willumsen, 2011)). On the other hand, the idea can be further adapted to do network–based service locations optimization or represent different service use cases. For example, Stiglic et al. (2015) shows the benefits of using meeting points in ridesharing services. The use case of meeting point–based AMoD ridesplitting can be represented using the virtual nodes' settings. Moreover, the allocation of service locations can be taken as an optimization problem targeting for the better ridesharing, reducing detours (passenger LOS), etc. Note that, in the current platform implementation, we implement the first idea of reducing the service nodes (equally spaced service locations) for representing average demand locations (TAZ connectors) but the implementation is easily extendable to do have an optimized meeting point–based service. Details on service stop allocation and demand generation are discussed in chapter 4.

Service operations and multi–modal person trips

Microscopic models allow link–level AMoD service operations. The passenger boarding and alighting operations are conducted on–road where functions like `Passenger_onboard` (section 3.5.3) insure the passenger pickups. For having dynamic rerouting of vehicles and reassignment of passengers, the scheduler only sends information till the next vehicle service stop. Hence, the information related to passenger assignment, their pick–up/drop–off stops are logged in python. Later, functions like `Person_generate` and `Person_reassign` is used for assigning and reassigning the passengers among vehicles. While, functions like `set_stops` are used after each interaction to add and validate the list of required stops. Similarly, the vehicle idle waiting behavior is modeled by moving it outside the network (by `Movein`/`Moveout` functions) for waiting to be called up (SUMO facilities are now available to do this automatically using taxi vehicle type).

To model discrete passenger trips, persons are generated in SUMO at the pick–up location (`Person␣generate` function) upon its request acceptance. These persons can be allocated to different transport modes using their `drivingstage` parameter, e.g., walking, waiting, driving, riding, and public transport usage. Hence, the current platform can be further used for multimodal use cases e.g., last–mile services. Note that, person reassignments (due to traffic conditions, new requests, etc.) and sudden stops for emptied vehicles also require appropriate facilities to model realistic behavior.

3.5.3 Code implementation

The AMoD modeling framework is implemented in python and consists of many different modules. In the section, we enlist all python modules alongside their code structures. Note that, many of these module functions are also referred during the chapter and the pseudo code also helps in understanding the interaction among different modules.

Scenario manager

`Scenario_manager` is the main interfacing module of the modeling framework and calls are other modules to conduct different tasks for overall scenario management. It doesn't include any of its own functions but only the calls for other modules.

Module 1 Scenario manager

```
Define AMoD scenario (fleet size, capacity, simulation time, preferences, ... )
if (Virtual nodes redefine) then
    Call schedulerNetwork from network module
end if
Call modeChoice from Demand module
Call networkStateUpdate from network module
Call Simulation_executor module for simulation
Process simulation outputs
```

Network module

`Network` module carries the code to add virtual nodes and write network states (link speeds) in the network loading file of the `Scheduler`. The pseudo-code is shown in module 2.

Module 2 Network

```
function SchedulerNetwork(Number of nodes, TAZ structure,...)
    Process TAZ to allocate service nodes spatially
    Simulate Sumocfg simulation to read network link speeds
    Write network file for Scheduler module
end function

function NetworkStateUpdate
    Read network state using TraCI
    Update network file for Scheduler module
    Call roadLoad from Scheduler module
end function
```

Demand module

`Demand` module carries the code to conduct mode choice and generate stochastic requests for AMoD ridesplitting service. Consisting of two functions, the `modeChoice` function executes

the logit model to generate OD demand matrox for the service, while the `requestGeneration` function generates discrete stochastic requests with varying origin-destination, depart times, and trip preferences. The pseudo-code is shown below in module 3.

Module 3 Demand

```
function modeChoice(OD demand, passenger preferences coefficients, )
    Evaluate service demand matrix using logit model
end function

function requestGeneration(Service demand)
    Call OD2Trips SUMO tool to generate stochastic requests
    Generate random trip time preferences for all passengers
end function
```

Scheduler module

`Scheduler` module contains the code for dynamic DARP scheduler written in C++ and wrapped as a dll for loading in python. The first `initializeScheduler` function is to initialize the scheduler memory (i.e., load service vehicles and network information), while the next procedure function `schedule` is to run optimization for an event and provide updated route and passenger assignment information. Another minor function `eventExists` is used to check for call ups of idle vehicles. The pseudo-code is shown below in module 4.

Module 4 Scheduler

```
procedure initializeScheduler                    ▷ Setup scheduler memory in DLL
    Load network routes in memory (a roadLoad function)
    Load passenger request history (a historyLoad function)
    Create vehicle objects in memory (a createVehicleObjects function)
end procedure

procedure Schedule                               ▷ Solve an event optimization problem
    function valueAssign(vehicle state, event type, request data)
        Update vehicle data, create request data, event information ....
    end function

    function handleEvent(optimization information)
        Solve optimization problem and generate vehicle routes
    end function

    function requestServed                       ▷ To get request ID served by each event
        Return the serving request ID for the call event
    end function
```

```
    function stopType                          ▷ Get the next stop purpose for all vehicles
        Return the stop purpose for all vehicles
    end function
end procedure

function eventExists(Stopped vehicles)         ▷ Call up check for idle vehicles
    Return events information for stopped vehicles (relocation, etc.)
end function
```

Simulation executor

`Simulation_executor` module acts as the online simulation controller module running the simulation on a while loop and serving ongoing service events. It follows the SUMO implementation architecture shown in figure 3.5. It contains only one function to initialize the AMoD vehicles, while the rest of the module runs the simulation under a `while` loop calling two event protocols to model the AMoD service. The pseudo-code of `Simulation_executor` module is shown below in module 5. The two event protocols (shown in module 6 and 7) either call the `scheduler` module for optimization or class methods from `Operations` module i.e., for `Passenger` and `Vehicle` classes to perform corresponding object's tasks during the AMoD simulation (e.g., `personGenerate`, `vehicleUpdate`, etc.).

Module 5 Simulation executor

```
Load Sumocfg via TraCI and the scheduler module

function InitializeAMoDvehicles
    Call createObjects from Operations module
    Call initVehiclePositions from Operations module
    Park all vehicles as idle (Method Vehicle.Moveout from Operations module)
end function
while Current step ≤ End simulation step do

    if Request event then
        Request event protocol
    end if

    if Vehicle stop then
        Stop event protocol
    end if

    if current step % x = 0  then                  ▷ Periodic checks
        Check event for idle vehicle
    end if
end while
Write simulation outputs
```

Module 6 Request event protocol

```
1: for n (vehicles) do
2:     Method Vehicle.vehiclePosition
3: end for
4:
5: Call Schedule from Scheduler module
6:
7: for n (vehicles rerouted) do                          ▷ Update simulation
8:     Method Vehicle.vehicleRoute
9:     Method Passenger.personGenerate
10:    if (passenger reassigned) then
11:        Method Passenger.personReassign
12:    end if
13:    Method Vehicle.vehicleUpdate
14: end for
```

Module 7 Stop event protocol

```
1: for n (vehicles) do
2:     Method Vehicle.vehiclePosition
3: end for
4:
5: for n (vehicle stopped) do
6:     Call Schedule from Scheduler module
7: end for
8: for n (vehicles rerouted) do
9:     Method Vehicle.vehicleRoute
10:    if Reassign then Method Passenger.personReassign
11:    end if
12:    Method Vehicle.vehicleUpdate
13:    for n (vehicle idle) do
14:        Method Vehicle.moveout
15:    end for
16: end for
```

Vehicle operations

`Vehicle_operations` module carries class definitions for two AMoD service objects i.e., passengers and vehicles. Both classes contain respective object parameters and methods to conduct service related operations in the simulation using TraCI API. The pseudo-code is shown below in module 8. The module also contains two functions i.e., `createObjects` to create objects instances from both classes and `initVehiclePositions` to optimize initial vehicle positions.

Module 8 Vehicle operations

```
1: function createObjects
2:     for n (vehicles) do create object from Class Vehicle
3:     end for
4: end function
5:
6: function initVehiclePositions                    ▷ Vehicle positioning methods
7:     for n (vehicles) do setup vehicle initial positions
8:     end for
9: end function
10:
11: Class Passenger
12:     Method PersonGenerate
13:         traci.person.(add, appendDrivingStage)
14:
15:     Method PersonReassign
16:         traci.person.(appendDrivingStage, removeStage, getStage)
17:
18: Class Vehicle
19:     Method VehicleRoute                         ▷ Translate to SUMO route
20:         Nodes to links ID translation for SUMO route
21:         traci.route.add
22:
23:     Method VehicleMoveIn/MoveOut                ▷ Idle vehicle behavior
24:         traci.vehicle.moveToXY
25:
26:     Method VehiclePositionUpdate                ▷ Index position in scheduler route
27:         traci.vehicle.(getRoadID, getLanePosition, getRoute)
28:         Links to nodes ID translation for route
29:         Index vehicle position in route
```

```
30:  Method VehicleUpdate                                  ▷ Update vehicle in SUMO
31:      if Pickup stop then
32:          Method passengerOnboard
33:      end if
34:      traci.vehicle.setRouteID
35:      if Idle vehicle then
36:          Method MoveIn
37:      end if
38:      Method setStops
39:
40:  Method setStops                                       ▷ Set route stops for vehicle
41:      Create stop array for pickups and dropoffs in current route
42:      traci.vehicle.setStop, traci.vehicle.moveTo for suddenstop
43:
44:  Method PassengerOnboard                               ▷ Check passenger pickup
45:      for (Persons waiting) do
46:          if person pickup stop = vehicle road ID then
47:              if traci.person.getStage ≠ 'driving' then
48:                  Continue vehicle waiting
49:              end if
50:          end if
51:      end for
```

3.6 Conclusion

This research focuses on modeling and optimizing AMoD ridesplitting in microscopic traffic assignment models. Ridesplitting has gained considerable popularity among other on–demand transport modes for being more sustainable. However, it requires extensive research and assessment tools, while considering the operational complexity and recent failures for many of its practical implementations. Ridesplitting service optimization is complex, because the requests are assigned considering the network conditions, service operational times, and in–service passenger time constraints. Hence, the accuracy of modeling service operations, traffic environment, service demand, and network information significantly affects the efficacy of service performance and experimental assessment. The current modeling methods lack such comprehensive modeling methods and focus either on optimization solutions or modeling large–scale service scenarios. This research fills the gap by developing modeling methods that allow efficient AMoD service optimization and realistic simulation. The proposed modeling platform models the service operations in a microscopic traffic environment and conducts robust service optimization by incorporating the problem information from the simulation.

The main contribution of this research is to propose an AMoD modeling framework that allows simulation–based modeling and optimization of the AMoD ridesplitting. The framework integrates a dynamic and stochastic DARP algorithm in a microscopic traffic assignment model and models stochastic trip–based AMoD demand suitable for the most state–of–the–art microscopic simulators. The integration of the dynamic DARP algorithm simultaneously optimizes new requests using the stochastic network and service information from the simulation model, therefore increasing the service optimization efficiency with more realistic request assignments and higher satisfaction of passenger time constraints. Similarly, microscopic traffic and service modeling represent detailed network dynamics, link–based interactive service operations, and multimodal person trip requests, replicating more naturalistic service behavior for realistic assessment. The platform can be used as a service planning and assessment tool or even a step closer to real–time service management. Additionally, since the trip–based demand modeling method is used as a base to estimate the service mode shares and generate stochastic requests, it allows easier adaptability for most traffic simulators and also enables utilization of well-developed OD demand estimation techniques not available for the novel agent–based modeling methods.

The research also include multiple supply enhancements for dynamic DARP integration and modeling AMoD service vehicle behavior. The framework is implemented in python using the microscopic traffic simulator SUMO and its TraCI simulation interface. The DARP algorithm is integrated online with the TraCI server to optimize the service simultaneously alongside the simulation. Different methods are also proposed for, e.g., compatibility of the dynamic DARP scheduler and SUMO by node–to–link transformation, AMoD vehicle operations, and service location optimization. Details on the platform code implementation are also given in section 3.5.3. It is also noteworthy to mention that the proposed methods can also be conveniently used towards modeling and optimizing different service use–cases like meeting point–based service setup or last–mile feeder service for public transport (using multimodal person trips). Moreover, microscopic traffic models also provide many additional advantages. First, they allow modeling of autonomous driving behavior and other automated concepts (e.g., platooning, signal coordination), which can significantly influence vehicle travel and operational times. Then, these models allow modeling of multiple network–based KPIs based on dynamic traffic assignment which can help to conduct detailed service impact assessment (e.g., vehicle/link–based emission modeling, VKT estimation, safety, vehicle/operations delays, etc.). Similarly, they also facilitate modeling of the network–based scenarios (e.g., construction, incidents, etc.) and multimodal scenario setup, e.g., AMoD-PuT integration or comparisons.

4 Demand modeling for AMoD ridesplitting

Contents

This chapter focus on demand modeling methods specific to the unique characteristics of AMoD ridesplitting. First, stated-preference experiments are presented to identify the factors affecting user travel behavior in the presence of high capacity ridesplitting as a separate transport mode. Then, a simplified trip-based demand estimation method is developed to allow easier adaptability of the microscopic AMoD platform by removing the requirement of iterative simulations. Similarly, a theoretic equilibrium model is developed to represent ridesplitting markets, interpreting the interactions of service decisions and the system's endogenous variables at the network level for catering to ridesplitting dynamic ride-matching and detouring. Finally, a utility-based compensation pricing method is also presented that helps to address the riders' uncertainty and inequity experience in ridesplitting, improving service perception and adaptability.

The content of this chapter has been partially presented in the following works, while part of the content is unpublished to date:

Tsiamasiotis, K., Chaniotakis, E., Qurashi, M., Jiang, H., & Antoniou, C. (2021). Identifying and Quantifying Factors Determining Dynamic Vanpooling Use. Smart Cities, 4(4), 1243-1258.

Qurashi, M., Jiang, H., & Antoniou, C., 2022. Microscopic modeling and optimization of autonomous mobility on–demand ridesplitting, *(Submitted)*

Lu, Q., Qurashi, M., & Antoniou, C., 2022. A utility-based compensation pricing method for ridesplitting services. Transportation *(Under revision)*

4.1 Introduction

4.1.1 Background and context

Ridesourcing demand modeling is unique from other transport modes due to the intertwined relationship of the service supply and demand. In a static context, the service availability indirectly affects the service utilization through its influence on attracting service demand; service utilization, in turn, directly affects the available supply (Manski and Wright, 1967; Yang et al., 2002). Therefore, it requires supply-demand interaction to dampen out an equilibrium under fixed conditions, where the equilibrium must be reevaluated for any change in, e.g., demand, supply, or network conditions. For this reason, literature works use market equilibrium (ME) models for static representation of ride-sourcing markets (mainly taxis) (Yang and Wong, 1998; Wong et al., 2001). ME models establish demand-supply equilibrium satisfying both demand and supply equations simultaneously. Since the ridesharing services have gained popularity much recently among research and industry, only a few works focus on modeling ridesharing ME models (covered in chapter 2), in which, as per our knowledge, none of the existing efforts model ME for ridesplitting. The supply-demand equilibrium for ridesplitting is even more complex and interactive. It also requires representation of dynamic trip utilities, i.e., additional waiting and detour times (due to dynamic matching and detouring) subject to service availability and utilization.

For modeling ridesplitting with time-dependent representation, the demand attraction (mode choice) is dynamic at the level of individual ride requests due to stochasticity (and uncertainty) in both service availability and possible trip utilities for each trip. Each person opts to call the service given its case-specific service availability and expected trip utility in competition with other transport modes, which is also affected by his preference towards the value of different trip attributes (cost, waiting, and travel time). For such dynamic cases, the requirement of attaining the equilibrium lies in damping out the difference between the expected and experienced trip utilities for the set of attracted demand, i.e., given all inputs fixed, the attracted service demand, when served, should experience similar trip utilities set as expected during demand estimation (Liu et al., 2019).

Figure 4.1 shows the illustration for an on-demand mode choice system. The AMoD demand is evaluated from the overall network travel demand depending on AMoD service features, pricing, and environment (network) specific variables. For equilibrium, first, the AMoD service features indirectly define expected trip utility, while later, the simulation results in actual (experienced) trip utility for the resulting AMoD demand. The experienced trip utilities are then set as expected to reestimate the AMoD demand, and the process runs with iterative simulations until convergence between the expected and experienced trip utilities. Note that using the equilibrium-based mode choice method is infeasible for DTA models (especially meso- microscopic models) due to several reasons. First, good convergence of such an equilibrium gets difficult due to the presence of stochasticity (noise) in simulations. Then, a computationally expensive iterative process is required at any change in input conditions, which includes any change in supply, demand, and network conditions.

Since AMoD demand modeling requires equilibrium with iterative simulations, most AMoD modeling efforts in literature are agent-based frameworks (Basu et al., 2018; Horni et al., 2016). These efforts do microscopic demand modeling, i.e., individual agents with certain activity chains travel through the network with dynamic (case-specific) mode choices to conduct each activity. The overall level-of-service (LOS) experience for each agent is measured by a score, and iterative simulations are run with changes in agent choices (departure, modes) to minimize the overall LOS scoring (attaining user equilibrium). Therefore, agent-based models can conveniently model demand for on-demand mobility services. However, due to its recent emergence and a high

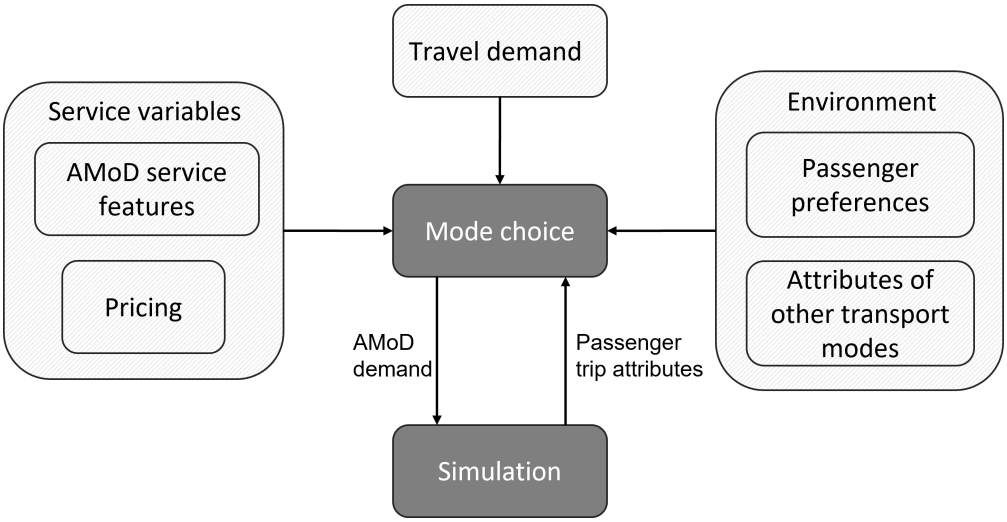


Figure 4.1: Mode choice for AMoD

degree of disaggregation and stochasticity, microscopic demand modeling lacks well-established methods. For example, the model calibration problem is much more complex, and no calibration approach can calibrate all modeling aspects simultaneously (Moeckel et al., 2020), stochastic nature causes significant variations in model outputs (Wegener, 2011)). Moreover, most current agent-based efforts also lack either detailed traffic congestion modeling or service optimization (chapter 3), compromising on model accuracy mainly either due to their focus on network-level evaluations or the computational requirements of iterative simulations. Moreover, demand modeling in the most state–of–the–art traffic simulators is modeled by well-established trip-based demand methods (i.e., origin-destination matrices having trip counts between aggregated traffic zones). However, these traffic models, in turn, lose the degree of disaggregation by modeling aggregated trips. In most cases, these models do not model iterative supply-demand interactions for changing individual travel choices (route choice in some cases is modeled with iterative nature to model equilibrium). Thus microscopic models with trip-based models remain infeasible for equilibrium-based AMoD demand modeling with repeated simulations.

In the context of the passenger preference input (figure 4.1), it defines how the travelers perceive the value of different trip attributes of each transport mode, e.g., waiting time, in-vehicle time, trip cost. Each on-demand service is unique from the other. Since they fill the gap between fixed-route public transport and flexible private cars, they differentiate in operations and their sense towards the users. One major limitation is that existing literature does not include ridesplitting (users with different origin-destination and departure times served with dynamic matching and detouring) as a discrete alternative in mode choice (and stated preferences) studies. However, finding the user travel preferences specific to ridesplitting as a separate mode can significantly improve demand estimation. Especially when these services have failed to establish (e.g., Bridj, Via, Kutsuplus, also others mentioned in Currie and Fournier (2020)), understanding the specific factors that influence their use by potential customers is very beneficial. It can also help develop more efficient ways to do demand management through, e.g., request assignment, dynamic vehicle dispatching, dynamic pricing (see chapter 2).

A supply-demand equilibrium represents a stable state under given fixed conditions, and any change in input variables can shift the stable equilibrium state. On the one hand, this effect makes efficient modeling methods crucial to plan and assess the service against any changes. On

the other hand, it provides service and demand management opportunities to achieve optimum output for a target/policy objective. Specifically, working with passenger preferences (mainly affecting monetary and time costs), the operator can influence the adaptability of the service (Qiu et al., 2018; Guan et al., 2019a). Among others, travel time (attributed with supply) and trip fare are the most critical indicators of the level of service (LOS) (de Ruijter et al., 2020). Therefore, planning/managing service supply and dynamic pricing can help optimize service usage significantly. Furthermore, since ridesplitting serves riders with different origin-destination and departure times through dynamic detours. The riders experience both uncertainty and inequity in their trip level of service (i.e., trip fare and travel time), which can significantly affect how riders perceive the service and its adaption. Apart from the racial and gender discrimination disclosed in the literature (Ge et al., 2016), this inequity aspect exists between distinctive individual ridesharing trips and requires mitigation through appropriate dynamic pricing methods to minimize the uncertainty effect in expected trip utilities for ridesharing riders.

4.1.2 Research contributions

Given the mentioned literature gaps in AMoD demand modeling, our research contributes by exploring and developing methods that can help solve and simplify the demand modeling needs specific to ridesplitting. First, we identify the passenger preferences distinctive to high capacity ridesplitting (dynamic vanpooling) by analyzing factors evaluated from a stated preference survey. Then, we propose a market equilibrium (ME) method that specifically represents dynamic trip utilities in AMoD ridesplitting. Furthermore, a simplified trip-based demand estimation method is also proposed to approximate AMoD attracted demand for microscopic simulations. Finally, we also develop a utility-based dynamic pricing method that addresses the uncertainty issue among passenger trip utilities through a compensation strategy. Below, we provide a list of contributions covered in this chapter:

1. **Passenger preferences:** This research investigates the factors affecting the travel preferences of individuals by the emergence of high occupancy ridesplitting (dynamic vanpooling) in the transportation system. A novel stated preference experiment is conducted and analyzed using discrete choice models (Multinomial Logit and Ordered Probit models) to explore the factors that affect the use of vanpooling and the willingness to pay. The obtained factors provide insight into identifying the potential users for this transportation service. Meanwhile, the analysis also helps quantify how users perceive the value of monetary cost and in-vehicle and waiting time. These preferences are crucial inputs for estimating AMoD demand and later establishing demand management strategies, e.g., request allocation or dynamic pricing.

2. **Ridesplitting market equilibrium:** This research contributes to developing a market equilibrium (ME) modeling method for ridesplitting markets. ME models can analytically represent the on-demand service markets by solving the supply-demand interaction using a system of simultaneous equations. Ridesplitting differentiates itself from other taxi and ridepooling markets due to the presence of dynamic matching and detouring. Therefore, we extend the literature-based ME methods to represent the ridesplitting service supply (passenger seats inside of vehicles) and trip utility attributes (expected detour and waiting times). The expected detour and waiting times are modeled as a function of service supply and demand along with hyperparameters that tune their relationships specific to the modeled market traits. We also propose a calibration method to estimate these hyperparameters and capture distinct market characteristics, e.g., network geometry.

3. **Trip–based mode choice methods:** Since AMoD demand modeling requires supply-demand equilibrium through iterative simulations, it is infeasible for microscopic models due to the involved stochasticity and computational times. This research proposes a simple trip–based AMoD demand modeling method which removes the requirement of modeling supply-demand equilibrium and allows modeling AMoD demand similar to a conventional mode choice setup. The method uses a simplifying assumption of considering the riders' trip time preferences as the expected service attributes, which is only valid due for AMoD ridesplitting, which integrates a dynamic DARP scheduler solving vehicle routing with explicit time windows constraints. Moreover, since trip-based demand modeling is widely adopted in traffic simulators, our method is much easier to adapt and model AMoD demand. It also provides the advantage of better model accuracy due to the presence of well-established model calibration methods for trip-based demand models.

4. **Utility–based dynamic pricing:** In ridesplitting, the riders experience both uncertainty and inequity in their trip level of service (LOS), significantly affecting how riders perceive and adapt to the service. This research proposes a utility-based dynamic pricing method that compensates the trip fares to reduce the standard deviation of trips' utilities. The method adds both equity and certainty among trips with varying LOS. It compensates trips with a utility less than a LOS threshold based on a predefined function. Applying utility-based dynamic pricing can also improve the trip-based demand approximation method by setting the LOS threshold as the (minimum) expected utility to reduce the effects of LoS inequity on expected utilities.

4.2 Passenger preferences

Passenger preferences define how users perceive using different transport modes. They weigh the user's value for each particular trip attribute (e.g., walking, waiting, riding, monetary cost) overall or specific to each transport mode. Two different methods are generally used to evaluate passenger preferences. One is 'Stated Preference' (SP) which is a survey-based method to estimate user choice preferences, while the other is 'Revealed Preference' (RP) is a data-based method to estimate (reveal) peoples' decisions by actual choice data. As mentioned in section 4.1.1, ridesplitting differentiates in terms of operations and its sense for the users due to the presence of dynamic matching and detouring. Therefore, to model its demand, it requires reevaluation of passenger preferences specific to considering it as a separate transport mode. This section describes the use of the SP method to evaluate passenger preferences for high capacity ridesplitting. For the simplicity of respondents, the survey uses a specific service name of dynamic vanpooling that refers to a ridesplitting van service providing point-to-point flexible travel upon users' preferred time windows. Below, we describe the stated preference survey along the model estimation methods and analysis results.

4.2.1 Stated preference survey

The survey consisted of a three-part questionnaire. The first part asks the respondents about their current travel characteristics, such as the most frequently used mode of transport, average commuting time (both ways), car availability, possession of driver's license, and their satisfaction with the current way of traveling. The second part presented hypothetical choice scenarios to the respondents to ask their preference in a 5-point rating scale (ranging from the strong preference for the first alternative to the strong preference of the second alternative). The alternatives considered in this study were a private car, public transport, and dynamic vanpool. The part

consisted of nine scenarios, and each scenario with two alternatives (the first three scenarios involved a private car and dynamic vanpool, the subsequent three involved public transport and dynamic vanpool, and the last three involved only dynamic vanpools). Three attributes defined the choice alternatives, i.e., in-vehicle travel time, total travel cost, walking/waiting time, or searching time for parking, each varying on three levels. Table 4.1 summarizes the values of attribute levels. The last part of the survey included standard demographic questions regarding gender, age, education level, principal occupation, and income. The full factorial design of each pair of alternatives includes 729 choice sets. Hence, an efficient design was applied to eliminate the choice situations using Federov's exchange algorithm (Wheeler, 2004).

Table 4.1: Summary table of alternatives, attributes and attribute levels.

Alternative	Attribute	Attribute levels
Private car	In vehicle travel time	12, 20, 28 min
	Total travel cost	5.00, 7.00, 9.00 €
	Walking and parking time	2, 6, 10 min
Public Transportation	In-vehicle travel time	16, 26, 36 min
	Total travel cost	1.50, 2.20, 2.90 €
	Walking and waiting time	7, 12, 17 min
Dynamic Vanpool	In-vehicle travel time	14, 24, 34 min
	Total travel cost	4.00, 6.00, 8.00 €
	Walking and waiting time	5, 10, 15 min

The survey was conducted between March and May 2019. The first distribution method was by using mailing lists and social media platforms. The drawback of this method is that the survey is not distributed randomly among the population, and potential biases may occur. Therefore the second method of using printed flyers was also used, and 2000 flyers were distributed in the chosen inner-city area of Munich. This method allowed targeting the population who do not use social media or similar networks. A total of 240 survey responses were collected, which were reduced to 208 after initial analysis (due to missing values or unsuccessful submission) and included 1872 choice observations. The survey included a major subsample of the Munich region (102 responses).

4.2.2 Model estimation

Two types of econometric models are estimated using the data collected from SP survey. These models include ordered probit models and multinomial logit (MNL) models. The models are estimated in PandasBiogeme software (Bierlaire, 2018). Note that, prior to model estimation, a priori expectations of the estimated coefficient signs and magnitudes were presumed, based on similar studies on mode choice (e.g., Kuppam et al. (1999)), and used to verify their estimated counterparts. In the sections below, the application and results of these two models are presented.

Ordered Probit model

The ordered probit (OP) models are developed to estimate the ordered choices as the dependent variable. Note that in our data each choice variable takes numerical values between 1 and 5 and considering this 5-level rating scale ("Certainly A", "Probably A", "Indifferent", "Probably B", "Certainly B"), there are 4 thresholds (k1 to k4) that separate the 5 choices. This means that respondents choose the alternative "Certainly A" if the utility is lower than k1, alternative "Probably A" if the utility is between k1 and k2, alternative "Indifferent" if the utility is between k2 and k3, alternative "Probably B" if the utility is between k3 and k4 and alternative "Certainly B" if the utility is greater than k4. Similarly, The values of the three attributes (i.e., in-vehicle travel time, travel cost and waiting/walking time) differ among the alternatives and, therefore, responses should be rearranged so that the fastest (and more expensive) option is always second, implying that an increase in threshold level corresponds to a higher preference for the faster alternative. The model specification started with a simple model with the main variables (in-vehicle travel time, travel cost and waiting/walking time); then, meaningful variables are added progressively. The variables with high significance are only retained (significance level above 95%). After evaluating different specifications, the final OP model is estimated. The major variables resulted from the model are presented in table 4.2.

Table 4.2: Ordered Probit estimation results.

Variables	**Coeff. estimate**	**Robust asympt. std. error**	**Robust t-Stat**	**Robust *p-Value***
In-vehicle travel time	-0.0673	0.00575	-11.7	0.00
Total travel cost	-0.335	0.0228	-14.7	0.00
Waiting/Walking time	-0.0448	0.00607	-7.38	0.00
PT	0.189	0.0563	3.36	0.00
Age: 18-25	-0.195	0.082	-2.36	0.00
Age: 46-65	-0.255	0.0935	-2.73	0.01
Car as commute mode	0.211	0.0684	3.08	0.01
60<Commuting time<90	0.246	0.0632	3.89	0.00
Employee	-0.437	0.116	-3.76	0.00
Income<3000€	-0.11	0.057	-1.90	0.03
Student	-0.562	0.12	-4.63	0.00
Household size: 2	-0.127	0.0594	-2.15	0.01
Household size>4	-0.272	0.0835	-3.56	0.02
Commute satisfaction	-0.083	0.0311	-2.68	0.01
Number of cars in household: 3	0.558	0.1425	3.92	0.00
Driving License	0.461	0.0784	5.88	0.00
Carsharing membership	0.183	0.070	2.63	0.02

Ordered Probit estimation results.

Bike-sharing membership	-0.265	0.0775	-3.41	0.00
Real-time information services	0.0652	0.0222	2.94	0.00
Affinity to technology	0.0686	0.0355	1.93	0.03
Social media	-0.048	0.0234	-2.05	0.04
Extraverted, enthusiastic	0.065	0.0314	2.07	0.04
Sympathetic, warm	-0.110	0.0358	-3.07	0.00
Threshold parameters for index model				
k1	-1.775	0.272	-6.56	0.00
k2	-0.821	0.2688	-3.05	0.00
k3	-0.659	0.2687	-2.45	0.00
k4	0.224	0.2687	0.83	0.00

Summary statistics

Number of observations: 1845

Number of estimated parameters: 27

Initial log-likelihood: -2780.77

Final Log-likelihood: -2507.13

Likelihood ratio test: 547.30

Rho-square for the final model: 0.10

The signs and magnitudes of the coefficients are reasonable and consistent with the respective a priori expectations. Starting off, the estimated coefficients of the main attributes (in-vehicle travel time, travel cost, and waiting/walking time) are negative (as expected). For socio-demographic characteristics, young (18 - 25 years old) and middle-aged (46 - 65 years old) respondents are more likely to choose a slower and less expensive mode of transport. Similarly employed (full- or part-time) and low income (less than 3000 €) respondents have a higher tendency to use a dynamic vanpool or public transport (PT), and students or those living in a household with at least two people have a tendency of choosing a less quick and cheaper mode of transport. Concerning user travel characteristics, respondents who commute daily with a private car and whose commuting time is between 60 and 90 min (both ways) have a higher tendency to use a faster and more expensive option. Similarly, the possession of a driving license along with more than two cars in the household leads to a higher likelihood of choosing a faster, more expensive option. Moreover, for respondents being members of a shared mobility service, bike-sharing users have a higher propensity to use a more affordable option, while the opposite effect is observed for car-sharing users. Other noting coefficients include participants who often use social media platforms tend to choose a slower and less expensive mode of transport.

Multinomial Logit Model

Multinomial logit model (McFadden et al., 1973) is also estimated to directly extract values of time and provide comparability with other studies' results. Since the respondents stated their preferences on a 5-point rating scale, MNL models could not be specified directly due to the violation of the Independence for Irrelevant Alternatives (IIA) (Antoniou and Polydoropoulou, 2015). Therefore, the 5-point scale of the response is transformed to a binary choice (Antoniou et al., 2007b). The responses with varying preferences for option alternative A (or B) were categorized as having a preference for choice A (or B, respectively). Responses with no preference between two options and responses gathered from the experiments which consider the same mode (dynamic vanpool) were excluded from the model. The model specification started with the derivation of an initial model, which consisted of the alternative specific constant and the main attributes of the alternatives. Subsequently, additional parameters are added, and those with a lower significance level (below 95%) are removed. Note that since model specification and the number of observations are different, a direct comparison of the Rho-square between the Ordered Probit and the MNL model cannot yield any meaningful conclusion for model performances. The major variables resulted from the model are presented in table 4.3.

To summarize the model results, we start with sociodemographic characteristics. The respondents aged 26 - 45 and 56 - 65 years are more likely to choose dynamic vanpooling instead of PT and private cars. Furthermore, participants with high monthly incomes (more than 7000 €) prefer to use dynamic vanpooling compared to PT. While students and those who hold a bachelor's or master's degree are more likely to choose PT than dynamic vanpooling. Then, concerning user travel characteristics, respondents who commute between 30 - 60 min tend to choose dynamic vanpooling instead of a private car. Besides, respondents who commute more than 30 - 90 min are significantly more likely to choose dynamic vanpooling in comparison with PT. Similarly, respondents who possess a driving license tend to choose the private car and dynamic vanpooling instead of PT. While a significant propensity to use dynamic vanpooling and private vehicles is identified by households that own at least three vehicles. Moreover, for respondents being members of a shared mobility service, bike-sharing users are less likely to use a private car and dynamic vanpooling. At the same time, car-sharing users or PT seasonal ticket holders are more likely to choose dynamic vanpooling.

Table 4.3: Multinomial Logit estimation results.

Variables	**Coeff. estimate**	**Robust asympt. std. error**	**Robust t-Stat**	**Robust *p-Value***
In-vehicle travel time (Car)	-0.137	0.0258	-5.20	0.00
Total travel cost (Car)	-0.445	0.112	-4.14	0.00
Walking/Parking time (Car)	-0.0938	0.0352	-2.65	0.01
In-vehicle travel time (Dynamic vanpool)	-0.147	0.0203	-7.33	0.00
Total travel cost (Dynamic vanpool)	-0.722	0.0769	-9.15	0.00
Waiting/Walking time (Dynamic vanpool)	-0.0755	0.0253	-3.03	0.00
In-vehicle travel time (PT)	-0.104	0.0206	-4.79	0.00
Total travel cost (PT)	-0.811	0.193	-4.00	0.00

Multinomial Logit estimation results.

Waiting/Walking time (PT)	-0.133	0.0316	-3.96	0.00
Age: 26-45 (PT)	-0.616	0.26	-2.42	0.02
Age: 56-65 (Car)	-1.25	0.524	-2.73	0.01
Monthly income >7000€ (PT)	-0.946	0.447	-2.10	0.04
Bachelor's or Master's degree (PT)	0.571	0.241	2.36	0.02
Student (PT)	0.582	0.294	2.00	0.05
PT as commute mode (PT)	1.31	0.312	4.05	0.00
Bike as commute mode (Car)	-0.817	0.409	-2.04	0.04
30<Commuting time<60 (Car)	-0.591	0.217	-2.75	0.01
30<Commuting time<60 (PT)	-0.816	0.249	-3.33	0.00
60<Commuting time<90 (PT)	-0.946	0.289	-3.39	0.00
Commuting time>90 (Car)	-0.656	0.333	-1.88	0.06
Driving license (Car)	0.968	0.274	3.55	0.00
Driving license (PT)	-1.42	0.364	-3.82	0.00
Available cars in household: 3 (PT)	-1.45	0.615	-2.18	0.03
Carsharing membership (PT)	-0.546	0.267	-1.99	0.05
Bike-sharing membership (Car)	-0.73	0.314	-2.14	0.03
Bike-sharing membership (PT)	0.797	0.343	2.13	0.03
PT seasonal ticket (Car)	-0.578	0.234	-2.54	0.01
PT seasonal ticket (PT)	-0.946	0.313	-2.94	0.00
Carsharing familiarity (Car)	-0.186	0.0939	-2.07	0.04
Uber familiarity (Car)	0.191	0.0942	2.05	0.04
Real-time information services (Car)	0.204	0.0857	2.48	0.01
Environmental awareness (Car)	-0.302	0.117	-2.56	0.01
Anxious, easily upset (PT)	-0.316	0.102	-3.14	0.00
Disorganized, careless (PT)	0.233	0.124	2.02	0.04
Conventional, uncreative (PT)	0.279	0.11	2.78	0.01
Sympathetic, warm (Car)	-0.298	0.12	-2.61	0.01
Sympathetic, warm (PT)	0.273	0.131	2.31	0.02

Summary statistics
Number of observations: 1182
Number of estimated parameters: 37
Initial log-likelihood: -819.30
Final Log-likelihood: -600.11
Likelihood ratio test: 438.38
Rho-square for the final model: 0.268
Rho-square-bar for the final model:0.222

Value of time

The coefficients estimated for cost and time can be applied to calculate the willingness-to-pay of the respondents to use the transport modes. The utility is generally unitless but the ratio of two cofficients for travel time (in minutes) and travel cost (€) would result in having the units of €/minutes which is the expected unit for value-of-time (VOT) measure. The willingness-to-pay (VOT) for in-vehicle time and waiting time is calculated using equation 4.1 and 4.2.

$$VOT_v = \frac{\beta_t}{\beta_r} * 60 \tag{4.1}$$

$$VOT_w = \frac{\beta_w}{\beta_r} * 60 \tag{4.2}$$

where VOT_v is value of in-vehicle time and VOT_w is value of waiting/walking or parking time. Also note that, β_t β_w β_r are estimated coefficients of in-vehicle time, waiting/waiting or parking time, and travel cost. Table 4.4 summarizes the VOT obtained from the OP and MNL models. Note that, in OP models a generalized VOT is calculated because the coefficients of travel times and travel cost are common.

Table 4.4: Summary table of VOT obtained from OP and MNL models.

	OP Model	**MNL Model**
Generalized VOT_v	12.05 €/h	-
Generalized VOT_w	8.02 €/h	-
VOT_v (Car)	-	18.47 €/h
VOT_w (Car)	-	12.65 €/h
VOT_v (Vanpool)	-	12.22 €/h
VOT_w (Vanpool)	-	6.27 €/h
VOT_v (PT)	-	7.69 €/h
VOT_w (PT)	-	9.84 €/h

The results in table 4.4 show a counterintuitive finding that the generalized value of in-vehicle travel time (12.05 €/h) is greater than the value of walking/waiting time savings (8.02 €/h). According to Wardman et al. (2016), the disutility for walking is 1.85 times higher than that for in-vehicle time savings [19]. One potential consideration of this result is that the experimental design's combination of waiting and walking time might have affected these attributes' influence on respondents' choice decisions. From the VOT estimated for each alternative, it is observed that people are willing to pay significantly more to use private cars (18.47 €/h) compared to dynamic vanpooling (12.22 €/h), while the VOT for PT (7.69 €/h) is the lowest among all alternatives, as expected. Concerning the value of walking/waiting time, it is observed that respondents are willing to pay more to reduce this time for PT compared to dynamic vanpooling or cost waiting/walking for PT higher than dynamic vanpools. Finally, the validity of the obtained values is also investigated from the literature. Wardman et al. (2016) reported values obtained by national studies, EIB (European Investment Bank), and meta-analysis models, indicating differentiation in VOT based on trip purpose. Note that this study does not include the trip purpose in the experimental design due to its dimensionality. Therefore, the magnitude of the obtained values from the models is within the estimated values by Wardman et al. (2016).

