

Simulation-Based Mode Choice Methods for Ride Sharing Services

Master's Thesis

A thesis presented in part fulfilment of the requirements of the Degree of Master of Science at the Department of Civil, Geo and Environmental Engineering, Technical University of Munich.

Supervisor	M.Sc. Moeid Qurashi Prof. Dr. Constantinos Antoniou Chair of Transportation Systems Engineering
Submitted by	Muhammad Ahmad Chaudhry
Submitted on	21.03.2021

I hereby confirm that this master's thesis was written independently by myself without the use of any sources beyond those cited, and all passages and ideas taken from other sources are cited accordingly.

Munich, 21.03.2021

Signature:

Abstract

This study proposes a simulation-based mode choice method for ride sharing services. Conventional mode choice for ride sharing services is a feedback loop based process where multiple simulation runs are expected to achieve an equilibrium state between assumed and simulated service attributes. The achieved equilibrium state is viable for the initial set of supply parameters. This thesis contributes in proposing the use of an analytical ride sharing market equilibrium model to perform the mode choice for simulating ride sharing services which is viable against any change in the supply, reducing the otherwise required regress computational time for running multiple simulations. The proposed methodology calibrates the market equilibrium model parameters using a set of observed service (attribute) data. The observed service data contains the service attributes for a range of fleet sizes serving a range of ride sharing demand. Whereas the analytical market equilibrium model proposed to be used in this study also outputs ride sharing demand and the network attributes detour and waiting time. This is taken as an optimization problem to be solved to calibrate the market equilibrium model parameters so that it outputs the observed service attributes for a given range of fleet sizes. A number of different goodness of fit errors are also utilized within the optimization problem. Two different case studies are used for the setup, first synthetic and then Munich network to calibrate the model parameters. The converged error results from both case study network shows that the analytical model fits to most of the extent with the observed data. Computational time of calibrated model is negligible compared to computationally expensive and regress process of simulation based equilibrium making the calibrated model as an effective alternate to perform mode choice of ride sharing services.

Acknowledgments

First of all, I would like to extend my sincerest gratitude towards my supervisor M. Sc. Moeid Qurashi, for his constant support throughout my thesis. His guidance has enabled me to work effectively in an improvisational manner.

I would like to thank Prof. Dr. Constantinos Antoniou, for his insightful feedback that aided me in the successful completion of my thesis.

Special thanks to Qinglong Lu for providing me the data related to ride sharing market equilibrium model.

My parents whose endless prayers and belief in me motivated me to keep moving always.

Last but not the least, I would like to specially thank my wife whose continuous support and scientific input helped me to perform better and successfully complete this study.

Contents

Abstract	iii
Acknowledgments	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives	2
1.3 Thesis Structure	3
2 Literature Review	4
2.1 Mode Choice	4
2.1.1 Sequential Models	4
2.1.2 Direct Demand Models	6
2.1.3 Discrete Choice Models	6
2.2 Mobility on-Demand Services	9
2.2.1 Modeling MoD services	9
2.2.2 Mode Choice of Ride-sharing Services	10
2.3 Market Equilibrium	12
2.3.1 Equilibrium Model for Taxi Market	13
2.3.2 Ride-Sharing Market Equilibrium	13
2.4 Literature Gaps	14
3 Methodology	15
3.1 Modeling Dynamic Van-pooling Services	15
3.1.1 Routing Algorithm	15
3.1.2 Simulation Platform	17
3.2 Ride Sharing Market Equilibrium Model	19
3.2.1 Demand Modelling	19
3.2.2 Integration with Simulation Platform	19
3.3 ME-based Mode Choice Method	21
3.3.1 General Methodology	21
3.3.2 Simulation Database	21
3.3.3 Calibration of ME Model	23

3.3.4	Discussion	24
4	Experimental Setup	26
4.1	Simulation Database	26
4.1.1	Ride Sharing Simulation Platform Setup	26
4.1.2	Issues/Errors	28
4.1.3	Standalone Scheduler	29
4.1.4	Data Collection	29
4.2	ME Model Calibration	30
4.2.1	ME Setup	30
4.2.2	ME Dataset	31
4.2.3	Data Comparison	32
4.2.4	Goodness of Fit	33
4.2.5	Optimization	34
4.3	ME-based Simulation Mode Choice	35
4.3.1	SUMO Inputs	35
4.3.2	Network Preparation	35
4.3.3	Demand Processing	36
5	Case Studies	39
5.1	Unterschleißheim - Synthetic	39
5.1.1	Network	39
5.1.2	Simulation Database	40
5.1.3	Optimization	41
5.2	Munich	47
5.2.1	Network	47
5.2.2	Simulation Database	48
5.2.3	Optimization	48
5.2.4	Mode Choice Methods Comparison	52
5.2.5	Model Applications	53
6	Conclusion	56
6.1	Summary	56
6.2	Limitations and Future Scope	57
	Bibliography	58

List of Figures

2.1	Multimodal Model Structures (de Dios Ortúzar & Willumsen, 2011)	5
2.2	A General Nested Logit Structure (de Dios Ortúzar & Willumsen, 2011)	8
2.3	Nested Logit model with several nests(de Dios Ortúzar & Willumsen, 2011)	8
2.4	A framework to optimize the supply-side parameters for MoD system with the integration of the mode choice model (Y. Liu et al., 2019)	11
2.5	Relationship between endogenous and exogenous variables (Lu, 2020)	14
3.1	Illustration of the evaluation procedure (Li et al., 2019)	16
3.2	Illustration of the scenario-based search (Li et al., 2019)	17
3.3	Modelling Architecture of Simulation Platform (Qurashi et al., 2020)	17
3.4	Scheduler Integration with Simulator (Qurashi et al., 2020)	18
3.5	Data Collection from Simulation Platform using Nested For Loop	22
3.6	Objective Function Workflow	23
3.7	Calibration of ME Model Parameters	24
4.1	Requisite inputs for Market Equilibrium Module	31
4.2	Data Collection from ME Module using for-loop	32
4.3	Ride Sharing Demand Processing	37
5.1	Unterschleißheim Network	39
5.2	Unterschleißheim TAZs	40
5.3	Convergence Plot - Initial Guess	44
5.4	Detour Time Comparison	45
5.5	Waiting Time Comparison	46
5.6	Convergence Plot - Best Optimization	46
5.7	Munich Network	47
5.8	Munich TAZs	48
5.9	Detour Time Comparison	49
5.10	Waiting Time Comparison	50
5.11	Convergence Plot - Best Optimization	50
5.12	Computational Time Comparison	52
5.13	Computational Time Comparison - Impact Assessment Analysis	53
5.14	Mode Choice Application - Mode Share	54
5.15	Mode Choice Application - Mode Probabilities	54

List of Tables

2.1	Summary of the literature on MoD services	12
4.1	Simulation platform modules summary	27
4.2	Simulation Database Format	30
4.3	ME Dataset Format	31
4.4	Comparison Dataset Format	33
5.1	Simulation Setup Fixed Parameters	41
5.2	ME Module Parameters - Initial Guess	41
5.3	ME Output (Initial Guess)	41
5.4	Comparison Dataset (Initial Guess)	42
5.5	GOF Evaluation - Initial Guess	43
5.6	Convergence Results - Initial Guess	43
5.7	Summary of Optimization Runs	44
5.8	Comparison Results - Best Optimization	45
5.9	Convergence Results - Best Optimization	47
5.10	Summary of Optimization Runs	49
5.11	Comparison Table	49
5.12	Convergence Results - Best Optimization	51
5.13	Run Times Comparison - Munich	52

1 Introduction

1.1 Background and Motivation

With the ever-increasing traffic related issues, it is impractical for each passenger to drive or be driven in an individual car. Population growth and relatively higher purchasing power points towards the immense challenge faced by transport planners of reducing the popularity of privately owned vehicles. As of 2017, 63 million motor vehicles are registered in Germany, among these cars constitute to 46 million, out of which around 90% are privately registered (Kuhnimhof, 2017). These vehicles do not only add up to already existing congestion issues but are also major contributors to air pollution (Colvile et al., 2001). The direct and indirect costs caused by congestion on German roads amounted to 80 billion euros in 2017, which amounts to about 1,770 euros per motorist (Local, 2018). With these figures in mind, it is evident that the solution does not lie in developing new infrastructures or issuing more cars rather a shift towards sustainable and environment-friendly modes is inescapable. To promote these modes a comprehensive approach enveloping, policy, design and psychological components is needed (Cools et al., 2009). In order to make a quantifiable change, similar or at the very least comparable, benefits as private transportation must be offered to individuals in other modes of transportation as well, to expect a shift in behavior.

A study conducted by (Linda, 2003) shows that fervent car users are unwilling to consider public transportation modes because of the independence and convenience a privately owned vehicle offers. Following timetables, managing access to bus, tram or subway stations and traveling with strangers especially during rush hours in over-crowded vehicles on a daily basis are a few traits not every individual is ready to undertake to solve climate change or congestion problems. Therefore, this is a high time that we re-think public transportation in the light of modern technology. Many new ideas have been introduced in the context of urban mobilities in the past couple of years. Recently, the large-scale usage of smart-phones and decrease in cellular communication costs has resulted in emergence of new on-demand mobility, which is also known as car-sharing, ride-sharing(car-pooling, van-pooling), ride-sourcing or e-hail services. With the advent of autonomous vehicles (AVs) it is envisioned in a study conducted by (Arbib & Seba, 2017), that by 2030, 95% of US passenger miles traveled will be served by on-demand autonomous electric vehicles owned by fleets. This business model is called *transport-as-a-service (TaaS)*. The fundamental idea of on-demand mobility is based on *shared economy*, where a desired commodity is shared amongst individuals rather than a single person owning it. These user-centric services are transforming the urban mobility by providing convenient transportation options to anyone, anywhere and anytime. Besides privately owned vehicles and taxis, other public transportation does not offer last-mile services. Taxis do not fall in the budget bracket for most of the daily commuters. By

providing taxi-like last-mile services and being reasonably priced the mobility on-demand (MoD) services can become very popular. These services can have tremendous positive effects not only in mitigating congestion but also by reducing other potential external negativities such as vehicular emissions (Alonso-Mora et al., 2017). MoD users can also benefit from shared travel costs while experiencing the convenience of a car, travel time can be reduced by using high-occupancy lanes and often perks such as preferential parking can be availed (Greenblatt & Shaheen, 2015).

The concepts of shared mobility, in their earliest forms, can be seen applied in transportation sector throughout the history (Shaheen et al., 1998). Ride-sharing services made their first appearance in North America as early as 1940s in an effort to preserve resources for war, they re-emerged again in 1970s during the oil crisis (Greenblatt & Shaheen, 2015). However, these concepts experienced radical change in their perception in the wake of advancements in mobile services. Smartphone and internet evolution opened new prospects for ride-sharing services, as communication was not dependent on time of the day or day of the week anymore, but the possibilities now extended to real-time access of large quantities of data. Numerous applications made their way to market which allowed activities to be scheduled ahead of time (Boutueil, 2018). These ideas revolutionized ride-sharing services such as carpool and van-pool, as they aim at increasing vehicle occupancy by grouping together users in one vehicle with similar origin or destination or both, resulting in reduced number of cars on road.

Nevertheless, for the most part, the implementation of MoD services in an urban mobility setup has still remained a disorganized activity, as a consequence such innovative services are still not well established and many startups/service providers failed to follow up for a sustainable business model. Hence, users also haven't established trust on using such services and rather choose more reliable modes. The emerging popular modes signify research needs from user's perspective to the opportunity to utilise ride-sharing services for personal commute. The conventional mode choice models rely on utility functions for each mode. These functions are derived based on the mode's service attributes and user's perception. However, this method does not give realistic results when applied to ride-sharing as a mode, the reason lies in their dynamic service attributes, making mode-choice not straightforward for ride-sharing modes. Wide adoption of these services can only be made certain after proper calculation of its mode choice based on waiting and detour times of users and fleet size required to serve the demand ensuring a minimum set Level of Service (LOS) served.

1.2 Objectives

Exhaustive research has been conducted trying to quantify the impact of MoD services in urban environment, determining the mode choice of these services has unfortunately not received the required attention and is still open to research.