4.3 Trip-based AMoD demand modeling

As discussed before (in section 4.1.1), mode choice for on-demand services requires iterative supply-demand interaction (equilibrium) due to their unique intertwined relationship. For ridesplitting, the supply-demand interlinkage further increases due to dynamic ride matching and detouring. The existing literature efforts related to the demand modeling of on-demand services, use either agent-based frameworks (Basu et al., 2018; Horni et al., 2016), market equilibrium models (section 4.4), or simpler and direct iterative methods for equilibrium estimation (Liu et al., 2019). Agent-based frameworks by their modeling nature allow iterative supply-demand interaction within the iterative agent scoring simulations. However, these models suffer in model accuracy due to the degree of stochasticity, disaggregation, and the lack of suitable calibration methods, therefore are mainly utilized for network-level assessment. Whereas, employing the iterative equilibrium-based mode choice setup is rather infeasible for microscopic models due to their stochasticity, computational requirements, and dynamic network modeling.

To solve the problem, we propose a naive trip–based method specifically applicable to AMoD ridesplitting which removes the iterative simulations requirement and can better approximate the ridesplitting demand. The method uses a conservative assumption of considering the passenger trip time preferences as the expected service attributes. This assumption is made considering the ridesplitting service concept and the integration of the dynamic and stochastic DARP scheduler which optimizes the service as a vehicle routing problem with explicit time window constraints. Note that, trip-based demand modeling is widely adopted by most state–of–the–art traffic simulators and therefore the proposed method is give easier adaptability of the AMoD modeling platform (discussed in chapter 3). Moreover, It also provides the advantage of better model accuracy due to presence of well-established mode calibration methods (see chapter 2 and 5).

Next, we describe the conventional mode choice setup for trip-based demand modeling, followed by the description of simplified ridesplitting mode choice method and the procedure to generate stochastic requests using time-dependent OD matrices.

Conventional trip-based mode choice

The conventional mode choice method includes classic discrete choice modeling based on random utility theory, in which, the attractiveness of choosing a transport mode is modeled by the concept of utility. This utility of a transport mode is calculated using equation 4.3.

$$U = V + \epsilon \tag{4.3}$$

where U is the utility, V is the deterministic component of the utility and ϵ is the disturbance.

The deterministic utility is derived from the characteristics of a transport mode and is defined with a linear combination of variables like travel time, cost, comfort, and waiting time. Although many factors can be closely related to the riders' intention for choosing ridesplitting (Wang et al., 2020), e.g., personal inventiveness, environmental awareness; time, and cost are always the primary indicators that influence riders' choice. Therefore, considering AMoD ridesplitting as other conventional models (ignoring supply-demand equilibrium) with the difference in perceptions among travel time, waiting time, and cost; the utility function of choosing the service (i.e., the deterministic part V) is constructed as:

$$V = \beta_t t + \beta_w w + \beta_r r \tag{4.4}$$

where t, w, and r denote the travel time (sum of direct trip time and detour time), waiting time, and trip fare, respectively. Meanwhile, β_t, β_w, and β_r correspond to the preference coeffi-

cients that are evaluated from the discrete choice models under the passenger preference analysis (through stated or revealed preference methods, see section 4.2.

To estimate a trip-based OD matrix for AMoD ridesplitting, its utility must be contrasted with the utility of other transport modes. In addition, we assume that all travelers make decisions objectively based on the perceived utilities of the available transport mode. If the error term ϵ is assumed to follow the Gumbel distribution, the utility of using AMoD ridesplitting can be transformed into a probability value. For this purpose, the simplest and popular discrete choice model, i.e., multinomial logit model (MNL), can be used which is commonly applied for transport mode choice behavior (Vrtic et al., 2010; Krueger et al., 2016). The probability $P_{ij,rs}$ of choosing ridesplitting as a transport mode for OD pair ij is given by:

$$P_{ij,rs} = \frac{e^{V_{ij,rs}}}{\sum_{m \in \mathbb{M}} e^{V_{ij,m}}} \tag{4.5}$$

where $\mathbb{M}$ is the set of available transport modes in the system. Finally, given the overall OD pair demand D_{ij}, the AMoD ridesplitting demand attraction Q_{ij} is estimated as:

$$Q_{ij} = D_{ij} P_{ij,rs} \tag{4.6}$$

Simplified AMoD ridesplitting mode choice

In AMoD ridesplitting, riders with different origin - destination and departure times are served together by dynamic matching and detouring. Therefore, the riders provide their trip time preferences of maximum waiting and arrival time in advance which act as the optimization constraints in the dynamic DARP algorithm solving the routing problem with explicit time windows (see chapter 3, section 3.3). Since, the DARP with explicit time windows consider the time preferences as hard constraints the trips are served within them. Further note that a conservative assumption can be made that riders only consider calling for the service when the preferred time windows can still result in the higher trip utility compared to other transport modes. Therefore, for simplification it is possible to set these preferences as expected service trip variables in equation 4.4.

The expected waiting and travel times w_{ij}, t_{ij} for an OD pair ij are modeled as a function of riders' flexibility and direct travel time t_i^d in the ridesplitting utility function (equation 4.4).

$$w_{ij} = t_{ij}^d w_p \tag{4.7}$$

$$t_{ij} = t_{ij}^d + t_{ij}^d d_p \tag{4.8}$$

where d_p, w_p are coefficients for riders' flexibility for detour and waiting times. Although both flexibility coefficients can be a factor of direct time t_d, waiting times can also be set directly as a fixed time compared to the walking-waiting for PuT and car-sharing SAVs or parking time for cars. Note that to model population heterogeneity or market segmentation, different user groups can be created with varying sets of time flexibility coefficients (d_p, w_p). These coefficients only provide an average time preference value for all requests, while time preferences for individual requests are assigned during stochastic request generation (discussed in the next section). The third component of trip price r_{ij} (equation 4.4) can be modeled as a function of direct time t^d with added compensation depending on varying riders' flexibility (equation 4.9). Due to varying passenger flexibility, compensations can help achieve equity of overall trip utility, e.g., equation 4.10 shows an example of setting trip fares based on passenger time preference.

$$r_{ij} = p t_{ij}^d - c_{ij} \tag{4.9}$$

$$c_{ij} = (t_{ij}^{d} d_p)\alpha_t + w_{ij}\alpha_w \tag{4.10}$$

where p is the unit price (per km) and α_t, α_w is the price compensations per unit time for the riders' detour and waiting time flexibility. Note that compensations can also be proposed as a dynamic pricing strategy to provide certainty and equity among served riders. One such dynamic pricing method is proposed in section 4.5

Stochastic requests

A ridesplitting service considers individual time-varying rider requests separately under DARP optimization. Thus, we need to translate the aggregated service OD matrix Q (equation 4.6) into individual requests. Furthermore, the request generation method also need to model the stochasticity among trip requests to replicate the real-world conditions. The stochasticity among trip requests can be present in three aspects, i.e., departure time, the origin/destination locations (within each subsequent TAZ), and trip time preference. Below, we discuss the method of generating each of the three request attributes separately:

- **Departure time:** Aggregated demand for specific time intervals is represented by time-dependent OD matrices that result from a travel demand model combining departure choice model and average daily OD demand matrix. For further disaggregation into individual trip departures statistical distributions like Uniform or Poisson distribution are generally employ among most traffic models. The traffic simulators already contain such facilities to generate departure times for individual trips, given time-dependent OD matrices are available as an input.
- **Origin/destination locations:** An OD matrix contains set of aggregated trips for each OD pair. To generate origin-destination locations for individual trips, a set of specific links or connectors are chosen for each TAZ. For AMoD, these connectors could either be all serviceable network links, a few optimally chosen network links covering the whole network, or optimized meeting points (see chapter 3, section 3.5.2). Different weights can be assigned to each connector to distribute the trip production and attraction within the TAZ. Note that, individual trips are randomly generated (on their departure times) between the connectors of subsequent origin and destination TAZ. All traffic simulators which model trip-based demand contain facilities to create connectors-based individual trips, e.g., OD2Trips in SUMO.
- **Trip time preferences:** The average time flexibility of all passenger requests is defined by d_p, w_p coefficients (equation 4.7 and 4.8) which can varying for different user groups catering for model population heterogeneity or market segmentation. To generate stochastic individual requests, the corresponding d_p, w_p coefficients are randomly perturbed using the Gaussian distributions $\mathcal{N}$ with mean μ and standard deviation σ. Furthermore, individual trip price r can also be adjusted (with compensation) depending upon the varying trip flexibility.

Platform implementation

The proposed ridesplitting mode choice method is part of the AMoD modeling architecture discussed in chapter 3. The method is coded in python taking in time-dependent travel OD matrices as input to first estimate AMoD demand matrices and later use a SUMO tool named "OD2Trips" to generate individual requests with departure time and origin-destination links. Note that the "OD2Trips" tool takes in the estimated AMoD demand matrix (written in SUMO

readable format) along the service links or connectors (written as an AMoD TAZ file "*.taz") to generate a trip file (*.trips.xml) holding all individual requests. These requests are then read in python to create their subsequent information i.e., time preference windows and trip price. Finally, the simulator executor module reads this information to trigger request events at each request's departure time. The trip request file is also later used to log the served trip attributes resulting from the simulation.

4.4 Ridesplitting market equilibrium

Market equilibrium for an on-demand service is described as the system state, when the supply-demand interaction eventually damps out under certain regulated conditions (e.g., trip fare). At this state, relationships between the system endogenous variables (e.g., passenger demand, average waiting and detour times) can be satisfied, under a specific operation strategy (i.e., vehicle fleet size, trip fare). Mathematically, the equilibrium is established when both demand and supply equations are satisfied simultaneously (Arrow and Debreu, 1954). Literature efforts does systematically investigate equilibrium models of taxis (Cairns and Liston-Heyes, 1996; Wong et al., 2001; Yang et al., 2002, 2010), ridematching (Bimpikis et al., 2019) and ridepooling (Ke et al., 2020) markets, however no such work exist which focuses on modeling ridesplitting (more details in chapter 2).

Modeling ridesplitting ME is more peculiar than taxi markets due to dynamic ride-matching and detouring. Therefore, it further requires representation of dynamic trip utilities, i.e., additional waiting and detour times also subject to service availability and utilization. Generally, the expected waiting time is deemed to be related to the number of available vehicles (Cairns and Liston-Heyes, 1996; Li et al., 2019b; Ke et al., 2020). However, given the nature of ridesplitting (any ride request can be matched with any vehicle having vacant seats upon feasibility), we need to modify this assumption and consider the expected waiting time to depend on the number of available seats subject to vehicle fleet size and service demand. Moreover, due to the interdependence of the ridesplitting demand, waiting time, and detour time, seat availability also indirectly affects the expected detour time. Further, it is important to emphasize ridesplitting differ from ridepooling by having no picking up status, since vehicles are always available in the matching pool even when they are on the way to pick up riders. Therefore, the pick-up time is accounted as part of rider's waiting time.

In the remainder of this section, we first interpret how to model the service supply, waiting time and detour time (modeling service demand is already discussed in section 4.3). Then, the method to calculate the ME is presented. Apparently, the ridesplitting ME provides an aggregated model of the ridesplitting market and the estimation accuracy of expected detour time and expected waiting time can significantly affect its effectiveness. Therefore, we also propose calibration methods for ME which can estimate its hyper-parameters to help better represent the real market conditions.

Modeling ridesplitting supply

The supply of a ride-sharing service is represented by the number of seats (instead of vehicles). Since the state of each seat can be either vacant or occupied, the summation of vacant seats H_v and occupied seats H_c should be equal to the total number of seats of all vehicles. For a given hour, the conservation equation of seat capacity is thus given by

$$Nn_s = H_v + H_c \tag{4.11}$$

where N is vehicle fleet size, n_s is the number of seats in a vehicle. The utilized seat capacity in one hour can be calculated as

$$H_c = \sum_{i \in \mathbb{Z}} Q_i t_i \tag{4.12}$$

where t_i is the expected travel time (combination of direct and detour trip times), Q_i represents the demand for OD pair i, and $\mathbb{Z}$ is the set of all OD pairs. Substituting equation 4.12 into equation 4.11 results in

$$N n_s = H_v + \sum_{i \in \mathbb{Z}} Q_i t_i \tag{4.13}$$

This seat capacity conservation equation (equation 4.13) bridges the demand and supply of ridesharing services and has to be satisfied at the ME.

Expected detour times

Following the findings from empirical data of real operations in several cities, Ke et al. (2020) assumed the average detour time between two riders is inversely proportional to the ridesharing demand. Mathematically, the average detour time between two riders can be estimated by $\tilde{t}^{(2)} = \tilde{A}/\sum_j Q_j$, where $\tilde{A}$ is a market-specific parameter. Intuitively, more ridesharing requests mean the average distance between two riders becomes shorter, manifested as a reduction in the average detour time on one hand. On the other hand, however, it also increases the possibility of pairing more riders and thus results in a growth of the detour time. Consequently, we can modify the assumption adopted in Ke et al. (2020), where the ridesharing service of pairing is restricted to at most two riders, to extend it to the general case. In addition, in ridesplitting, vehicles and riders can be matched anytime (even en route to serving other riders), and vehicles are allowed for detouring within the neighborhood to pick up new requests. It significantly complicates modeling the pairable influential demand, otherwise limited to a given OD pair. Nevertheless, we introduce a simpler modification, i.e., $t_i^d/\bar{t}^d$, that captures the spatial network difference in the detouring probability allowing to extend the previous assumption and keeping a more general pairable demand influence based on the length of a rider's trip. Mathematically the detour time can be modeled as:

$$\tilde{t}_i = \frac{A_i \sum_j Q_j t_j^d}{N \sum_j Q_j} \tag{4.14}$$

where $A_i = A t_i^d/\bar{t}^d$, i.e., the ratio of direct time t_i^d of the OD pair i with average direct time of the network $\bar{t}^d$, given by $\bar{t}^d = \sum_{j \in \mathbb{Z}} Q_j t_j^d / \sum_{j \in \mathbb{Z}} Q_j$. Note that A is a market-specific constant.

Expected waiting times

Assuming the matching process of riders and vehicles follows the Cobb-Douglas type production function (Li et al., 2019b), the expected waiting time can be derived to be inversely proportional to the square root of the number of idle vehicles. Considering the sharing nature of ridesplitting services, we assume to model the waiting time as inversely proportional to the square root of the available seat capacity. In addition, we adapt this assumption for fitting network-level modeling by considering (i) the effect of demand over the supply offered to the respective OD pair using Q_i^θ ($\theta > 0$), (ii) the spatial difference of vehicle allocation using $\eta_i \in (0,1)$, and (iii) the supply attraction relative to demand in the neighborhood using $\Omega_i (\Omega_i > 0)$. Mathematically, the expected waiting time for trips from i^o to i^d is estimated by

$$w_i = \frac{BQ_i^{\theta}}{\Omega_i \sqrt{\eta_i H_v}} \tag{4.15}$$

where B is a market-specific parameter. Below are details for each consideration

- Q_i^{θ} ($\theta > 0$) represents demand over the supply affect offered to i, where θ measures the intensity of the influence. Higher demand adds in the "competition" among riders for the given supply, especially when the demand is greater than the supply, also commonly used in waiting time model for network-level e-hailing taxi market, as in He et al. (2018).
- $\eta_i \in (0,1)$ is a percentage value for representing spatial difference of vehicle allocation and measures the vacant seat capacity assigned to i. Note that total capacity is $\sum_j \eta_j = 1$. The vacant seats for each OD pair are allocated by its spatial characteristics, i.e., origin and destination distance to the city center (denoted by λ_{i^o} and λ_{i^d}) and the OD distance (denoted by d_i). The waiting time for OD pairs near the city center is shorter since more vehicles drive through the city center and thus more supply (Li et al., 2019c). Tu et al. (2021) also found that the distance to city center is one of the key influencing factors of ridesplitting ratio. Likewise, the distance between i^o and i^d also determines if the vehicles are willing to detour to catch these requests. To calculate the seats distribution, we apply the form of inverse distance weighting function as below:

$$\eta_i = \frac{\left((\lambda_{i^o} + \lambda_{i^d})^{\rho_c} d_i^{\rho_d}\right)^{-\rho}}{\sum_j \left((\lambda_{j^o} + \lambda_{j^d})^{\rho_c} d_j^{\rho_d}\right)^{-\rho}} \tag{4.16}$$

  where ρ_c and ρ_d are positive parameters for measuring the influence of the proximity to the city center and OD distance, respectively, and ρ ($\rho > 0$) is the power parameter.
- $\Omega_i (\Omega_i > 0)$ is measures the supply attraction caused by the relatively high demand for the neighboring pairs, which essentially depends on the temporal demand patterns of the market. To clarify, ridesplitting can match riders with either closer origins or destinations or both (Wang et al., 2019). Specifically, $\Omega_i > 1$ means more vehicles are coming to serve the neighboring pairs of i and vice versa. Ω_i, named as supply attraction factor, is given by

$$\Omega_i = \frac{n_z \sum_{j \in \mathbb{Z}_i} Q_j}{\sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}_k} Q_j} \tag{4.17}$$

  where n_z is the number of OD pairs in $\mathbb{Z}$, $\mathbb{Z}_i$ is the set of neighboring pairs of i which is a subset of $\mathbb{Z}$. Ω_i also reflects that ridesplitting vehicles are allowed deviating from a given path within a service area, which is a commonality with the Mobility Allowance Shuttle Transport (MAST) service (Quadrifoglio et al., 2008).

Modeling ridesplitting market equilibrium

Both expected detour $\tilde{t}_i$ and waiting time w_i are modeled as a function of vehicle supply i.e., fleet size N and ridesplitting demand $\boldsymbol{Q}$. $\boldsymbol{Q} = [Q_{i_1}, Q_{i_2}, \cdots, Q_{i_m}, \cdots]^T$ is the vector of all OD-pairs containing ridesplitting demand. Thus, we can rewrite them as $\tilde{t}_i(\boldsymbol{Q}, N)$ and $w_i(\boldsymbol{Q}, N)$, respectively. Recall that the expected travel time is the sum of direct trip time and expected detour time, such that we can rewrite the travel time as $t_i(\boldsymbol{Q}, N)$. Then the utility function (equation 4.4) for the ridesplitting service can be rewritten as:

$$V_i(\boldsymbol{Q}, N) = \beta_t t_i(\boldsymbol{Q}, N) + \beta_w w_i(\boldsymbol{Q}, N) + \beta_r r_i \tag{4.18}$$

Substituting equation 4.18 into the multinomial logit model for service demand estimation (equation 4.5), the ridesplitting passenger demand becomes an implicit function of itself.

$$Q_i = \frac{D_i e^{V_i(\boldsymbol{Q},N)}}{e^{V_i(\boldsymbol{Q},N)} + \mu_i} \tag{4.19}$$

Consequently, under certain operation strategies (i.e., given the value of vehicle fleet size N and trip fare r_i for all OD pairs), an equilibrium in a ridesplitting market is a set of values for $\tilde{t}_i, w_i$ and Q_i that satisfies equation 4.13, 4.14, 4.15, 4.18, and 4.19 for all i in $\mathbb{Z}$. It is worth pointing out that, equation 4.13 and the set of equation 4.19 given different i describes the supply of and demand for ridesplitting services, respectively. In other words, the interplay between system endogenous variables (expected detour time, expected waiting time and ride-sharing demand) at equilibrium given the values of exogenous variables (vehicle fleet size and trip fare) is described by a simultaneous equations system written as below.

$$\begin{cases} Q_{i_1} = \frac{D_{i_1} e^{V_{i_1}(\boldsymbol{Q},N)}}{e^{V_{i_1}(\boldsymbol{Q},N)} + \mu_{i_1}} \\ Q_{i_2} = \frac{D_{i_2} e^{V_{i_2}(\boldsymbol{Q},N)}}{e^{V_{i_2}(\boldsymbol{Q},N)} + \mu_{i_2}} \\ \vdots \qquad\qquad \vdots \\ Q_{i_m} = \frac{D_{i_m} e^{V_{i_m}(\boldsymbol{Q},N)}}{e^{V_{i_m}(\boldsymbol{Q},N)} + \mu_{i_m}} \\ \vdots \qquad\qquad \vdots \end{cases} \tag{4.20}$$

This equations system can be solved via a hybrid method for nonlinear equations proposed in Powell (1970). The numerical experiments indicate that the resultant solutions are always unique under rational operation strategies.

Calibration method for market equilibrium

The accuracy in modeling expected detour $\tilde{t}_i$ and waiting time w_i can significantly affect the efficiency of the proposed ridesplitting market equilibrium model. Therefore, it is critical to provide plausible values for ME hyper-parameters A and B that define their relation with service supply Nn_s and demand Q. Since the regularity of ridesplitting service operations can get affected by the change in network geometry, traffic conditions, and demand levels, the values of A and B should also variate among different markets (or conditions within a market) to capture the distinct market traits and thus provide a reliable market model for the relevant analysis.

The ME hyperparameters can be calibrated using real-world service operational data for ridesplitting. Figure 4.2 demonstrates the calibration idea in which the initial guess A_0 and B_0 are calibrated for a set of fleet sizes Nn_s at different time-of-day settings. The ME is solved for each fleet size and time-of-day setting, resulting in the attracted demand Q and ME waiting and detour times. Overall, the combined set of expected detour $\tilde{t}_i$ and waiting times w_i is compared with the observed set of values from real data to optimize the ME hyperparameters A and B until a certain convergence criterion is met. Note that since in most cases the real world service data is not available, especially for planning case studies where ME models are better suited, simulation-based modeling of ridesplitting (see chapter 3) can be an alternative to arbitrary generate such datasets. The data generated from simulations should be able to approximate the effects of network geometry, traffic conditions, and different demand levels (time-of-day) (considering the models are well calibrated, see chapter 5) .

Although a single set of A and B coefficients can be estimated using the procedure (from figure 4.2), however, they would only represent averaged market conditions captured in the data.

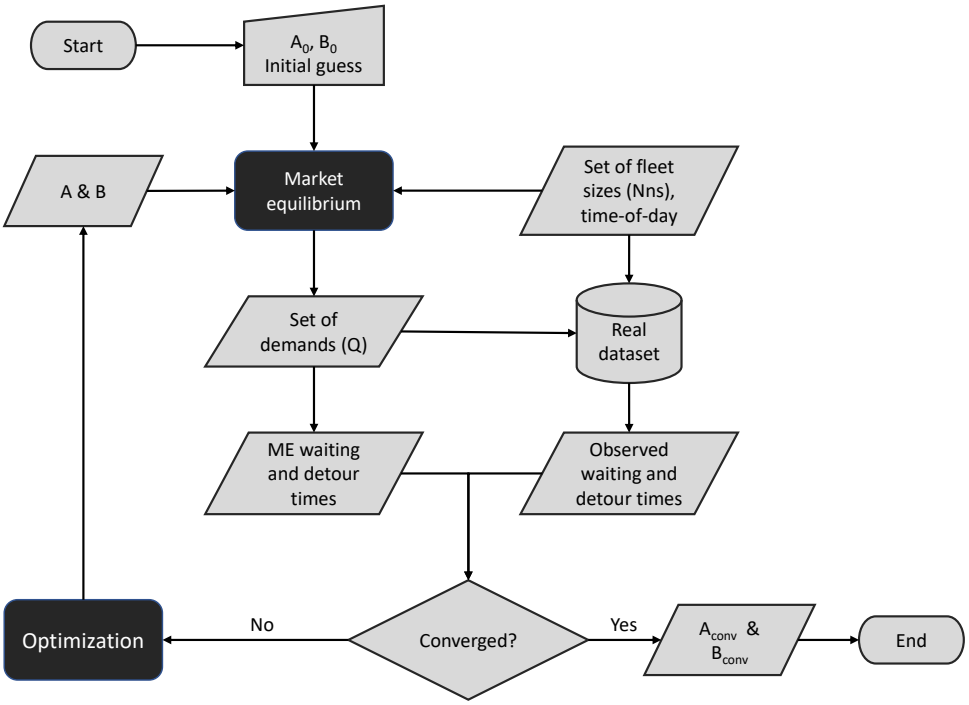


Figure 4.2: Calibration process of market equilibrium (ME) hyper-parameters

The relationship of expected detour $\tilde{t}_i$ and waiting time w_i with service supply Nn_s and demand Q can especially variate by time-of-day (i.e., morning, evening peak hours, off-peak hours), where different OD demand and traffic congestion patterns should uniquely affect the service regularity. Likewise, it could also be calibrated for large differences in service supply (fleet sizes). The most detailed and dynamic capturing of network demand patterns could include defining the hyperparameters at OD pair level, i.e., the A and B coefficients becomes vectors as $\boldsymbol{A} = [A_{i_1}, A_{i_2}, \cdots, A_{i_m}, \cdots]^T$ and $\boldsymbol{B} = [B_{i_1}, B_{i_2}, \cdots, B_{i_m}, \cdots]^T$ for all OD-pairs containing each OD pair specific relation with service supply and demand.

4.5 Utility-based dynamic pricing

Ridesplitting serve riders with different origin-destination and departure times through dynamic matching and detouring. Riders experience detours and different waiting times to share their rides and therefore experience both uncertainty and inequity in their trip level of service (LoS). It significantly affects how riders perceive the service and its adaption. LoS of a trip is generally represented by its corresponding utility (Wang et al., 2018), which is a linear combination of trip fare, waiting time, and travel time (equation 4.4), i.e., the main factors affecting the riders' perception. Therefore, compensation methods (as one described in equation 4.10) which compensate on trip fares based on the time flexibility utilized from a passenger trip allow to add both equity and certainty among trips with varying experienced LoS. This section proposes a utility-based dynamic pricing method to compensate riders and reduce the standard deviation of

trips' utilities. The compensation approach compensates trips with a utility less than a threshold based on a predefined function. In the remainder of this section, we define the compensation principle with more detail alongside discussing its effects on mode choice and the opportunity to do efficient mode choice and demand management.

Compensation principle

To define the compensation principle, we need to specify the threshold value (termed a compensation reference point, CRP) and the method to calculate the amount of compensation (termed as compensation function). Trips below CRP will be compensated with an amount of money determined by the compensation function. Moreover, in order to connect CRP with actual utilities of trips, we define CRP as a proportion of the mean of trips utilities, which can be written as equation 4.21.

$$a = \alpha \bar{V} \tag{4.21}$$

where α is named as compensation reference factor (CRF).

In reference to equation 4.4, utility of an individual trip i is given as:

$$V_i = \beta_t t_i + \beta_w w_i + \beta_r r_i \tag{4.22}$$

while, trip fare of an individual trip i (in reference to equation 4.9) is given as:

$$r_i = p d_i + c_i \tag{4.23}$$

The compensation method redefines the compensation component c_i of the trip fare per individual trip utility. The new compensation is given as:

$$c_i(V_i) = \frac{V_i^a(V_i) - V_i}{\beta_r} \tag{4.24}$$

where β_r is the monetary preference coefficient, V_i is the trip utility before compensation, and $V_i^a(V_i)$ is the compensated utility function for calculating the utility after compensation, which is a function of V_i. The compensated utility function describes the relationship between the utilities of trips before and after compensation. It is the base to calculate compensation for every trip. The compensated utility function should satisfy the following conditions.

1. The utility after compensation should not be larger than the compensation reference point, i.e., $V_i^a \leq a$.
2. The order of trips sorted by utility should not change after compensation, i.e., if $V_{i_1} \leq V_{i_2}$, then $V_{i_1}^a \leq V_{i_2}^a$.
3. Trips with a utility farther below the compensation reference point should get more compensation than those closer, i.e., if $V_{i_1} \leq V_{i_2} \leq a$, then $c_{i_1} \geq c_{i_2} \geq 0$.

Effect on mode choice

Since with dynamic compensations the value of average trip fare r can variate, it changes the expected trip utility and the attracted demand matrix Q. Equation 4.25 defines the new aggregated trip fare $\check{r}$:

$$\check{r} = \frac{1}{D} \sum r_i \tag{4.25}$$

while the new expected trip utility $\check{V}$ is defined as:

$$\check{V} = \beta_t t + \beta_w w + \beta_r \check{r} \tag{4.26}$$

Finally, the new ridesplitting demand is then calculated as:

$$\check{Q} = D \frac{e^{\check{V}}}{e^{\check{V}} + \mu} \tag{4.27}$$

Note that the static compensation method (equation 4.10) compensate by passenger preference times regardless of the actual trip utility. Therefore, it can not solve the issue of inequity among riders. Whereas the dynamic compensations can directly improve the equity among experienced rider utilities. Moreover, for trip-based demand method, the utility set using the CRP threshold can be directly used to more accurately define the expected utility of the service which by dynamic compensations is kept true.

Demand management

Utility-based compensation pricing is an efficient dynamic pricing method to attain better certainty and equity for passenger trips. Moreover, it also provides the opportunity to do efficient demand management. Note that the compensations directly change the average trip fares $\check{r}$ and trip utility of using ridesplitting, which subsequently affects the overall attracted demand Q. Therefore, the CRP threshold a that defines the trip utility threshold for all served trips can also dynamically manage attracted demand. Although, the idea of directly operating the unified price per distance p coefficient is much simpler (which will also control the trip fare and indirectly attracted demand). The CRP (compensation) coefficient should be much more efficient as it will effectively improve the perception of ridesplitting by removing worse trip utility experiences. It also adds more certainty to the expected trip utility (reducing the trip utility variance) and keeps the similar average trip fare effect on demand attraction. Dynamic CRP coefficients can be strategized temporally based on historical service usage data. Note that there is more potential to exploit better the concept of ridesplitting for demand management with other similar dynamic pricing strategies, e.g., dynamic spatial pricing, which could exploit network travel demand patterns and subsidize OD pairs with higher trips improving service ride-sharing with minimum detours.

4.6 Conclusion

This chapter focuses on exploring and developing methods specific to modeling demand for AMoD ridesplitting. AMoD ridesplitting, similar to other on-demand (taxi) services, requires an iterative supply-demand equilibrium process to estimate the passenger demand (equating the expected and experienced trip utilities) given the service fleet attributes and network conditions. Such an equilibrium process is infeasible for microscopic models due to the involved modeling stochasticity, computational times, and traffic dynamics, especially considering any change in input requires reestimating the equilibrium. Moreover, ridesplitting differentiates in operations and the sense it creates for users due to the dynamic ride matching and detouring for sharing rides. Therefore, it requires estimation of passenger preference models, which consider it a separate transport mode to understand better user perception and adaptability for the service, which still lacks in the existing literature. Another problem specific to ridesplitting exists in the user perception towards experienced trip utilities of the service. Due to dynamic detouring, there

exists uncertainty for eventual trip utility and inequity among all passenger trips. This research contributes to developing suitable modeling methods that can solve these limitations.

The first contribution of this research is to propose a simplified trip-based demand modeling method that removes the requirement of attaining equilibrium iteratively with any change in model input. The proposed method uses a simplification assumption of considering the rider trip time preferences as the expected service attributes, that is only valid due to the concept of AMoD ridesplitting and integration of the dynamic DARP algorithm that considers the rider time preferences as explicit/hard time constraints. Furthermore, since trip-based demand modeling is widely adopted among traffic simulators, the proposed method allows more effortless adaptability of the proposed AMoD modeling methods. The second main contribution of this research is to conduct a stated preference experiment and develop discrete choice models to explore the factors that affect the use of ridesplitting and understand users' willingness to pay. The evaluated preferences are crucial inputs for estimating AMoD demand and later establishing demand management and pricing strategies.

Furthermore, this research also develops a utility-based dynamic pricing method to add certainty and equity among individual passenger trips. The method compensates for the trip fares of riders who experience worse trip utilities than a certain threshold. These compensations reduce the standard deviation of trips' utilities, adding certainty in users' perception towards using ridesplitting. Note that the compensation threshold can also be used to accurately define the minimum expected utility in the demand estimation method. Finally, this research also extends the concept of market equilibrium (ME) for ridesplitting. ME models allow the supply-demand interaction analytically using a system of simultaneous equations. We extend the ME method to represent the service supply and trip utility attributes (expected detour and waiting times) specific to ridesplitting. Moreover, a ME calibration method is also defined to tune the model for specific market characteristics, e.g., service definitions, network features, or traffic effects.

5 Dynamic demand estimation

Contents

The recent literature on dynamic demand estimation focuses on improving the estimation scalability towards large-scale networks. For this, Principal Component Analysis (PCA) establishes itself as the new state-of-the-art to tackle the dimensionality and non–linearity issues. However, its application requires the presence of historical estimates that are seldom available due to the limited applicability of conventional methods. This chapter focuses on solving such practical implementation problems for PCA–based calibration techniques and proposes a data–assimilation framework to generate OD historical data–sets. Further, it also proposes a simplified problem formulation by exploiting PCA application properties and explores PCA's implementation properties and its combination with Simultaneous Perturbation Stochastic Approximation (SPSA) to assess the toll and benefits of using PCA and establish practical application guidelines.

The content of this chapter has been presented in the following works:

Qurashi, M., Lu, Q., Cantelmo, G., Antoniou, C., 2022. Dynamic demand estimation on large scale networks using Principal Component Analysis: the case of non-existent or irrelevant historical estimates, *accepted for publication in Transportation Research Part C: Emerging Technologies*

5.1 Introduction

5.1.1 Background and context

Dynamic Origin-Destination (OD) Estimation (DODE) is a problem that searches for a set of time-dependent OD demand matrices which are able to best fit the measured traffic data (Cascetta and Postorino, 2001). In general, mobility demand is unobservable and any network related observation only measures its effect on the network, hence it is mostly generated using demand generation models (McNally, 2007). These models only generate an average demand estimate which can substantially differ for daily demand patterns due to partially predictable phenomena such as weather conditions (Balakrishna, 2006). DODE problem aims to correct these deviations using traffic measurements, such as loop detectors, to update the existing (a-priori) OD matrix. The complexity of calibrating a Dynamic OD Estimation (DODE) problem depends on the amount of disaggregation (time intervals for estimation/prediction or ODs), network size and complexity, and available data. Depending on the specific DTA application, several formulation frameworks have been proposed in the literature to solve the DODE problem (see chapter 2). A first distinction is between *offline* and *online* models (Antoniou et al., 2009), where the former focus on medium-long term planning, while the latter are frequently adopted for real-time applications, such as route guidance. Similarly, we can divide existing models into assignment–matrix based and assignment–matrix free algorithms (Cantelmo et al., 2014b), where assignment–matrix based algorithms explicitly use an analytical representation of the relationship between demand and traffic flows (Cascetta and Postorino, 2001; Toledo and Kolechkina, 2012). However, this relationship is usually assumed to be linear. As this is not the case in reality, other authors proposed assignment–matrix free algorithms, using the DTA model to indirectly capture this correlation (Balakrishna et al., 2007b; Vaze et al., 2009).

Recent years have witnessed a shift towards assignment matrix–free methods. They solve two of the main issues common to all DODE formulations. First, they allow to accurately model the relationship between supply and demand. Second, they allow to incorporate any data source and do not require defining an analytical relationship between data and observations (e.g., between Bluetooth data and mobility demand). The possibility to include additional data is in fact crucial since the DODE is highly under–determined problem. One such approach, named 'Simultaneous Perturbation Stochastic Approximation' (SPSA) (Spall, 1998a), has been one of the most popular algorithms for DTA model calibration (Qurashi et al., 2022). SPSA, due to its ability to deal with non–linear and stochastic systems, a generalized problem formulation, and ease of implementation, has been used frequently by many researchers (Balakrishna et al., 2005; Cantelmo et al., 2014a; Barceló et al., 2010; Ros-Roca et al., 2021). However, DTA models are highly non-linear and the complexity of the DODE problem rapidly increases with the size of the transport network. Hence, conventional algorithms, including the SPSA, often fail in convergence with large–scale problems, because their performance deteriorates rapidly with the increase of the problem scale and complexity. For example, SPSA's gradient approximation gets highly sensitive against: 1) definition of hyper–parameters (objective function gets more expensive, making trail–based setup infeasible); 2) more varying OD magnitudes, which increase exponentially with DTA model size and are also sparsely correlated with traffic measurements.

Most of the literature, which aims to improve the application scalability of DTA model calibration, has followed two major domains i.e., reducing problem dimensions or reducing problem non–linearity. Within the dimension reduction domain, approaches tend to reduce the number of estimation variables by e.g., using a statistical technique i.e. Principal Component Analysis (PCA) (Djukic et al., 2012; Prakash et al., 2018; Qurashi et al., 2019), using a correlation assumption i.e., quasi dynamic (Cascetta et al., 2013; Cantelmo et al., 2014b), clustering the model parameters (Tympakianaki et al., 2015), redefining the problem formulation i.e., utility–based

formulations (Cantelmo et al., 2018, 2020). While, in the other domain of catering problem non–linearity, approaches tend to add additional structural/correlation information spatially or temporally among model parameters and traffic measurements e.g., adding a weight matrix for correlation between ODs and network (Cantelmo et al., 2014a; Lu et al., 2015; Antoniou et al., 2015), using response surface methods or (physical) metamodels which approximate the DTA simulation's input/output relationship using differentiable analytical functions (Zhang et al., 2017; Osorio, 2019b). Among all such efforts, the application of PCA stood out for being significantly more efficient in reducing both problem dimensions and non–linearity. It transforms the OD vector into a lower dimensional space (from the scale of 10^3 to 10^1) which is defined by orthogonal/uncorrelated PCs extracted from the variance of historical OD estimates. Given the strong properties of PCA application on DODE, it has been widely adopted for both offline and online calibration problems to do dimension reduction. For DTA model calibration, it is first proposed by Djukic et al. (2012), followed by many other approaches e.g., PC–GLS (Prakash et al., 2017), PC-EKF (Prakash et al., 2018), and PC–SPSA Qurashi et al. (2019). In all these PCA-based OD estimation frameworks, given a series of historical estimates, PCA leverages strong patterns and correlations to represent the problem with a few orthogonal/uncorrelated Principal Components (PCs) in a low dimensional space. PC–based methods, although being powerful and intuitive, strongly rely on the presence and quality of the historical estimates, by which they extrapolate estimation patterns. PCA provides a considerable advantage through dimension reduction, providing a lower dimensional search space based on PCs evaluated from historical data–set. Hence, application and performance of PCA–based methods is limited by the presence and quality/relevance of the historical data–set relative to the target solution. This in general is not possible for large–scale DTA models, for which such PCA–based methods are proposed, because conventional calibration techniques struggle to calibrate them and PCA application requires historical estimates. Apart from the availability limitation of historical estimates for PCA, less focus has been given in literature to explore the application properties of PCA for calibration large–scale DODE, e.g., identifying its robustness against dimensionality, non–linearity, or estimation algorithm (like SPSA) setup.

5.1.2 Research contributions

Properties of Principal Component Analysis

1. **Generation of historical OD estimates:** This study defines a data-assimilation framework for both generating historical estimates data–set and controlling their quality. As mentioned before, the application and performance of any PCA–based method is limited by the presence and quality of OD historical estimates.The data–assimilation framework proposed in this study explores all possible correlation in the existing demand matrix and generates a set of (artificial) historical estimates from a given historical OD matrix. In addition, this method also provides the possibility to derive these correlations from different available data sources which can help further reduce the residual errors.

2. **Simplified problem formulation:** This study proposes a simplified problem formulation for DODE. Since OD demand is unobservable and the DODE problem is under–determined, literature always include an error term between calibrated and initial OD estimate in DODE objective function. Although it constraints the OD solution near the initial estimate and reduce the over fitting on traffic data, it also limits the calibration performance due to added noise and complexity. In this research, we show that the application of PCA does not require such demand constraint, as it allows to include information about

the historical matrix directly into the objective function. Hence, we propose to simplify the DODE problem formulation using only the error term between traffic measurements.