One has to understand that for services dynamic in nature such as ride sharing, their utility depends on the service they provide and it is perceived by users mainly in terms of detour they are facing or the the waiting time spent for the pickup. The current state of the system

keeps changing thus the utility of the service. To apply conventional mode choice models, an equilibrium state has to be achieved where for a certain supply, demand of the ride sharing services is ascertained. In a nutshell, the standard mode choice model might have to be repeated multiple times until an equilibrium state is achieved and final mode choice is performed. This multiple repetition of the process means that several simulation runs are required in order to obtain the equilibrium state, which is computationally expensive, especially keeping in mind the fact that the equilibrium tends to change for every change in a supply, therefore even if the fleet increases or decreases for a same network, the ride sharing demand will be different thus a new equilibrium state needs to be calculated. Therefore, the main objective of this thesis is to develop a mode choice method for dynamic van-pooling services where the need to run an equilibrium for every change in supply and extensive computational efforts shall be avoided. Specifically it includes the following:

- Understanding the concept of mode choice for mobility on-demand services.
- Developing a framework to integrate an existing ride-sharing market equilibrium model with a simulation platform.
- Developing a calibrated analytical model viable for varying service attributes to perform mode choice.

1.3 Thesis Structure

The overall approach adopted for this research and the results concluded are presented in the rest of this thesis. Chapter 2 reviews the literature on available mobility on-demand services, modal-split models' evolution over the years and the models currently in practice to represent mode-choice for MoD services. It also provides an extensive review on works done in the area of market equilibrium specifically talking about ride sharing market equilibrium model, highlighting the literature gaps at the end. Chapter 3 formulates the methodology used for this study, it describes the framework to integrate an existing ride-sharing market equilibrium model with a simulation platform to perform the mode choice of ride sharing services and it describes the process of calibrating the model's parameters. Chapter 4 describe the experimental design setup, problems and errors faced during the execution process with the platform and adoption of the alternative approach to complete the study. Chapter 5 represents the results obtained for a synthetic network and then the case study network of Munich. Finally the chapter 6 concludes the results and provide direction for the possible future research.

2 Literature Review

This chapter is further divided into three main sections. First section addresses in detail the mode choice, the types of mode choice models and how these have evolved over time. Second section discusses the research that has been conducted on ride-sharing services, their modeling and determining the mode-choice techniques used to determine their modal-split. The third section goes in depth about Market Equilibrium (ME), how over the years researchers have tried to establish ME for taxi-markets and only recently for ride-sharing services. The aim of this chapter is also to comprehend and highlight the research gaps in the area of determining mode choice and establishing market equilibrium for ride-sharing services.

2.1 Mode Choice

Mode choice is one of the most critical steps of the famous four step model. It is substantial to transport planners as it affects how much space on ground will be allocated to each mode of transportation. Being able to identify and model those attributes which influence traveler's mode choice is paramount. (de Dios Ortúzar & Willumsen, 2011) classified the factors influencing the mode choice into three groups; Characteristics of the traveler, characteristics of the trip and finally mode characteristics.

In USA, previously, mode choice models were created with the assumption that the most deterministic feature is traveler's personal characteristics, which gave little authority to policy makers to influence their modal decision. It was required to incorporate the characteristics of other modes in the model additionally to make traveler's mode choice more policy sensitive. This approach also changed the sequence of four step modeling, as mode choice was calculated immediately after trip generation which was in contrast to what was being practiced in Europe. In Europe, modal-split was determined after the distribution step. This procedure had the advantage of including the attributes of alternate available modes as well as the trip itself but traveler's personal characteristics were lacking as they may have already been aggregated in trip matrix. (de Dios Ortúzar & Willumsen, 2011)

2.1.1 Sequential Models

Sequential models use four-step modeling as their fundamental procedure. However, in the past, trip characteristics were not included in the model other than the elemental ones, such as travel time (Goodwin, 1977). Diversion curves were used to detect the proportion of trips made using similar mode against cost or difference in time. Another initial heuristic modal split model utilized a version of Kirchoff's electricity principal which resulted in graphs very similar to the graphs obtained as a result of Logit equations (Gaudry & Quinet, 2012). The

main drawback of these Logit models was their inability to link trip attributes with trip's nature. There were also constraint in capturing the detailed characteristics of other modes available to individual users.

Entropy-maximisation models estimated trip distribution and modal-split simultaneously. A big positive of this approach was the possibility that these models could easily be extended for multiple modes (Wilson, 1969). However, accurate results could only be expected if the equation proposed by (H. C. Williams, 1977) held true, which inherently indicated that cost plays a relatively more important role in choosing the mode than it does in choosing the destination. Modelers failed to satisfy this equation in abundance which resulted in models producing irrational results. In case the aforementioned equation was not satisfied, modal-split had to be done immediately after trip generation for realistic results. This was contrary to the well accepted way of starting with trip generation, which is succeeded by trip distribution, proceeded by modal split and finally trip assignment takes place (Solli-Sæther & Gottschalk, 2010).

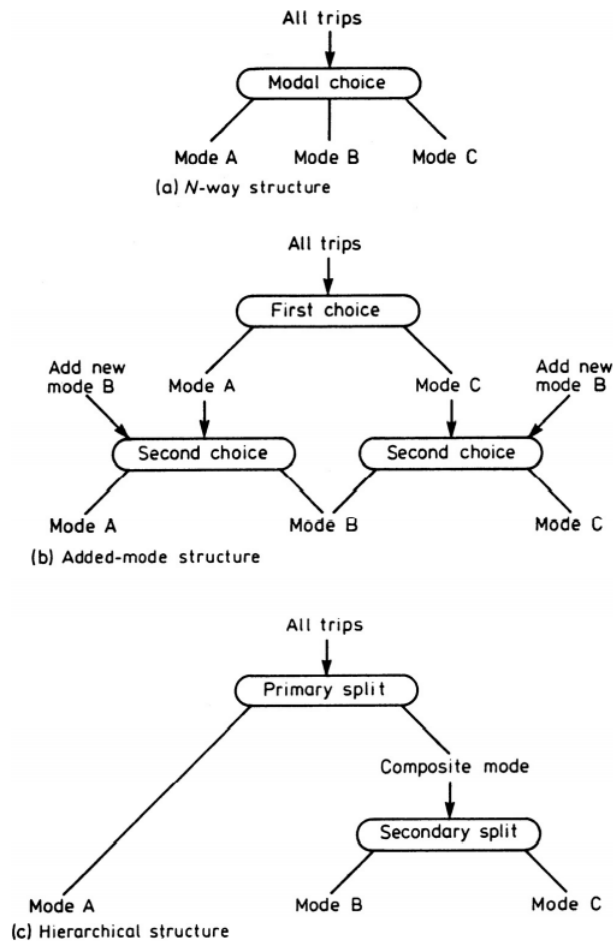


Figure 2.1: Multimodal Model Structures (de Dios Ortúzar & Willumsen, 2011)

Gradually travelers were equipped with a wider range of choices and it was crucial to incorporate these choices in model structures as well. Three multimodal-split models are discussed in (Ortuzar, 1980a), namely N-way structure, added-mode structure and hierarchical structure. N-way structure being the simplest, makes the mistake of not taking into account the extent of correlations of different modes, problems arise when some modes are more similar in nature than others. The second type discussed is added-mode structure, which adds another mode in the secondary level but it has shown inconsistent results depending on which mode is taken as the added one (Langdon, 1976). The third structure discussed is the hierarchical or nested structure, which groups together modes having similar features such as Public Transport.