Implementation properties of PC–SPSA

1. **Ease of hyper–parameters tuning:** This study performs sensitivity analyses for robustness of PC–SPSA against SPSA hyper–parameters. There is no set rule to define hyper–parameters generically for SPSA and its variants i.e., previous studies often do sensitivity analysis to identify case–specific hyper–parameters (Cantelmo et al., 2014a). Even other large scale calibration approaches require regress effort to set up specific to a DTA model (e.g., defining physical metamodel functions (Zhang et al., 2017; Osorio, 2019b), creating correlation weight matrices (Antoniou et al., 2015)). In this research, we show that PC-SPSA is significantly robust in converging on high quality solutions to a range of different hyper–parameter values. This advantage of skipping problem-specific manual input, especially with large–scale DTA models, is the reduction of additional computational effort for running simulations repeatedly during this trial/definition phase.
2. **Value of added structural information:** Literature approaches use different techniques to add information within the objective function for improving their application scalability (Antoniou et al., 2015; Cantelmo et al., 2014a; Tympakianaki et al., 2018; Osorio, 2019b). Similarly, PCA also incorporates OD structural patterns from the historical estimates to reduce non–linearity and computational requirements. In this study, we perform multiple sensitivity analyses to measure the impact of varying historical data–set characteristics (i.e., size, variance, and number of PCs) on PC–SPSA calibration performance. The analysis helps to understand the value of structural information added in the objective function and provides directions to control model over fitting.
3. **Computational efficiency:** Most calibration methods struggle due to high computational efforts for simulation run–times, large set of estimation variables, and iterative nature. Methods proposed in this study help address this practical challenge, and calibrate one of the largest calibration experiment for DODE to date i.e., the Munich network. First, the results show the direct benefits of PCA i.e., the increase in dimensionality and non–linearity/complexity for Munich network doesn't directly translate into an equivalent increase in optimization complexity and estimation variables. Moreover, exploiting PCA properties, the ease of SPSA hyper–parameters tuning eliminate the need of recursive simulations for trail–based setting. Similarly, we also eliminate the requirement of using multiple gradient replications in SPSA (otherwise used in all SPSA methods to remove gradient biased). Also, the simplified problem formulation provides significant improvements for the required number of iterations. Hence overall, the calibration runtime is significantly reduced for the Munich case study (2-6 iterations with practically almost 1 simulation run–time) making PC–SPSA even feasible for online calibration.

5.1.3 Outline

The rest of the chapter is structured as follows. Section 5.2 describes the overall methodology followed in this research. After introducing PCA in the OD estimation context, we discuss the proposed data-assimilation framework for historical data matrix generation, the simplified problem formulation, and our implementation of PC–SPSA. Then, section 5.3 describes the experimental setup, network case study, and the calibration results for PC–SPSA. It also includes the comparisons for different historical OD generation methods and conventional versus simplified problem formulations. Later, section 5.4 covers the sensitivity analyses performed on PCA and

PC–SPSA implementation properties alongside the guidelines for their setup. Finally, section 5.5 concludes with describing the overall contributions and findings of the research alongside its future implications and possible research directions.

5.2 Methodology

5.2.1 DODE problem formulation

The DTA calibration problem is generally formulated as an optimization problem, minimizing the specified objective function by optimizing the model parameter values with the given constraints (to decide a feasible parameter space). A generic problem formulation for DTA model calibration is given as:

$$\underset{\boldsymbol{\beta},\boldsymbol{x}}{Minimise}\ z(\boldsymbol{y^{obs}}, \boldsymbol{y^{sim}}, \boldsymbol{x}, \boldsymbol{x^{p}}, \boldsymbol{\beta}, \boldsymbol{\beta^{p}}) \tag{5.1}$$

Where $\boldsymbol{y^{sim}}/\boldsymbol{y^{obs}}$ represent the simulated/observed traffic measurements, $\boldsymbol{x}$ and $\boldsymbol{\beta}$ indicate the current values for the origin–destination demand flows and for the behavioural parameters, respectively, while $\boldsymbol{x^{p}}$ and $\boldsymbol{\beta^{p}}$ are their historical (or prior) estimates. In this study, we solve the traditional DODE problem, which focuses on only estimating time–dependent OD flows $\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_h}$, while other model parameters $\boldsymbol{\beta}$ are kept constant. The objective function formulation for time-dependent DODE problem is reformulated as:

$$\underset{\boldsymbol{x}}{Minimise}\ \sum_{H}^{h=1}[w_y z_1(\boldsymbol{y_h^{obs}}, \boldsymbol{y_h^{sim}}) + w_{od} z_2(\boldsymbol{x_h}, \boldsymbol{x_h^{p}})] \tag{5.2}$$

subject to:

$$\boldsymbol{y_h^{sim}} = f(\boldsymbol{x_1}, ...\boldsymbol{x_h}; \boldsymbol{\beta}; \boldsymbol{G_1}, ..., \boldsymbol{G_h})$$
$$\boldsymbol{l_x} \leq \boldsymbol{x} \leq \boldsymbol{u_x}$$

where the calibration time period is defined in intervals $\mathcal{H} = \{1, 2, ..., H\}$ and:

$\boldsymbol{y_h^{obs}}/\boldsymbol{y_h^{sim}}$: Observed and simulation time–dependent traffic measurements
$\boldsymbol{x_h}/\boldsymbol{x_h^{p}}$: Current and prioir values for time–dependent demand parameters i.e., OD flows
$\boldsymbol{\beta}$: Other fixed DTA model parameters
$\boldsymbol{G_h}$: Road network and other supply parameters

The minimization of the DODE objective function (equation 5.2) heavily relies on z_1, which measures the goodness of fit between observed and simulated traffic measurements, while z_2 (i.e., the goodness of fit between estimated and prior OD demand) help to restrain the estimated solution closer to the prior/starting OD. The weight factors w_1 and w_2 are used to scale the reliance (or reflect uncertainty) on both observed traffic measurements $\boldsymbol{y_h}$ and prior OD flows $\boldsymbol{x_h^{p}}$ information. The simulated traffic data $\boldsymbol{y_h^{sim}}$ detected in time interval h are explicitly modelled through a (non-linear) function $f(\cdot)$ (i.e., DTA simulator) of all OD flows $\boldsymbol{x}$, model parameters $\boldsymbol{\beta}$ and the road network/supply parameters till time interval h. Using this optimization–based problem formulation with any non-assignment based approach provides an advantage of including any available traffic data y_h for estimation (requiring $f(\cdot)$ to be a DTA simulator).

5.2.2 PCA application for DODE

Principal Component Analysis (PCA) is already a standard for problem dimension reduction. Principal Component Analysis (PCA) is already a standard for problem dimension reduction. It allows to dimensionally reduce a large set of decision variables $\boldsymbol{\theta}$ or $\boldsymbol{x}$ (i.e., the starting OD vector for DODE) into few number of PC–scores using a lower dimensional space. This space is defined by a set of Principal Components (PCs) estimated by the application of PCA on the time series historical data of the decision vector. For OD estimation, Djukic et al. (2012) is the first to apply PCA on the time series OD matrices, extracting the spatial–temporal correlation among different OD pairs. Although the idea of PCA's application is of dimension reduction, it also gives other favorable properties. For example, it gives an orthogonal/uncorrelated OD demand representation which otherwise is sparsely correlated and it keeps the search space limited in the variance captured from historical estimates resulting in good quality OD solutions. Below, we describe the estimation process of principal components, OD matrix transformation and PCA–based DODE formulation.

Principal components estimation

Principal Components (PCs) are linear vectors combinations containing the variance information of a time series data. All PCs have their subsequent coefficients (named 'PC–directions') which define the amount of variance captured by them. The value of these PC–directions decrease in an ascending order i.e., the first PC captures the highest sample variance in the data followed by the second PC with the second–highest variance captured and soon. The estimation of PCs requires a time series OD demand information which can be supplemented using historical OD estimates (calibrated offline or online). Given the availability of historical estimates, they are set in a data matrix D with dimensions $[n_k \times n_x]$, where n_k is the number of historical data points and n_x is the size of OD vector estimate. Then, Singular Value Decomposition (SVD) is applied on this historical data matrix D to evaluate the PCs, given as:

$$D = U\Sigma V^T \tag{5.3}$$

The unitary matrix V with dimension $[n_x \times n_x]$ contains vectors of the orthogonal PCs and their corresponding PC–directions are stored in the rectangular–diagonal matrix Σ with dimension $[n_k \times n_x]$. U is a $[n_k \times n_k]$ unitary matrix with orthogonal vectors. A time series historical estimates data–set of n_k data points result in n_k PCs (Djukic et al., 2012), hence the first n_k columns of unitary matrix V are PCs and the diagonal n_k values of matrix Σ are their PC-directions. The evaluated PCs can be further reduced to retain only the first few significant PCs n_d, which can explain most of the time series variance from the historical estimates (Djukic et al., 2012), hence V is further reduced to $\hat{V}$:

$$\hat{V} = [\, v_1 \;\; v_2 \;\; v_3 \;\; ... \;\; v_{n_v} \,] \tag{5.4}$$

PCA–based DODE

The starting OD vector $\boldsymbol{x}$ (otherwise used directly for estimation) is transformed into a lower dimensional PCs space. The reduced $\hat{V}$ unitary matrix containing n_v significant PCs is used to transform $\boldsymbol{x}$ into to set of PC scores z of dimension $[n_v \times 1]$, as:

$$z = \hat{V}^T x \tag{5.5}$$

These estimated PC scores z are then estimated instead of the OD flow vector x, the DODE problem formulation (equation 5.2) can be rewritten as:

$$\underset{\boldsymbol{z}}{Minimise} \sum_{H}^{h=1} [w_y z_1(\boldsymbol{y_h^{obs}}, \boldsymbol{y_h^{sim}}) + w_{od} z_2(\boldsymbol{x_h}, \boldsymbol{x_h^p})] \tag{5.6}$$

For objective function evaluations and final solution transformation, the OD vector x can be re–approximated as:

$$x \approx \hat{V} z \tag{5.7}$$

5.2.3 Historical data matrix generation

Historical OD estimates used for estimating PCs are critical for application of PCA–based methods. These historical estimates should be relevant temporally (i.e., day–to–day historical estimates of the same time intervals ($\mathcal{H} = \{1, 2, ...h\}$), to ensure similar OD spatial/structural patterns as of the target solution. This implies that different historical data–sets should be constructed between e.g., morning and evening peak hours, peak and off–peak hours, weekdays and holidays. If relevant estimates are not available then PCA–Based models will give poor quality solutions. Setting the relevance property aside, the existence/availability of historical OD estimates is even more critical (especially for large scale DTA models). It is evident from the literature that conventional models, such as SPSA, are in fact not capable of being used to calibrate large–scale networks and therefore the presence of calibrated/estimated historical OD data–set is impractical, hence limiting the use of PCA–based techniques in practice.

In this section, we propose a data-assimilation framework for applicability of PCA–methods in scenarios of irrelevant or non–existing historical estimates. In such scenarios, there exists a possibility to synthetically generate historical OD estimates using the available OD estimate. As discussed previously, PCA limits the search space by projecting each OD pair into a few principal components capable of explaining their variance. Traditionally, principal components are obtained from time series of data - i.e. the historical estimates. The data assimilation framework allows to incorporate historical information from one single historical demand matrix into the principal components of the problem. This means that, while previous approaches rely on historical estimates, in this case the Principal Components represent the historical (seed) matrix, which can be easily obtained with any demand model, from the gravity model to Synthetic Population. Given a single demand matrix x, we perturb the demand and artificially generate variations within the data. Different types of demand fluctuations are considered, such as spatial, temporal, and day–to–day variations. This allows us to use PCA–based algorithms, without even the need to first obtain the historical estimates, which is often infeasible in practice. Additionally, by artificially perturbing the demand in three different dimensions, the proposed approach allows to have control over the search space definition (e.g., define a narrow search space if small variations are assumed and hence good OD quality is retained in reference to the initial OD estimate; or a broader search space with more variance is considered in case the model error does not converge to a good solution).

Correlations among time–dependent OD flows

Dynamic OD demand is mostly represented as time-dependent OD flows ($\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_h}$), which are individual sets of OD matrices $\boldsymbol{x}$ each representing a single time interval h. Demand fluctuations among such time–dependent OD flows can correlate in three possible dimensions. Figure 5.1 presents the conceptual directions for each of these three correlation dimension in a OD demand time series plot, where each vertical vector represents a single time–dependent OD for a given time interval. Further, we describe these dimensions as:

- **Spatial correlation:** The spatial correlation presents the spatial structure of the OD demand over the network, i.e., how all the OD pairs $x_{n_{ij}}$ are spatially correlated among themselves. This correlation dimension should help in capturing the demand fluctuations triggered spatially e.g., the changes in trip distribution among different OD pairs. The source of these fluctuations can variate from long–term changes of land-use to short–term changes in trip attractions and distributions among OD pairs due to consistently varying network travel times or traffic congestion patterns.
- **Temporal correlation:** The temporal correlation presents the times series evolution of demand, i.e., the time–dependent fluctuations of each OD pair $x_{n_{ij}}$ between all time intervals t (or previously said $\mathcal{H} = 1, 2, ...h$). This correlation dimensions helps in capturing the demand fluctuations or distributions for departure time choice of the overall demand for each OD pair. Individual departure time choice decisions depend on factors such as trip purpose/activity, network state/congestion and person demographics.
- **Day–to–day correlation:** Mobility demand is correlated to the demand for activities. As such, it follows a structure and day–to–day variations are likely to occur. Hence, day–to–day correlations presents the correlation of each OD pair $x_{n_{ij}}$ among different days d. This correlation dimension should capture the day–to–day demand fluctuations for individual OD pairs due to change in their trip generation/attractions for different trip activities which are influenced by e.g., day–of–the-week, weather conditions, seasons, special events like sales, festivals, sport events etc.

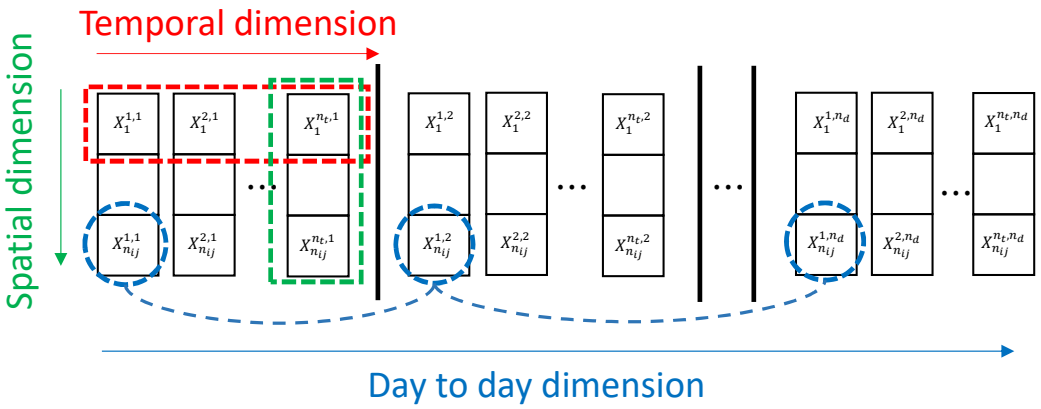


Figure 5.1: Different correlation dimensions among time–dependent OD flows

Historical data–set generation methods

After developing our understanding on the above mentioned correlation dimensions for time–dependent ODs, we consider that the demand fluctuations within the historical OD estimates should naturally follow these correlations. Hence, synthetic historical data–sets can be generated by perturbing the starting OD vector $\boldsymbol{x}$ among them. Since, these three correlation dimensions cover the possible user behaviors, we propose six different historical OD generation methods exploiting them. Intuitively, more correlations should lead to a more realistic representation

of the behaviour. However, this will also requires a larger time series, which also means more principal components and therefore more variables to be calibrated. To mathematically express the proposed methods, we first define the utilized notations in table 5.1, followed by the definitions of all methods.

Table 5.1: List of Symbols

$\boldsymbol{D}$	Historical data matrix with dimensions $[n_{ij} \times (n_t n_d)]$
$\boldsymbol{\Delta_T}$	Perturbation matrix for correlation of type T
$\boldsymbol{x}$	Current/prior OD estimate matrix with dimensions $[n_{ij} \times n_t]$
$\boldsymbol{X}$	Augmented matrix of multiple x sets with dimensions $[n_{ij} \times (n_t n_d)]$
$\boldsymbol{\mathcal{N}_{od}}$, $\boldsymbol{\mathcal{N}_t}$,$\boldsymbol{\mathcal{N}_d}$	Gaussian distributions of size n_{ij}, n_t and n_d, mean μ and standard deviation σ
R_{od}, R_t	Perturbation/weight coefficient for sizing the effect of spatial and temporal correlation variance
R_{min}	The smaller value within R_{od} and R_t
n_{ij}, n_t, n_d	Number of OD pairs, time intervals and historical days

- **Method 1: Spatial correlation**
  This method considers the spatial correlation to generate the historical OD data–set D. The perturbation matrix $\boldsymbol{\Delta_{od}}$ is generated using $\boldsymbol{\mathcal{N}_{od}}$ Gaussian distribution. The mathematical expression is given as:

$$\boldsymbol{D} = (\mathbf{1} + R_{od}\boldsymbol{\Delta}_{od}) \odot \boldsymbol{X} \tag{5.8}$$

  where $\boldsymbol{X}$ is an augmented matrix given by:

$$\boldsymbol{X} = \underbrace{(x|x|\ldots|x)}_{n_d} \tag{5.9}$$

  while x is the initial OD estimate matrix of n_{ij} OD pairs and n_t time intervals. The $\odot$ operation achieves the Hadamard (element-wise) product to perturb the augmented matrix X. Note that, R_{od} is the perturbation factor for sizing the effect of perturbation matrix $\boldsymbol{\Delta_{od}}$.

- **Method 2: Temporal correlation**
  This method considers the temporal correlation to generate the historical data–set D. The perturbation matrix $\boldsymbol{\Delta_t}$ is generated using $\boldsymbol{\mathcal{N}_t}$ Gaussian distribution. The mathematical expression is given as:

$$\boldsymbol{D} = (\mathbf{1} + R_t\boldsymbol{\Delta}_t) \odot \boldsymbol{X} \tag{5.10}$$

  where R_t is the perturbation factor for sizing the effect of perturbation matrix $\boldsymbol{\Delta_t}$.

- **Method 3: Spatial and temporal correlation**
  This method considers both spatial and temporal correlations to generate the historical data–set D. The perturbation matrix $\boldsymbol{\Delta_{od,t}}$ is generated using the Gaussian distributions

$\boldsymbol{\mathcal{N}_{od}}$ and $\boldsymbol{\mathcal{N}_{t}}$ in spatial and temporal directions (see fig. 5.1). The mathematical expression is given as:

$$\boldsymbol{D} = (\mathbf{1} + R_{min}\boldsymbol{\Delta}_{od,t}) \odot \boldsymbol{X} \tag{5.11}$$

where R_{min} is the lowest of the perturbation factors R_{od} and R_t for sizing the effect of perturbation matrix $\boldsymbol{\Delta_{od,t}}$.

- **Method 4: Spatial and day–to–day correlation**
  This method considers both spatial and day–to–day correlations to generate the historical data–set D. The perturbation matrix $\boldsymbol{\Delta_{od,d}}$ is generated using the Gaussian distributions $\boldsymbol{\mathcal{N}_{od}}$ and $\boldsymbol{\mathcal{N}_{d}}$ in spatial and day–to–day directions (see fig. 5.1). The mathematical expression is given as:

$$\boldsymbol{D} = (\mathbf{1} + R_{od}\boldsymbol{\Delta}_{od,d}) \odot \boldsymbol{X} \tag{5.12}$$

- **Method 5: Temporal and day–to–day correlation**
  This method considers both temporal and day–to–day correlations to generate the historical data–set D. The perturbation matrix $\boldsymbol{\Delta_{t,d}}$ is generated using the Gaussian distributions $\boldsymbol{\mathcal{N}_{t}}$ and $\boldsymbol{\mathcal{N}_{d}}$ in temporal and day–to–day directions (see fig. 5.1). The mathematical expression is given as:

$$\boldsymbol{D} = (\mathbf{1} + R_{t}\boldsymbol{\Delta}_{t,d}) \odot \boldsymbol{X} \tag{5.13}$$

- **Method 6: Spatial, temporal and day-to-day correlation**
  This last method considers all possible correlation dimensions possible in time–dependent ODs. To estimate the historical data–set D. The perturbation matrix $\boldsymbol{\Delta_{od,t,d}}$ is generated using the Gaussian distributions $\boldsymbol{\mathcal{N}_{od}}$, $\boldsymbol{\mathcal{N}_{t}}$ and $\boldsymbol{\mathcal{N}_{d}}$ in spatial, temporal and day–to–day directions (see fig. 5.1). The mathematical expression is given as:

$$\boldsymbol{D} = (\mathbf{1} + R_{min}\boldsymbol{\Delta}_{od,t,d}) \odot \boldsymbol{X} \tag{5.14}$$

The six proposed generation methods capture all possible combinations between spatial, within–day temporal and day–to–day temporal correlations. Note that, in current methodology we use Gaussian distributions with zero mean to define the perturbation matrices Δ_T but an additional value of these generation formulations is that these correlation distributions (currently $\boldsymbol{\mathcal{N}_{od}}$, $\boldsymbol{\mathcal{N}_{t}}$, and $\boldsymbol{\mathcal{N}_{d}}$) can be derived by other data sources, such as mobile phone network data and survey data. Finally, this leads to a framework that is more general - as it does not depend on an historical database - and is more flexible - as the structure of the PCs would reflect both OD flows as well as other spatial-temporal dynamics.

5.2.4 Simplification of DODE problem formulation

The tradition DODE problem formulation (equation 5.2) comprises two minimizing error terms i.e., z_1 between traffic measurements (y' and y) and z_2 between prior x_h^p and calibrated OD estimates x_h. Since the DODE problem is considered highly underdetermined (far more estimation variables against traffic measurements), reliance on using z_2 can be seen throughout the literature. The idea for using z_2 is to keep the calibrated OD estimates close to the prior/starting estimate, which is considered being the most reliable available estimate.

For PCA–based models, we propose to simplify the DODE formulation releasing the z_2 error term. This is a generalization of PCA–based models where the use of PCs help us include

historical OD information in the objective function (allowing us to release z_2 and simplify the DODE problem formulation) and also simplify the problem through dimension reduction (as we solve it in PC space). The new simplified problem formulation is given as:

$$\underset{z}{Minimise} \sum_{H}^{h=1} [z_1(\boldsymbol{y_h}, \boldsymbol{y'_h})] \tag{5.15}$$

This simplification is possible only by the use of PCA, where previously in the standard approach presented in equation 5.2 the term z_2 included prior information about the historical demand. This information, however, is already included within the PCA components, where the vector of Principal Components $\hat{V}$ is in fact directly obtained by the time series historical demand, which means that the PCs defined search space is already constrained within the variance present in the historical estimates. This keeps all the patterns of the calibrated OD estimate within those present in historical estimates, which is also the purpose of using the error term z_2. Hence, for all PCA–based methods, the purpose of using the error term z_2 is already fulfilled by PCA's dimension reduction.

5.2.5 Estimation setup

As discussed before in section 5.1, SPSA is arguably the most popular assignment matrix–free method due to its generalized problem formulation and ability to deal with non–linear and stochastic systems. Therefore, to demonstrate the significance of the proposed PCA methods, we choose it as the optimization problem solver to estimate the DODE problem formulated in PC space (Qurashi et al., 2019). Below, we describe the SPSA setup for PCA–based DODE and emphasize on the ease in requirement of defining SPSA hyper–parameters alongside proposing some modifications to exploit the properties of PCA application. Similarly, we also discuss the PCA application setup to understand the role of new hyper–parameters required to define the characteristics of historical data matrix and dimension reduction.

SPSA for PCA–based estimation

SPSA (Spall et al., 1992) is a Stochastic Approximation (SA) algorithm with a unique advantage of approximating a noisy gradient with only two objective function evaluations using simultaneous perturbation. Qurashi et al. (2019) proposed a modified SPSA to solve PCA–based DODE problem. Equation 5.16 shows the modified gradient estimation method to estimate PC–scores z, where Δ is a p-dimensional vector generated randomly from a ± 1 Bernoulli distribution (where P is the length of decision vector $\boldsymbol{z_k}$).

$$\boldsymbol{g'} = \frac{f(\boldsymbol{z_k} + \boldsymbol{z_k} \times c_k \boldsymbol{\Delta_k}) - f(\boldsymbol{z_k} - \boldsymbol{z_k} \times c_k \boldsymbol{\Delta_k})}{2c_k} \left[\Delta_1 \ \Delta_2 \ . \ . \ \Delta_p\right]^T \tag{5.16}$$

The estimated gradient is used to minimize the solution using a modified form of SA approach (equation 5.17).

$$\boldsymbol{z_{k+1}} = \boldsymbol{z_k} - \boldsymbol{z_k} \times a_k \boldsymbol{g'_k}(\boldsymbol{z_k}) \tag{5.17}$$

$$c_k = c/k^{\gamma} \qquad a_k = a/(k+A)^{\alpha} \tag{5.18}$$

Note that, the coefficients of perturbation c_k and minimization a_k evolve over the number of iterations $\mathcal{K} = \{1, 2, 3, ...k\}$ and are evaluated based on the set of pre–defined hyper–parameters

c, a, γ, α, and A (equation 5.18). Apart from the general guidelines proposed by Spall (1998a), their is no set rule to define these hyper–parameters for SPSA or any of its variants. Hence, it requires a trail–based method to find appropriate values which can result in good convergence. When combining PC-SPSA with the data–set generation method proposed in Section 5.2.3, the number of hyper–parameters further increases, as the model requires to define both the number of historical estimates n_d as well as the mean and the variance for the spatio/temporal distributions $\boldsymbol{\mathcal{N}_{od}}, \boldsymbol{\mathcal{N}_t}, \boldsymbol{\mathcal{N}_d}$, which regulate the link between historical demand and PCs. However, the application of PCA drastically reduces the required number of iterations (Qurashi et al., 2019) and the modified SPSA (equation 5.17) applies a percentage change instead of absolute increase/decrease in estimation variables $\boldsymbol{z_k}$, as in the traditional SPSA. Therefore, the sensitivity of the model to changes in the hyper–parameter decreases significantly. as shown in Section 5.4. Additionally, by combining the proposed data–set generation method with the simplified formulation discussed in Section 5.2.4, the number of iterations of PC-SPSA futher decreases making the calibration of the parameters γ, α and A unnecessary, as the model converges for a low value of k. Finally, SPSA requires multiple gradient replications for DODE (Balakrishna et al., 2007b) and almost all SPSA based literature works use it to reduce gradient bias (e.g., Cantelmo et al. (2014a); Tympakianaki et al. (2015)) due to correlations and non–linearity present in DODE variables. We show in this study that this becomes unnecessary with PCA because all PCs are orthogonal and uncorrelated. Hence, we also propose to remove this requirement and all experiments ran in this study use only a single gradient estimate per SPSA iteration. A step–wise PC–SPSA algorithm is given in appendix 1.

PCA application setup

Recalling from section 5.2.2, to transform the OD flows in lower dimensional space, PC–directions V^T are used. These PC–directions are evaluated from the historical data matrix D (see equation 5.3) and represent the variance present in it. Note that, the optimization search space for PCA–based methods is confined within this variance. In other words, it is the additional demand information added to the DODE objective function. Hence, it is important to better understand the impact of this added variance information and control it characteristics accordingly. The variance present in PC–directions can be controlled by certain parameters which define the characteristic of historical data matrix D. These parameters include the number of historical estimates i.e., size n_d of data matrix D (equation 5.9), number of PCs retained n_v (equation 5.4), and control of the variance present in historical estimates (defined by R and σ from equation 5.8-5.14, i.e., in case of using historical generation methods). Note that, both in case of availability or unavailability of historical OD estimates, the variance information can be controlled. But, it also increases the overall set of required hyper–parameters for manual setup.

PC–SPSA algorithm

The step-by-step PC–SPSA algorithm (including the steps of generating historical estimates D) is described below:

Algorithm 1: PC–SPSA

Initialization at iteration 0
Define historical data–set generation parameters: n_d, R_{min}, $\boldsymbol{\mathcal{N}}$ (μ and σ) Generate historical data–set D Estimate PCs: $D = U\Sigma V^T$ Definition SPSA hyper–parameters: c,a A, γ, α OD transformation to PC–scores: $z_0 = \hat{V}^T x_0$
Gain sequence update at iteration k
$c_k = c/k^\gamma$ $a_k = a/(k+A)^\alpha$
Perturbation
$\boldsymbol{z_k^\pm} = \boldsymbol{z_k} \pm \boldsymbol{z_k} \times c_k \Delta$
OD approximation
$\boldsymbol{x_k^\pm} \approx \hat{V} \boldsymbol{z_k^\pm}$
Gradient evaluation
$\boldsymbol{g'_k(x_k)} = \frac{f(\boldsymbol{x_k^+}) - f(\boldsymbol{x_k^-})}{2c_k} \left[\Delta_1 \ \Delta_2 \ . \ . \ \Delta_p\right]^T$
Minimization
$\boldsymbol{z_{k+1}} = \boldsymbol{z_k} - a_k \boldsymbol{g'_k(x_k)}$
OD approximation at convergence iteration $\mathcal{K}$
$\boldsymbol{x_\mathcal{K}} \approx \hat{V} \boldsymbol{z_\mathcal{K}}$

5.3 Case study: Munich city

5.3.1 Experimental setup

Network and simulation setup

We implement the case study on the Munich regional network (about 900 km^2). As shown in Figure 5.2, the network is divided into 73 zones resulting in 5,329 OD pairs, including 10 external zones (green circles) at major radial motorways entering the city. The network consists of a total of 8,761 links (Figure 5.4), excluding residential roads to reduce the route choice burden for the simulation experiment. A total of 507 detector locations are used for the case study. As described previously, this leads to a highly *underdetermined system* (5,329 *unknowns* per interval with only

507 *traffic measurements*) and renders the application difficulties of conventional calibration methods.

An open–source traffic simulator, Simulation of Urban MObility (SUMO, Lopez et al. (2018)), is assembled with the proposed calibration algorithm for experiments. All simulations are implemented at the mesoscopic level via the trip–based stochastic user route choice assignment method. To focus on DODE problem, we fix the route choice and supply side parameters (e.g., jam threshold). Also, to cater for the stochasticity of the traffic simulations we used outputs averaged from 10 simulation replications. Overall, the run–time for a single simulation (for morning peak hours i.e., 6am - 10 am) is 12 minutes and the 10 simulation replications are ran in parallelization. Given the sizes of the network, SPSA cannot be used to calibrate the DTA model. Additionally, historical estimates are not available as the network has never been calibrated before. Therefore, we use the procedure explained in section 5.2.3 combined with the PC–SPSA algorithm to calibrate the network under the simplified problem formulation, described in section 5.2.4.

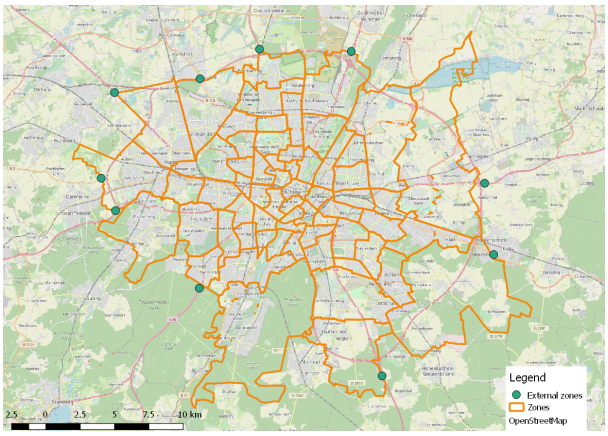


Figure 5.2: Traffic zones of Munich major region.

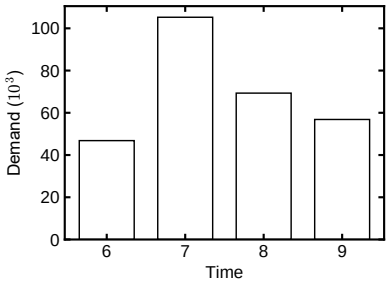


Figure 5.3: Network demand (6 am to 10 am)

Demand scenarios

To explore the effectiveness and efficiency of PC–SPSA on the DTA model calibration problem, we apply PC–SPSA to calibrate the demand from 6 am to 10 am represented in 15–minute intervals, which contains characteristics of very low demand (6 am – 7am), normal off–peak (8 am – 10 am) and peak traffic (7 am – 8 am). To process the procedure, we specify the demand scenario following the benchmarking framework standardized by Antoniou et al. (2016) for testing calibration algorithms. The method has also been used in many recent works on developing calibration algorithms (Qurashi et al., 2019; Cantelmo et al., 2020). To create the scenario, the target/true demand is synthetically perturbed with the latest previous estimate x_{p_1} and its simulated outputs are taken as true outputs. Two coefficients of reduction (*Red*) and randomization (*Rand*) are used for perturbation. Different values of these two coefficients are used to create different types of true demands as in reality. The demand scenario generation is specifically expressed as:

$$x_c = (Red + Rand \times \delta) \times x_{p_1} \tag{5.19}$$

where δ is the random perturbation vector following Gaussian distribution. In this case study, we apply $Red = 0.7$ and $Rand = 0.15$ (i.e., $x_c = (0.7 + 0.15\delta) \times x_{p_1}$), and $\delta \sim N(0, 0.333)$ (99.7%

Figure 5.4: Used Munich network overview

of values located in [-1,1]), resulting in the demand distribution shown in Figure 5.3 (aggregated into one hour for easy illustration).

PC–SPSA algorithm settings

Recall Equation (5.18), we need to update the gains for perturbation (c_k) and minimization (a_k) to control the step size and convergence at each step. In all following experiments, A, α and γ are set to be 25, 0.3 and 0.15, respectively. For the experiments within this section, c and a are set to be 0.15 and 1, respectively. Note that, c and γ control the perturbation percentage of the PC–scores. For example, at the first step, the PC-scores are perturbed with $\pm(15\%)$. On the other hand, a, A and α control the actual moving step in the searching space. All historical data–set generation methods introduced in Section 5.2.3 are applied for comparison, for which R_{od}, R_t, and R_d are set as 0.3, 0.4 and 1, respectively, while the Gaussian distributions $\boldsymbol{\mathcal{N}_{od}}$, $\boldsymbol{\mathcal{N}_t}$, and $\boldsymbol{\mathcal{N}_d}$ are generated using $\sim N(0, 0.333)$ setting. The demand of 100 historical days is thus generated. Furthermore, to reserve enough variance contained in the historical data–set for tracking the patterns and achieve the goal for dimension reduction at the same time, the number of PCs expressing 95% of the total variance are used.

Goodness of fit

Given that PC–SPSA is a non-assignment matrix based algorithm it requires the DTA model simulation to map the OD matrix into measurable traffic measurements, such as vehicle counts recorded by detectors. These generated traffic counts are then compared with the observed traffic counts to evaluate their difference which is used as an indicator for DODE minimization (i.e., z_1 in equation 5.15). In this study, we apply Root Mean Square Normalized error (RMSN) to measure the Gof of the simulated traffic counts and thus evaluate the estimated OD matrix. RSMN is specifically used extensively for DODE problem (Qurashi et al., 2019; Antoniou et al., 2015) because it finds the normalized root mean distance between all counts helpful to estimate closer patterns towards the target solution. The calculation of RMSN is given by:

$$RMSN = \frac{\sqrt{n \sum_{i=1}^{n} (y_i^{sim} - y_i^{obs})^2}}{\sum_{i=1}^{n} y_i^{obs}} \tag{5.20}$$

where y^{obs} and y^{sim} are the observed traffic counts and simulated traffic counts, respectively. n is the number of detectors.

5.3.2 Results

Convergence analysis and calibration quality

Figure 5.5 displays PC–SPSA's convergence results for calibrating 15-minute demand intervals of the peak hour from 7 am to 8 am as shown in fig. 5.3. The results include convergence plots for all six historical OD generation methods described in section 5.2.3. Despite the large study area, PC–SPSA is able to converge to a low RMSN error values within the first few iterations, confirming the improved application scalability of PC–SPSA on large scale DTA models. Figure 5.6 illustrates the quality of model calibration comparing observed and simulated traffic counts at all detector locations using a 45° plot. The results depicted are only for method 6 (figure 5.5). We refer to section 5.3.2 for the discussion on the differences between the six method. Since all points are aligned closer to the 45° line, it is confirmed that the low error convergence is achieved at all detector locations. Figure 5.5 also shows that, while all generation methods perform fairly well, some of the proposed methods obtain drastic improvements in only one or two iterations.

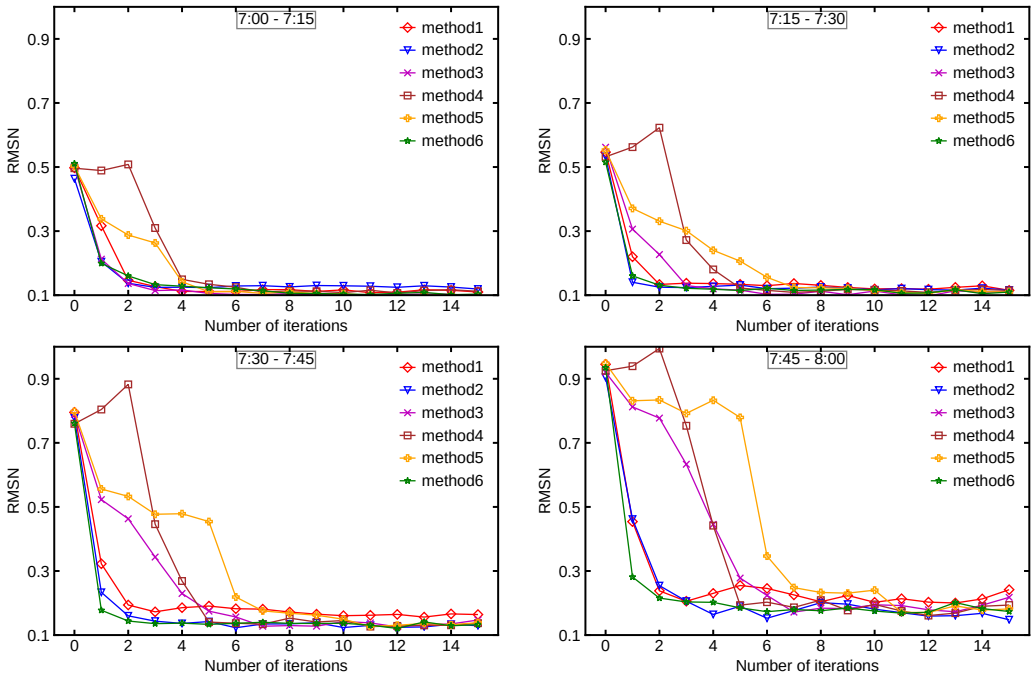
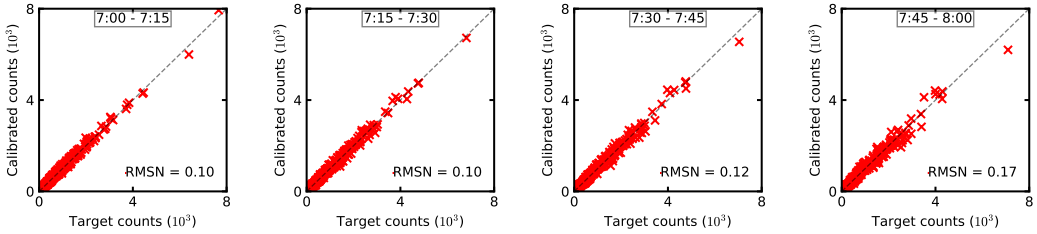


Figure 5.5: Comparison between generation methods for specific intervals.

Figure 5.7 and 5.8 are also plotted for method 6 and depict the quality of calibrated OD matrices by comparing it with the target and initial OD matrices on 45° plots. Overall, PC–SPSA is able to find a good quality solution and as per the property of PCA application (i.e.,

confining search space in historical OD variance) all OD pairs are close to the 45° line. Note that PC-SPSA is able to calibrate the reduction change of the target demand (i.e., plots in figure fig. 5.7 are around the 45 %) but it is not able to entirely converge the error due to random fluctuations (*Rand* in equation 5.19). This is an expected result when using PCA, as the PCs constraint the search space allowing for limited structural changes in the OD demand matrix. To understand better this behavior, we conduct a sensitivity analysis for different demand scenarios in section 5.4), and also compare the results from different historical OD generation methods which actually do behave differently for converging the random fluctuations (section 5.3.2).



(a) 7:00 - 7:15 **(b)** 7:15 - 7:30 **(c)** 7:30 - 7:45 **(d)** 7:45 - 8:00

Figure 5.6: Comparison of target and calibrated traffic counts.