Calibration of nested structures was started at a secondary split and moved up till main mode. It used to be done using maximum likelihood method as this method was found to be more reliable (Hartley & Ortuzar, 1980) and efficient than least square methods (Domencich & McFadden, 1975).

2.1.2 Direct Demand Models

This approach does not follow the conventional four-step sequential modeling, consequently it avoids some of the drawbacks of the regular method. This model is further divided into two types; purely direct and quasi direct. In purely direct a single equation is used to estimate the relation between travel demand and mode, trip and user attributes. Quasi direct method segregated the mode choice and origin-destination travel demand. These models were originally based on multiplicative functions. One of the first models, SARC used a multiplication function to link travel demand to socio-economic attributes of each zone, incorporating the level of service provided by each mode (Kraft, 1968). The only turn down of demand model is the fact that a vast number of parameters are required to produce utilizable results. (Domencich et al., 1968) suggested another form of direct demand model by introducing linear and exponential terms besides multiplicative ones.

Direct Demand models are particularly useful for areas which have larger zone areas. In their recent versions, household data is used to evaluate a combined frequency-mode-destination using the structure of Nested Logit Model (de Dios Ortúzar & Willumsen, 2011).

2.1.3 Discrete Choice Models

(Warner, 1962) and (Oi & Shuldiner, 1962) highlighted the downsides of using the conventional aggregate models early in 1960s. In spite of that, aggregate models were still commonly used in transport projects until 1980s (H. Williams, 1979), when finally discrete choice models or Disaggregate Models (DM) as they were then called, started receiving attention because of an extensive comparative study conducted by (Spear, 1977). In his study it was concluded that Disaggregate Models had precedence over aggregate models, as they could be applied at any aggregation level, and were transferable in time and space. All variability of data was utilised in DM, as individual data was used, resulting in more efficient and less biased models. Utility

functions used in these models allowed for any number and specifications of explanatory variables which resolved issues such as inclusion of policy variables.

Discrete Choice Models dictate that *the probability of an individual choosing an option is a function of his socioeconomic characteristics and the relative attractiveness of the mode* (de Dios Ortúzar & Willumsen, 2011). The term *utility* was used, which was defined as a theoretical construct that the individual seeks to maximise while choosing an alternative. The utility function has constants assigned to particular features defining their weights, these features are both related to the individual and the alternative. An additional *alternative specific-constant* tries to capture the net-influence of all the unperceived characteristics. An individual selects the maximum utility alternative.

Random utility theory is the most common hypothetical structure of discrete choice model (Domencich & McFadden, 1975), and is based on equation 2.1. Net utility U_{jq} is composed of two parts; a measurable part V_{jq} and a random part ε_{jq} that considers the idiosyncrasies, where j refers to an alternative and q to an individual.

$$U_{jq} = V_{jq} + \varepsilon_{jq} \quad (2.1)$$

Multinomial Logit Model (MNL)

Multinomial Logit Model (MNL) is the most pragmatic discrete choice model (Domencich & McFadden, 1975) as it satisfies the axiom of IIA (Independence of Irrelevant Alternatives) and has a simple covariance matrix. It focuses on the difference of utilities between two alternatives rather than the utilities themselves. The unique feature of MNL regarding IIA contributed to its popularity in the past, as MNL was availed to discover the new mode share after a new alternative was introduced to public. However, in late 2000s serious disadvantages of this property came to light in the presence of correlated alternatives (de Dios Ortúzar & Willumsen, 2011).

Nested Logit Model (NL)

In scenarios where alternatives are dependent, or the variance of error terms is not uniform MNL results in far too simplistic models. To overcome this drawback, an extension of MNL was proposed which came to be known as Nested Logit (NL) Model. NL can accommodate varying degrees of similarities amongst alternatives. First exhaustive analysis on NL structures was done by (H. C. Williams, 1977) with a focus on composite utilities. Meanwhile (Daly & Zachary, 1978) developed the fundamental theories elucidating Nested Logits. Figure 2.2 depicts a very popular application of NL with two levels.

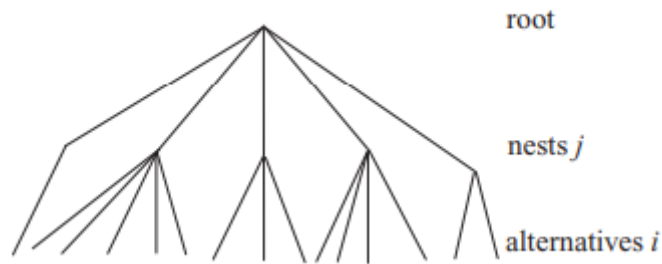


Figure 2.2: A General Nested Logit Structure (de Dios Ortúzar & Willumsen, 2011)

The practical implementation of NL was studied in detail by (Ortuzar, 1980b) and (Sobel, 1980). Each nest is represented by a *composite alternative*, whose utilities is used as a means to carry the details from lower nest to the next higher nest. In case of more nesting levels, NL models can be extended to more hierarchical levels as shown in figure 2.3.

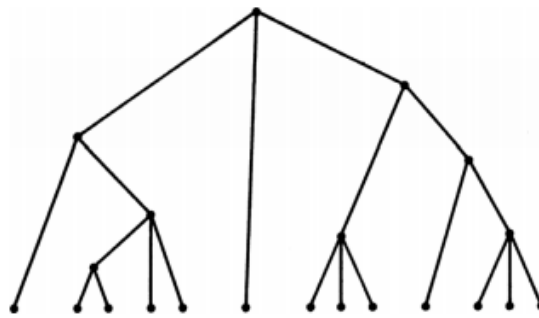


Figure 2.3: Nested Logit model with several nests(de Dios Ortúzar & Willumsen, 2011)

One of the major disadvantages of NL model is its dependency on hit and trial method while deciding the best nesting pattern, as the number of possible structures increases geometrically with the number of alternatives (Sobel, 1980).