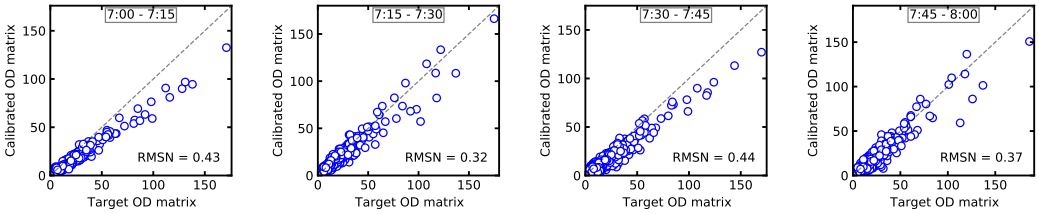


Figure 5.7: Comparison of target and calibrated OD matrices.

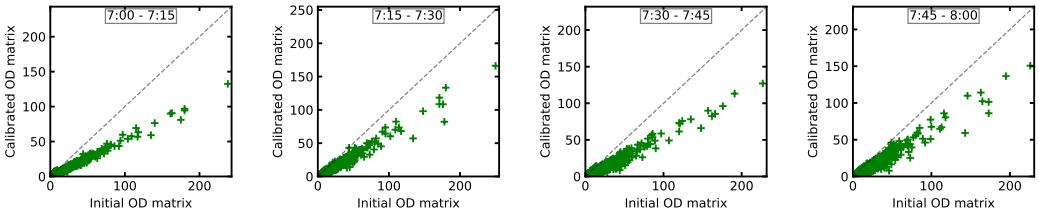
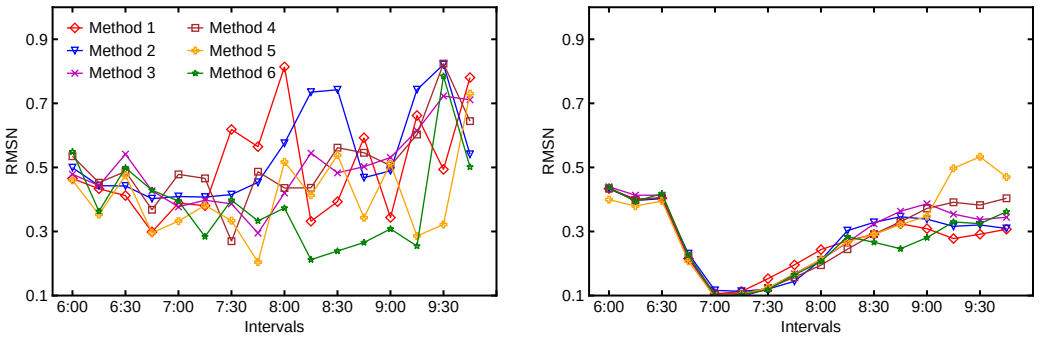


Figure 5.8: Comparison of initial and calibrated OD matrices.

Comparing different historical OD generation methods

Figure 5.5 deploys PC–SPSA's convergence results using all historical OD generation methods and despite that all methods show different converging speed, they can converge to almost the same level of error. This indicates restricted requirements and robustness of PC–SPSA on the historical OD estimates with respect to final error convergence. However, in terms of

the converging speed, the method capturing most correlations (method 6) and the methods considering only one–dimension correlation (method 1 and 2) outperform the others. For the latter, it is easy to understand as searching the pattern in a single correlated direction would be faster because the defined search space have more noise and randomness (local minimums). In contrary, when the correlations of two of three dimensions are fused (method 3, 4 and 5), they construct the search space with more accurate and sufficient information. Although the noise and randomness is reduced, now its presence probably hinders the SPSA algorithm to struggle finding the minimized solution. Surprisingly, method 6 which combines three dimension information, however, also leads to a fast convergence as method 1 and 2. This behavior may be due to the expectation that the space constructed by this method is more comprehensive and thus it directs the algorithm to find a faster direction compared with the ones with only two–dimensional information.



(a) Best RMSN for OD matrix **(b)** Best RMSN for counts

Figure 5.9: Comparison between all generation methods.

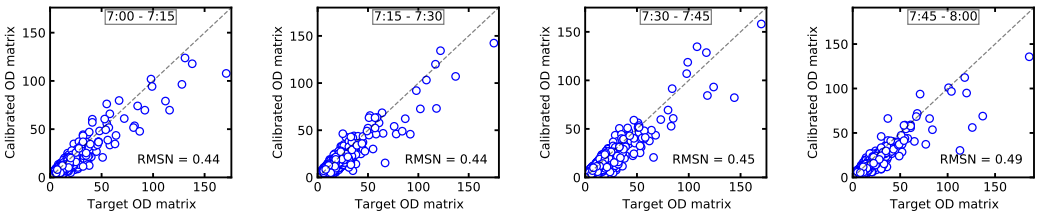


Figure 5.10: Comparison of target and calibrated OD matrices (method 2).

To better understand the above stated comparison, we further compare all the historical OD generation methods by their calibration quality. Figure 5.9 illustrates the quality of calibration for all generation methods with fig. 5.9(a) showing the quality of calibrated OD (RMSNs comparing with target OD) and fig. 5.9(b) showing the final convergence error achieved for the whole demand period. Moreover, as mentioned previously, literature efforts only considered temporal correlations for historical OD generation i.e., method 2, and hence we also show comparison of its calibrated OD with the target OD in figure 5.10. By analyzing fig. 5.9(a), we can validate the above mentioned arguments about the effects of using more correlated information in generation methods. In general, considering multiple correlations leads to a reduction in the demand error (fig. 5.9(a)) and a similar error in term of traffic counts (fig. 5.9(a)). Method 6, the one considering the highest number of spatio/temporal correlations, not only shows a faster convergence but it

is also the most consistent in terms of OD calibration quality (i.e., least RMSN error from target OD). At the other end of the scale, the methods considering only one correlation dimension (method 1 and 2) are the most inconsistent with poor quality OD estimates (see time intervals from 7 to 9 am), meaning that the faster convergence is mostly due to the model over fitting the data. Especially, figure 5.10 shows that the calibrated OD from method 2 is more scattered as compared to figure 5.7 for method 6 (further comparison of the calibrated OD quality for method 2 and 6 is shown in section 5.4). The methods with two correlations (method 4 and 5) have a medium range of OD quality. Perceiving these results, it can be established that creating the OD estimates with more correlation information helps in better calibration quality and having lesser random perturbation or noise also pushes towards faster convergence. Lastly, analyzing fig. 5.9(b), it can be seen that all different historical OD generation methods are able to eventually converge on very similar RMSN errors, validating the robustness of PC–SPSA algorithm convergence performance with different methods.

Conventional versus simplified problem formulation

Simplified problem formulation removes the error term z_2 (between the calibrated and prior OD) from the conventional problem formulation equation 5.2. This is similar as setting the w_{od} weight as 0 % in equation 5.2, which otherwise if set as w_{od} >0 is following the conventional problem formulation. Figure 5.11 shows the convergence performance of PC–SPSA at different weight settings (i.e., 0%, 20%, 40% and 60% weight w_{od} for z_2).

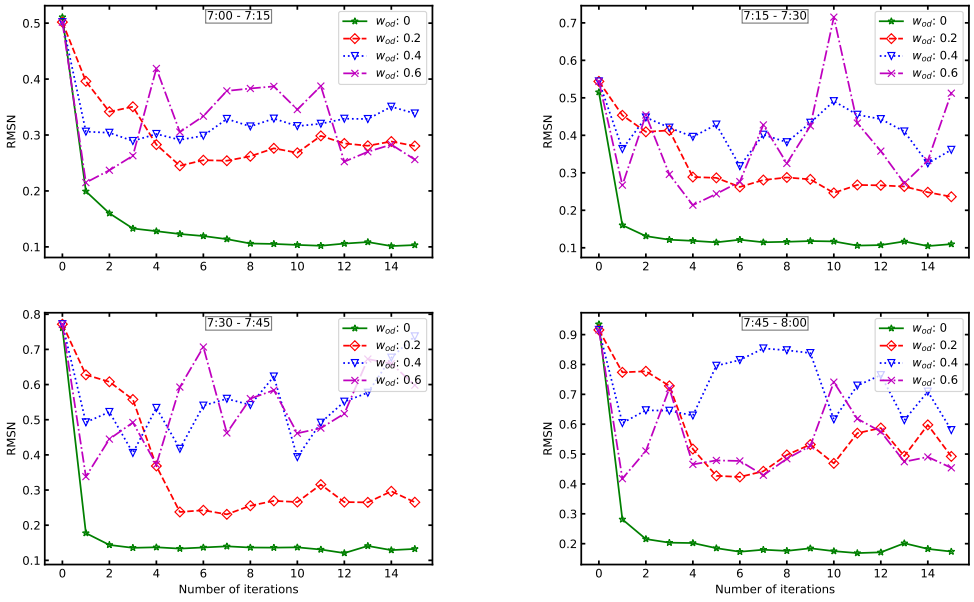
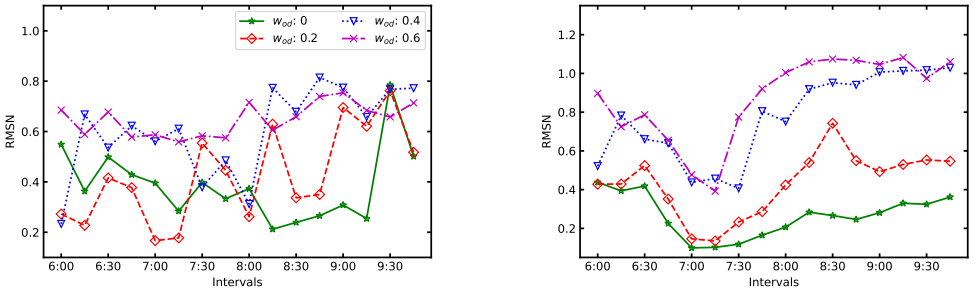


Figure 5.11: Comparison between objective weights for specific intervals.

Similarly, figure 5.12(b) shows the least RMSN error achieved for traffic counts and figure 5.12(a) shows the OD solutions' quality for all different weight settings. It is clearly evident that the simplified problem formulation outperforms all other weight settings for much faster convergence towards the least RMSN error. Another surprising outcome is from figure 5.12(a) where the simplified problem formulation also results good OD solution quality consistently.

Only 20% w_{od} gives better solution quality for some intervals but this comes at the cost of an increased error in the traffic counts.



(a) Best RMSN for OD matrix **(b)** Best RMSN for counts

Figure 5.12: Comparison between different weights combination in the objective function.

Note that, all the mentioned results confirm that we can utilize the benefits of using PCA application (i.e., limiting SPSA search space within the variance of historical OD estimates) for simplifying the DODE objective function. Since PCA application adds the required demand information in the PCs, further constraining the calibrated OD with prior/starting OD will have a double restraining effect adding unnecessary burden in the objective function. Moreover, even adding the weight w_{od} does not result in better OD solution quality indicating that PCA includes the OD information in a more structural way. Note that, as we increase the w_{od} weight for OD error term, the performance of the algorithm detoriates either it is in terms of convergence (figure 5.11), the least RMSN error (figure 5.12(b)) or the OD solution quality (figure 5.12(a)). Lastly, Figure 5.7 and 5.8 (plotted for method 6) also provide supplementary results for simplified problem formulation (showing the quality of calibrated OD matrices by comparisons with the target and initial OD matrices on 45° plots), where both plots show that the patterns of calibrated OD estimates are well estimated and are close to the target solution.

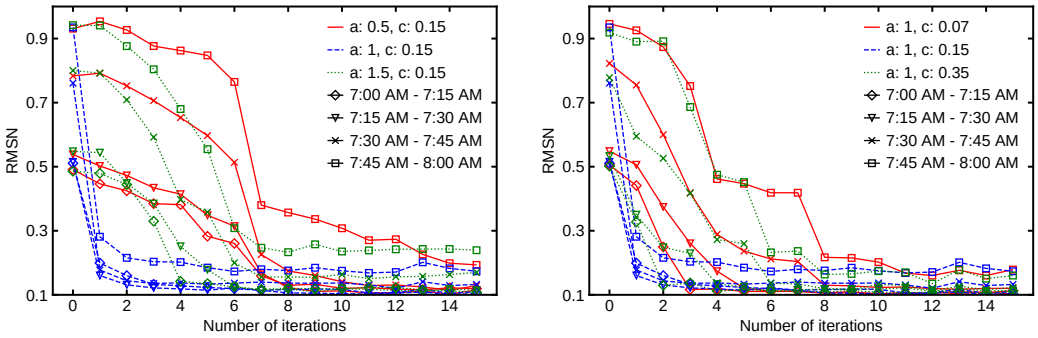
5.4 Sensitivity analysis

In this section, we perform sensitivity analysis on PC–SPSA with respect to SPSA parameters, demand conditions, and quality of historical estimates, respectively. The historical estimates are generated using method 6 (as per our analysis in section 5.3.2). Note that, the other parameters not specifically mentioned here remain the same as that in the previous section.

Robustness against SPSA parameters definition

In this section, we analyze the robustness of PC–SPSA against definition of SPSA hyper–parameters. SPSA is a random search stochastic algorithm and requires an appropriate definition of its hyper–parameters. These hyper–parameters can vary significantly for different problems and don't have any universally identified set of values (guidelines are given by Spall (1998a)). Since SPSA parameters are only defined by trial–and–error method during implementation, we observe its sensitivity for the PC–SPSA algorithm. Figure 5.13 shows the convergence plots for calibrating the Munich network case study with different set of c and a hyper–parameters. c is used for defining the perturbation step size, while a is used for minimization step (equation

5.18). Analyzing the results from fig. 5.13, PC–SPSA appears to be significantly less sensitive to varying SPSA hyper–parameters. The values used for both c and a vary significantly since they act as a percentage change instead of an absolute change. Although, the convergence rate is different among these hyper–parameter settings, all experiments converge to the a similar RMSN error value within a few iterations.



(a) Robustness against a

(b) Robustness against c

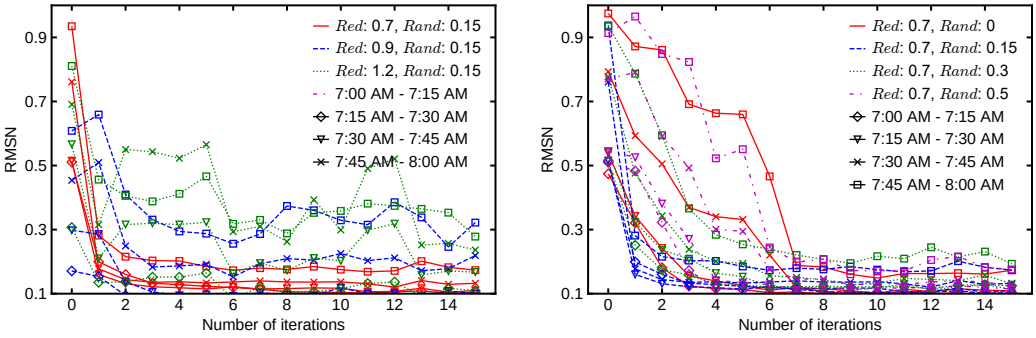
Figure 5.13: Comparison of using different SPSA parameter values (c and a).

We consider two reasons for PC–SPSA robust behavior, 1) the hyper–parameters act as the percentage change in perturbation and minimization (equation 5.16); and 2) faster convergence of PC–SPSA and properties of PC scores vector (i.e., very few estimation variables with even lesser being more significant). Also, since the rest of SPSA hyper–parameters i.e., γ, α and A are used for evolving the gain sequence parameters over the number of iterations, we do not add their sensitivity analysis as PC–SPSA converges in a handful number of iterations; making it insensitive to their definition (we use the default values given by Spall (1998a)). Overall, we can establish that PC–SPSA being robust, requires significantly less manual input or trail–and–error method for setup.

Performance in different traffic conditions and demand fluctuations

In this section, we analyze the performance of PC–SPSA in different traffic conditions and demand fluctuations. More specifically, we define different demand scenarios using eq. (5.19) and analyze PC–SPSA convergence. Here its noteworthy to mention that, the historical demand matrix D is created using method 6 (section 5.2.3) with R_{min} as 0.3 and $\Delta_{od,t,d} \sim N(0, 0.333)$.

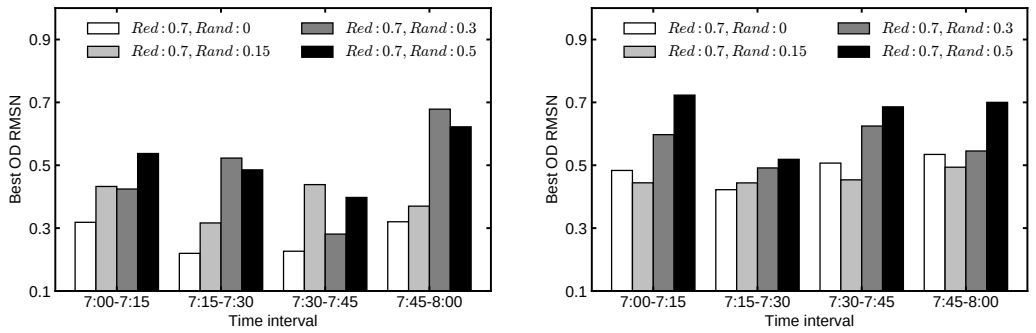
Figure 5.14(a) shows the PC–SPSA performance under different network conditions, where *Red* coefficient (from eq. (5.19)) are set to 0.7 (70%), 0.9 (90%) and 1.2 (120%) in reference to starting/current OD matrix while keeping the *Rand* coefficient constant as 0.15 (15%). These set of variables result in target demands with three different traffic conditions i.e., less–congested, normal/congested, highly congested. Analyzing fig. 5.14(a), PC–SPSA converges well for the first two scenarios converging to a low RMSN error, but struggles to calibrate the highly congested scenario. The zig–zag behavior of its convergence is due to the use of traffic counts in congested state, which adds more noise in the objective function. This is a known result for demand calibration and this is why, for practical implementation, it is suggested to always use a matrix that is less congested than the target one. This can be easily done by comparing the simulated and observed traffic data. Still overall, PC–SPSA is able to converge the RMSN errors for all different traffic conditions.



(a) Reduction sensitivity **(b)** Randomness sensitivity

Figure 5.14: Demand scenarios sensitivity (method 6)

Similarly, fig. 5.14(b) shows PC–SPSA performance while calibrating against different magnitudes of random fluctuations in target demand generated using multiple *Rand* values in *eq.* (5.19), while fig. 5.15(a) illustrates the subsequent OD solution quality for all scenarios. As mentioned above the D historical data–set is generated with R_{min} as 0.3, hence the target demand generated equal or above $Rand = 0.3$ should contain more significant demand fluctuations than what are present in D data–set. Analyzing the results from fig. 5.14(b) and fig. 5.15(a), PC–SPSA using method 6 with 30% R_{min} is able to converge all demand fluctuations scenarios resulting in a low RMSN error but with varying solution quality (i.e., RMSN between calibrated and target OD). Comparing the scenarios results individually, $Rand = 0$ scenario has the target demand without any pattern changes and gets the best OD solution quality but PC–SPSA convergence is quite slower because the algorithm is still directly perturbing the OD patterns hence it also requires a few iterations to get back to closer solution (a reduced clone of initial OD). A similar convergence trend can been seen in $Rand = 0.5$ scenario, since the target demand patterns are highly fluctuated and is even more than the variance within historical demand D, hence it requires more time for converging to a low RMSN error and with poor OD solution quality (i.e., the possible solution within the variance of historical estimates satisfying the traffic measurements).



(a) Method 6 **(b)** Method 2

Figure 5.15: Best OD RMSN of scenarios with different randomness.

Note that, with the increase in *Rand* values both the OD solution quality and algorithm convergence performance deteriorates because the target solution has more demand fluctuations (i.e.,

higher *Rand* component) from initial OD. Hence, we can say that overall PCA–based methods have limited performance against estimating higher random demand fluctuations especially because the OD solution quality deteriorates significantly. Furthermore, figure 5.15 also compares the OD solutions' quality for method 2 and 6, where the latter is able to result better OD solutions consistently against all scenarios. This comparison validates the argument that using all three correlations dimensions (method 6) helps in establishing the search space more structurally around the initial OD. It is also noteworthy to mention that, the fact that PC–SPSA has limited performance against random demand fluctuations also signifies the importance of the proposed data–assimilation framework which allows derivation of the correlations from other data sources to form more realistic search space for PCA–based calibration.

Historical estimates setup

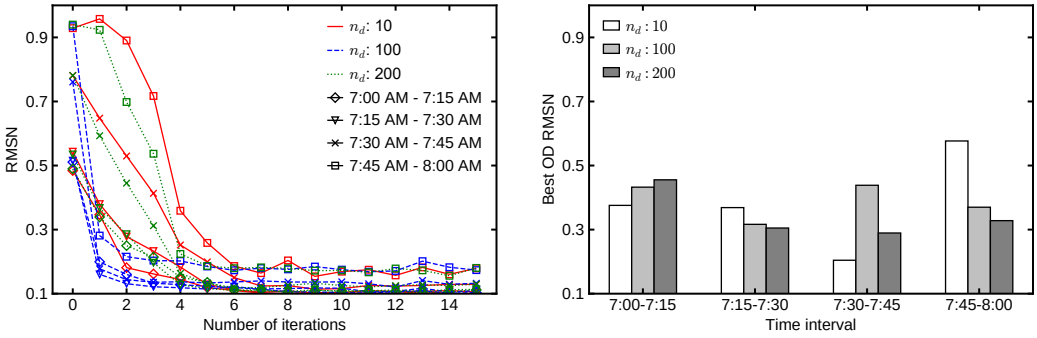
As already established, all PCA–based methods heavily rely on the quality of historical estimates. Previously, in section 5.2.3, we proposed a data assimilation framework to create estimates from an initial historical matrix for scenarios where they are unavailable or irrelevant. These established generation methods should also allow to control the quality of historical estimates and calibrated OD solution (in reference to starting/available OD estimates). In this section, we explore the effects of historical data–set D generation variables i.e., n_d the number of days historical data–set contains, R_{min} for resizing the variance within historical estimates and σ (standard deviation) for Δ (i.e. the correlated random matrix) defining the shape of variance.

Size of historical data–set

The number of historical observations n_d is an additional parameter to be calibrated when using PCA in the context of the DODE. Figure 5.16 illustrates the PC–SPSA performance upon using three different sizes of D data–set. Analyzing these results, it is evident that the size of D data–set influences the convergence plots (fig. 5.16a) as if the n_d is too small or large, the convergence gets slower. Comparing the OD solution qualities for different D data–set sizes (fig. 5.16b), the increase in size seems to improve both the consistency and quality of estimated OD solution. The convergence results can be explained such that the size of D data–set defines the amount of variance which if is too small or large the algorithm needs more iterations for convergence, while given an appropriate set of n_d historical estimates, the algorithm performs faster. This is proven by the fact that for $n_d = 10$ both the convergence results and OD solution quality show larger fluctuations while on the other hand, a larger number of observations ($n_d = 200$) shows a much more consistent quality, which is explained by capability of the model to better incorporate the structure of the demand. Overall, it can be established that small size of D data–set contains less variance directing the algorithm to converge slower and with random OD estimate quality, while as the number of observations in D data–set increase the amount of variance generated also increases which till a certain optimum value improves convergence but later with further increase the convergence requires more time due to larger search space. But enlarging the variance or search space always helps to improve the consistency in OD solution quality.

Variance within historical data–set

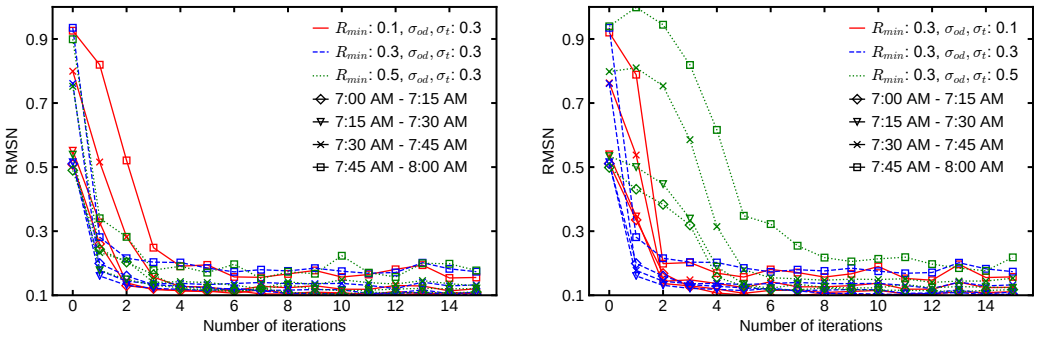
Next, we perform the sensitivity analysis on defining the variance of historical data–set D. Different set of values are used for R_{min} and σ (i.e., the standard deviation for the Gaussian distributions defining Δ_T coorelation) to generate historical data–set using method 6. Note that, the effect of changing both R_{min} and σ is quite similar with a minor difference, where R_{min} widens/shrinks the shape of Gaussian distribution with increasing/decreasing the values



(a) Traffic counts RMSN **(b)** OD RMSN

Figure 5.16: Historical data matrices size sensitivity.

of random distribution, σ directly effects the distribution of random numbers. We also perform the analysis for calibrating two different target demand fluctuations, setting *Rand* in eq. (5.19) as 0.15 and 0.3, while the σ is set to 0.333 (i.e., $\delta \sim N(0, 0333)$. Figures 5.17 and 5.18 illustrates the convergence plots for both demand scenarios subsequently, while Figures 5.19 and 5.20 show the OD solution qualities for varying R_{min} and σ experiments.

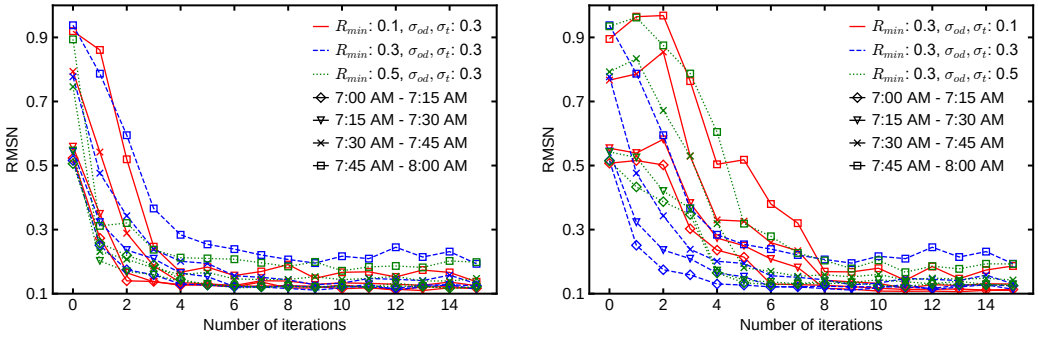


(a) Robustness against R_{min} **(b)** Robustness against σ_{od}, σ_t

Figure 5.17: Demand scenarios sensitivity (scenario: $Red = 0.7, Rand = 0.15$).

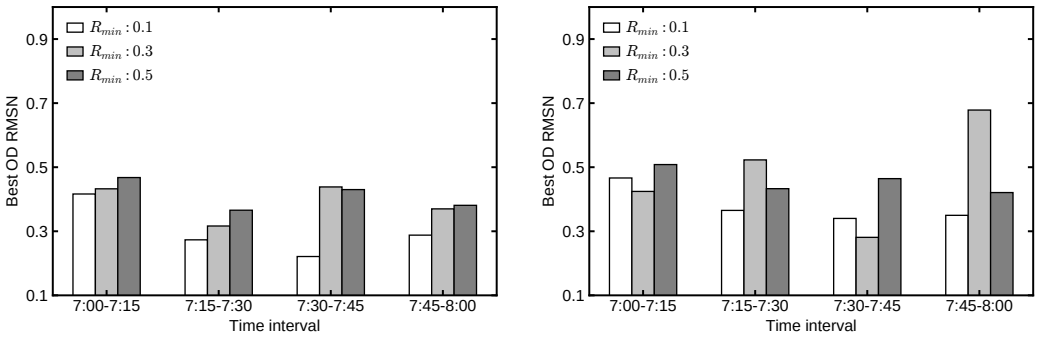
First analyzing the effect of varying R_{min} values, the calibration convergence plots are similar to the demand fluctuation experiment from fig. 5.14(b) i.e., for scenarios where $R_{min} > Rand$ the convergence is much faster (see $R_{min} = 0.5$) and for $R_{min} \leq Rand$, the convergence is slower (see $R_{min} = 0.3$ for $Rand = 0.3$ scenario). While fig. 5.19 illustrates that lower R_{min} setting results in better OD solution quality and as we increase R_{min}, the error between target and calibrated OD also increase. The performance for varying R_{min} is consistent with the previous results from section 5.4, i.e., if we use larger values, the variance space increases and the algorithm converges faster but to a poor quality solution (see fig. 5.19). Hence, given the results it can be said that the use of lower values for R_{min} is more efficient unless either the solution is not converging and more variance space is required or a faster convergence is desired.

Next, analyzing the effect of varying σ values, the algorithm convergence is slower for both the smaller and larger σ values and is more optimum for middle value of $\sigma = 0.3$. Considering the OD solution qualities, note that similar to R_{min}, lowest value of σ result in the best calibrated ODs



(a) Robustness against R_{min} **(b)** Robustness against σ_{od}, σ_t

Figure 5.18: Demand scenarios sensitivity (scenario: $Red = 0.7, Rand = 0.3$).



(a) scenario: $Red = 0.7, Rand = 0.15$ **(b)** scenario: $Red = 0.7, Rand = 0.3$

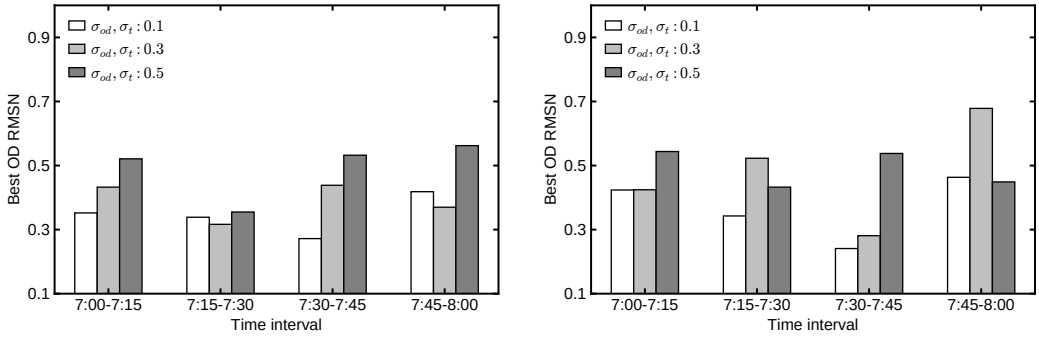
Figure 5.19: OD RMSN with different R_{min}.

relative to the target solution. Hence, to achieve better calibration efficiency in solution quality, lower amount of variance is desirable. The convergence behavior of varying σ is similar as of varying sizes of D data–set (fig. 5.16) which also control the amount of variance and the middle optimum size gave faster convergence. But, it is noteworthy to understand that controlling the variance through R_{min} or σ is more systematic which create a more restrictive search space around initial OD estimate generating better OD solution qualities.

Comparing the results of varying R_{min} and σ experiments, first it is interesting to see that lower values of both parameters can converge much more fluctuating demand scenarios (i.e., with $Rand = 0.5$ and σ =0.333). Then, also note that, in comparison to the lower σ value of 0.1 (with $R_{min} = 0.3$), the setting of $R_{min} = 0.1$ and $\sigma = 0.3$ gives much faster convergence. Hence, we can conclude that restricting the generated variance by directly reducing the random vector distribution is less efficient than keeping the random vector generation more distributed using higher σ and than tuning down the amount of variance by use of smaller R_{min} values.

Remarks

The combination of PCA's dimension and complexity reduction with simplified problem formulation gives significant boost to SPSA calibration performance. Also, the proposed framework for data-assimilation generation of historical estimates gives the flexibility to control the size and



(a) scenario: $Red = 0.7, Rand = 0.15$ **(b)** scenario: $Red = 0.7, Rand = 0.3$

Figure 5.20: OD RMSN with different σ_{od} and σ_t.

quality of generation historical variance i.e., the algorithm search space or directions for PCA–methods. Overall, the set of inputs required to use PC–SPSA in our proposed framework include: SPSA hyper–parameters (c, a, α, γ, A), historical data–set generation parameters (generation method, R_{min}, n_d, σ) and PCA application parameters (amount of dimension reduction i.e., V to $\hat{V}$, temporal limits for combined PCA application). In sections 5.3.2 and 5.4, we perform a set of experiments on different parameter inputs for PC–SPSA setup. Analyzing the empirical outputs of these experiments, we enlist the guidelines in below sections which can be followed for efficient calibration setup.

SPSA hyper–parameters

Although Spall (1998a) gave guidelines for defining appropriate SPSA parameters, their definition remain problem specific with no universal values for different DTA models. For PC–SPSA, the perturbation c_k and minimization a_k coefficient behave as percentage change instead of absolute, hence they can be set similar for varying DTA models even having different magnitudes of the decision variable. The results from section 5.4 depict that PC-SPSA is even robust for significantly varying values of c and a but they still effect the convergence speeds. Hence, for efficient performance of the algorithm, c can be set in range of 0.1-0.2 resulting in c_k with 10-20% change at first iteration. Similarly, for setting a parameter, a range between 0.8-1.2 is optimum for the current network and using the RMSN as estimator but it depend on the resulting gradient values. Due to fast convergence, the other SPSA parameters α, γ and, A are insignificant because they only control the evolution of gain sequence parameters (c_k, a_k) over the increasing number of iterations.

Historical data–set generation

The proposed data-assimilation framework generate historical OD data–sets using all different correlations present in time-dependent ODs. The set of inputs given in these generation methods include number of correlation dimensions or generation method, size n_d of the historical estimates, R_{min} to control size of generated variance and σ to define the correlation distributions used to generate Δ_T perturbation matrices. In 5.4, sensitivity analysis on each of these stated parameters are performed to understand their effect on calibration convergence and OD solution quality. Below are the stated guidelines to be followed for each parameter:

- **Generation method:** Given the results in figure 5.9, method 6 which generates D data–set with all correlation dimensions outperform because of its consistency in convergence speeds and OD solution quality. Hence, it is recommended to use method 6 for implementing PCA–methods with the proposed data-assimilation framework.

- **Size of historical data–set:** In section 5.4, analysis upon different sizes of historical data–set is performed. For faster convergence of DODE, the optimum size of generated D data–set should be around 3-4 months (90-120 prior days). Further, to improve the quality and consistency of OD solution quality D data–set can be further extended to higher size but at an expense of reducing convergence speed.

- **Variance of historical data–set:** Section 5.4 gives the analysis on defining different variance characteristics within generated D data–set. Two parameters (i.e., R_{min} and σ) are set to control the variance. Individually, smaller values of both parameters (around 0.1) result in optimum OD solution qualities as they restrict the generated variance closer to the seed OD matrix. In terms of convergence, higher values of R_{min} always result in faster convergence but at an expense of more nosier/poor OD estimate, while very low or high σ values show slower convergence, hence optimum value of $\sigma = 0.3$ can result in faster convergence.
  For combined set of values for both R_{min} and σ, it is recommended to use larger σ value in range $0.3 - 0.5$ with smaller value of R_{min} in range $0.1 - 0.15$. This helps to generate a more distributed variance with higher σ but with a much smaller size contained by lower values of R_{min}. If convergence error results are not satisfactory, gradually increasing the R_{min} value is recommended due to probabilities of larger fluctuations in target demand. Note that higher value of R_{min} in such case with always reduce the OD solution quality.

PCA application

The application of PCA on DODE has been covered previously in literature. Djukic et al. (2012) showed the detailed concept of PCA application on OD estimation. Later many other approaches followed the use of PCA to develop variants of conventional approaches (Prakash et al., 2017, 2018; Qurashi et al., 2019). Once the historical OD estimates are available, two main inputs are required for PCA application: 1) the amount of dimension reduction or the number of PCs retained 2) Temporal settings of historical estimates to apply PCA.

The first input of PCs retained during dimension reduction (i.e, changes V to $\hat{V}$ in 5.4) is commonly given in terms of the level of variance explained by the retained PCs. Since mostly the first few PCs are the most significant, explaining the majority of variance, a cumulative variance of 95% is set for reducing the PCs matrix V. The second input about temporal settings of historical data–set D is defined inside matrix x of eq. (5.9) in our proposed framework. This input is the number of n_t time intervals set together for application of PCA. It is recommended to apply PCA for the time intervals which have a single activity pattern (e.g., morning or evening peak hours separately). It is also a work in progress for future research to do more systematic PCs extraction from discrete activity patterns and then use the combination of these PC-directions to do more efficient OD estimation.

5.5 Conclusion

This chapter propose and evaluate practical implementation methods for the application of Principal Component Analysis (PCA) for model calibration. PCA application has become a standard

for improving the scalability of conventional algorithms towards large–scale DTA models. However, to use PCA requires the availability of historical estimates, which are usually not available in practice, especially for large–scale networks. This is a major limitation of current PCA–based methodologies, which is addressed in this study. In addition, while current approaches mostly focus on using PCA for dimension reduction, this study also focuses on exploiting the properties of PCA based model calibration for simplifying the structure of the calibration process and understanding how the quality of historical estimates influence prediction accuracy.

The major contribution of this research is to propose a data–assimilation framework which allows to incorporate the structure of the historical (seed) demand into the Principal Components (PCs) without the need for historical estimates. Such a framework allows the use of all PC-based algorithms proposed in the literature when historical data is irrelevant or unavailable (a standard case for large–scale networks). Meanwhile exploiting the properties of PCA, a simplified problem formulation for Dynamic Origin Destination matrix Estimation (DODE) is also presented, which allows removing the demand from the objective function. These extensions are tested using PC-SPSA that combines PCA with the well known Simultaneous Perturbation Stochastic Approximation (SPSA) model. The study shows that the enhanced algorithm achieves much faster convergence and provide more robust results even on large urban networks. Different historical OD generation models are proposed and tested in this study, each of which accounts for different types of correlations between the variables. The results suggest that the method with most correlations outperforms others for convergence speed, robustness of the results, and calibrated OD solution quality. The proposed framework also provides the flexibility to include data–driven spatial–temporal correlations extracted from other data sources, representing more realistic structure of PCs which can better reflect the historical OD flows' dynamics.

In this study, we tested the model on the network of Munich, one of the largest DTA models ever used as a calibration case study. Even on such a scale (above 8000 links and 20.000 variables), the results indicate that a very low number of iterations is required for convergence (requiring around 10 simulation runs) which is very low in comparison to conventional techniques like SPSA (almost 150-300 simulation runs) on much smaller networks. Further, the PC–SPSA implementation used in this research shows robustness towards the definition of SPSA hyper–parameters indicating ease of algorithm setup. The proposed approach allows to introduce domain specific information within the PCA algorithm by using probability distributions to describe spatial and temporal correlations. These distribution are characterized by mean and variance parameters which also require pre–definition and become additional hyper–parameters. But, these parameters also provide more control over the OD solution quality. These findings are summarized in section section 5.4, which introduces implementation guidelines for PC-SPSA and can also be used to combine enhanced SPSA algorithms, such as the W–SPSA, and PCA.

This research introduces the first building block to move PCA–based calibration models from theory to practice. Existing works in fact rely on historical estimates of the demand, which are not necessarily always available. The proposed concept of data-assimilation opens many promising research directions e.g., to incorporate synthetic populations, activity based models, and, in general, more information about the travel demand without increasing the complexity of the problem. Similarly, it can also allow incorporating different data sources, such as mobile phone network data, GPS trajectory data, and even social media data. Finally, traditional PCA–based are linear in their nature. However, there is not guarantee that data are linearly correlated, specifically when using different data sources or complex representations of travel behaviour, such as synthetic populations. Therefore, non linear PCA–based frameworks should also be investigated in the future.

6 AMoD ridesplitting case study

Contents

This chapter models the case studies for the microscopic AMoD ridesplitting platform and the market equilibrium model. The first case study focus on exploring the effects of microscopic modeling of AMoD ridesplitting and the relations between service demand, rider waiting flexibility, ridesharing occupancy, and service benefits. While the second case study shows the efficacy of ridesplitting market equilibrium model under varying operational objectives. It also further utilizes the ME model to assess the benefits and impacts of utility-based compensation pricing in improving ridesplitting service perception and adaptability.

The content of this chapter has been presented in the following works:

Qurashi, M., Jiang, H., & Antoniou, C., 2022. Microscopic modeling and optimization of autonomous mobility on–demand ridesplitting, *(Submitted)*

Lu, Q., Qurashi, M., & Antoniou, C., 2022. A utility-based compensation pricing method for ridesplitting services. Transportation *(Under revision)*

6.1 Introduction

This chapter includes two case studies specific to AMoD ridesplitting service. First, a simulation-based case study employing the AMoD ridesplitting platform developed in chapter 3 and, second, an analytical demand-based case study that employs the ridesplitting market equilibrium (ME) model and explores the benefits of utility-based compensation pricing method both developed in chapter 4. Note that both case studies represent ridesplitting services in different contexts. The first case study models the ridesplitting service time-dependently using microscopic traffic models. The second case study models an aggregated representation of the service market for modeling demand-supply equilibrium.

The AMoD ridesplitting platform proposed in chapter 3 models AMoD ridesplitting using microscopic traffic models with integration of a dynamic and stochastic DARP algorithm for service optimization. The platform allows modeling link-level vehicle driving and service operations, detailed traffic emissions, different vehicle driving behaviors, and individual rider trips for AMoD ridesplitting, all in a dynamic traffic environment. Therefore, the case study is set up to understand the effects of microscopic service modeling, ridesplitting service performance and benefits, and the effects of varying ridesharing occupancies. The ridesplitting service is modeled standalone as a separate transport mode with a service area around Munich city center, i.e., serving short distance trips. The case study experiments model different demand and rider flexibility scenarios and also compare different service vehicle types, i.e., electric-automated and human-driven petroleum vehicles, for their driving behaviors and emissions. Meanwhile, the ridesplitting service benefits are evaluated by comparing it with private transport.

The second case study is used to model the ridesplitting market equilibrium (ME) and utility-based compensation pricing method that are proposed in chapter 4. Since the ME model represents an aggregated ridesplitting market, the second case study employs a larger Munich city area (also employed in chapter 5). It assesses the optimum service penetration and price rates (affecting demand attraction) against profitability and social welfare objectives. The ridesplitting market is modeled in competition with two other transport modes, i.e., private car and public transport. It employs a user preference survey (from section 4.2) to scale how users perceive ridesplitting in model its demand. Further, the case study also explores how utility-based compensation pricing adds certainty and equity among ridesplitting riders and proposes suitable opportunities for smart subsidy schemes that can help promote ridesplitting.

6.2 Case study for AMoD ridesplitting platform

This section evaluates the performance and efficacy of the AMoD ridesplitting platform, specifically modeling of the ridesplitting service microscopically with integrated service optimization (proposed in chapter 3). For this purpose, the case study of the Munich center region is employed, modeling the AMoD ridesplitting as a standalone service in the region.

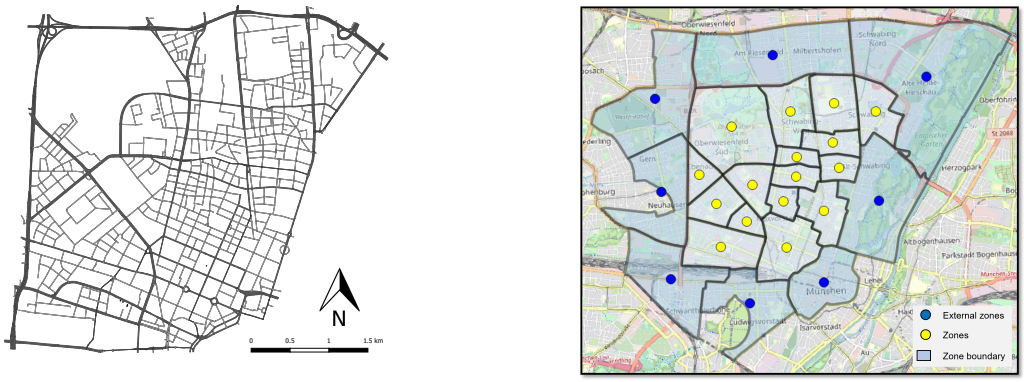
To describe the outline, section 6.2.1 describes the experimental setup, including details on network and simulation setup, the model calibration procedures, and the scenarios set up to test the platform. Then, section 6.2.2 observes the effects of employing microscopic models by, i) comparison against the use of simpler time-dependent travel times (as in Li et al. (2019a)) for rider serviceability and level of service distributions, ii) depicting different microscopic vehicle profiles that show microscopic vehicle network propagation, iii) modeling and comparing two different vehicle types of electric AV and petroleum vehicle for both ridesplitting and private rides. Further, section 6.2.3 does an occupancy analysis to observe the effects of ridesplitting service type on triggering ridesharing under different demand and rider flexibility scenarios. Finally, section 6.2.4 explores the traffic and environmental benefits resulting from the use of

ridesplitting instead of private rides. Apart from the aggregated benefits, benefits per rider trip are also evaluated for different ridesharing occupancy levels to help observe the efficacy of high occupancy ridesharing.

6.2.1 Case study setup

6.2.1.1 Network and simulation setup

We define an AMoD ridesplitting modeling case study using a Munich city center region network. The network covers the area of the Maxvorstadt and Schwabing regions and is partially surrounded by the inner ring of Munich (i.e., Mittlerer Ring or Bundesstraße 2R). It comprises of 1,249 links, shown in fig. 6.1a. Note that the passenger preference survey mentioned in chapter 4 was also conducted in this region, and the region is also a part of the operational area for Isartiger, i.e., a pilot experiment for ridesplitting service in Munich (MVG, 2022). Furthermore, to model the trip-based network demand, the network is spatially divided into 16 TAZs, resulting in 256 OD pairs. At the same time, eight external zones are also added to realistically calibrate and represent the traffic conditions (shown in fig. 6.1b).



(a) Network **(b)** Traffic zones

Figure 6.1: The case study of Munich city center area

The AMoD service experiments are run simulating the morning time between 8 to 9 am with a 15 min pre–simulation phase for a warm-up and 15 min post-simulation phase for concluding in-service requests. The traffic model is simulated at the microscopic resolution to allow modeling of the on-demand service operations and driving behavior of individual AMoD vehicles. The traffic assignment is carried out via the stochastic user assignment method (one-shot assignment in SUMO). Moreover, two different vehicle types, autonomous (AV) and a regular passenger vehicle, are modeled by using different driving behavior setups (discussed in section 6.2.1.3). Similarly, the emissions are modeled by using HBEFA3 based model (INFRAS, 2022) for regular petroleum vehicles and an electric vehicle model developed by Kurczveil et al. (2013) to model electricity consumption, both available as default types in SUMO.

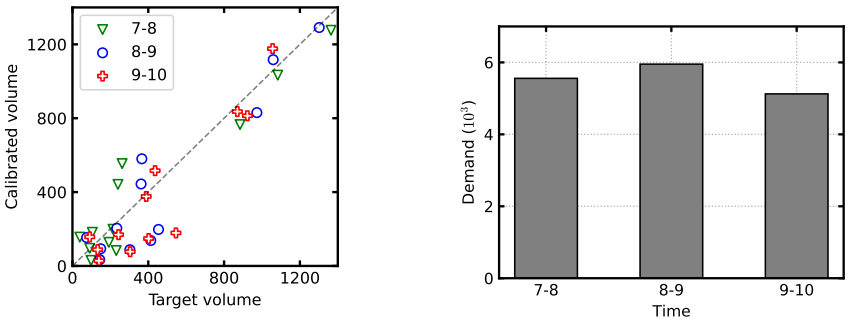
6.2.1.2 Model calibration

Among different modeling steps, model calibration is a crucial step to help better represent reality, e.g., realistic traffic congestion and vehicle driving behaviors. Estimating both supply and demand parameters is required to calibrate a dynamic traffic assignment model. For microscopic

models, supply includes the calibrating driving behavioral parameters. While since the travel demand is represented using a trip-based OD matrix, calibrating it requires estimating all different OD pairs. Chapter 5 discusses the process of dynamic OD demand estimation in detail. Therefore, in this section, we mention it briefly.

To formulate the demand calibration problem, we simulate the morning peak hours between 6 am and 11 am, where the first and last hour is used as warm-up and dissipation time intervals. Since the study area represents a part of the Munich city center region, it contains only a portion of the Munich inner ring (Mittlerer Ring). Thus, the amount of traffic that uses the Mittlerer ring for bypassing tends to use the inner network, increasing the overall traffic in the inner network, contradicting reality. We tackle this problem by first under-fitting the traffic demand from/to external zones for which the bypassing should occur. Since the number of estimation variables were significantly less, we used the SPSA algorithm for the estimation.

Moreover, to retain the minimum addition of noise in the demand pattern, SPSA calibrates the demand of all intervals simultaneously. Then, we employ the PC-SPSA algorithm (following the process as in chapter 5) to calibrate the overall OD matrix demand for all OD pairs. The estimation problem is aggregated at one-hour interval, and the Root Mean Square Normalized (RMSN) error is used as a goodness-of-fit measure between the observed and simulated traffic counts. Figure 6.2a depicts the detector-based calibration error by comparing the calibrated and observed traffic counts. The results show that the demand is considerably well calibrated, especially against the high-volume locations. Figure 6.2b provides the information for the total calibrated demand of the network by each hour.



(a) Calibrated traffic vs. observed traffic

(b) Calibrated demand

Figure 6.2: Calibration results.

The process of calibrating the driving behavioral parameters for human-driven vehicles in this case study is discussed under Dinar (2020). The Wiedemann99 car-following model is used for modeling the driving behavior, while its parameters are calibrated using the SPSA algorithm setup. Note that the calibration is set up using a set of collected data and a corresponding network model of an arterial road in the Munich-Maxvorstadt region (i.e., Ludwigstraße and Leopoldstraße).

6.2.1.3 Scenario Setup

Since the focus of this case study is testing the AMoD modeling platform and exploring aspects related to microscopic modeling and ridesplitting service behavior, we mainly explore different service vehicle types with subsequent emission models and driving behaviors that specifically

require microscopic models. In addition we also explore different level of service demand and riders' flexibility that mainly dictates the levels of ridesharing occupancy in ridesplitting. In other words, the modeled scenarios explore the ridesplitting performance and benefits at different levels of ridesharing occupancy and vehicle type scenarios. Below, we provide further details on the ridesplitting supply and ridesplitting demand characteristics set for the experiments.

Service characteristics

Below we discuss the definitions of each of the different service characteristic. Note that as mentioned, in this case study we only explore different vehicle types while all the rest service characteristics are set fixed and are explored in a more detailed experimental setup in chapter 7.

- **Vehicle types and driving behaviors:** Two different vehicle types are modeled to replicate autonomous vehicles and human driven vehicles (table 6.1). Driving behavior for both vehicle types is modeled using Wiedermann99 model, where AV parameters (listing 6.1) are set based on the guidelines provided by Sukennik et al. (2018) and human driving parameters (listing 6.2) are calibrated from an arterial road case study of Munich-Maxvorstadt region (Dinar, 2020).

Table 6.1: Case study vehicle types

Vehicle type	Car following model	Emissions model
PV-Petroleum vehicle (human-driven)	Wiedermann99	HBEFA3
AV-Autonomous electric vehicle	Wiedermann99	Electric vehicle model

```
1  <vType id="AV" vClass="passenger" carFollowModel="W99"
2  cc1="0.9" cc2="0" cc3="-8" cc4="-0.10" cc5="0.10"
3  cc6="0" cc7="0.10" cc8="3.5" cc9="1.5"
4  length="7" personCapacity="8" maxSpeed="22.22"
5  type="ElectricVehicle" emissionClass="Energy/unknown"/>
```

Listing 6.1: AMoD vehicle definition in additional XML format

```
1  <vType id="W99_manual" carFollowModel="W99"
2  cc1="1.5" cc2="4" cc3="-8" cc4="-0.40" cc5="0.35"
3  cc6="11.44" cc7="0.25" cc8="4" cc9="1.5"
4  vClass="passenger" personCapacity="8" probability ="1"/>
```

Listing 6.2: Human driven vehicle definition in XML format

- **Fleet size and vehicle capacity:** The fleet size is set as 8 service vehicles each with a capacity of maximum 8 passengers sharing their rides.
- **Initial fleet positioning:** The vehicle initial positions are set using the network demand patterns. We identify the trip production ratios for all zones against the cumulative network trip origins and use the same ratios to divide the fleet vehicles over the network.

- **Service locations:** As discussed in section 3.5.2, modeling virtual internal nodes is a requirement to model link-based service operations through node-based DARP optimization. Specifically a few internal nodes can be used to do optimum network coverage (fig. 3.6), an idea also used in microscopic traffic modeling to produce and attract traffic vehicles on selected connectors (network links) that approximate the detailed network effect.

  To model AMoD ridesplitting, we also use a selected set of service locations. The predefined n number of connectors per zone are identified by bisecting the zonal spatial area in n sub-areas and finding the nearest link to the centroid of each sub-area. Such a technique allows equal distribution of service location spatially. Another smarter method is to divide each zone into sub-areas with an equal walking or driving diameter, which can also cater to varying spatial size of different zones. The final set of service locations or connector are used as the origin-destination of the ridesplitting rider requests.

Demand and rider flexibility

Ridesplitting is unique from other ridesourcing services due to dynamic ride-matching and detouring. Without these specific characteristics (as in other services), the number of trips served is defined by the attracted demand and available supply. However, specific to ridesplitting, riders' flexibility (in waiting and detouring) also directly defines the extent of ridesharing occupancy and total trips served because the service vehicles are scheduled considering riders' trip time preferences as constraints (see chapters 3 and 4).

Considering the above arguments, we define three different demand scenarios for ridesplitting services (table 6.2), changing the ridesplitting mode share and riders' waiting time flexibility. Note that the mode share directly defines the number of requests/opportunities for ridesharing, whereas the waiting flexibility scales the possibility of detouring. Therefore, all three scenarios are in ascending order for resulting ridesharing occupancy. Specifically, S1 is with low demand - medium detouring possibility, S2 has high demand - medium detouring possibility, and S3 has medium demand - high detouring possibility.

Table 6.2: Ridesplitting demand scenarios

Scenario	Mode share-Q (%)	Waiting time-w (min)
S1	5	10
S2	15	10
S3	10	15

Note that, although the riders' flexibility is defined by both waiting and detour flexibility, the results from the detailed experimental analysis (chapter 7, section 7.1) conclude that waiting time flexibility is more influential, directly affect the ridesharing occupancy, whereas different detour levels are supplementary to waiting time for ridesharing benefits. Therefore, to limit the number of scenarios allowing more detailed analysis, the detour flexibility is fixed at 100% (equal to direct trip time).

6.2.2 Effects of microscopic AMoD modeling

This section explores the effects of modeling AMoD ridesplitting in microscopic traffic models. These models simulate discrete vehicle movements on the network to replicate detailed traffic

dynamics. Thus, AMoD vehicles operate in the dynamic traffic environment, driving with autonomous behavior models and conducting naturalistic stop operations on network links. To assess the microscopic modeling effects, we compare the AMoD platform with the method of using time-dependent travel times for representing the network and vehicle movements, named 'Scheduler' (used by Li et al. (2019a)). The comparisons are made for the total amount of trips served under multiple scenario replications and the distributions of experienced trip attributes.

Moreover, we also represent profile plots to show the microscopic driving behavior of a random service vehicle and timelines of ridesharing occupancy for the whole fleet. Finally, we also compare driving behaviors of human-driven and autonomous vehicles to measure latency in trip times under three possible vehicle type combinations for ridesplitting versus private rides.

Rider trips served

Figure 6.3 plots the total trips served by both scheduler and AMoD platform for all three demand scenarios against multiple replications. Note that trips served from both S2 and S3 scenarios are comparable, where S2 has more rider requests while S3 has more rider waiting flexibility. However, the S1 scenario results in fewer trips served due to both lesser demand and waiting flexibility. Moreover, comparing different scenario replications, a significant variance is evident for total trips served due to the stochasticity in trip locations and depart times dynamically affecting the ridesplitting opportunities, as in reality. Finally, the total trips served also variate between the scheduler and the AMoD platform. The main reason is the modeling of dynamic traffic congestion and vehicle driving interactions with the infrastructure, which would change rider acceptance patterns (or opportunities) stochasticity. A detailed comparison can be seen in the next section comparing rider trip attributes.

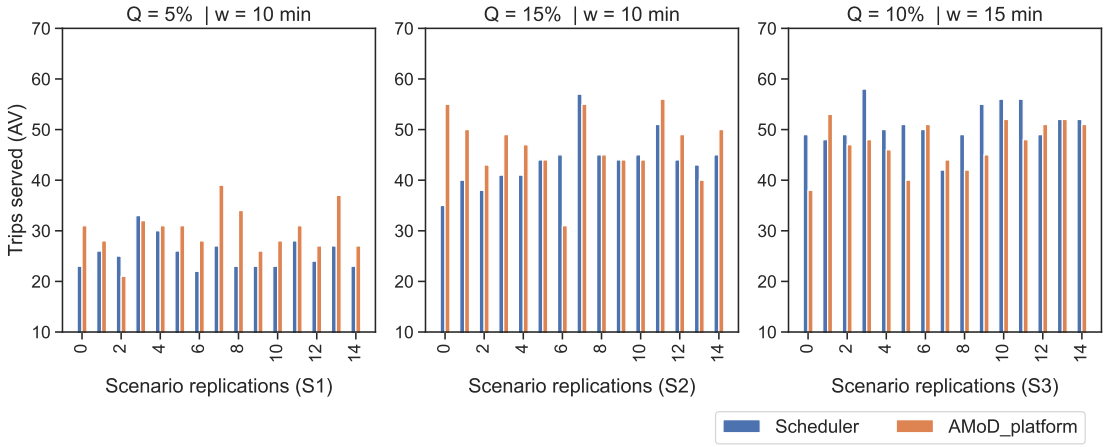


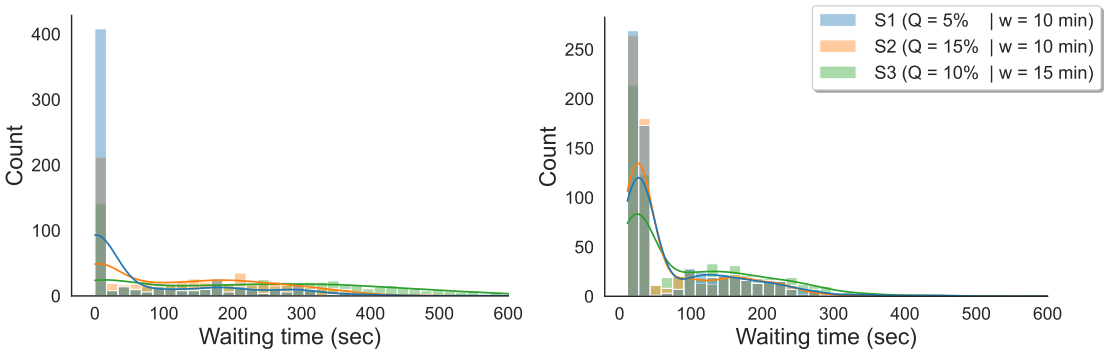
Figure 6.3: Request served in scheduler versus AMoD platform (multiple scenario replications)

Level of service distributions

Ridesplitting passengers experience both dynamic waiting and detouring to share rides with other passengers. Figures 6.4 to 6.6 show the distribution of riders' waiting, detour, and total additional times incurred in modeling by the scheduler and AMoD platform. Before discussing each figure in detail, all three figures show the presence of uncertainty and inequity among all ridesplitting riders for their experienced trip attributes. While the riders do state the maximum waiting and arrival times beforehand, their actual trip attributes are uncertain till the completion

of the trip, and the travel utility variate among fellow riders. Therefore, we proposed a utility-based pricing compensation method in chapter 4 which can help add both certainty and equity in rider travel utilities through dynamic compensations (results are discussed in section 6.3.4).

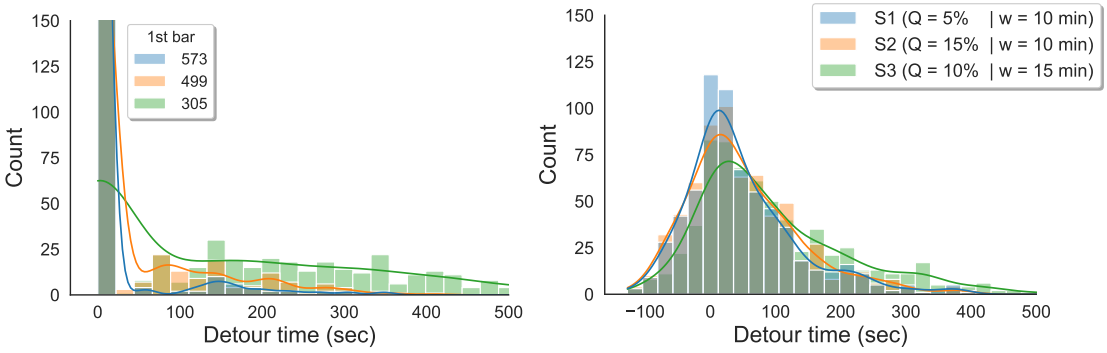
Figure 6.4 compares the waiting time distribution from the scheduler and the AMoD platform for all three demand scenarios. Note that in comparison with S1 and S2, the waiting times in the S3 scenario have more variance due to higher waiting time flexibility. Whereas, compared to the scheduler, waiting times in the AMoD platform are slightly more distributed due to traffic model stochasticity.



(a) Scheduler **(b)** AMoD platform

Figure 6.4: Comparison in waiting times distribution from scheduler and AMoD platform

Figure 6.5 compares the detour time distribution from the scheduler and the AMoD platform for all three demand scenarios. Both the figs. 6.5a and 6.5b are scaled equally to better compare the distributions. Moreover, to compute the detour times in the AMoD platform, the ridesplitting vehicle type is set as AV and the riders' in-vehicle times are subtracted from direct trip times taken from modeling a private human-driven vehicle or HV (details in section 6.2.1.3).



(a) Scheduler **(b)** AMoD platform

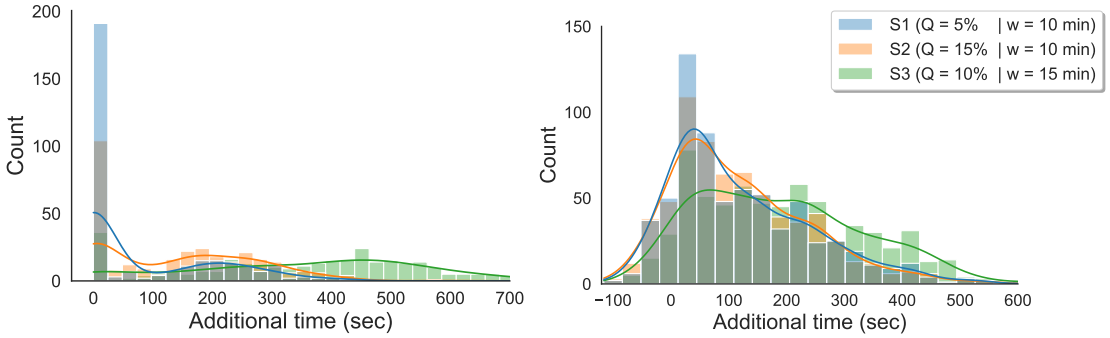
Figure 6.5: Comparison in detour times distribution from scheduler and AMoD platform

The first prominent difference is that the scheduler setup clearly evaluates travel times analytically without any dynamic traffic influence. Therefore the large set of direct trips (without any detouring) shows minimum detour. Whereas in the AMoD platform, even the AV and HV direct times vary due to vehicle driving behaviors and dynamic traffic congestion (ridesplitting

trips are offset by the waiting times). Therefore the detour shows a normal distribution with a mean close to zero. Rest, the detours for riders experiencing ridesplitting show somewhat similar patterns.

Comparing different demand scenarios, it is evident that both S1 and S2 scenarios have similar variance due to the same waiting time flexibility regardless of the size of attracted demand. Whereas the S3 scenario shows higher variance with more frequency of larger detours, indicating that higher waiting time flexibility allows higher detouring and helps achieve better ridesharing occupancy. As also indicated in section 7.1.2.3, it suggests that detouring flexibility is less influential than waiting time flexibility to trigger ridesplitting.

Finally, fig. 6.6 compares the combined additional times, i.e., the combination of waiting and detour times, for all three demand scenarios and the scheduler versus AMoD platform setups. Similar to fig. 6.5a, the distributions of all three demand scenarios in fig. 6.6a clearly differentiate the effect of change in demand and waiting flexibility. For example, the ridesplitting rides incurring additional times are higher in S2 and S3 than in S1 due to higher available demand, whereas the total ridesplitting rides (that incur additional times) and average additional time are much higher for S3 than other two scenarios due to higher available waiting flexibility.



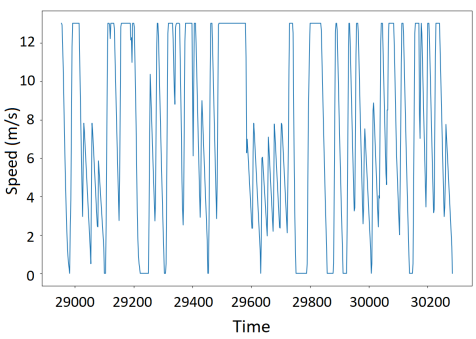
(a) Scheduler **(b)** AMoD platform

Figure 6.6: Comparison in additional times distribution from scheduler and AMoD platform

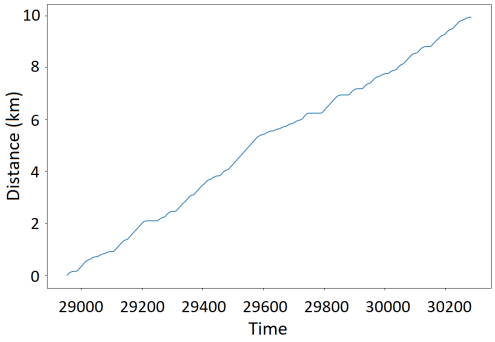
Vehicle profiles

Microscopic traffic models simulate individual vehicles using driving behavior models that dictate the vehicle driving decisions based on the interaction with neighboring vehicles and the infrastructure. For example, Figure 6.7 shows a sample of multiple temporal profiles of a randomly selected ridesplitting service vehicle depicting its microscopic driving motion. Both the speed and acceleration profile plots depict the detailed vehicle motion through the network, making multiple service stops and consistent speed changes interacting with neighboring traffic. Note that modeling such microscopic driving behavior allows replicating realistic (stochastic) information, e.g., vehicle travel times, service operations, and modeling traffic emissions.

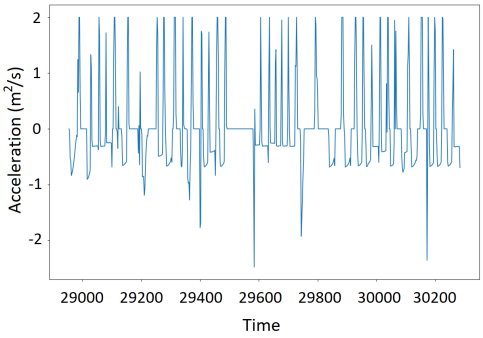
Figure 6.8 shows a sample ridesharing occupancy profile of the AMoD ridesplitting fleet for a randomly selected scenario. The plot depicts different occupancy levels for each vehicle during the simulation runtime. Note that a vehicle occupancy of up to 6 riders is achieved during the service operations, whereas some vehicles have much higher vehicle occupancy than others. Later, in sections 6.2.3 and 6.2.4 we further utilize these occupancy profiles of the ridesplitting service fleet to understand the effect of demand and waiting flexibility on triggering high occupancy ridesharing and will compare the actual resulting benefits at different occupancy rates.



(a) Speed-time profile



(b) Distance-time profile



(c) Acceleration-time profile

Figure 6.7: AMoD vehicle travel profiles in AMoD platform (randomly selected)

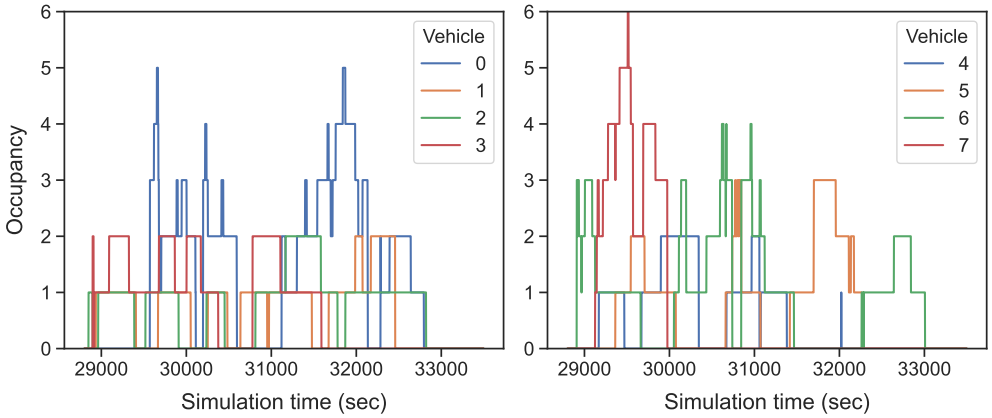


Figure 6.8: AMoD fleet occupancy profile

Different driving behaviors

Microscopic traffic models are suitable for modeling different driving behaviors. This behavior can vary by vehicle types, i.e., passenger, freight, or emergency vehicles, by automation, i.e., AV versus human-driven vehicles, or to represent specific driving characteristics like varying aggressiveness, response rates, and crossing or merging gap acceptance. Figure 6.9 compares the influence of the two driving behaviors used in this experiment, i.e., AV and PV (see table 6.1), using the detour distribution plots. Note that since the detours are computed by subtracting ridesplitting in-vehicle times with the direct trip time by private transport (PrT), both ridesplitting and private vehicle can be either AV and PV, resulting in a minimum of three comparison scenarios.

The most prominent difference is in the AV — PV scenario, where the ridesplitting vehicles are AVs. The detours seem to have a shift increase, indicating that the AV driving behavior results in slower or cautious driving with higher travel times. In comparison, the other two scenarios behave somewhat similarly since they compare similar driving behaviors for both ridesplitting and PrT. Note that these results are only subjected to the current case study and rely on the mentioned resources of driving parameters for both AV and PV. However, the fig. 6.9 does show a use case of using the developed microscopic AMoD platform for studying driving behavior impacts.

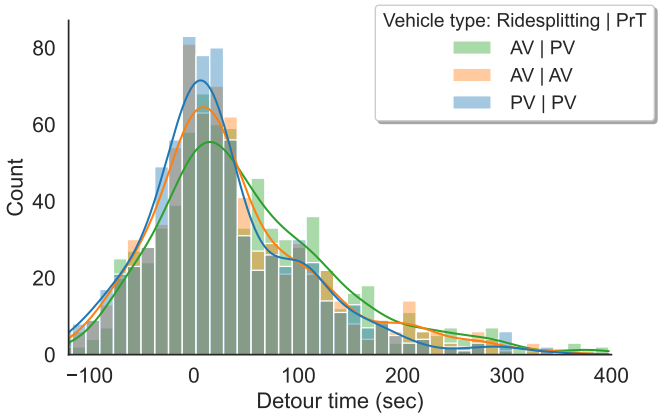


Figure 6.9: Comparison of three different vehicle type scenarios on detours

6.2.3 Occupancy analysis

Ridesplitting extends the on-demand service concept by dynamic ride-matching and detouring to achieve better ridesharing occupancy and offer a cheaper travel option. This section compares the effects of three different demand scenarios upon ridesharing occupancy. Below, we discuss three plots: average fleet occupancy, occupancy percentile, and share of requests served at different occupancy levels.

Figure 6.10 shows the vehicle-wise average occupancy of the AMoD fleet sorted in a descending order (averaged for all scenario replications). Note that the S3 scenario has higher average occupancy compared to the other two scenarios, especially compared to the S2 scenario, which has higher attracted demand and an equal amount of requests served (fig. 6.3). This is because S3 has higher waiting time flexibility than S2. Therefore, it can be concluded that ridesharing occupancy is more sensitive to waiting flexibility than attracting higher demand. However, only specific to the used case study because the effect of both demand and waiting is subjective to the service network shape and geometry.

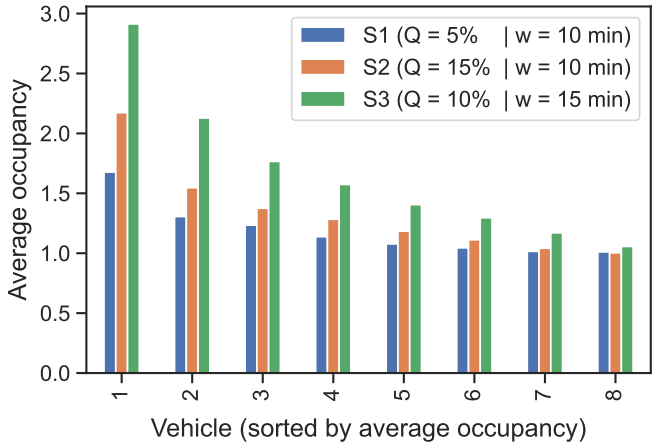


Figure 6.10: Average vehicle occupancy of AMoD fleet (sorted by descending order)

Figure 6.11 shows the fleet occupancy percentiles for all three demand scenarios (averaged of all scenario replications), depicting the time percentage of different ridesharing occupancy levels. Even in the conservative S1 scenario, occupancy of up to 5 riders is achieved. Whereas, S3 scenario, with the highest waiting time flexibility, occupancy of up to 8 riders is achieved (both are subject to the stochastic sequential availability of ridesharing opportunities). Moreover, similar to average occupancy results (fig. 6.10), the S3 scenario shows larger percentiles for ridesharing occupancy than S2 due to higher waiting flexibility, i.e., 53% of the service time with ridesharing (¿2 riders) against 25.2% in S2 scenario.

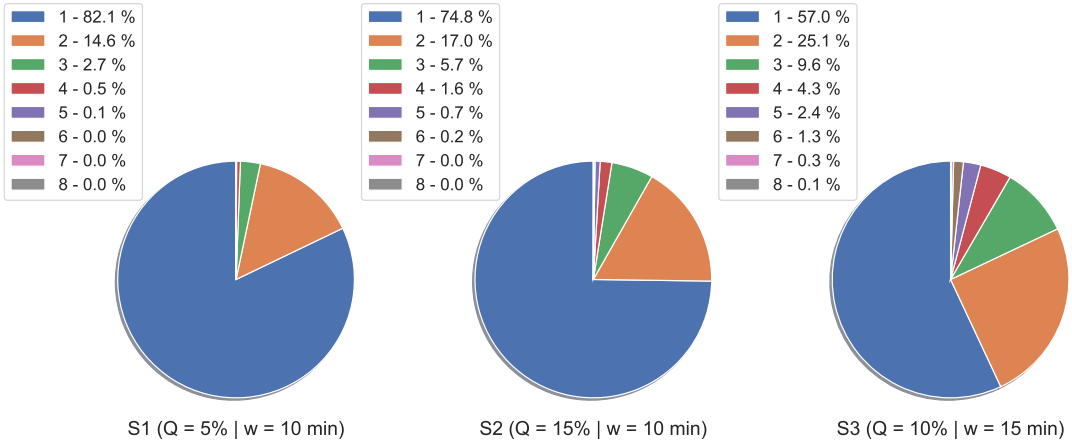


Figure 6.11: AMoD fleet occupancy percentiles for all three demand scenario

Another interesting plot is shown in fig. 6.12 that depicts the share of riders served against different average service vehicle occupancy levels. First, note that the average vehicle occupancy differs among all three scenarios, where the average occupancy of almost all vehicles in S1 is up to 2 riders, and for S2 and S3 scenarios, the average vehicle occupancy goes up to 3 riders and 5 riders, respectively. Further, the plot shows the share of riders served by varying average vehicle occupancy. For example, for S2 and S3 scenarios, about 11.5% and 22.4% of riders are

served by vehicles with above 2 average vehicle occupancy. Note that higher percentages of high vehicle occupancy are generally desirable to achieve better profits and benefits from ridesplitting (discussed later).

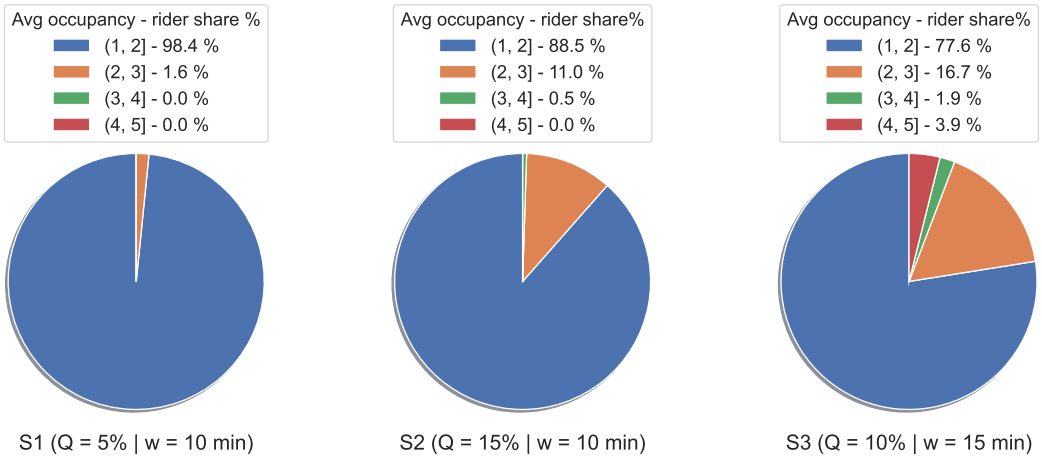


Figure 6.12: Share of riders served by average vehicle occupancy for all three demand scenarios

6.2.4 Ridesplitting benefits

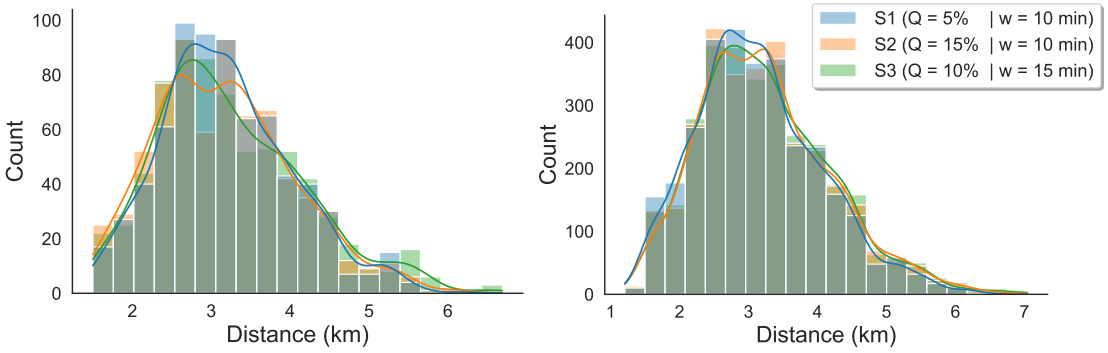
Ridesplitting services are becoming popular due to their dynamic sharing nature that promises to reduce traffic and environmental imprints alongside improving passenger affordability for on-demand travel. Serving riders with shared rides provide saving in terms of vehicle kilometers traveled, emissions, and energy. However, it is also important to understand if ridesplitting services achieve more benefits against, e.g., higher demand, waiting flexibility, or ridesharing occupancy. Therefore, this section explores the benefits against the three demand scenarios (table 6.2) and two vehicle types (table 6.1). Further, it also explores the relation of ridesplitting benefits with different occupancy levels.

Trip length distributions

Figure 6.13 shows the trip length distribution plots for all trips attracted and served under the three different scenarios. For both the trip attracted and served, the trip lengths range between 2 to 7 km, while trips between 2.5 to 3.5 km show the highest frequency. Note that the trip length distributions are well aligned for all three scenarios and are also similar between the requests attracted and served. It shows the absence of almost any bias based on trip length for the request acceptance.

Vehicle kilometers saved

Figure 6.14 shows a scatter plot for vehicle kilometer (VK) saved against the rider requests served for all three demand scenarios and subsequent scenario replications. Note that all scenarios show a linear relationship between the VK saving and the request served. For example, both S1 and S2 seem aligned, showing that the increase in attracted demand linearly increases both the rider request served and VK saved. Whereas S3 replications show a shifted increase of VK saved against requests served, i.e., it has more VK saved for the equal number of requests served compared to S2. This depicts that higher waiting flexibility and ridesharing occupancy rates trigger slightly better VK saved.