Other Mode-Choice Models

Multinomial Probit Model (MNP) was introduced to address the problem of simple covariance matrix in MNL by normally distributing the stochastic residual with mean zero and an arbitrary covariance matrix. As a result, simulation is needed to solve this model (Van Can, 2013).

The widely-used model currently is Mixed Logit Model (ML). It deals with the logit model limitations by allowing for the random taste variation among individuals, which is a very serious issue as shown by (Horowitz, 1981). The original formulation of ML dates back to 1970s and is attributed to (N. Cardell & Reddy, 1977) and (N. S. Cardell & Dunbar, 1980).

However, its present form is the outcome of research of (Bolduc & Ben-AkiWand, 1996) and (McFadden & Train, 2000).

Hybrid Choice Models have been proposed in an effort to include subjective elements in discrete choice models. They account for the intangible attributes of alternatives such as individual's personal emotions etc. by introducing latent variables (Morikawa & SASAKI, 1998), (Ashok et al., 2002), (Abou-Zeid & Ben-Akiva, 2010).

While these models are fairly suitable to determine the mode-choice for traditional modes of transportation, they do not produce practical results when applied to new emerging modes such as Mobility on Demand (MoD). Therefore, there is a dire need to make adjustments to already existing models to fit to new developing modes, as the utility function for such modes does not remain constant but changes for every run because of their dynamic service attributes.

2.2 Mobility on-Demand Services

Mobility on-demand (MoD) have gained popularity over recent years since the advancement in technology sector, as now various on-demand mobility services can be accessed instantaneously through hand-held devices. These services are rapidly changing urban mobility, providing an alternative to public transport. By the term ride-sharing we refer to a system in which the goal is to utilize an individual vehicle for multiple user trips, hence increasing its utility. A variety of services such as ride-hailing, Car sharing, ride sharing and taxi-sharing can be categorized as on-demand mobility. These services also have a measurable impact in the reduction of traffic congestion on road and have numerous environmental benefits over private cars and ride-alone taxis (Cai et al., 2019).

2.2.1 Modeling MoD services

Over the years many attempts have been made to simulate on-demand services effectively. In the study conducted by (Zhang et al., 2015b) SAVs are simulated using agent based modeling with dynamic ride-sharing. Based on available household data, similar travel profile trips are assigned to SAVs in the modelling system. (Ma et al., 2013) tried to develop a practical framework for managing a large-scale MoD fleet, he devised a heuristic-based taxi dispatching strategy and fare management system that could handle operations of large urban scaled fleets. (Levin et al., 2017) developed a framework for existing traffic simulation models to incorporate the behavior of SAVs. The framework compares the SAVs scenarios with personal vehicle trips for the study network. (Bischoff et al., 2017) modeled the taxi-sharing services by extending a MATSim model which was previously used for non-sharing taxi trips only. MATSim extensions for simulating on-demand transport modes have been used for ride-sharing taxi service. Similarly, (Lokhandwala & Cai, 2018) used AnyLogic for agent-based modeling of shared autonomous taxi services in New York City. An agent-based model for SAVs was introduced by (Fagnant & Kockelman, 2014). In their work they model SAVs for a given OD-demand with a reasonable fleet size and investigate the relocation strategies for

SAVs to minimize waiting times faced by passengers. (Fagnant & Kockelman, 2018) further extended this model to incorporate the concept of Dynamic Ride sharing (DRS), in which a single SAV can be pooled by multiple users having similar trip attributes.

(Dia & Javanshour, 2017) studies the impact of SAVs on Melbourne mobility patterns using agent-based modeling. Another agent-based MATSIM model developed for SAVs is studied by (J. Liu et al., 2017) for Austin Texas. Their work studies the effect of different fare levels for the SAVs service on person trips modal split. (Zhang et al., 2015a) study the impact of SAVs on urban parking demand. A Similar simulation based study by (Fiedler et al., 2017) explores the SAVs impacts on congestion.

Extensive research has been conducted studying the impacts of implementing shared taxis in urban mobility environments. However, not the same can be stated in case of ride sharing services despite their promising characteristic to reduce the traffic congestion substantially. According to the concluded results of the study conducted by (Alonso-Mora et al., 2017), the total fleet of 13,000 taxis for New York city can be replaced by 2,000 10-seater vehicles, with 98% service rate. Considering such precedence over other modes, some efforts have been made to integrate vanpooling services in urban mobility setting.

In a study conducted by (Martinez & Viegas, 2017), on assessing impacts of shared mobility on urban mobility infrastructure of Lisbon Portugal, an autonomous taxi-bus in conjunction with autonomous shared-taxis was used. The concept of taxi-bus used in this study is similar to a vanpool service, where the users can reach their destination without transfers and travel time is comparable to a private car's. The effect of shared mobility is studied in a scenario where autonomous shared-taxis and taxi-bus services replaced all other transport modes. (Alazzawi et al., 2018) implemented the concept of self-driven robo-taxis as shared autonomous vehicles to study their impact on urban mobility. The seating capacity of these robo-taxis has been predefined as 6 passengers per taxi. The concept of DRS introduced by (Fagnant & Kockelman, 2018) for SAVs in which multiple users with same origins, destinations and departure times can pool a single vehicle. Although these studies give us an insight about the impact of vanpool-like ride-sharing services on urban mobility, however the implementation of these services have some limitations, such as smaller capacity (Alazzawi et al., 2018), inability to accommodate detour times (Fagnant & Kockelman, 2018) or the fact that the study cannot be extended beyond its scope (Martinez & Viegas, 2017).

2.2.2 Mode Choice of Ride-sharing Services

With the increase in popularity of ride-sharing services, many efforts have been made over the last few years to understand, design and implement these services in a more efficient way. It is crucial to recognize here that mode-choice of ride-sharing is not a one time process as it does not remain constant rather is highly dependable on the service attributes provided to the users dynamically, based on the current state of the system.