(a) Trips served **(b)** Trips attracted

Figure 6.13: Trip length distributions for trips served and attracted

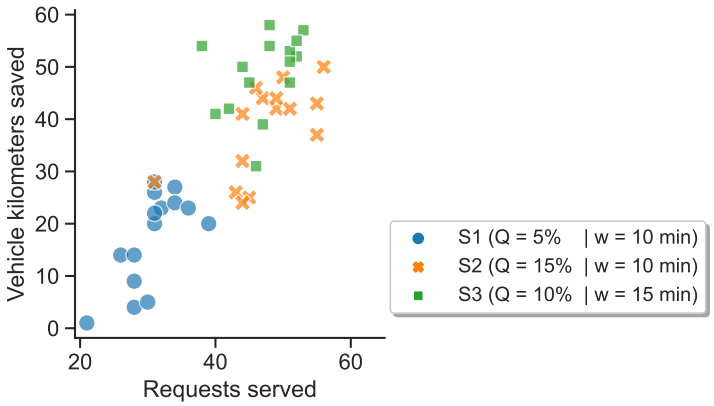


Figure 6.14: Vehicle kilometers saved for all three demand scenarios (multiple replications)

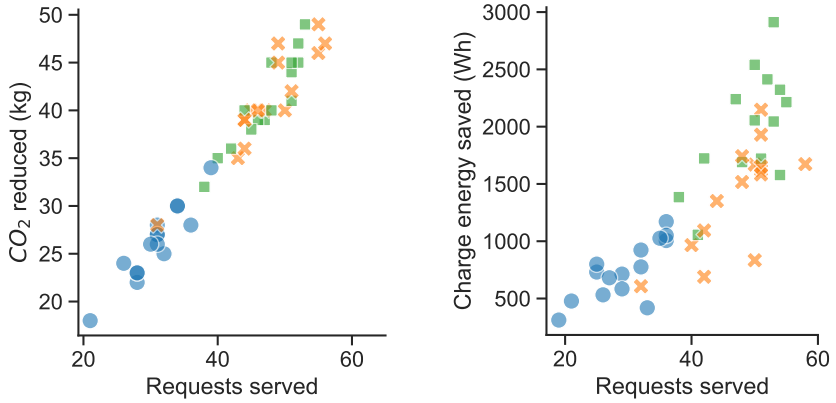
Benefits by varying vehicle types

Figure 6.15 shows the plots for emissions or energy saving in different vehicle type scenarios. As also compared in fig. 6.9, the three possible vehicle type scenarios include AV versus PV, AV versus AV, and PV versus PV usage for ridesplitting and private transport (PrT), respectively. Note that since AV vehicle type is considered electric (table 6.1) with zero emissions, the AV — AV scenario is compared by electricity or charge energy saved, whereas the AV — PV and PV — PV scenarios are compared by CO_2 emissions reduction.

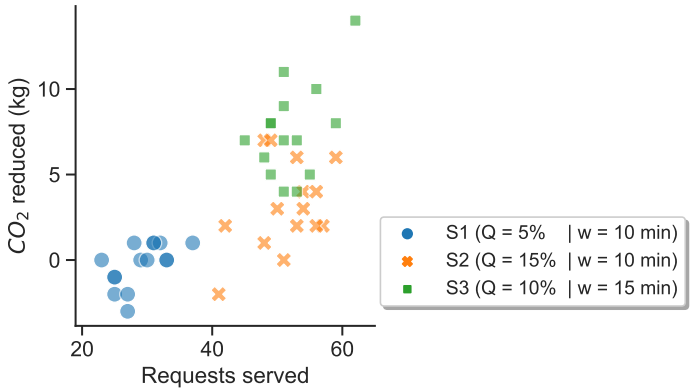
Figure 6.15a compares AV — PV scenario, i.e., ridesplitting service vehicle is electric AV and petroleum vehicle is for private transport. Therefore the total amount of CO_2 emitted by private rides of the corresponding rider requests is taken directly as CO_2 emissions reduced. The plot shows a linear relationship of emissions reduced against the rider request served without any prominent effect due to change in demand or waiting flexibility among three different scenarios.

Then, fig. 6.15b compares the AV — AV scenario, showing the amount of charge energy saved due to lesser VK traveled. Whereas, fig. 6.15c compares PV — PV scenario, showing the number of emissions reduced by ridesharing. Note that the trends in both figs. 6.15b and 6.15c are somewhat similar, i.e., S2 scenario replications have less amount of benefits compared to S3 with an equal number of requests served. It indicates the benefit of high occupancy ridesharing

triggered due to higher waiting flexibility present in S3. In contrast, the higher demand in S2 does increase the number of requests served but with a lesser amount of high occupancy ridesharing. Further, it is possible that for fig. 6.15a a similar trend is present; however less prominent due to the scale of values for CO_2 emissions saved by electric vehicles.



(a) CO_2 saved for AV versus PV **(b)** Energy saved for AV versus AV



(c) CO_2 saved for PV versus PV

Figure 6.15: Emission and energy benefits for different service and private vehicle types (AV - Electric autonomous vehicle, PV - petroleum vehicle)

Benefits at different vehicle occupancy rates

This section explores the per rider trip benefits for ridesplitting at different vehicle occupancy rates. These results are evaluated further from fig. 6.12 which shows the share of rider requests served at different levels of average vehicle occupancy. The resulting benefits from the request shares are divided by the number of requests to evaluate the benefit rates at varying levels of average vehicle occupancy. These benefit rates help depict the value of different ridesharing occupancies and the impacts of varying demand, waiting flexibility, and vehicle types. Since the three demand scenarios vary by their resulting ridesharing occupancies, benefit values for some occupancy rates do not exist against the respective demand scenarios.

Figure 6.16 shows benefit rates from AV — PV scenario, where both VK and CO_2 saved per rider trip increase slightly for both S2 and S3 scenarios with the increase in average vehicle oc-

cupancy. Surprisingly, the benefit rates also increase between S1, S2, and S3 at equal occupancy rates, showing that higher demand and waiting flexibility also improve the efficiency of ridesplitting ride combinations, i.e., minimizing detours. Note that CO_2 benefits are much higher since the comparison is between electric versus petroleum vehicles.

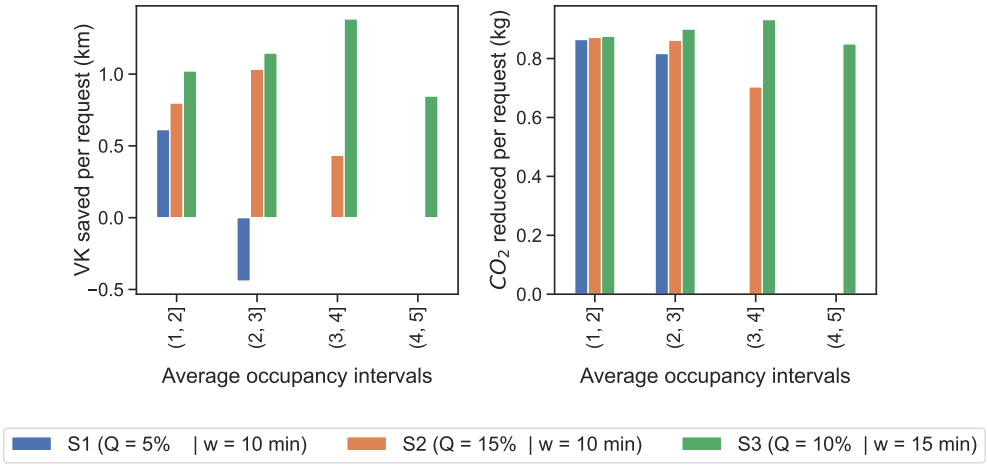


Figure 6.16: Per request benefit for AV ridesplitting versus petroleum car

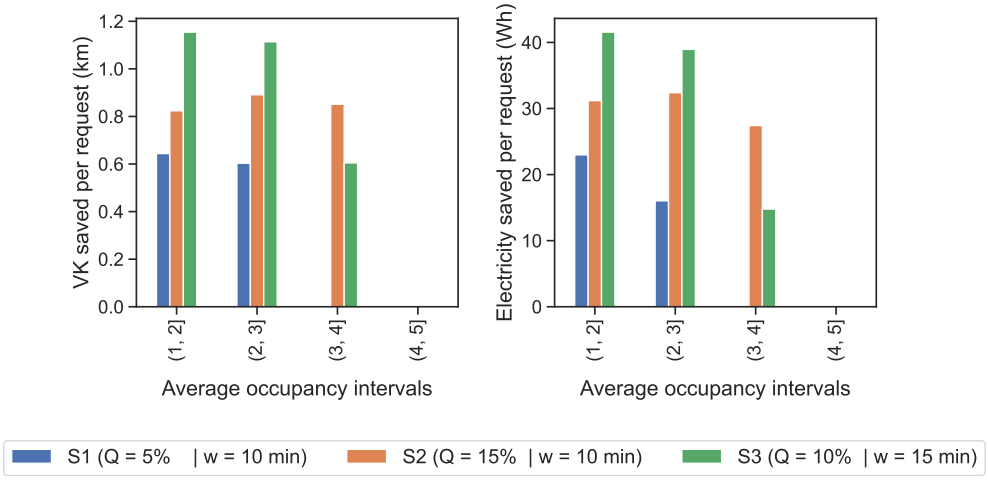


Figure 6.17: Per request benefit for AV ridesplitting versus AV car-sharing

Figures 6.17 and 6.18 show benefit rates from AV — AV and PV — PV scenarios, respectively. The above finding that the benefit rates increase sequentially between S1, S2, and S3 at equal occupancy rates due to higher demand and waiting flexibility is consistent in these results. Whereas, fig. 6.17 does not show any prominent trend of increase in benefit rates with the increase in average vehicle occupancy. However, the trend is present in fig. 6.18 for average occupancy of up to three riders. Note that since the shares of average vehicle occupancy above three riders are much lower (see fig. 6.12) their results can also have randomness (requiring more

scenario replications). Eventually, it can be suggested that the benefits per ride trip does not necessarily or significantly increase with higher average vehicle occupancies.

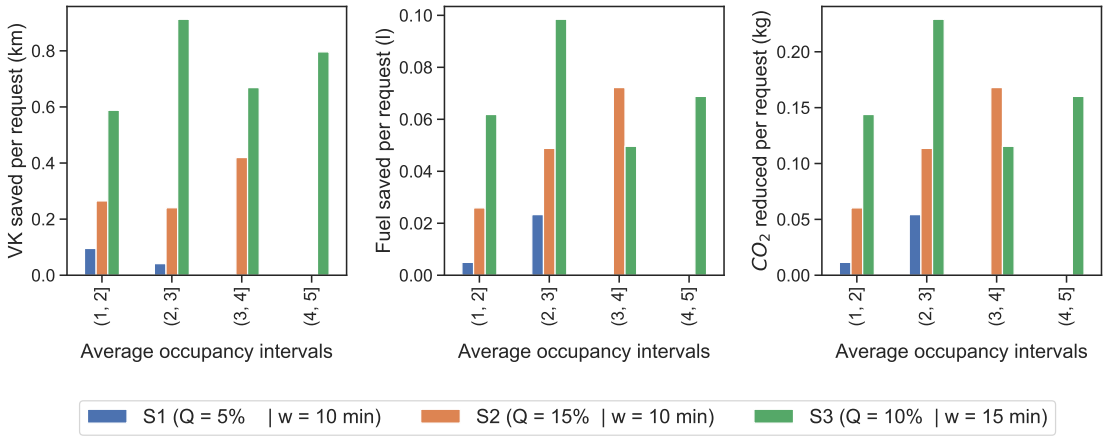


Figure 6.18: Per request benefit for petroleum van ridesplitting versus petroleum car

6.3 Ridesplitting market equilibrium and utility-based compensation pricing

This section evaluates the performance of ridesplitting market equilibrium (ME) and utility-based compensation pricing method (both proposed in chapter 4). For this purpose, the case study of Munich city (from chapter 5, fig. 5.4) is employed. Details on the experimental setup are discussed below. Further, two different operation objectives are defined to help evaluate the performance of the ME model under a base case of unified pricing. Finally, these results are used further to evaluate the performance and impacts of utility-based compensation pricing.

6.3.1 Case study setup

Network demand

The demand setup of Munich city is redefined by aggregating it into 20 zones with 380 OD pairs. Since both public transport and private transport are considered, we scale up the road traffic demand (private transport) based on the modal split of the Munich network. Furthermore, to mitigate the randomness of the ridesplitting market, we only consider the OD pairs whose travel demand is greater than 100 trips, restricting the service to 45 ODs with 7,726 trips in total (we focus on an off-peak period between 5 a.m. to 6 a.m.). The direct travel times and distance data are taken by averaging all trips of each OD pair, generated by SUMO simulation running at the mesoscopic level through the non-iterative dynamic stochastic user route choice assignment. The simulation outputs are also averaged over ten simulation replications to cater to the simulation stochasticity.

Mode choice

The mode choice setup is based on the data from the preference survey reported in chapter 4, which is also conducted in and around the Munich city region. The passenger preference coefficients (i.e., $\beta_t, \beta_w, \beta_r$, used to define the travel mode utilities, eq. (4.4)) are estimated using the preference survey data employing ordered logit model (Train, 2009). Table 6.3 lists the estimated

coefficients with p-values approximated to zero, indicating the estimation result to be significant with a confidence level of 99%. At the same time, the attributes for competing transport modes of public transport and private car are also set similar to what has been used in the preference survey (shown in table 6.4).

Table 6.3: Estimation of preference coefficients.

Coefficient	Value	Standard error	t-test	p-value
β_r (/Euro)	-0.589	0.0509	-11.6	0
β_t (/min)	-0.128	0.0139	-9.18	0
β_w (/min)	-0.113	0.0134	-8.46	0

Table 6.4: Attributes of public transport and private vehicles.

Mode	Waiting time (min)	Travel time (min)	Trip fare (Euro)
Public transport	10	$1.5t_i^d$	$0.8d_i$
Private car	3	t_i^d	$0.5d_i + 3$

Ridesplitting service setup

For ridesplitting, we assume the operating cost per vehicle per hour $\phi = 15$ Euro/h. Then, to calculate the ridesplitting market model parameters (A and B), we assume the average detour time and average waiting time of the ridesplitting services in Munich as 30% of the average direct trip time and 4 minutes, respectively (when the vehicle fleet size is $\hat{N} = 400$ and the unit price is $\hat{p} = 1.00$ Euro/km). Such a market leads to $A = 120.546, B = 0.026$ set parameters by simpler approximations, which are used in all hereafter experiments. In practice, one can calibrate the parameters with real operational data to characterize the market of interest (as discussed in chapter 4).

When generating the individual trips for calculating the compensations, the standard deviations of waiting time and detour time are set to be one-third of the means. We apply the following compensated utility function to calculate the utility after compensation in this study.

$$V^a = \begin{cases} V & \text{if } V > a \\ -\sqrt{2aV - a^2} & \text{otherwise} \end{cases} \tag{6.1}$$

As a result, the compensation function is given by

$$c_{i,k} = \begin{cases} 0 & \text{if } V_{i,k} > a \\ \frac{1}{\beta_r}(-\sqrt{2aV_{i,k} - a^2} - V_{i,k}) & \text{otherwise} \end{cases} \tag{6.2}$$

6.3.2 Modeling monoploy and social optimum scenarios

Ridesplitting services can differ by their operational objective depending upon the operator either being a private company or public transportation agency. Therefore, the market is generally

analyzed under two representative scenarios, extensively discussed in the literature, i.e., monoploy and social optimum scenarios. Each of the type operate as below.

The ridesplitting service is generally operated with for-hire drivers and vehicles, which differs it other ridesourcing types. A ridesplitting monopolist attempts to maximize its profit by optimizing the vehicle fleet size N and trip fare r. Profit is the difference between revenue and operating cost. The problem can then be formulated as

$$(\mathbf{P1}) \quad \text{maximize} \quad \Pi(N, r) = \sum_i D_i P_i r_i - \phi N \tag{6.3}$$

where ϕ is the operating cost of a vehicle in one hour, r is the vector of trip fare of all OD pairs, D_i is the demand for OD pair i, and P_i is the subsequent probability of choosing ridesplitting.

Social welfare also known as social surplus, equals the sum of consumers' and producers' surplus (Cairns and Liston-Heyes, 1996). Mathematically, the social welfare maximization problem can be constructed as

$$(\mathbf{P2}) \quad \text{maximize} \quad S(N, p) = \sum_i \int_0^{Q_i} F_i(x)dx - \phi N \tag{6.4}$$

where $F_i(\cdot)$ is the inverse of the demand function given in eq. (4.6) and Q_i is the ridesplitting demand.

6.3.3 Ridesplitting operations under unified pricing

Figure 6.19 shows the operation performance of ridesplitting services for the Munich case study under the distance-based unified pricing. It shows the iso-profit contours and iso-welfare contours in a two-dimensional space of vehicle fleet size (x-axis) and unit price (y-axis). Meanwhile, the monopoly optimum (MO) and social optimum (SO) are also marked in the figure. Note that these optimums can be evaluated (e.g., by gradient search methods) for varying market conditions like network, demand levels, and time of day. As Yang and Wong (1998) mentioned that a steady-state equilibrium solution for small fleet sizes might not exist in a network-based equilibrium model, we also observe an empty region in the lower left of the fig. 6.19.

Further, it can be seen that the optimal unit price of MO is higher than that of SO, while the optimum fleet size of SO is greater than that of the MO fleet size. This makes sense, as to benefit the public, the services should be operated more widely and cheaply. Also, as per Figure 6.19, both profit and welfare first increase with the unit price and fleet size and then decrease. Note that the joint influence of decision variables on profit and welfare is similar for higher unit price values. However, when the unit price is relatively small, the movements of the two contours become significantly different. It implies that the design of operation strategies should be explicitly dedicated to a market with particular consideration of its characteristics and objectives.

6.3.4 Benefits of utility-based pricing

To improve the equity and certainty of expected rider level of service (LoS) for ridesplitting services, we propose a utility-based compensation pricing method (in chapter 4). Both LoS and equity are represented by the mean and standard deviation of trip utilities, respectively. This section evaluates the market performance under varying compensation reference factors (CRFs) α on the basis of the optimum MO (i.e., $N^*_{mo} = 356, p^*_{mo} = 0.53$) and SO operation (i.e., $N^*_{so} = 623, p^*_{so} = 0.36$) strategies separately.

Figure 6.20a depicts the profit, social welfare, and mean of utilities under varying CRFs based on the MO solution with CRF on the x-axis, profit/welfare on the left y-axis, and mean utility on the right y-axis. Clearly, profit and welfare increase with CRF, ending up with the respective base

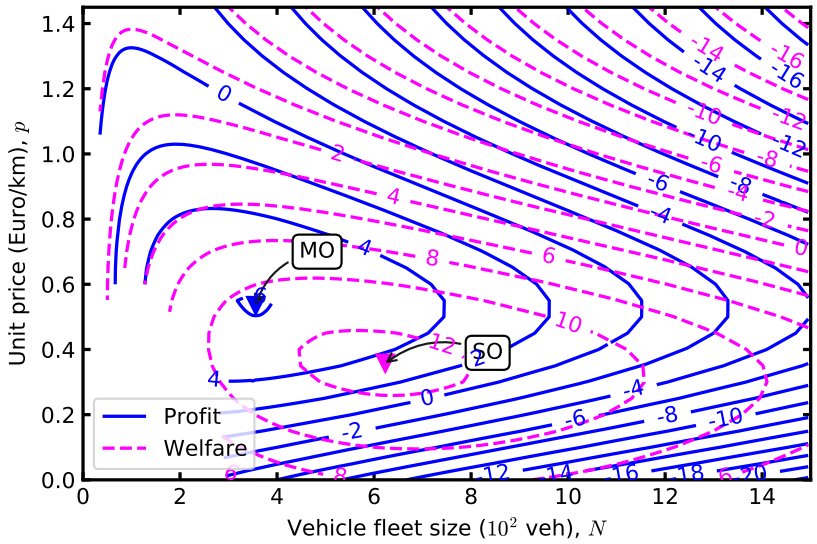
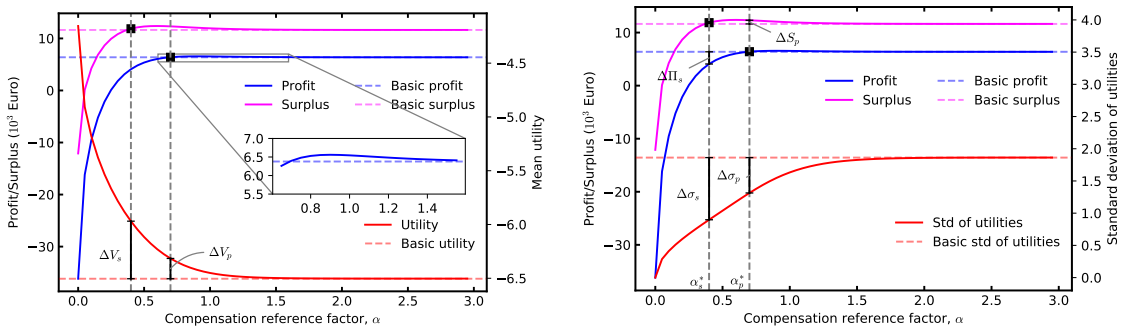


Figure 6.19: Profit and welfare in a two-dimensional space of vehicle fleet size and unit price.

values, i.e., the equilibrium values of the MO solution for the unified pricing scenario. Note that the peaks of the surplus and profit curves are higher than the base, implying that the proposed compensation pricing approach can benefit both profit and welfare if the objective of improving LoS and equity is disregarded. The maximum profit and welfare increase by 2.9% (from 6,378 Euro to 6,560 Euro) and 6.5% (from 11,630 Euro to 12,388 Euro), respectively. Let us denote the CRF of the first meeting points between the base profit (dashed blue) and the profit curve (solid blue) as α_p^*, the first meeting point between the base surplus and the surplus curve as α_s^*, and the corresponding improvement in mean utilities as ΔV_s and ΔV_p. Then, note that $\alpha_s^* < \alpha_p^*$, while $\Delta V_s > \Delta V_p$, depicting that optimum SO extends towards higher compensations allowing both better service utilities and overall service equity.



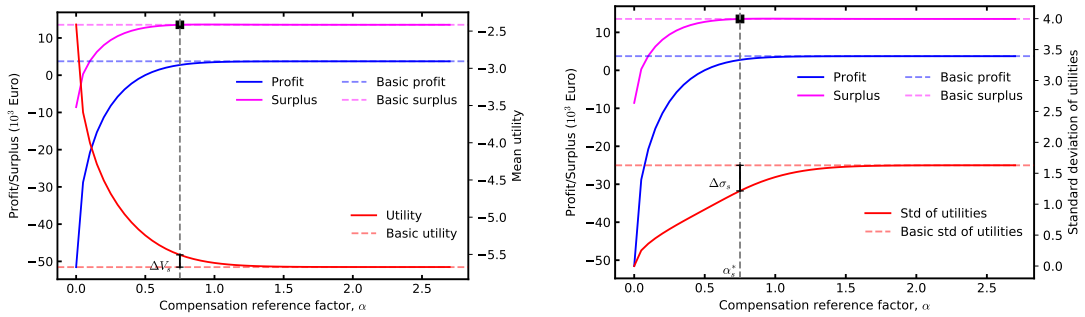
(a) LoS (mean of utilities) **(b)** Equity (standard deviation of utilities)

Figure 6.20: Performance of the utility-based compensation pricing method under different CRFs based on the MO operation strategy.

Further, as per Figure 6.20b, implementing compensation under α_s^* will lead to a reduction of profit by $\Delta\Pi_s$. It provides a reference for developing a smarter subsidy policy for ridesplitting operators that maximizes the LoS and equity of ridesplitting services without sacrificing any profit and social welfare. Even for the case without any subsidy, i.e., α_p^*, one can not only improve the LoS and equity (though less than that of α_s case) but also contribute to additional welfare of ΔS_p (5.9%, from 11,630 Euro to 12,320 Euro). Note that the profit and welfare also increase in the range between α_p and the second meeting point of the base profit and the profit curve. Further, table 6.5 provides the influence on the system endogenous variables when applying compensation pricing under the two mentioned CRF points.

Table 6.5: The performance of compensation pricing on the system endogenous under MO.

CRF	Ridesplitting demand	Seats occupancy rate	Waiting time	Detour time
α_p^*	10.3%	14.5%	15.0%	6.1%
α_s^*	33.6%	41.0%	48.5%	9.0%



(a) LoS (mean of utilities)

(b) Equity (standard deviation of utilities)

Figure 6.21: Performance of the utility-based compensation pricing method under different CRFs based on the SO operation strategy.

Figure 6.21 illustrates the effects of the proposed compensation approach in the SO operation scenario. Since α_p^* (meeting of base profit and the profit curve) does not exist, a subsidy is necessary for the operator to implement the compensation method. Otherwise, it will produce a profit loss compared to the unified pricing. Likewise, there is also nearly no increase in the maximum welfare (only +0.6%). Moreover, the improvement of LoS and equity under α_s^* is diminished compared to that in the case of MO. Therefore, we state that the proposed compensation pricing method is more beneficial for a market aiming at maximizing profit. However, to a certain extent, this also implies the inefficiency of a monopoly market.

6.4 Conclusion

This chapter presents two case studies of AMoD ridesplitting employing the microscopic modeling platform and the market equilibrium (ME) model. The AMoD modeling platform proposed in chapter 3 comprehensively models the ridesplitting service in a microscopic traffic model integrated with a dynamic DARP service optimization algorithm. Since microscopic models

model link-level vehicle driving and service operations in a dynamic traffic network, the case study assess its effects and depicts comparison results of the platform with simpler travel time-based network representation. The amount of rider request served (fig. 6.3) stochasticity vary among the two setups. However, the detour time distributions (fig. 6.5) show the effect of dynamic traffic and vehicle driving behaviors with well-distributed trip times even for non-ridesharing trips. Similarly, the vehicle distance, speed, and acceleration time profiles (fig. 6.7) and the comparison of automated versus human driving behavior scenarios (fig. 6.9) also show the efficacy of microscopic modeling. Finally, another advantage of employing microscopic traffic models is shown by estimating and comparing emissions from ridesplitting and private transport using the HBEFA3-based emission model for different vehicle types.

The AMoD ridesplitting platform case study also assesses the ridesplitting benefits and its performance sensitivity against change in demand and riders' waiting flexibility. Ridesplitting shows positive service benefits, including vehicle kilometers (VK), emissions, and energy saved that increase linearly with the amount of request served. Moreover, these benefits also have a linear relation with both demand and waiting flexibility. However, the increase in waiting flexibility results in higher benefits (figs. 6.14 and 6.15). Similarly, the emissions and energy savings are also modeled against multiple vehicle type scenarios to show the benefits of the use of electric vehicles against petroleum (fig. 6.15). Further, the case study also analyzes ridesplitting with respect to ridesharing occupancies, where first it is found that both increase in demand and waiting flexibility trigger better ridesharing. However, waiting flexibility better encourages higher ridesharing occupancies (figs. 6.10 to 6.12). Further, the benefits per ride trip analysis shows that the higher occupancies do not necessarily result in better benefits per request. However, at equal occupancy rates, the presence of higher demand and higher waiting flexibility generates more ridesplitting benefits, where again, waiting flexibility has a more significant effect (fig. 6.16). It is important to mention that the stated results are subjective to the case study setup, i.e., the service setup, network geometry, and the demand, all of which directly affect the ridesplitting service performance. Therefore the outcomes could vary between different service networks and demand levels. However, the case study does show the efficacy of the developed AMoD ridesplitting platform.

Moreover, the second case study assesses the ridesplitting market equilibrium (ME) and utility-based pricing methods using two different operational objectives of maximizing profits or social welfare. First, the ME model is utilized to generate a contour plot against varying unit prices and fleet sizes for the ridesplitting market in the Munich network (fig. 6.19). The plot includes two layers of profit and social welfare contours and also depicts the monopoly (profit) and social optimum states. Note that these optimums can be evaluated (e.g., by gradient search methods) for varying market conditions like network, demand levels, and time of day. Similarly, ME models can also help evaluate the impacts of different market strategies. Therefore it is also utilized to explore the performance of utility-based compensation pricing (modeled for monopoly and social optimum market states). For the monopoly case, the pricing method can significantly improve maximum profit and social welfare by 2.9% and 6.5%, respectively. Further, it also improves the mean and standard deviation (i.e., equity) of all riders' utility significantly by almost 8% and 50%, respectively (fig. 6.20). The pricing method also provide opportunities for smart subsidy schemes for ridesplitting operators which can maximize the certainty and equity without sacrificing any profit and social welfare margins (fig. 6.20b). However, for the social welfare case, external subsidies are necessary to implement the compensation method, which otherwise results in profit loss compared to simpler unified pricing (fig. 6.21).

7 AMoD ridesplitting impact assessment

Contents

This chapter further extends the ridesplitting service exploration and models a larger experimental setup to assess the relations and impacts of multiple service-related variables on passenger serviceability, occupancy, and related benefits under both exogenous and endogenous demand scenarios.

The content of this chapter has been partially presented in the following works, while part of the content is unpublished to date:

Qurashi, M., Jiang, H., & Antoniou, C., 2022. Microscopic modeling and optimization of autonomous mobility on–demand ridesplitting, *(Submitted)*

7.1 Ridesplitting assessment with exogenous demand

This section explores the ridesplitting service performance and impacts with fixed mode shares, where the demand attraction is not effected by changes in other experimental variables. Ranges of multiple supply and demand variables are explored to see their impact on passenger serviceability, ridesharing occupancy, and related benefits. Alongside plotting different plot types (scatter and bar plots) and regression lines to show the relation among the experimental variables and outputs, we also develop regression models using the Random Forest Regression (RFR) method to better visualize and quantify the impacts and relationships. The interpretation of the RFR models is shown using the SHAP value summary plots that help show both the amount of impact and positive or negative relationships of the independent variables with the dependent variable.

To describe the outline, section 7.1.1 discusses the overall experimental setup, while, section 7.1.2.1 shows a combined plot for all scenario results with the number of requests served against all experimental variables. Then, section 7.1.2.3 explores the effect of rider flexibility on the ridesplitting benefits and level of service, and section 7.1.2.4 compares the benefits against the loss in rider level of service. Similarly, section 7.1.2.5 explores the service environmental impacts, and finally, section 7.1.2.6 discusses the effect of pricing on request acceptance and ridesharing occupancy.

7.1.1 Experimental setup

The variables explored in this experimental setup are fleet size, waiting and detour time flexibility, profit, and mode share. To model the service area, we continue to use the case study of Munich city center region. The network covers the area of Maxvorstadt and Schwabing regions and is partially surrounded by inner ring of Munich (details are presented in section 6.2.1.1). Regarding ridesplitting service setup, the service penetrations rates are varied by using different fleet sizes between 6 to 14 vehicles (set based on the network size and scheduler optimization capacity), while the vehicle initial positions are set based on the OD matrix demand patterns, i.e., in zones with more trip origins (as discussed in section 6.2.1.3). Vehicle capacity is set to have a maximum of 8 rider trips sharing the service vehicle. Note that the current experiments don't model trip occupancy which is planned to be explored in future research.

For ridesplitting demand setup, the network demand is represented by 256 OD pairs (with 4851 trips within 8-9 am). Since we compare varying rider flexibility, short trips less than 2 km are not considered (leaving 2845 trips for mode choice). Since this set of experiments assess ridesplitting with exogenous demand, fixed mode share at three different levels are considered, i.e., 6%, 12%, and 18%, resulting in 170, 341, and 512 number of rider requests, respectively. Similarly, since pricing directly effects mode shares which are fixed, a single pricing unit cost of 1.5 €/km is used (only effecting scheduler's profit-based operations), while multiple profits ratios are explored separately in 7.1.2.6, ranging between 30 % to 100 % of the service cost.

Finally, we also explore the effects of riders' flexibility by varying d_p and w_p coefficients which define the individual rider detour and waiting time flexibility or pickup $[e_{r_i^p}, l_{r_i^p}]$ and delivery $[e_{r_i^d}, l_{r_i^d}]$ time windows for scheduling constraints (discussed in section 3.3 and 4.3). As per the literature related to DRT travel preference studies (Frei et al. (2017); Tsiamasiotis et al. (2021)), we use two levels for waiting/walking times (for short to medium trips), i.e., 5 and 10 minutes. Similarly, the rider detour flexibility d_p is set at three levels of 50 %, 100%, and 150 % of direct trip time (where travel time is sum of direct t_d and detour time d).

7.1.2 Results

7.1.2.1 Rider requests served

Figure 7.1 plots the amount of requests service by the ridesplitting service against different experimental variables. Taking each variable separately, the fleet size is shown by individual sub–graphs and their increase subsequently rise the amount of request served; mode share is shown by varying bar colors and also similar positive effect; rider flexibility which is represented by both waiting time and detour percentage, shows that increase in detour is more effective at lower waiting time preference (5 min), while at higher waiting time (10 min) the increase in detour preference get insensitive, especially with the increase in mode shares (since more ridesharing opportunities are available). The noticeable outcomes include, increase in detour flexibility is less effective at higher waiting times; higher waiting and detour times can allow more ridesharing (request served) at lower mode shares; increase in mode share has more prominent effects for larger fleet sizes. All results shown are taken as an average of multiple scenario replications (as also used in the previous experiment in chapter 6) and the black lines on each bar depict the variance among different replications (due to stochasticity in request and network information).

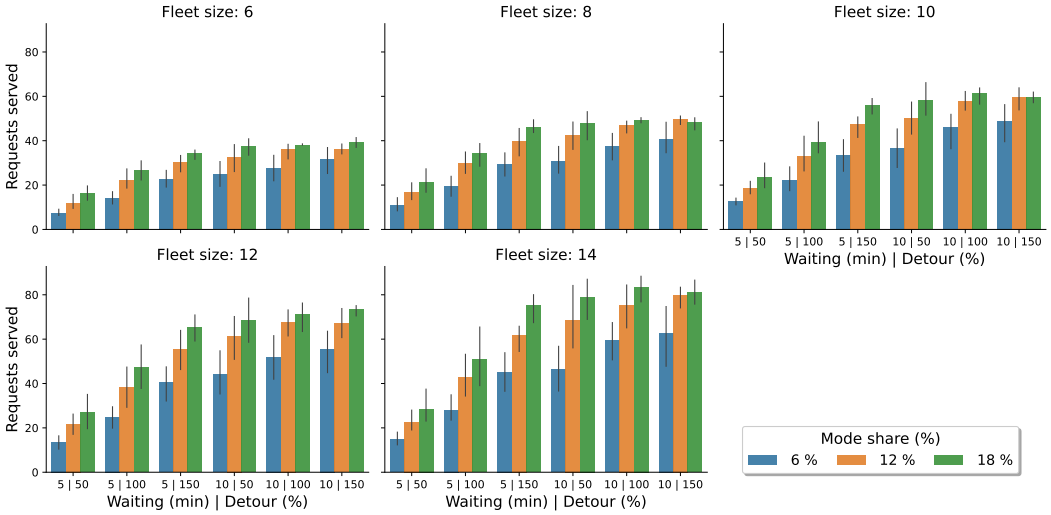


Figure 7.1: Requests served against different experimental variables

Figure 7.2 shows the SHAP summary plots that help interpret the Random Forest Regression (RFR) model developed for the number of riders served as the dependent variable. The independent variables include average vehicle/ridesharing occupancy, fleet size, waiting time flexibility, mode share, and detour time flexibility. Note that section 7.1.2.1 scales the mean amount of impact for each variable and section 7.1.2.1 shows their positive and negative relationship with the number of riders served. Note that all variables show positive impacts on the number of riders served and differ by the amount of impact. The most influential variables are ridesharing occupancy and fleet size since the occupancy directly defines the amount of ridesharing, and an increase in fleet size directly increases the amount of available supply. Moreover, the impact of detour flexibility is lower than that of waiting time flexibility, which is coherent with our previously mentioned findings. Another interesting finding is that the impact of mode share is lesser than waiting time flexibility and fleet size, for which it can be argued that although the increase in mode share does provide more ridesharing requests, their matching and acceptance possibility is influenced by the fleet size and waiting time flexibility. However, it is also noteworthy to men-

tion that these findings are subjected to the characteristics of the modelled case study and can vary due to changes in network size and geometry, traffic congestion and other similar variables.

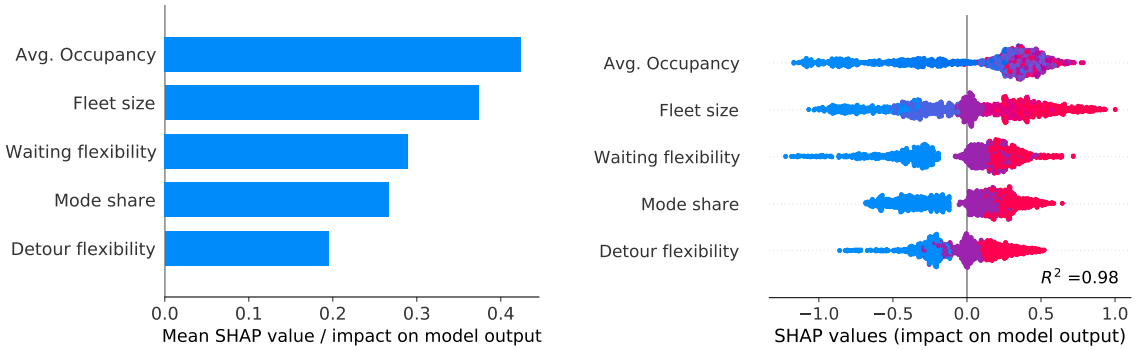


Figure 7.2: SHAP summary plots to show the impact of different service variables on amount of riders served

7.1.2.2 Ridesharing occupancy

Figure 7.3 shows the SHAP summary plots of the RFR model for ridesharing occupancy, which indirectly indicates the amount of ridesharing in the ridesplitting service. The main influential variables include the waiting time flexibility, the detour time flexibility, and the mode share, while the change in fleet size does not seem to have much influence. Note that the waiting time flexibility has much more impact on achieving higher ridesharing occupancy than the other two variables, while the detour flexibility impacts much lower and acts more complementary to the waiting time flexibility. The model findings of having the waiting time flexibility as the most influential variable are also coherent with the findings in chapter 6. It is also noteworthy to mention that in section 7.1.2.2 the variable values seem to be clustered with certain gaps because they act more as the categorical variables with multiple levels.

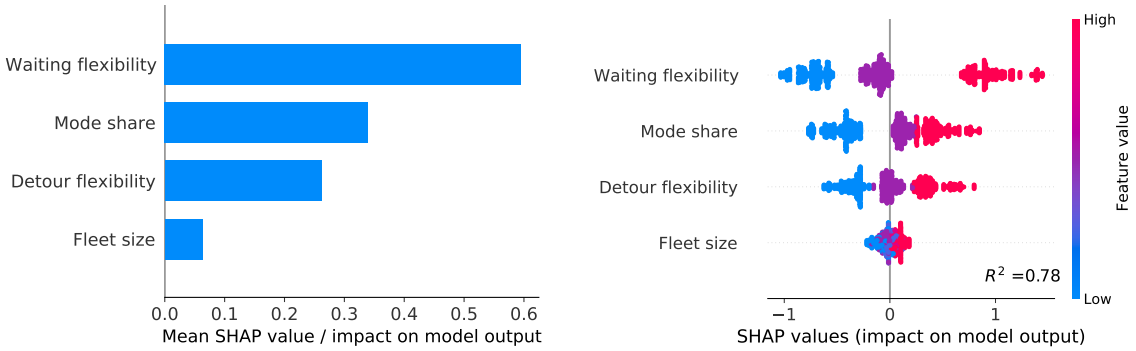


Figure 7.3: SHAP summary plots to show the impact of different service variables on ridesharing occupancy

7.1.2.3 Rider flexibility and ridesplitting

Figure 7.4 and 7.5 show scatter plots to represent the effect of varying rider flexibility at different mode shares and fleet sizes. Figure 7.4 plots the request served against ridesharing benefits i.e., vehicle kilometers (VK) saving), while figure 7.5 plots the request served against the total additional times incurred for all served requests. To distinguish among different variables and their levels: the mode shares have different styles; detour levels have varying colors; waiting times are divided by rows; fleet sizes have different columns. Note that the main influential variable for both VKT reduction and total additional times seems to be the waiting times' preference (hence divided by rows for better comparison). Similarly, figure 7.6 shows the SHAP summary plots of the RFR model for vehicle kilometers saving. Recall that, AMoD car–sharing have been reported to have a high increase in VKT against private cars due to empty kilometers traveled (Narayanan et al., 2020), while here the effect is opposite having a reduction in VKT against private cars but the amount of said benefits can depend on available demand, rider flexibility, and the scheduling algorithms (section 7.1.2.6).

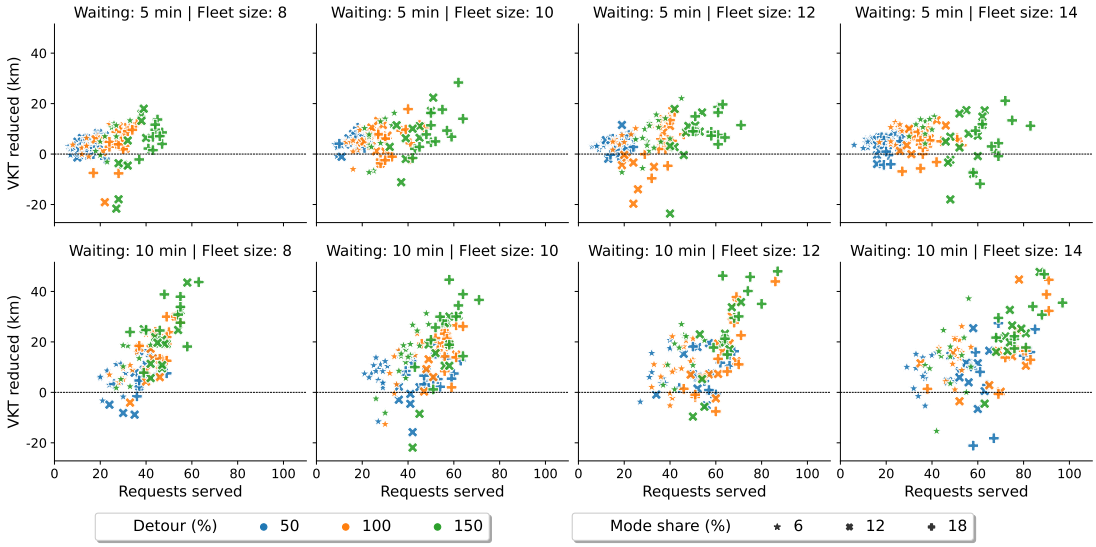


Figure 7.4: VKT benefits at varying time flexibility, request served, and mode shares

Considering figure 7.4, the results depict that at lower waiting time preference (5 min), the increase in detour flexibility, fleet sizes, and mode shares also increases the number of requests served but with little improvements in VKT reduction (i.e., slight increases in ridesharing). While, for the higher waiting time preference (10 min), the amount of VKT reduction increases significantly against the number of requests served. In other words, an increase in waiting time not only shifts the request served but also pushes for higher ridesharing. A similar finding is also evident from fig. 7.6, in which waiting time flexibility is far more influential than detour flexibility and mode share. This observed behaviour is expected since the higher waiting times increase the chances to avail of the ridesharing service, while detour flexibility defines the amount of possible detour given that the new riders have enough waiting times. Note that in this experiment, the mode share is much less influential, similar to detour flexibility, and both remain supplementary to enough waiting time availability for ridesharing benefits. Figure 7.6 depicts two more interesting findings; first that the fleet size, which, although has the least impact, negatively influences the ridesharing benefits (further discussed in section 7.1.2.4). Then, the

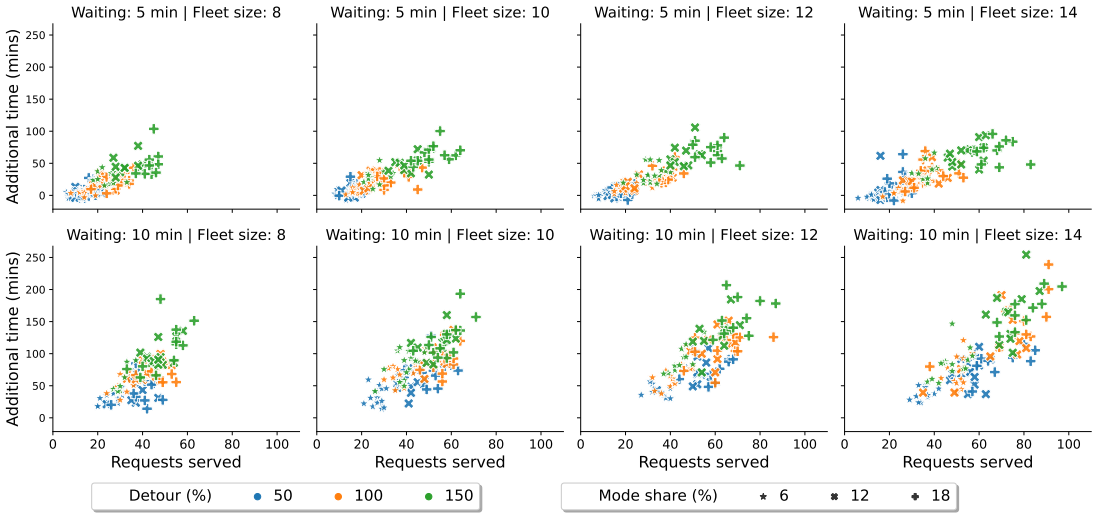


Figure 7.5: Total additional times at varying time flexibility, request served, and mode shares

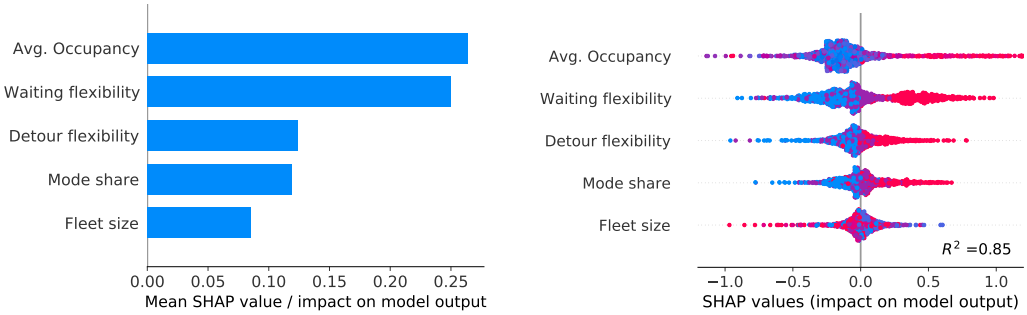


Figure 7.6: SHAP summary plots to show the impact of different service variables on vehicle kilometers saving

ridesharing occupancy also has the most impact on gaining the VK saving and is positively related.

Moreover, the results from figure 7.5 show that the additional times are influenced similar to VKT benefits. They remain low for lower waiting time while slightly increasing against all other variables, hence riders experience less amount of additional trip times. For higher waiting times, additional times increase significantly (against same amount of request served), depicting the influence of more ridesharing. Note that, since the request served increases with the increase in fleet size, the additional times are also increasing simultaneously, while the VKT benefits seem rather constant. This also suggests that given fixed demand, the increase in fleet size dilutes the potential for exploiting ridesharing, or the overall loss in rider level of service is increasing without additional ridesharing benefits.

7.1.2.4 Level of service against VKT benefits

Figure 7.7 shows the comparison between the level of service lost for AMoD ridesplitting riders (as additional trip times) against the ridesharing (VKT) benefits for different rider flexibility. The results are combined for different mode shares and fleet sizes, while segregated for preference

in detour flexibility (sub-graphs) and waiting time (by color), since they have more significant influence (as in figure 7.4 and 7.5). Note that all regression plots linearly increase and both variables are positively correlated i.e., the increase in additional times do increase the VKT reduction. But, also note that the both increase in waiting and detour flexibility result in steeper slopes (the effect of higher detour is less for higher waiting times). Overall, these results illustrate the importance of rider flexibility for ridesharing, i.e., if riders are willing to show more flexibility, higher benefits can be attained at lower rider time costs. However an important aspect is that this could add more inequity in individual trip level of service (since the trip can be served with wider time flexibility, but only when necessary), therefore stronger dynamic pricing strategies as the utility-based compensation pricing (covered in chapter 4) are beneficial to add equity among the riders through price compensations.

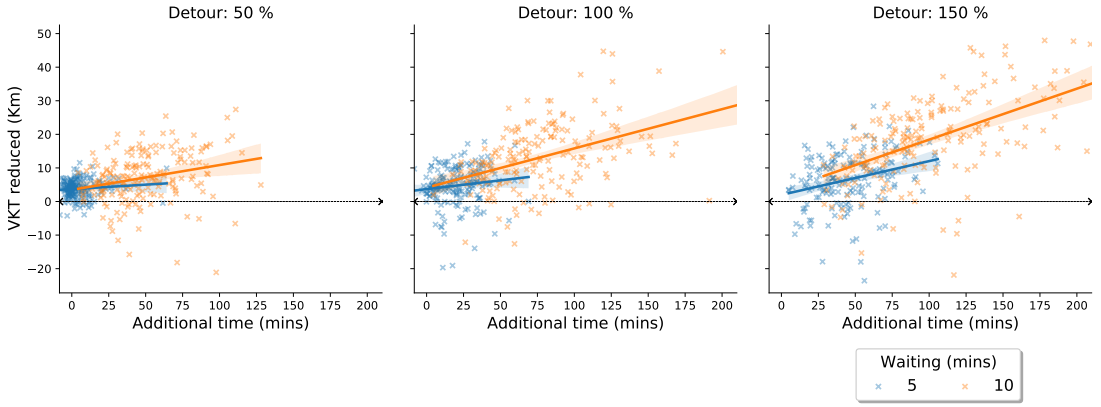


Figure 7.7: Vehicle kilometers reduced against rider additional times

Similar results as of fig. 7.7 are shown in figure 7.8 which are plotted differently to analyze the effect of different fleet sizes. These results show a similar effect as discussed in section 7.1.2.3, i.e., the increase in fleet size dilutes the potential of ridesharing opportunities, showing decreasing slopes of regression lines for the increase in VK savings against additional times.

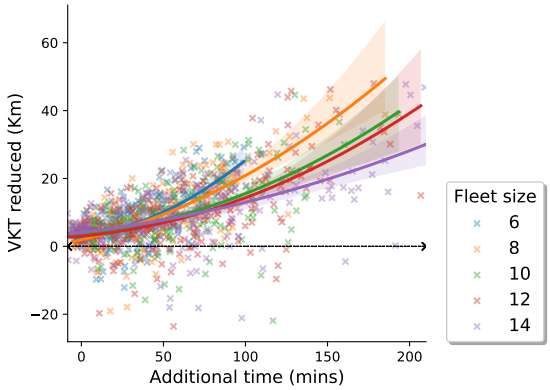


Figure 7.8: Vehicle kilometers reduced against rider additional times

7.1.2.5 Environmental impacts

Emissions from road transport account for one-fifth of Europe's greenhouse gas emissions with 75% from rider transport (Fontaras et al., 2017). While, the traffic-related ridesharing benefits

can be measured by vehicle kilometers (VK) savings, figure 7.9 shows the environmental impacts of AMoD ridesplitting in terms of CO_2 emission saving. These emissions are modeled by using the HBEFA3 based model (INFRAS, 2022) for regular petroleum vehicles (as in chapter 6). Figure 7.9 compare the CO_2 emission saving with the varying number of request served, the results are combined for different mode shares hence showing the effect of an increase in demand attraction (which directly translates in the requests served), while the regression plots show the performance of different fleet sizes at different rider flexibility. Similar to previous findings, significant improvements are seen for higher waiting time preference and higher detours, while overall the rate of per request benefits decrease with increase in fleet sizes although serving more number of requests.

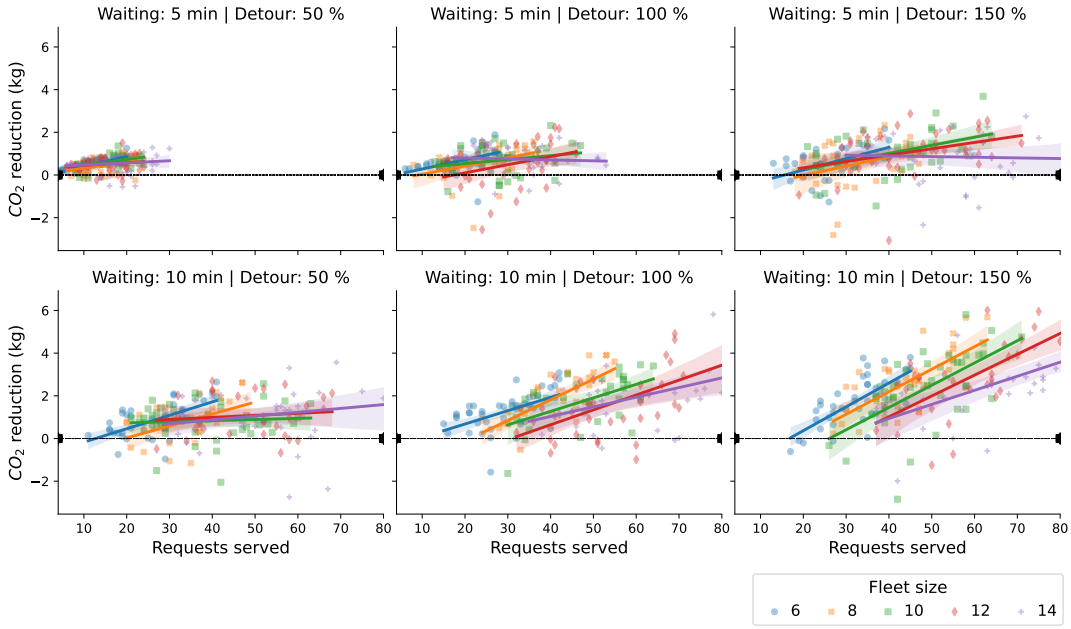


Figure 7.9: CO_2 emissions saving against varying rider flexibility

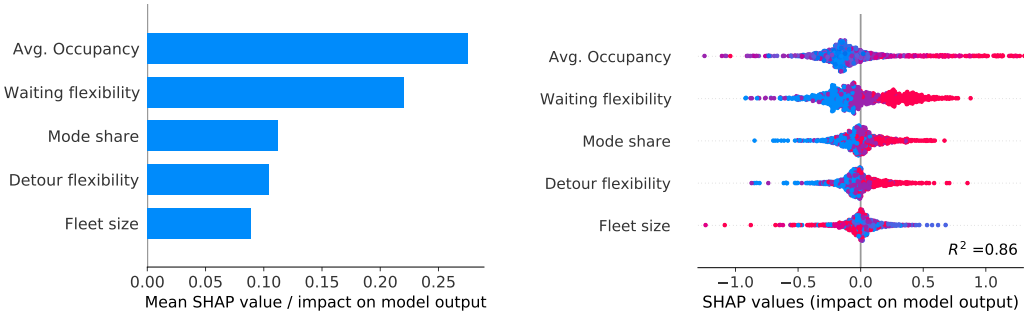


Figure 7.10: SHAP summary plots to show the impact of different service variables on CO_2 saving

Figure 7.10 shows the SHAP summary plots of the RFR model for emissions saving, in which the impacts and relations of all independent variables are similar to that found in fig. 7.6 (for VK saving). The findings include, waiting time flexibility is much more influential than detour flexibility and mode share; fleet size, although with less impact, negatively influence the CO_2

savings; and the ridesharing occupancy has the most impact in gaining the CO_2 saving and is positively correlated with it.

7.1.2.6 Effect of profit shares

For exogenous demand scenarios, pricing doesn't effect the mode shares but it does effect the performance of the integrated scheduler (see chapter 3 for scheduler description) because it is an operator based optimization algorithm generally targeting to maximize profit. Hence, we explore different profit ratios over the fixed unit cost. Figure 7.11 shows the effect of varying trips pricing (i.e., profit % over the service cost) on the number of requests served and Vehicle Kilometer Travelled (VKT) reduced against the private car. The results from different mode shares and rider flexibility are combined to show box plots of the request served (left) and VKT reduced (right) at varying pricing and fleet sizes. Note that, it is evident (similar to figure 7.1) that higher profits increase the number of request served (more request acceptance), but effects the ridesharing benefits (VKT reduced) negativity, due to a reduction in the amount of shared rides. The effect occurs due to increase in acceptance of the incoming requests due to higher profit margin, while the algorithm reduce its consideration of riders' flexibility for potential ridesharing.

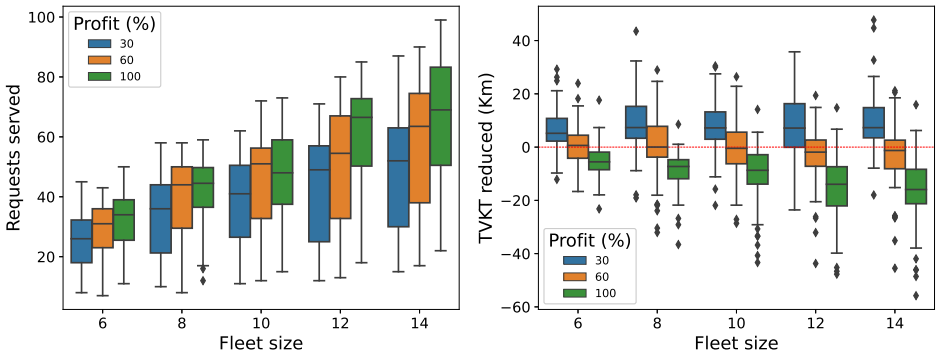


Figure 7.11: Effect of profit on ride–splitting

7.2 Ridesplitting assessment with endogenous demand

This section explores the ridesplitting service performance and impacts with endogenous demand modeling, where the ridesplitting demand is estimated using the trip-based mode choice method proposed in chapter 4. The ridesplitting mode shares are estimated as per the change in any other experimental variable. To describe the outline, section 7.2.2.1 shows a combined plot for the number of requests served at all different settings, while, section 7.2.2.2 discusses the effect of pricing and rider flexibility on service performance and ridesharing. Finally, to analyze the efficacy of change in rider flexibility and service cost, section 7.2.2.3 compares the VKT benefits against additional trip times.

7.2.1 Experimental setup

The variables explored in this experimental setup are fleet size, waiting and detour time flexibility, service cost, and profit. While, we continue to use the case study of Munich city center region

and the service setup with varying fleet sizes between 6 to 14 vehicles along with other settings discussed earlier in section 7.1.1.

Since this section models endogenous demand, the mode choice setup is based on the ridesplitting preference survey discussed in chapter 4. The assumed travel mode attributes for public transport and private car along with the resulting estimation of rider's preference coefficients is already mentioned in section 6.3.1 and therefore not repeated again. To model varying demand attributes within this experimental setup, we model two levels of service cost, i.e., 1.3 and 1.5 €/km, two levels of profit (30% and 60%), two levels of rider waiting times (5 and 7 min) and three levels of rider detour flexibility (50%, 100%, 150%). Note that, the mentioned values are also set considering the results from exogenous demand experiment, e.g., maximum waiting time preference is set to 7 min since 10 min preference overrides the influence of detour flexibility (see figure 7.1).

Figure 7.12 shows the requests attracted against all different sets of mode attributes, where the mode shares are mainly affected by riders' flexibility and service cost. Note that riders' flexibility highly affects the mode shares due to the conservative assumption of taking the rider trip time preference directly as the expected trip attributes.

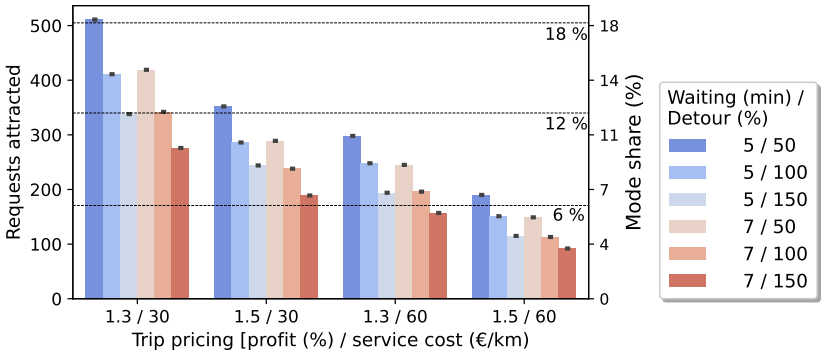


Figure 7.12: Demand attracted at varying service pricing and rider flexibility

7.2.2 Results

7.2.2.1 Rider requests served

Figure 7.13 shows the overall results for the AMoD ridesplitting services in terms of the number of requests served against all sets of scenarios (bar shows the average value and black lines show variance in scenario replications). Note that, compared to exogenous demand scenarios, both the overall request served and its variance among different scenarios is lesser due to the mode choice being an endogenous factor, where apart from pricing the higher riders' flexibility also significantly reduce the mode shares.

While analyzing each variable, increasing fleet sizes (varying by subplots) show a consistent increase in the number of requests served; increasing rider flexibility also consistently increases the number of requests served; higher service cost results in the lower request served due to the decrease in mode share; higher profit also reduces the mode share and results in the equal or lesser amount of request served among two different service costs. Note that, even higher profits do not result in an increase in the request served, but it still effects the ridesharing benefits negatively (see figure 7.14).

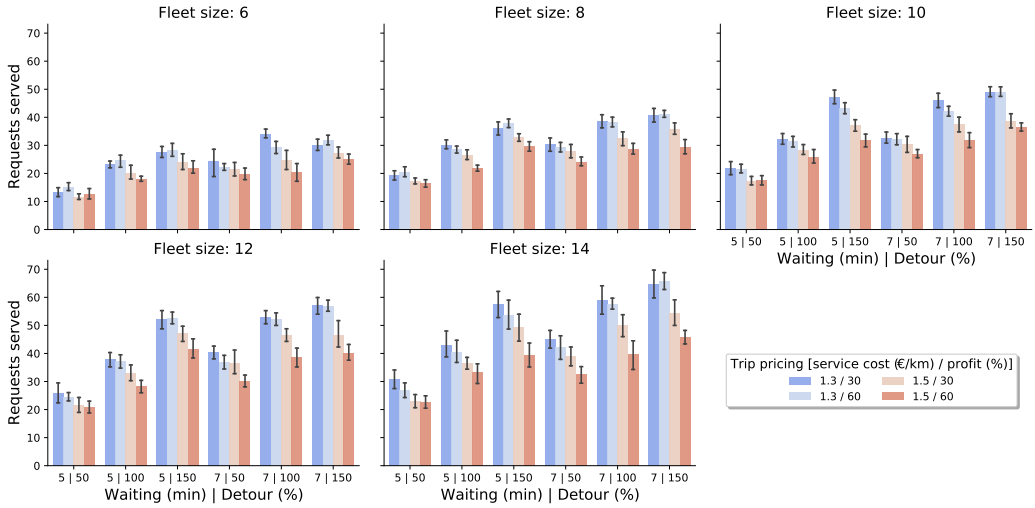


Figure 7.13: Requests served against different experimental variables

7.2.2.2 Effect of trip pricing and rider flexibility

Figure 7.14 shows the effect of the service cost and profit levels on the number of requests served and their ridesharing benefits through scatter plots. Note that a higher profit level clearly increases the VKT instead of reduction, indicating that the service vehicle accepts farther requests and/or incurs lesser ridesharing, however profit level 30 % results in positive VKT reduction in most scenarios. Furthermore, the difference between higher and lower service costs is also clear, since lower cost results in higher mode shares both the number of requests served and VKT benefits are higher. Note that, since only lower profit results in positive ridesharing benefits, we only use the lower profit scenarios in further results.

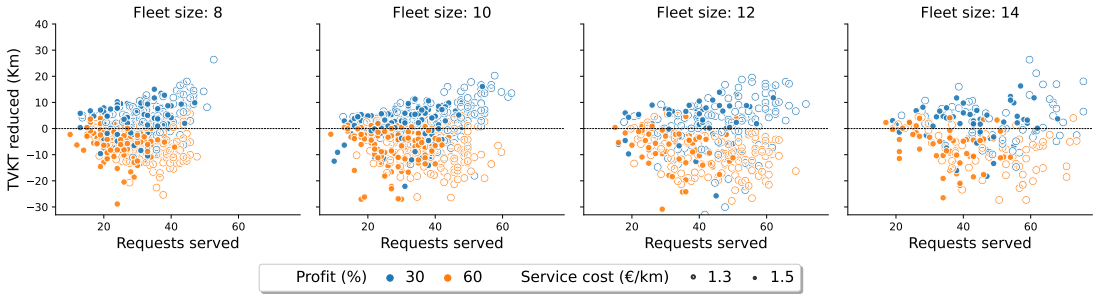


Figure 7.14: Vehicle kilometers reduced against varying service pricing

Figure 7.15 and 7.16 show the amount of ridesharing benefits and the additional times incurred against the number of requests served at varying rider flexibility. As per figure 7.15, it seems that an increase in waiting time doesn't result in a significant increase in ridesharing benefits but only increases the number of requests served (note that the increase also reduces the mode shares). While a similar effect is also evident for detour flexibility. Next, analyzing figure 7.16 which shows cumulative additional times, no change in slopes appears for different waiting times while the slope gets milder as the fleet size increases depicting more requests are served at a lesser amount of additional times.

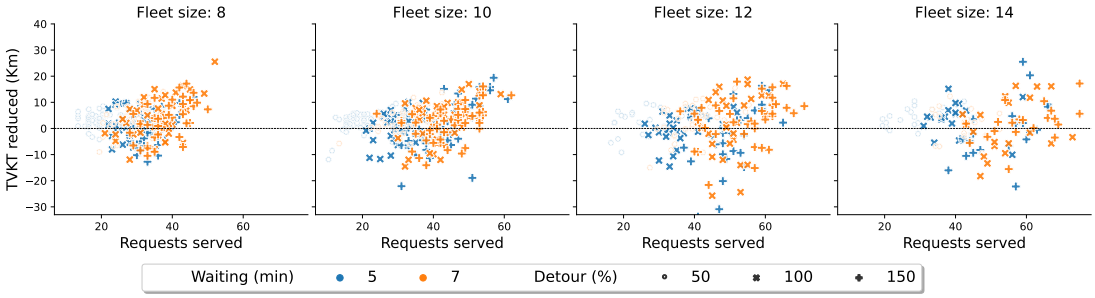


Figure 7.15: Vehicle kilometers reduced against varying rider flexibility

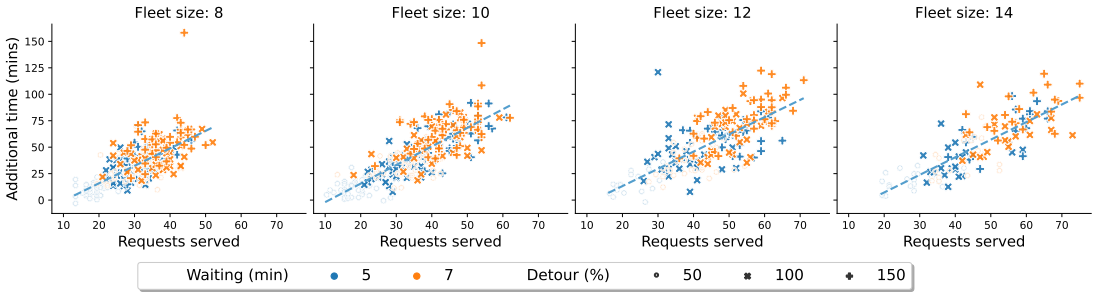


Figure 7.16: Total additional times against varying rider flexibility

7.2.2.3 Level of service against VKT benefits

Similar to the results from section 7.1.2.4, figure 7.17 plots the comparison between cumulative additional times and VKT benefits (but with the difference of segregating the service costs and analyzing the trend due to an increase in rider flexibility). Note that, the plots show that the increase in flexibility doesn't translate into better ridesharing given relative demand attraction and current mode choice setup. The effect of lower service cost (that results in higher mode share) is just adding more dispersion in scatter plot and a slight improving trend for some fleet sizes.

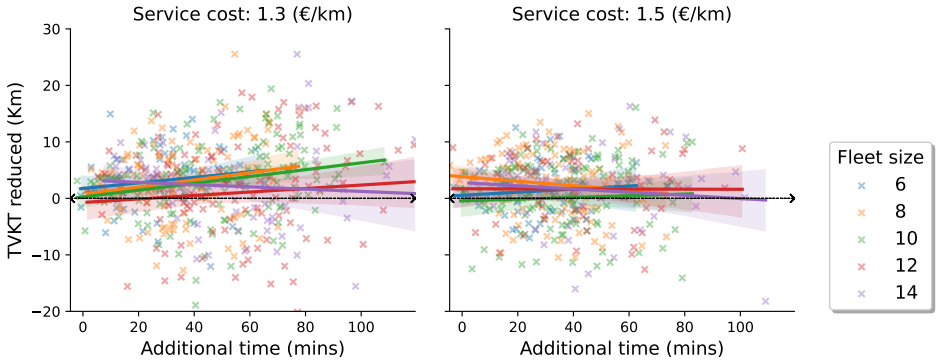


Figure 7.17: Vehicle kilometers reduced against rider additional times

Note that, although there are no significant improvements due to an increase in flexibility (other than resulting in more requests served, see figure 7.13) and AMoD ridesharing service seems rather ineffective, but the results are subjective to the case study setup which has been simplified to demonstrate the usage of the developed platform. While future studies can focus on improvements that can help better define the mode choice setup.

7.3 Discussion

This chapter further explores the ridesplitting service to assess the impacts and relation of different supply and demand variables on ridesplitting service performance and benefits under exogenous and endogenous demand scenarios. The experimental setup uses the case study of the Munich inner city region. The exogenous demand experiment allows more independent exploration of the impacts from different levels of mode shares, fleet size, pricing, and riders' waiting and demand flexibility on ridesplitting occupancies and benefits. In comparison, the endogenous demand experiment models ridesplitting mode choice as a separate transport mode in competition with private car and public transport. Apart from exploring the impact of different fleet sizes, the demand attraction is subjected to varying riders' flexibility and pricing using the conservation assumption of considering riders' waiting and detour flexibility directly as the expected trip utility.

The exogenous experiment shows that the riders' waiting flexibility is the most influential variable in obtaining higher ridesharing occupancies and service benefits from the ridesplitting service (figs. 7.3, 7.6 and 7.10). Similarly, the increase in mode shares and detour time flexibility directly translate into higher rider trips served; however, they seem supplementary to waiting time flexibility for higher ridesharing benefits. Moreover, comparing riders' additional trip times and resulting ridesplitting benefits depict that higher service benefits can be attained at similar additional time costs when riders show higher trip time flexibility fig. 7.7. However, higher flexibility can also result in more inequity among riders. In the endogenous experiment, decreasing service cost attracts more demand and increases the request served and ridesplitting benefits figs. 7.13 and 7.14. While although the service achieves positive benefits for vehicle kilometres (VK) and emissions savings. These benefits do not increase with the increase in waiting and detouring flexibility and only translate into more requests served fig. 7.15.

The findings related to fleet size and profitability are common in both experiments. Where the increase in fleet size always translates into more requests served but somewhat decreasing the benefits figs. 7.6, 7.9 and 7.17. In other words, while serving an equal amount of demand, the increase in fleet size decreases the possible ridesharing opportunities. Since the service optimization objective is set for higher profitability (operator-owned service), the increase in profit margin results in higher rider request acceptance and trips served. However, it reduces the ridesplitting benefits because requests with lesser ridesharing potential are accepted given enough profit margin figs. 7.11 and 7.14.

8 Conclusion

Contents

This chapter concludes the thesis and provides the thesis summary, finding and shortcomings, and future research directions.

8.1 Summarizing research scope

The ridesplitting services have emerged much recently, showing invaluable market potential. However, they have had both successes and failures, posing different uncertainties for all stakeholders, including the operators, policymakers, and service users. The purpose of this thesis is to develop ridesplitting specific methods that improve service modeling, operations, and assessment and help reduce the service-related uncertainties for all stakeholders. To pursue our objective, we specifically seek methods to address the following different concerns stated with respect to the subsequent stakeholders.

- ***Operators:*** For operators, planning and managing ridesplitting is rather complex. They charge much less than other ridesourcing services while aiming to establish sufficient ridesharing occupancy through detours for sustainable/profitable operations. Therefore, the operators must cater to service operational complexity (requiring dedicated dynamic optimization) and its sensitivity towards the involved dynamic and stochastic information for demand and supply (Wang and Yang, 2019). Thus far the ridesplitting ridership is relatively low (Tu et al., 2021; Li et al., 2019c), and more than 70% similar startups have failed to establish (Currie and Fournier, 2020). Therefore, from the operator's perspective, it requires both robust models and efficient operational strategies specific to the service characteristics to better plan and manages the service.

  Given the above statement:

  - Chapter 3 proposes a comprehensive modeling framework to model AMoD ridesplitting in microscopic models with integration of dial-a-ride (DARP) optimization. It allows detailed modeling of network dynamics, link-based service operations, and incorporation of the stochastic network, service, and demand information in service optimization, resulting in efficient modeling and optimization of AMoD ridesplitting operations.
  - Chapter 5 solves practical implementation problems for large-scale model calibration using Principal Component Analysis, allowing it to tackle the higher dimensionality and non–linearity of traffic models in the absence of relevant historical estimates. Note that efficient dynamic demand estimation helps achieve better accuracy in both network congestion modeling and service demand estimation.
  - Chapter 4 proposes a market equilibrium (ME) model that specifically caters to ridesplitting characteristics, interpreting supply and demand interactions at the network level. ME allows to model an aggregated demand and market assessment for wider ranges of varying service attributes and strategies.
  - Chapter 4 proposes a simple and more practical trip-based demand modeling method for ridesplitting, exploiting its specific service characteristics (i.e., hard/explicit rider time constraints). The method helps remove the requirement of iterative simulations and allows much easier adaptability among most traffic simulators.
  - Chapters 6 and 7 does ridesplitting service exploration to help understand the value of ridesplitting specific traits, e.g., varying penetration rates, demand attraction, and passenger flexibility to achieve better profitability and higher ridesharing occupancies.

- ***Policymakers:*** For policymakers, the rapid emergence and popularity of ridesplitting pose many doubts that require deeper understanding. For example, i) can the ridesharing nature help solve the urban problem by reducing excess traffic volumes and emissions? ii) how would ridesplitting affect existing transportation systems? iii) is high occupancy

ridesplitting fruitful, i.e., what are its costs versus benefits? iv) what are the prospects of autonomous (AMoD) and electric ridesplitting vehicles?

Given the above statement:

- Chapter 7 explores the ridesplitting-specific benefits at varying penetration rates, mode shares, passenger flexibility, and service costs for both exogenous and endogenous demand scenarios to help understand the ridesplitting service impacts on improving urban traffic problems. Similarly, Chapter 6 explores the value of high occupancy ridesharing for attaining better benefits since ridesplitting differs from other ridesourcing by using dynamic matching and detouring to attain higher occupancies.
- Chapter 3 models the AMoD ridesplitting with microscopic traffic models that aim to model individual vehicle traits, e.g., driving behaviors and emissions. Therefore, in Chapter 6, it allows modeling and exploring detailed impacts of autonomous and electric vehicles on ridesplitting benefits against the conventional human driving and petroleum vehicles.
- The developed market equilibrium (ME) model in chapter 4 is also well suited for policymakers to, e.g., seek possible market states at different penetration rates, find states with optimum social welfare, evaluate smarter subsidy methods and see ridesplitting impacts on other modes by evaluating mode shares. Note that dynamic demand estimation in chapter 5 is a major contributing method to any similar demand modeling method since it improves the estimation process of the network demand, and any evaluations made rely on its accuracy.

- ***Service users:*** For service users, ridesplitting is unique from other ridesourcing services. Due to dynamic ride-matching and detouring, on the one hand, they perceive uncertainty and inequity, while on the other, they find a service most affordable among others with equal flexibility. Therefore, it is also important to understand how users perceive this uniqueness and also propose operational strategies to minimize its adverse effects on service adoption.
  - Chapter 4 conducts stated-preference experiments to identify the factors affecting user travel behavior in the presence of high capacity ridesplitting as a transport mode. Multiple models are specified to understand user preferences and value toward the ridesplitting trip attributes.
  - Chapter 4 proposes a utility-based compensation pricing method to reduce both uncertainty and inequity in riders' trip level of service (LoS). It reduces the standard deviation of experienced trip utilities to improve certainty in traveling ridesplitting service and subsequently its adoption. In addition, smart subsidy methods are also proposed for policymakers to encourage ridesplitting usage.

8.2 Main findings

Microscopic modeling and optimization:

- Microscopic traffic models show stochastic variations in the sequence and amount of requests served against the use of simpler time-dependent travel times, depicting the effects of modeling dynamic traffic congestion and driving interactions. The effect is further evident in riders' travel and detour times, which are well-distributed in microscopic modeling even for non-ridesharing trips, whereas the fixed travel time-based setup shows more constant values for all non-ridesharing trips.

- Microscopic traffic models allow comparison of autonomous and human-driven vehicles using their subsequent microscopic driving models. The results depict slightly slower or less aggressive driving by AVs with longer average trip times. Whereas the models also allow modeling of detailed emissions for both electric and petroleum vehicles, which clearly indicate ridesharing emission benefits.
- Although microscopic models can better replicate the stochasticity in network and service information in the integrated DARP optimization to allow more efficient service optimization. The result does not clearly indicate this due to the dynamic nature of the service, where the efficiency does not translate into, e.g., higher requests served, but only translates into different riders' matching sequences by dynamic matching and detouring.

Dynamic demand estimation:

- For PCA-based estimation methods, the presence of relevant historical estimates can significantly improve the estimation quality, where the increase in relevance first translates into better reproduction of traffic measurements and then to better OD solution quality.
- Among the three possible correlations in demand, i.e., spatial, temporal, and day to day, results suggest that the method that uses all three correlations outperforms others for convergence speed, the robustness of the results, and calibrated OD solution quality. Whereas the use of only one correlation dimension can provide good results in reproducing the traffic measurements, however, the PCA–models are more likely over-fit the data, as the PCs cannot model ODs correlations properly.
- A better exploitation of the PCA properties like the proposed simplified problem formulation leads to an enhanced algorithm that achieves faster convergence and provides more robust results even on large urban networks.
- The proposed PC-SPSA implementation is significantly robust against varying estimation setups, i.e., defining SPSA hyperparameters and historical dataset's size, mean, and variance. Therefore it is much more convenient to adopt it for large-scale dynamic demand estimation with minimum manual input, which is otherwise a cumbersome trial and error-based manual process.
- In the historical dataset generation setup, the algorithm convergence is slower for both the smaller and larger dataset sizes and standard deviations while being faster for certain optimum middle values. Whereas enlarging the dataset size continues to improve the consistency in OD solution quality, however, the lower values of dataset standard deviation result in the best OD solution quality.

Ridesplitting preferences, demand modeling, and pricing:

- The ridesplitting preference models indicate that respondents aged 26-35 years or public transport commuters or members of bike-sharing services are more likely to choose ridesplitting or PT than the private car. Similarly, respondents concerned with climate change and willing to spend more on environmentally friendly products are more likely to use ridesplitting against the private car. Meanwhile, the mode-specific values of in-vehicle travel time are 7.7 €, 12.2 €, and 18.5 € per hour for public transport, ridesplitting, and private car, respectively. While for the survey subsample of Munich, the in-vehicle travel time values are slightly lower, i.e., 5.0 €, 13.4 €, and 16.9 € per hour for public transport, ridesplitting, and private car, respectively.

- The ridesplitting market equilibrium analysis of the Munich case study indicates that the unit price of optimum profitability (0.53 €/km) is higher than that for social welfare (0.36 €/km), while the fleet size that indicates available supply it is the opposite (i.e., 356 vehicles for profit and 623 vehicles for welfare). Similarly, the joint influence of unit price and fleet size on profit and welfare is similar for high unit price values and considerably different for low unit price values.
- The utility-based compensation pricing helps reduce the presence of uncertainty and inequity in ridesplitting trips, i.e., it improves the mean and standard deviation of trip utilities by about 8% and 50%, respectively (in the Munich case study). While disregarding the objective to improve equity, the pricing method also shows the potential to increase profit and social welfare by 2.9% and 6.5%, respectively, through better demand attraction.
- The utility-based compensation pricing provides opportunities for smart subsidy schemes for ridesplitting operators where the subsidies can be utilized as dynamic compensations to maximize the certainty and equity of ridesplitting trips with higher ridesplitting demand attraction without any sacrifice in profit and social welfare margins.