However, most of the literature mentioned in section 2.2.1 ,assumes a fixed exogenous demand for ride-sharing services with the underline assumption that demand for these services remains the same over a period of time. In reality these services co-exist with other transport modes and based on the service attributes the mode choice should be updated

dynamically. Some of the studies can be found in literature (mentioned subsequently) which have tried to include the concept of dynamic demand based on the state of the system.

The effects of MoD on urban mobility have also been tried to quantify and control better by simulating these services on MATSim (Hörl, 2017) and SimMobility platforms (Araldo et al., 2018). They have used historical data to update the demand unlike many studies where the demand is fixed. In the works done by (Djavadian & Chow, 2017a) and (Djavadian & Chow, 2017b), a day-to-day process is implemented to have supply-demand interactions dynamically update the demand for the MoD service.

Integrating mode choice in the design of MoD system is a study by (Y. Liu et al., 2019), where a unified framework is developed which integrates the mode choice models with an on-demand mobility system. In their study the users can choose between available MoD options with varying seating capacity and public transport. The underline assumption of the MoD services is that they are all run by the same operator with no competition amongst them. Their proposed framework can be separated into inner and outer loops as shown in figure 2.4. The outer loop iterates and optimizes the supply-side parameters using Bayesian Optimization, the objective function of outer loop is based on the equilibrium criterion. The inner loop keeps updating the mode-specific attributes and feeds them to the choice model, based on which new mode choice is generated for each iteration until equilibrium is achieved. This study uses data from stated preference survey conducted on New York population for on-demand mobility services.

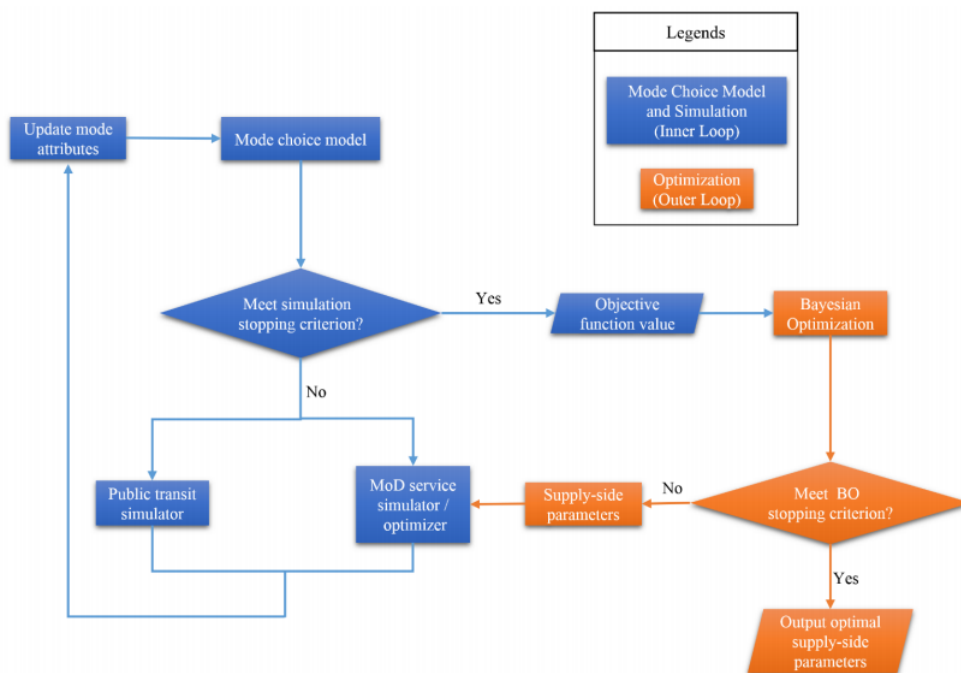


Figure 2.4: A framework to optimize the supply-side parameters for MoD system with the integration of the mode choice model (Y. Liu et al., 2019)

However, a simulation-based mode choice framework as developed by these studies restricts the mode choice equilibrium for the given set of service attributes, its viability against changing service attribute values is very less and requires computationally expensive simulation runs to achieve the equilibrium.

Table 2.1: Summary of the literature on MoD services

Literature	MoD Service	RS	SB	DSI
Ma et al., 2013	Taxi			
Fagnant and Kockelman, 2014	SAVs			
Zhang et al., 2015a	SAVs			
Zhang et al., 2015b	SAVs	+		
Dia and Javanshour, 2017	SAVs	+	+	
Levin et al., 2017	SAVs			
Bischoff et al., 2017	Taxi		+	
Fiedler et al., 2017	SAVs		+	
J. Liu et al., 2017	SAVs		+	
Martinez and Viegas, 2017	SAVs	+		
Hörl, 2017	SAVs		+	+
Djavadian and Chow, 2017a	SAVs			+
Djavadian and Chow, 2017b	SAVs			+
Fagnant and Kockelman, 2018	SAVs	+		
Lokhandwala and Cai, 2018	SATs		+	
Alazzawi et al., 2018	SATs	+	+	
Araldo et al., 2018	SAVs		+	+
Y. Liu et al., 2019	SAVs	+		+

Note: RS Ride Sharing (≥ 2 sharing customers per vehicle); SB Simulation-based modelling; DSI Demand Supply Interaction; SAVs, Shared autonomous vehicles; SATs Shared autonomous Taxis; '+' means that the criteria is true

2.3 Market Equilibrium

Market equilibrium is a stable state of the system where the supply-demand interaction is balanced out mathematically. Not much literature can be found in the area of establishing an equilibrium market state for ride-sharing services, however, some substantial works on taxi market equilibrium have been published. In this section we will discuss slightly the works done in taxi market equilibrium, and discuss a recently proposed market-equilibrium for ride-sharing services also considering passenger preference.

2.3.1 Equilibrium Model for Taxi Market

(S. C. Wong & Yang, 1998) were one of the first ones to circulate the idea that aggregate models cannot capture the true influence of road network structure, since taxi movements take place over and over and a network model can better reflect reality. Hence, they initially formulated an optimization network model, based on shortest travel path adopted by taxi driver to drop off and pick up a customer, thereby reducing empty taxi time. From this, gravity distribution of empty taxis is derived. This model was further improved by (K.-I. Wong et al., 2001), by incorporating network congestion and demand elasticity first. Later on, (K. Wong et al., 2002) proposed a sensitivity-based solution algorithm to solve the congestion model more efficiently. However, with the advent of smart taxi systems, these models failed to deliver reality-based results.