Ridesplitting service exploration and impact assessment:

- ***Ridesplitting benefits:*** Ridesplitting as a short-distance and stand-alone transport mode in the Munich city area tends to show positive benefits in the form of vehicle kilometers (VK), emissions, and energy saved. However, the benefits are subject to the service setup, demand attracted, and riders' flexibility. For the Munich city center case study with medium to high demand attraction and riders' trip flexibility served by small fleet size, the vehicle kilometer saving range between 0.5 to 1.2 km per trip. Similarly, the CO_2 emissions reduction (estimated by the HBEFA3 model) is around 0.8 kg per trip for the use of electric ridesplitting vehicles and 0.05 to 0.2 kg per trip for the use of petroleum ridesplitting vehicles, both against petroleum cars.
- ***Effect of demand and waiting for flexibility:*** Ridesplitting benefits have a positive linear relationship with both demand and riders' waiting flexibility, where the benefits show higher sensitivity against the increase in riders' waiting flexibility. Similarly, both increase in demand and riders' waiting flexibility also help trigger better ridesharing; however, the waiting flexibility can better encourage higher ridesharing occupancies.
- ***Effect of ridesharing occupancy:*** Benefits per ride trip analysis indicates that the higher ridesharing occupancies do not necessarily result in better benefits per request. However, at equal occupancy rates, the presence of higher demand and higher waiting flexibility generates more ridesplitting benefits, with waiting flexibility having a more significant effect.
- ***Waiting versus detour flexibility:*** Riders' waiting flexibility shows a direct effect on achieving higher ridesharing occupancy and benefits, whereas higher travel detour flexibility seems to increase the number of requests served but is supplementary to waiting time for higher ridesharing benefits.
- ***Riders' time loss against service benefits:*** The comparison of riders' additional trip times and resulting ridesplitting benefits depict that higher service benefits can be attained at similar additional time costs when riders show higher trip flexibility. However, higher travel flexibility can also result in more inequity among riders making dynamic pricing like utility-based compensation pricing a viable solution.

- ***Endogenous demand:*** Under the conservation assumption of considering riders' waiting and detour flexibility directly as the expected trip utility, ridesplitting as a separate transport mode in competition with private car and public transport also shows positive benefits for vehicle kilometers (VK) and emissions saving. However, the benefits do not increase with the increase in waiting and detouring flexibility which only translates into more number of requests served.

8.3 Limitations and future works

While this thesis sought methods specific to ridesplitting characteristics for improving the service modeling, operations, and assessment, it leaves many future research directions and further required efforts to enhance modeling ridesplitting supply, dynamic network and ridesplitting demand estimation, and ridesplitting service assessments.

First considering the AMoD ridesplitting platform, due to its comprehensive nature, it is restricted by the scale of the service optimization problem, i.e., only a limited amount of service vehicles and network size can be setup. Future extensions are possible to scale the service implementation by smarter network representation and scheduler integration methods, i.e., simplifying scheduler network definition with node limited only to service locations or meeting points and using SUMO path assignment for routing finding, or setting up multiple subdivided ridesplitting services to do parallel optimization. Similarly, future research prospects include, extending the proposed ridesplitting platform towards modeling and optimizing different service use-cases like meeting point-based setup or last–mile feeder service for public transport (using multimodal person trips), and exploiting the use of microscopic traffic models to model advance vehicle automation and connectivity concepts (like platooning, signal coordination, and prioritization) for ridesplitting services.

Next in ridesplitting demand related aspects, the future work for riders' preference experiments could include having larger number of alternatives and attributes, such as comfort, safety and trip purpose, as well as combination of RP and SP methods to cater for a more realistic trips representation. Meanwhile, issues of heterogeneity and data collection biases could be addressed, e.g., by conducting market segmentation. Moreover, the impact of automation on riders' preference is also a crucial aspect and should be included in future research. Similarly, in ridesplitting demand modeling, the mode choice setup utilized for market equilibrium and endogenous demand experiments assumes simplifications for service costs and competitive modes' attributes which should be modeled more comprehensively upon data availability. Moreover, assessments on the proposed trip-based mode choice method for ridesplitting are also part of future work to better understand its viability. Similarly, another interesting aspect is to model and assess different levels of rider trip occupancies with varying pricing setup and their ride-sharing related impacts.

Likewise, future works related to the ridesplitting service assessment can focus on deploying different networks with varying size and typologies and modeling variations in demand structures and resulting traffic congestion patterns to assess their impact on ridesplitting. Similarly, extensive modeling of the network–level KPIs, e.g., modeling network–level emissions and traffic impacts due to vehicle kilometers saving or modeling other indirect benefits like increase in parking availability, reduction in required service fleet, and consumer–based economical impacts are also interesting future directions. Another research direction is to integrate the utility-based dynamic compensation pricing within the endogenous demand experiments to assess its efficacy and impacts under time-dependent service modeling.

Finally, with respect to dynamic demand estimation, the presented research introduces the first building block to move PCA–based calibration models proposed in the literature from theory

to practice. While, the proposed data-assimilation framework is used to incorporate historical information within the PCs of the problem, it opens many promising research directions. Future researches can plan to use the same concept to incorporate synthetic populations, activity based models, and, in general, more information about the travel demand without increasing the complexity of the problem. Likewise, different data sources, such as mobile phone network data, GPS trajectory data, and even social media data can be incorporated into the framework in a similar fashion. Another advantage of the proposed framework is that, beside reducing the number of variables, the proposed model drastically reduces the number of simulation runs required to calibrate the model. This is an important observation when the objective is to calibrate multimodal transport systems, where the number of variables to be calibrated as well as the simulation time are prohibitive already for small sized systems. Finally, traditional PCA–based are linear in their nature. However, there is not guarantee that data are linearly correlated, specifically when using different data sources or complex representations of travel behaviour, such as synthetic populations. Therefore, non linear PCA–based frameworks should also be investigated in the future.

Bibliography

Adnan, M., Pereira, F.C., Azevedo, C.M.L., Basak, K., Lovric, M., Raveau, S., Zhu, Y., Ferreira, J., Zegras, C., Ben-Akiva, M., 2016. Simmobility: A multi-scale integrated agent-based simulation platform, in: 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record.

Agatz, N., Erera, A.L., Savelsbergh, M.W., Wang, X., 2011. Dynamic ride-sharing: A simulation study in metro atlanta. Procedia-Social and Behavioral Sciences 17, 532–550.

Aissat, C., Oulamara, A., 2014. A priori approach of real-time ridesharing problem with intermediate meeting locations. Journal of Artificial Intelligence and Soft Computing Research 4.

Alam, M.J., Habib, M.A., 2018. Investigation of the impacts of shared autonomous vehicle operation in halifax, canada using a dynamic traffic microsimulation model. Procedia computer science 130, 496–503.

Alazzawi, S., Hummel, M., Kordt, P., Sickenberger, T., Wieseotte, C., Wohak, O., 2018. Simulating the impact of shared, autonomous vehicles on urban mobility-a case study of milan. EPiC Series in Engineering 2, 94–110.

Alonso-González, M.J., Cats, O., van Oort, N., Hoogendoorn-Lanser, S., Hoogendoorn, S., 2021. What are the determinants of the willingness to share rides in pooled on-demand services? Transportation 48, 1733–1765.

Alonso-González, M.J., van Oort, N., Cats, O., Hoogendoorn-Lanser, S., Hoogendoorn, S., 2020. Value of time and reliability for urban pooled on-demand services. Transportation Research Part C: Emerging Technologies 115, 102621.

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, 462–467.

Antoniou, C., 2004. On-line calibration for dynamic traffic assignment. Ph.D. thesis. Massachusetts Institute of Technology.

Antoniou, C., Azevedo, C.L., Lu, L., Pereira, F., Ben-Akiva, M., 2015. W–spsa in practice: Approximation of weight matrices and calibration of traffic simulation models. Transportation Research Procedia 7, 233–253.

Antoniou, C., Balakrishna, R., Koutsopoulos, H.N., Ben-Akiva, M., 2009. Off–line and on–line calibration of dynamic traffic assignment systems. IFAC Proceedings Volumes 42, 104–111.

Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., Montero, L., Nigro, M., Perarnau, J., Punzo, V., Toledo, T., van Lint, H., 2016. Towards a generic benchmarking platform for origin-destination flows estimation/updating algorithms: Design, demonstration and validation. Transportation Research Part C: Emerging Technologies 66, 79–98.

Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2007a. Nonlinear kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. IEEE Transactions on Intelligent Transportation Systems 8, 661–670.

Antoniou, C., Matsoukis, E., Roussi, P., 2007b. A methodology for the estimation of value-of-time using state-of-the-art econometric models. Journal of public transportation 10, 1.

Antoniou, C., Polydoropoulou, A., 2015. The value of privacy: Evidence from the use of mobile devices for traveler information systems. Journal of Intelligent Transportation Systems 19, 167–180.

Arrow, K.J., Debreu, G., 1954. Existence of an equilibrium for a competitive economy. Econometrica: Journal of the Econometric Society , 265–290.

Ashok, K., Ben-Akiva, M.E., 2000. Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows. Transportation science 34, 21–36.

Atasoy, B., Glerum, A., Bierlaire, M., 2011. Mode choice with attitudinal latent class: a swiss case-study, in: Second International Choice Modeling Conference.

Azevedo, C.L., Marczuk, K., Raveau, S., Soh, H., Adnan, M., Basak, K., Loganathan, H., Deshmunkh, N., Lee, D.H., Frazzoli, E., et al., 2016. Microsimulation of demand and supply of autonomous mobility on demand. Transportation Research Record 2564, 21–30.

Bai, W., Quan, J., Fu, L., Gan, X., Wang, X., 2017. Online fair allocation in autonomous vehicle sharing, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE. pp. 1–6.

Balakrishna, R., 2006. Off-line calibration of dynamic traffic assignment models. Ph.D. thesis. Massachusetts Institute of Technology.

Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., Wen, Y., 2007a. Calibration of microscopic traffic simulation models: Methods and application. Transportation Research Record 1999, 198–207.

Balakrishna, R., Ben-Akiva, M., Koutsopoulos, H.N., 2007b. Offline calibration of dynamic traffic assignment: simultaneous demand-and-supply estimation. Transportation Research Record 2003, 50–58.

Balakrishna, R., Koutsopoulos, H.N., Ben-Akiva, M., 2005. Calibration and validation of dynamic traffic assignment systems, in: Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic TheoryUniversity of Maryland, College Park.

Barceló, J., Montero, L., Marqués, L., Carmona, C., 2010. Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. Transportation research record 2175, 19–27.

Basu, R., Araldo, A., Akkinepally, A.P., Nahmias Biran, B.H., Basak, K., Seshadri, R., Deshmukh, N., Kumar, N., Azevedo, C.L., Ben-Akiva, M., 2018. Automated mobility-on-demand vs. mass transit: a multi-modal activity-driven agent-based simulation approach. Transportation Research Record 2672, 608–618.

Bauer, G.S., Greenblatt, J.B., Gerke, B.F., 2018. Cost, energy, and environmental impact of automated electric taxi fleets in manhattan. Environmental science & technology 52, 4920–4928.

Bent, R.W., Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research 52, 977–987.

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, 343–355.

Bierlaire, M., 2018. Pandasbiogeme: a short introduction. EPFL (Transport and Mobility Laboratory, ENAC) .

Bimpikis, K., Candogan, O., Saban, D., 2019. Spatial pricing in ride-sharing networks. Operations Research 67, 744–769.

Bischoff, J., Maciejewski, M., 2016. Autonomous taxicabs in berlin–a spatiotemporal analysis of service performance. Transportation Research Procedia 19, 176–186.

Bischoff, J., Maciejewski, M., Nagel, K., 2017. City-wide shared taxis: A simulation study in berlin, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE. pp. 275–280.

Boesch, P.M., Ciari, F., Axhausen, K.W., 2016. Autonomous vehicle fleet sizes required to serve different levels of demand. Transportation Research Record 2542, 111–119.

Bösch, P.M., Ciari, F., Axhausen, K.W., 2018. Transport policy optimization with autonomous vehicles. Transportation Research Record 2672, 698–707.

Cairns, R.D., Liston-Heyes, C., 1996. Competition and regulation in the taxi industry. Journal of Public Economics 59, 1–15.

Cantelmo, G., Cipriani, E., Gemma, A., Nigro, M., 2014a. An adaptive bi-level gradient procedure for the estimation of dynamic traffic demand. IEEE Transactions on Intelligent Transportation Systems 15, 1348–1361.

Cantelmo, G., Qurashi, M., Prakash, A.A., Antoniou, C., Viti, F., 2020. Incorporating trip chaining within online demand estimation. Transportation Research Part B: Methodological 132, 171–187. doi:doi: https://doi.org/10.1016/j.trb.2019.05.010. 23rd International Symposium on Transportation and Traffic Theory (ISTTT 23).

Cantelmo, G., Viti, F., Cipriani, E., Nigro, M., 2018. A utility-based dynamic demand estimation model that explicitly accounts for activity scheduling and duration. Transportation Research Part A: Policy and Practice 114, 303–320.

Cantelmo, G., Viti, F., Tampère, C.M., Cipriani, E., Nigro, M., 2014b. Two-step approach for correction of seed matrix in dynamic demand estimation. Transportation Research Record 2466, 125–133.

Carrese, S., Cipriani, E., Mannini, L., Nigro, M., 2017. Dynamic demand estimation and prediction for traffic urban networks adopting new data sources. Transportation Research Part C: Emerging Technologies 81, 83–98.

Cascetta, E., 2009. Transportation systems analysis: models and applications. volume 29. Springer Science & Business Media.

Cascetta, E., Papola, A., Marzano, V., Simonelli, F., Vitiello, I., 2013. Quasi-dynamic estimation of o–d flows from traffic counts: Formulation, statistical validation and performance analysis on real data. Transportation Research Part B: Methodological 55, 171–187.

Cascetta, E., Postorino, M.N., 2001. Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks. Transportation science 35, 134–147.

Castiglione, M., Cantelmo, G., Qurashi, M., Nigro, M., Antoniou, C., 2021. Assignment matrix free algorithms for on-line estimation of dynamic origin-destination matrices. Frontiers in Future Transportation 2, 3.

Chen, T.D., Kockelman, K.M., 2016. Management of a shared autonomous electric vehicle fleet: Implications of pricing schemes. Transportation Research Record 2572, 37–46.

Chen, T.D., Kockelman, K.M., Hanna, J.P., 2016. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. Transportation Research Part A: Policy and Practice 94, 243–254.

Chen, X., Zheng, H., Wang, Z., Chen, X., 2021. Exploring impacts of on-demand ridesplitting on mobility via real-world ridesourcing data and questionnaires. Transportation 48, 1541–1561.

Cheng, Q., Wang, S., Liu, Z., Yuan, Y., 2019. Surrogate-based simulation optimization approach for day-to-day dynamics model calibration with real data. Transportation Research Part C: Emerging Technologies 105, 422–438.

Chiappone, S., Giuffrè, O., Granà, A., Mauro, R., Sferlazza, A., 2016. Traffic simulation models calibration using speed–density relationship: An automated procedure based on genetic algorithm. Expert Systems with Applications 44, 147–155.

Chiu, Y.C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., Hicks, J., 2011. Dynamic traffic assignment: A primer. Dynamic Traffic Assignment: A Primer .

Cipriani, E., Florian, M., Mahut, M., Nigro, M., 2011. A gradient approximation approach for adjusting temporal origin–destination matrices. Transportation Research Part C: Emerging Technologies 19, 270–282.

Cobos, C., Erazo, C., Luna, J., Mendoza, M., Gaviria, C., Arteaga, C., Paz, A., 2016. Multi-objective memetic algorithm based on nsga-ii and simulated annealing for calibrating corsim micro-simulation models of vehicular traffic flow, in: Conference of the spanish association for artificial intelligence, Springer. pp. 468–476.

Currie, G., Fournier, N., 2020. Why most drt/micro-transits fail–what the survivors tell us about progress. Research in Transportation Economics 83, 100895.

Dinar, Y., 2020. Impact of Connected and/or Autonomous Vehicles in Mixed Traffic. Master thesis. Technical University of Munich. Munich.

de Dios Ortúzar, J., Willumsen, L.G., 2011. Modelling transport. John wiley & sons.

Djukic, T., Van Lint, J., Hoogendoorn, S., 2012. Application of principal component analysis to predict dynamic origin–destination matrices. Transportation research record 2283, 81–89.

DMR, 2022a. Didi statistics and facts. URL: https://expandedramblings.com/index.php/didi-chuxing-facts-statistics/.

DMR, 2022b. Uber statistics and facts. URL: https://expandedramblings.com/index.php/uber-statistics/.

Do, W., Rouhani, O.M., Miranda-Moreno, L., 2019. Simulation-based connected and automated vehicle models on highway sections: a literature review. Journal of Advanced Transportation 2019.

EEA, 2022. Greenhouse gas emissions from transport in europe. URL: https://www.eea.europa.eu/ims/greenhouse-gas-emissions-from-transport.

Fagnant, D.J., Kockelman, K., 2015. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice 77, 167–181.

Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transportation Research Part C: Emerging Technologies 40, 1–13.

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, 143–158.

Farhan, J., Chen, T.D., 2018. Impact of ridesharing on operational efficiency of shared autonomous electric vehicle fleet. Transportation Research Part C: Emerging Technologies 93, 310–321. doi:doi: https://doi.org/10.1016/j.trc.2018.04.022.

Fielbaum, A., Bai, X., Alonso-Mora, J., 2021. On-demand ridesharing with optimized pick-up and drop-off walking locations. Transportation research part C: emerging technologies 126, 103061.

Firnkorn, J., Müller, M., 2012. Selling mobility instead of cars: new business strategies of automakers and the impact on private vehicle holding. Business Strategy and the environment 21, 264–280.

Flötteröd, G., Bierlaire, M., Nagel, K., 2011. Bayesian demand calibration for dynamic traffic simulations. Transportation Science 45, 541–561.

Fontaras, G., Zacharof, N.G., Ciuffo, B., 2017. Fuel consumption and co2 emissions from passenger cars in europe–laboratory versus real-world emissions. Progress in energy and combustion Science 60, 97–131.

Frederix, R., Viti, F., Corthout, R., Tampère, C.M., 2011. New gradient approximation method for dynamic origin–destination matrix estimation on congested networks. Transportation Research Record 2263, 19–25.

Frei, C., Hyland, M., Mahmassani, H.S., 2017. Flexing service schedules: Assessing the potential for demand-adaptive hybrid transit via a stated preference approach. Transportation Research Part C: Emerging Technologies 76, 71–89.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.E., Wang, X., Koenig, S., 2013. Ridesharing: The state-of-the-art and future directions. Transportation Research Part B: Methodological 57, 28–46.

Ge, Y., Knittel, C.R., MacKenzie, D., Zoepf, S., 2016. Racial and gender discrimination in transportation network companies. Technical Report. National Bureau of Economic Research.

Ghilas, V., Demir, E., Van Woensel, T., 2016. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. Transportation Research Part B: Methodological 91, 34–51.

Gora, P., Katrakazas, C., Drabicki, A., Islam, F., Ostaszewski, P., 2020. Microscopic traffic simulation models for connected and automated vehicles (cavs)–state-of-the-art. Procedia Computer Science 170, 474–481.

Guan, Y., Annaswamy, A.M., Tseng, H.E., 2019a. Cumulative prospect theory based dynamic pricing for shared mobility on demand services, in: 2019 IEEE 58th Conference on Decision and Control (CDC), IEEE. pp. 2239–2244.

Guan, Y., Annaswamy, A.M., Tseng, H.E., 2019b. A dynamic routing framework for shared mobility services. ACM Transactions on Cyber-Physical Systems 4, 1–28.

Guo, S., Liu, Y., Xu, K., Chiu, D.M., 2017. Understanding ride-on-demand service: Demand and dynamic pricing, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE. pp. 509–514.

Gurumurthy, K.M., Kockelman, K.M., 2018. Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from orlando, florida. Computers, Environment and Urban Systems 71, 177–185.

Gurumurthy, K.M., Kockelman, K.M., Simoni, M.D., 2019. Benefits and costs of ride-sharing in shared automated vehicles across austin, texas: Opportunities for congestion pricing. Transportation Research Record 2673, 548–556.

Häll, C.H., Lundgren, J.T., Voss, S., 2015. Evaluating the performance of a dial-a-ride service using simulation. Public Transport 7, 139–157.

He, F., Wang, X., Lin, X., Tang, X., 2018. Pricing and penalty/compensation strategies of a taxi-hailing platform. Transportation Research Part C: Emerging Technologies 86, 263–279.

Henderson, J., Fu, L., 2004. Applications of genetic algorithms in transportation engineering, in: 83rd annual meeting of the Transportation Research Board.

Ho, S.C., Szeto, W.Y., 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, 395–421.

Holland, J.H., et al., 1992. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

Hörl, S., 2017. Agent-based simulation of autonomous taxi services with dynamic demand responses. Procedia Computer Science 109, 899–904.

Hörl, S., Erath, A., Axhausen, K.W., 2016. Simulation of autonomous taxis in a multi-modal traffic scenario with dynamic demand. Arbeitsberichte Verkehrs-und Raumplanung 1184.

Horni, A., Nagel, K., Axhausen, K.W., 2016. The multi-agent transport simulation MATSim. Ubiquity Press.

Hosni, H., Naoum-Sawaya, J., Artail, H., 2014. The shared-taxi problem: Formulation and solution methods. Transportation Research Part B: Methodological 70, 303–318.

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. Journal of educational psychology 24, 417.

Huang, Y., Kockelman, K.M., Garikapati, V., Zhu, L., Young, S., 2021. Use of shared automated vehicles for first-mile last-mile service: micro-simulation of rail-transit connections in austin, texas. Transportation research record 2675, 135–149.

Hyland, M., Mahmassani, H.S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign avs to immediate traveler demand requests. Transportation Research Part C: Emerging Technologies 92, 278–297.

Hyland, M.F., Mahmassani, H.S., 2017. Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility. Transportation Research Record 2653, 26–34.

INFRAS, 2022. Handbook Emission Factors for Road Transport (HBEFA). URL: https://www.hbefa.net/e/index.html.

Jäger, B., Agua, F.M.M., Lienkamp, M., 2017. Agent-based simulation of a shared, autonomous and electric on-demand mobility solution, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE. pp. 250–255.

Jäger, B., Brickwedde, C., Lienkamp, M., 2018. Multi-agent simulation of a demand-responsive transit system operated by autonomous vehicles. Transportation Research Record 2672, 764–774.

Kang, S., Mondal, A., Bhat, A.C., Bhat, C.R., 2021. Pooled versus private ride-hailing: A joint revealed and stated preference analysis recognizing psycho-social factors. Transportation Research Part C: Emerging Technologies 124, 102906.

Kattan, L., Abdulhai, B., 2006. Noniterative approach to dynamic traffic origin–destination estimation with parallel evolutionary algorithms. Transportation research record 1964, 201–210.

Ke, J., Yang, H., Li, X., Wang, H., Ye, J., 2020. Pricing and equilibrium in on-demand ride-pooling markets. Transportation Research Part B: Methodological 139, 411–431.

Kiefer, J., Wolfowitz, J., 1952. Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics , 462–466.

Kim, S.J., Kim, W., Rilett, L.R., 2005. Calibration of microsimulation models using nonparametric statistical techniques. Transportation Research Record 1935, 111–119.

Ko, E., Kim, H., Lee, J., 2021. Survey data analysis on intention to use shared mobility services. Journal of Advanced Transportation 2021.

Kolarova, V., Steck, F., Cyganski, R., Trommer, S., 2018. Estimation of the value of time for automated driving using revealed and stated preference methods. Transportation research procedia 31, 35–46.

Krueger, R., Rashidi, T.H., Rose, J.M., 2016. Preferences for shared autonomous vehicles. Transportation research part C: emerging technologies 69, 343–355.

Kuppam, A.R., Pendyala, R.M., Rahman, S., 1999. Analysis of the role of traveler attitudes and perceptions in explaining mode-choice behavior. Transportation Research Record 1676, 68–76.

Kurczveil, T., López, P.Á., Schnieder, E., 2013. Implementation of an energy model and a charging infrastructure in sumo, in: Simulation of Urban MObility User Conference, Springer. pp. 33–43.

Lavieri, P.S., Bhat, C.R., 2019. Modeling individuals' willingness to share trips with strangers in an autonomous vehicle future. Transportation research part A: policy and practice 124, 242–261.

Levin, M.W., 2017. Congestion-aware system optimal route choice for shared autonomous vehicles. Transportation Research Part C: Emerging Technologies 82, 229–247.

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, 373–383.

Li, D., Antoniou, C., Jiang, H., Xie, Q., Shen, W., Han, W., 2019a. The value of prepositioning in smartphone-based vanpool services under stochastic requests and time-dependent travel times. Transportation Research Record 2673, 26–37.

Li, S., Tavafoghi, H., Poolla, K., Varaiya, P., 2019b. Regulating tncs: Should uber and lyft set their own rules? Transportation Research Part B: Methodological 129, 193–225.

Li, W., Pu, Z., Li, Y., Ban, X.J., 2019c. Characterization of ridesplitting based on observed data: A case study of chengdu, china. Transportation Research Part C: Emerging Technologies 100, 330–353.

Li, Y., Chung, S.H., 2020. Ride-sharing under travel time uncertainty: Robust optimization and clustering approaches. Computers & Industrial Engineering 149, 106601.

Liu, Y., Bansal, P., Daziano, R., Samaranayake, S., 2019. A framework to integrate mode choice in the design of mobility-on-demand systems. Transportation Research Part C: Emerging Technologies 105, 648–665.

Lokhandwala, M., Cai, H., 2018. Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of nyc. Transportation Research Part C: Emerging Technologies 97, 45–60.

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using sumo, in: The 21st IEEE International Conference on Intelligent Transportation Systems, IEEE.

Lowalekar, M., Varakantham, P., Jaillet, P., 2018. Online spatio-temporal matching in stochastic and dynamic domains. Artificial Intelligence 261, 71–112.

Lu, L., Xu, Y., Antoniou, C., Ben-Akiva, M., 2015. An enhanced spsa algorithm for the calibration of dynamic traffic assignment models. Transportation Research Part C: Emerging Technologies 51, 149–166.

Ma, J., Li, X., Zhou, F., Hao, W., 2017. Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. Transportation Research Part C: Emerging Technologies 84, 124–141.

Ma, Z., Koutsopoulos, H.N., 2022. Near-on-demand mobility. the benefits of user flexibility for ride-pooling services. Transportation Research Part C: Emerging Technologies 135, 103530.

Maciejewski, M., Bischoff, J., Hörl, S., Nagel, K., 2017. Towards a testbed for dynamic vehicle routing algorithms, in: International Conference on Practical Applications of Agents and Multi-Agent Systems, Springer. pp. 69–79.

Manski, C.F., Wright, J.D., 1967. Nature of equilibrium in the market for taxi services. Technical Report.

Marczuk, K.A., Hong, H.S.S., Azevedo, C.M.L., Adnan, M., Pendleton, S.D., Frazzoli, E., et al., 2015. Autonomous mobility on demand in simmobility: Case study of the central business district in singapore, in: 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), IEEE. pp. 167–172.

Marković, N., Nair, R., Schonfeld, P., Miller-Hooks, E., Mohebbi, M., 2015. Optimizing dial-a-ride services in maryland: benefits of computerized routing and scheduling. Transportation Research Part C: Emerging Technologies 55, 156–165.

Martinez, L.M., Viegas, J.M., 2017. Assessing the impacts of deploying a shared self-driving urban mobility system: An agent-based model applied to the city of lisbon, portugal. International Journal of Transportation Science and Technology 6, 13–27.

Marzano, V., Papola, A., Simonelli, F., 2009. Limits and perspectives of effective o–d matrix correction using traffic counts. Transportation Research Part C: Emerging Technologies 17, 120–132.

Marzano, V., Papola, A., Simonelli, F., Papageorgiou, M., 2018. A kalman filter for quasi-dynamic od flow estimation/updating. IEEE Transactions on Intelligent Transportation Systems 19, 3604–3612.

McFadden, D., et al., 1973. Conditional logit analysis of qualitative choice behavior .

McNally, M.G., 2007. The four-step model. Emerald Group Publishing Limited.

Milanés, V., Shladover, S.E., 2014. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. Transportation Research Part C: Emerging Technologies 48, 285–300.

Moeckel, R., Kuehnel, N., Llorca, C., Moreno, A.T., Rayaprolu, H., 2020. Agent-based simulation to improve policy sensitivity of trip-based models. Journal of Advanced Transportation 2020.

Molenbruch, Y., Braekers, K., Caris, A., 2017. Typology and literature review for dial-a-ride problems. Annals of Operations Research 259, 295–325.

Moreno, A.T., Michalski, A., Llorca, C., Moeckel, R., 2018. Shared autonomous vehicles effect on vehicle-km traveled and average trip duration. Journal of Advanced Transportation 2018.

MVG, 2022. MVG ridepooling Service on Demand - IsarTiger. URL: https://www.mvg.de/ueber/mvg-projekte/mvg-sod.html.

Nahmias-Biran, B.h., Oke, J.B., Kumar, N., Basak, K., Araldo, A., Seshadri, R., Akkinepally, A., Lima Azevedo, C., Ben-Akiva, M., 2019. From traditional to automated mobility on demand: a comprehensive framework for modeling on-demand services in simmobility. Transportation Research Record 2673, 15–29.

Nahmias-Biran, B.h., Oke, J.B., Kumar, N., Lima Azevedo, C., Ben-Akiva, M., 2021. Evaluating the impacts of shared automated mobility on-demand services: an activity-based accessibility approach. Transportation 48, 1613–1638.

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transportation Research Part C: Emerging Technologies 111, 255–293.

Omrani, R., Kattan, L., 2013. Simultaneous calibration of microscopic traffic simulation model and estimation of origin/destination (od) flows based on genetic algorithms in a high-performance computer, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), IEEE. pp. 2316–2321.

Osorio, C., 2019a. Dynamic origin-destination matrix calibration for large-scale network simulators. Transportation Research Part C: Emerging Technologies 98, 186–206.

Osorio, C., 2019b. High-dimensional offline origin-destination (od) demand calibration for stochastic traffic simulators of large-scale road networks. Transportation Research Part B: Methodological 124, 18–43.

Pearson, K., 1901. Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science 2, 559–572.

Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. European Journal of Operational Research 225, 1–11.

Pisano, C.O., 2010. Mitigating network congestion: analytical models, optimization methods and their applications. Ph.D. thesis. Citeseer.

Powell, M.J., 1970. A hybrid method for nonlinear equations. Numerical methods for nonlinear algebraic equations .

Prakash, A.A., Seshadri, R., Antoniou, C., Pereira, F.C., Ben-Akiva, M., 2018. Improving scalability of generic online calibration for real-time dynamic traffic assignment systems. Transportation Research Record 2672, 79–92.

Prakash, A.A., Seshadri, R., Antoniou, C., Pereira, F.C., Ben-Akiva, M.E., 2017. Reducing the dimension of online calibration in dynamic traffic assignment systems. Transportation Research Record 2667, 96–107.

Qian, X., Ukkusuri, S.V., 2017. Time-of-day pricing in taxi markets. IEEE Transactions on Intelligent Transportation Systems 18, 1610–1622.

Qiu, H., Li, R., Zhao, J., 2018. Dynamic pricing in shared mobility on demand service. arXiv preprint arXiv:1802.03559 .

Quadrifoglio, L., Dessouky, M.M., Ordóñez, F., 2008. Mobility allowance shuttle transit (mast) services: Mip formulation and strengthening with logic constraints. European Journal of Operational Research 185, 481–494.

Qurashi, M., Cantelmo, G., Antoniou, C., 2022. Towards the use of ai in model calibration, in: AI in intelligent transportation systems. CRC Press.

Qurashi, M., Ma, T., Chaniotakis, E., Antoniou, C., 2019. Pc–spsa: employing dimensionality reduction to limit spsa search noise in dta model calibration. IEEE Transactions on Intelligent Transportation Systems 21, 1635–1645.

Ronald, N., Thompson, R., Haasz, J., Winter, S., 2013. Determining the viability of a demand-responsive transport system under varying demand scenarios, in: Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science, pp. 7–12.

Ronald, N., Thompson, R., Winter, S., 2017. Simulating ad-hoc demand-responsive transportation: a comparison of three approaches. Transportation Planning and Technology 40, 340–358.

Ros-Roca, X., Montero, L., Barceló, J., 2021. Investigating the quality of spiess-like and spsa approaches for dynamic od matrix estimation. Transportmetrica A: Transport Science 17, 235–257.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arxiv 2016. arXiv preprint arXiv:1609.04747 .

de Ruijter, A., Cats, O., Alonso-Mora, J., Hoogendoorn, S., 2020. Ride-sharing efficiency and level of service under alternative demand, behavioral and pricing settings, in: Transportation Research Board 2020 Annual Meeting.

Saidallah, M., El Fergougui, A., Elalaoui, A.E., 2016. A comparative study of urban road traffic simulators, in: MATEC Web of Conferences, EDP Sciences. p. 05002.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S.H., Ratti, C., 2014. Quantifying the benefits of vehicle pooling with shareability networks. Proceedings of the National Academy of Sciences 111, 13290–13294.

Sarriera, J.M., Álvarez, G.E., Blynn, K., Alesbury, A., Scully, T., Zhao, J., 2017. To share or not to share: Investigating the social aspects of dynamic ridesharing. Transportation Research Record 2605, 109–117.

Sayarshad, H.R., Chow, J.Y., 2015. A scalable non-myopic dynamic dial-a-ride and pricing problem. Transportation Research Part B: Methodological 81, 539–554.

Schilde, M., Doerner, K.F., Hartl, R.F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers & operations research 38, 1719–1730.

Schilde, M., Doerner, K.F., Hartl, R.F., 2014. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. European journal of operational research 238, 18–30.

Schultz, L., Sokolov, V., 2018. Practical bayesian optimization for transportation simulators. arXiv preprint arXiv:1810.03688 .

Sha, D., Ozbay, K., Ding, Y., 2020. Applying bayesian optimization for calibration of transportation simulation models. Transportation Research Record 2674, 215–228.

Shaheen, S., Cohen, A., 2019. Shared ride services in north america: definitions, impacts, and the future of pooling. Transport reviews 39, 427–442.

Shladover, S.E., Su, D., Lu, X.Y., 2012. Impacts of cooperative adaptive cruise control on freeway traffic flow. Transportation Research Record 2324, 63–70.

Simonetto, A., Monteil, J., Gambella, C., 2019. Real-time city-scale ridesharing via linear assignment problems. Transportation Research Part C: Emerging Technologies 101, 208–232.

Spall, J.C., 1998a. Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Transactions on aerospace and electronic systems 34, 817–823.

Spall, J.C., 1998b. An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins apl technical digest 19, 482–492.

Spall, J.C., 2000. Adaptive stochastic approximation by the simultaneous perturbation method. IEEE transactions on automatic control 45, 1839–1853.

Spall, J.C., 2003. Stochastic approximation and the finite-difference method. Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control, number , 150–175.

Spall, J.C., et al., 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE transactions on automatic control 37, 332–341.

Stathopoulos, A., Tsekeris, T., 2004. Hybrid meta-heuristic algorithm for the simultaneous optimization of the o–d trip matrix estimation. Computer-Aided Civil and Infrastructure Engineering 19, 421–435.

Steck, F., Kolarova, V., Bahamonde-Birke, F., Trommer, S., Lenz, B., 2018. How autonomous driving may affect the value of travel time savings for commuting. Transportation research record 2672, 11–20.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2015. The benefits of meeting points in ride-sharing systems. Transportation Research Part B: Methodological 82, 36–53.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2016. Making dynamic ride-sharing work: The impact of driver and rider flexibility. Transportation Research Part E: Logistics and Transportation Review 91, 190–207.

Stocker, A., Shaheen, S., 2018. Shared automated mobility: early exploration and potential impacts. Road Vehicle Automation 4 , 125–139.

Sukennik, P., Group, P., et al., 2018. Micro-simulation guide for automated vehicles. COEXIST (h2020-coexist. eu) .

Tafreshian, A., Abdolmaleki, M., Masoud, N., Wang, H., 2021. Proactive shuttle dispatching in large-scale dynamic dial-a-ride systems. Transportation Research Part B: Methodological 150, 227–259.

Toledo, T., Kolechkina, T., 2012. Estimation of dynamic origin–destination matrices using linear assignment matrix approximations. IEEE Transactions on Intelligent Transportation Systems 14, 618–626.

Train, K.E., 2009. Discrete choice methods with simulation. Cambridge university press.

Tsao, M., Milojevic, D., Ruch, C., Salazar, M., Frazzoli, E., Pavone, M., 2019. Model predictive control of ride-sharing autonomous mobility-on-demand systems, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE. pp. 6665–6671.

Tsiamasiotis, K., Chaniotakis, E., Qurashi, M., Jiang, H., Antoniou, C., 2021. Identifying and quantifying factors determining dynamic vanpooling use. Smart Cities 4, 1243–1258. URL: https://www.mdpi.com/2624-6511/4/4/66, doi:doi: 10.3390/smartcities4040066.

Tu, M., Li, W., Orfila, O., Li, Y., Gruyer, D., 2021. Exploring nonlinear effects of the built environment on ridesplitting: Evidence from chengdu. Transportation Research Part D: Transport and Environment 93, 102776.

Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2015. c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation. Transportation Research Part C: Emerging Technologies 55, 231–245.

Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2018. Robust spsa algorithms for dynamic od matrix estimation. Procedia computer science 130, 57–64.

Vaze, V., Antoniou, C., Wen, Y., Ben-Akiva, M., 2009. Calibration of dynamic traffic assignment models with point-to-point traffic surveillance. Transportation Research Record 2090, 1–9.

Vosooghi, R., Puchinger, J., Jankovic, M., Vouillon, A., 2019. Shared autonomous vehicle simulation and service design. Transportation Research Part C: Emerging Technologies 107, 15–33.

Vrtic, M., Schuessler, N., Erath, A., Axhausen, K.W., 2010. The impacts of road pricing on route and mode choice behaviour. Journal of Choice Modelling 3, 109–126.

Wang, H., Yang, H., 2019. Ridesourcing systems: A framework and review. Transportation Research Part B: Methodological 129, 122–155.

Wang, X., Agatz, N., Erera, A., 2018. Stable matching for dynamic ride-sharing systems. Transportation Science 52, 850–867.

Wang, Y., Wang, S., Wang, J., Wei, J., Wang, C., 2020. An empirical study of consumers' intention to use ride-sharing services: using an extended technology acceptance model. Transportation 47, 397–415.

Wang, Z., Chen, X., Chen, X.M., 2019. Ridesplitting is shaping young people's travel behavior: Evidence from comparative survey via ride-sourcing platform. Transportation research part D: transport and environment 75, 57–71.

Wardman, M., Chintakayala, V.P.K., de Jong, G., 2016. Values of travel time in europe: Review and meta-analysis. Transportation Research Part A: Policy and Practice 94, 93–111.

Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., Hubaux, J.P., 2008. Traci: an interface for coupling road traffic and network simulators, in: Proceedings of the 11th communications and networking simulation symposium, pp. 155–163.

Wegener, M., 2011. From macro to micro—how much micro is too much? Transport Reviews 31, 161–177.

Wei, C., Wang, Y., Yan, X., Shao, C., 2017. Look-ahead insertion policy for a shared-taxi system based on reinforcement learning. IEEE Access 6, 5716–5726.

West, D.B., et al., 2001. Introduction to graph theory. volume 2. Prentice hall Upper Saddle River.

Wheeler, R., 2004. Algdesign. the r project for statistical computing.

Wong, K.I., Wong, S.C., Yang, H., 2001. Modeling urban taxi services in congested road networks with elastic demand. Transportation Research Part B: Methodological 35, 819–842.

Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. European Journal of Operational Research 185, 534–551.

Yang, H., Leung, C.W., Wong, S.C., Bell, M.G., 2010. Equilibria of bilateral taxi–customer searching and meeting on networks. Transportation Research Part B: Methodological 44, 1067–1083.

Yang, H., Wong, S.C., 1998. A network model of urban taxi services. Transportation Research Part B: Methodological 32, 235–246.

Yang, H., Wong, S.C., Wong, K.I., 2002. Demand–supply equilibrium of taxi services in a network under competition and regulation. Transportation Research Part B: Methodological 36, 799–819.

Yang, Q., Koutsopoulos, H.N., 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transportation Research Part C: Emerging Technologies 4, 113–129.

Zhang, C., Osorio, C., Flötteröd, G., 2017. Efficient calibration techniques for large-scale traffic simulators. Transportation Research Part B: Methodological 97, 214–239.

Zhang, L., Liu, Z., Yu, L., Fang, K., Yao, B., Yu, B., 2022. Routing optimization of shared autonomous electric vehicles under uncertain travel time and uncertain service time. Transportation Research Part E: Logistics and Transportation Review 157, 102548.

Zhang, W., Guhathakurta, S., Fang, J., Zhang, G., 2015. Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. Sustainable Cities and Society 19, 34–45.

Zhao, Y., Kockelman, K.M., 2018. Anticipating the regional impacts of connected and automated vehicle travel in austin, texas. Journal of Urban Planning and Development 144, 04018032.

Zhu, Y., Qurashi, M., Ma, T., Antoniou, C., 2021. Joint calibration for dta model using islands-ga and pc-spsa. Transportation Research Procedia 52, 716–723.