In a relatively recent research by (Qian & Ukkusuri, 2017), taxi service is divided into two groups namely, Traditional Taxi Service (TTS) and App-based Third-party Taxi Service (ATTS). This study, models the taxi market at network level as a multiple-leader-follower game and investigates the equilibrium with competition (TMC Equilibrium). Based on numerical results, it is observed that fleet size and pricing policy are closely associated with the level of competition in the market and may have significant impact on total passengers' cost, average waiting time, and fleet utilization. However, passenger preference is still not incorporated in this model rather customer's greedy behavior is assumed.

2.3.2 Ride-Sharing Market Equilibrium

In a recent study, (Ke et al., 2020) extended a taxi market based equilibrium model for ride-sharing market with the inherent assumption that the ride sourcing vehicles will be providing the ride pooling services. However for simplification the model only paired two users at a time for a single ride. (Lu, 2020) in his recent study proposes a market equilibrium model for ride-sharing services in the multi-modal transport context by applying the Multinomial Logit (MNL) model to present the ride sharing demand for the given passenger preference data. The model also estimates the detour and waiting times for the network analytically. Figure 2.5 illustrates the relation between exogenous and endogenous variables of this equilibrium model.

In this market equilibrium model, trip fare and vehicle fleet size are exogenous variables, while detour time, waiting time, vacant seats and ride-sharing requests generated are all endogenous variables. Passenger preference is dependent on trip fare and other transport modes' attributes. Waiting time is assumed to be dependent on seat availability keeping in mind the sharing nature of this service, vacant seats available are directly related to fleet size, ride-sharing requests generated control the detour time. Because of high inter-dependencies, vacant seats available can also indirectly affect the detour time.

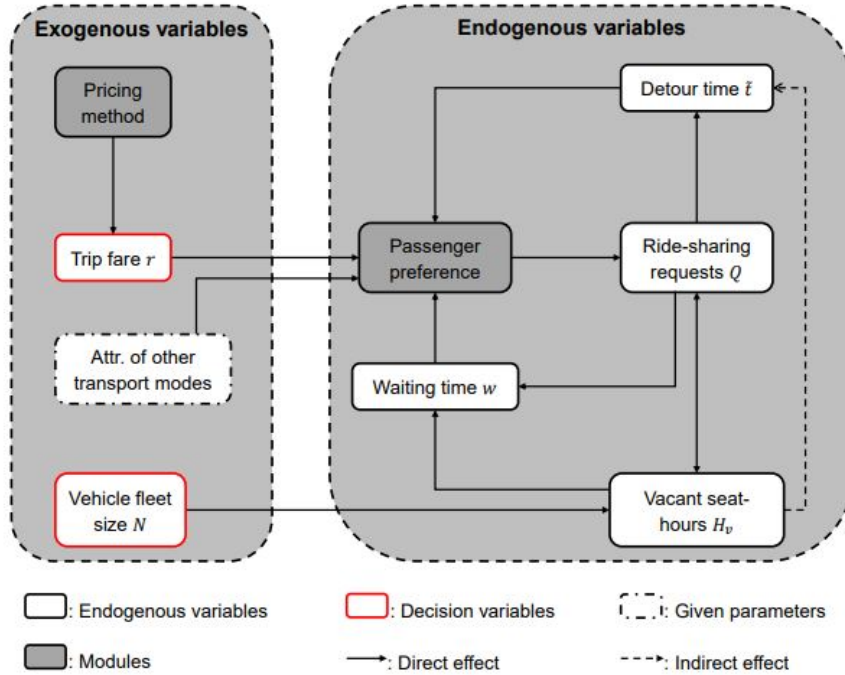


Figure 2.5: Relationship between endogenous and exogenous variables (Lu, 2020)

Market Equilibrium model is doing service availability based equilibrium as well by performing the mode choice. To find how many requests are served and what are the service attributes. An equilibrium state between the service attraction and service attributes is calculated by the ME model. This is the similar idea of performing a mode choice what this study aims to achieve for the ride sharing services and to replace regress simulation computations by a calibrated analytical model.

2.4 Literature Gaps

Following the available literature review of ride sharing services in section 2.2 and the ride sharing equilibrium model in section 2.3.2. These literature gaps can be extracted

- Few studies are available on the modelling of micro-transit services in a multi-modal network having a higher capacity than conventional taxis.
- The framework of dynamic mode choice for MoD services is still an open question with very less research efforts.
- Current mode choice methods for MoD services requires extensive computational effort (requires equilibrium state each time).
- None of the studies in discussion integrated and calibrated a market equilibrium model with a conventional simulation based mode choice method.

3 Methodology

This chapter presents the methodology adopted to achieve the objectives of this thesis. The methodology is explained in three sections. First section deals with modelling of dynamic van-pooling services, briefly explaining the routing optimization algorithm and functionality of the simulation platform used. In section two we discuss the market equilibrium model used for this study. Finally the last section covers the details of ME-based mode choice method, explaining the steps involved in developing the method, how a simulation database is created and is used to calibrate the market equilibrium model.

3.1 Modeling Dynamic Van-pooling Services

Many efforts have been made to model the mobility on-demand services but there is very less effort done to model the dynamic van-pooling services which involves a dynamic traffic assignment and real-time service optimization (Qurashi et al., 2020). Dynamic and stochastic nature of dvanpool service requires a continuous route scheduling. As the new requests are generated, the van needs to be routed to the nearest request and serve the request with best route possible to generate maximum revenue. The scheduler is in place to perform this routing optimally. The following section will discuss the general framework of scheduler and the simulation platform used to model the dynamic van-pooling services.

3.1.1 Routing Algorithm

Scheduler used is based on algorithms for dynamic van-pooling services developed by (Li et al., 2019). Its basic framework can be divided into two parts, evaluation procedure and scenario based search. Evaluation procedure handles the requests that are sent to scheduler while the scenario based search determines the best route for each individual van.

Initially the scheduler receives demand requests of new passengers and by means of real-time interaction with simulation environment, scheduler optimizes and finds the closest van best suitable for the pickup. Scheduler functionality includes the evaluation criteria which decides whether the request is to be served or not. At an event of a new request, it is by default rejected by the scheduler. The evaluation procedure as described in Algorithm 1 by (Li et al., 2019) compares the average objective function values of accepting and rejecting the requests respectively. If accepting the request results in a higher value of average objective function, the request is accepted otherwise remains rejected.

Algorithm 1: Evaluation Procedure (Li et al., 2019)

```

state(t) ← < V, R >
for each scenario  $s_k(t) \in S(t)$  do
  |  $obj_k(t) \leftarrow$  objective function value of SOLVE(state(t),  $s_k(t)$ )
end
return the average value of each  $obj_k(t)$  as  $\overline{obj}_k(t)$ 

```

Figure 3.1 illustrates the functionality of Algorithm 1. When the average objective function of accepting the request is higher than rejecting it, scheduler then accepts the requests and move on the the next step of scenario based search where it decides the route of the selected van.

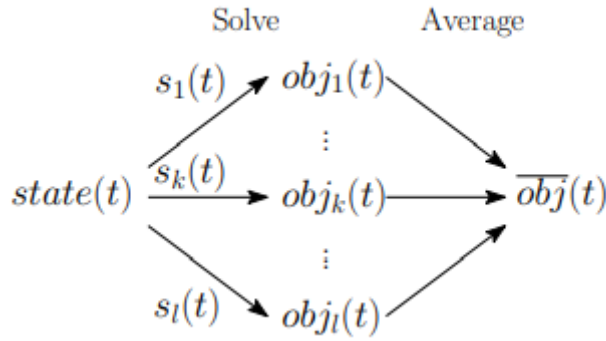


Figure 3.1: Illustration of the evaluation procedure (Li et al., 2019)

Deciding on the route that shall be assigned to van is done in scenario-based search. In brief words, the general methodology of this algorithm as shown in Algorithm 2 is to take into account the current state of the system , and then look at all the possible routes which can be assigned to vans, The algorithm loops through all the possible scenarios, updates the state of the system according to each scenario and then evaluates it based on the objective function value. The best scenario is then selected as the route to be assigned to the van.

Algorithm 2: Scenario-based search (Li et al., 2019)

```

state(t) ← < V, R >
for each scenario  $s_k(t) \in S(t)$  do
  |  $decision_k(t) \leftarrow$  decision of SOLVE(state(t),  $s_k(t)$ )
  |  $state_k(t + \Delta t) \leftarrow$  UPDATE(state(t),  $s_k(t)$ ,  $decision_k(t)$ )
  |  $\overline{obj}_k(t + \Delta t) \leftarrow$  EVALUATE(state $_k(t + \Delta t)$ )
end
return  $decision_k(t)$  with best  $\overline{obj}_k(t + \Delta t)$  as the final decision  $decision(t)$ 

```

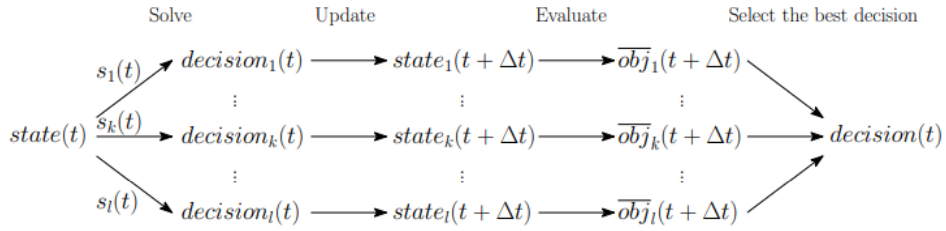



Figure 3.2: Illustration of the scenario-based search (Li et al., 2019)

3.1.2 Simulation Platform

In addition to scheduler, a ride sharing simulation platform developed by (Qurashi et al., 2020) is used to model the dynamic van-pooling services. Simulation platform takes in the operational parameters such as fleet size, seating capacity, unit price and initializes the vans on network at selected initial meeting points. On the other hand, demand requests are generated dynamically by the demand module for given network OD demand. The passenger preference module inside demand module takes in the scenario inputs such as the service attributes and based on the utility functions of the different transport modes, it performs the mode choice for the dynamic van pooling service. The demand is then written in trip request files and communicated with the scheduler during simulation runs. Based on this initialization information, simulator in conjunction with scheduler serves the requests.

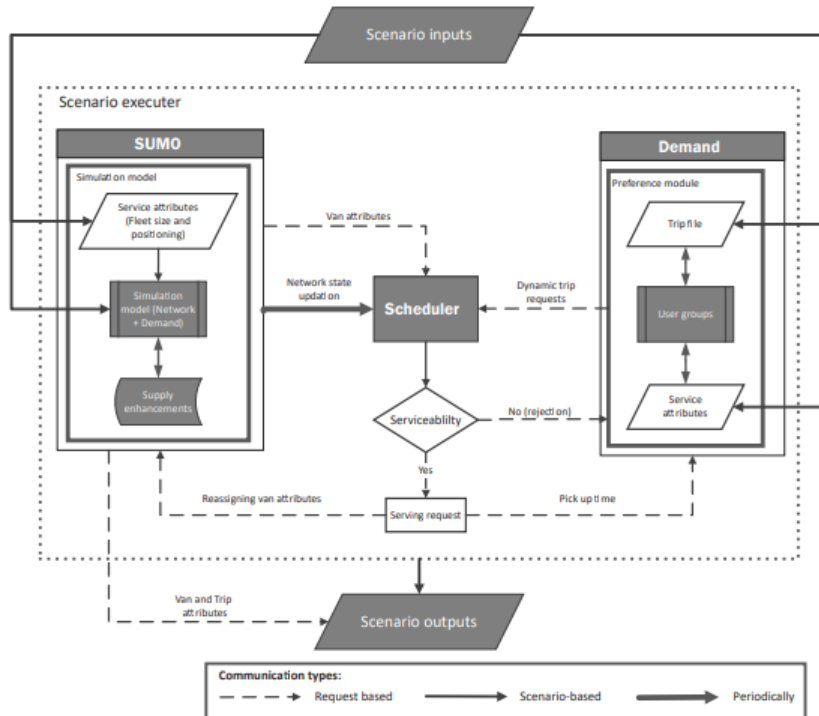


Figure 3.3: Modelling Architecture of Simulation Platform (Qurashi et al., 2020)

The python-based scenario executor of the simulation platform interacts with C++ based scheduler via dynamic link library. The further integration of scheduler with the simulation platform and translation of data among the two platforms is illustrated in figure 3.4.

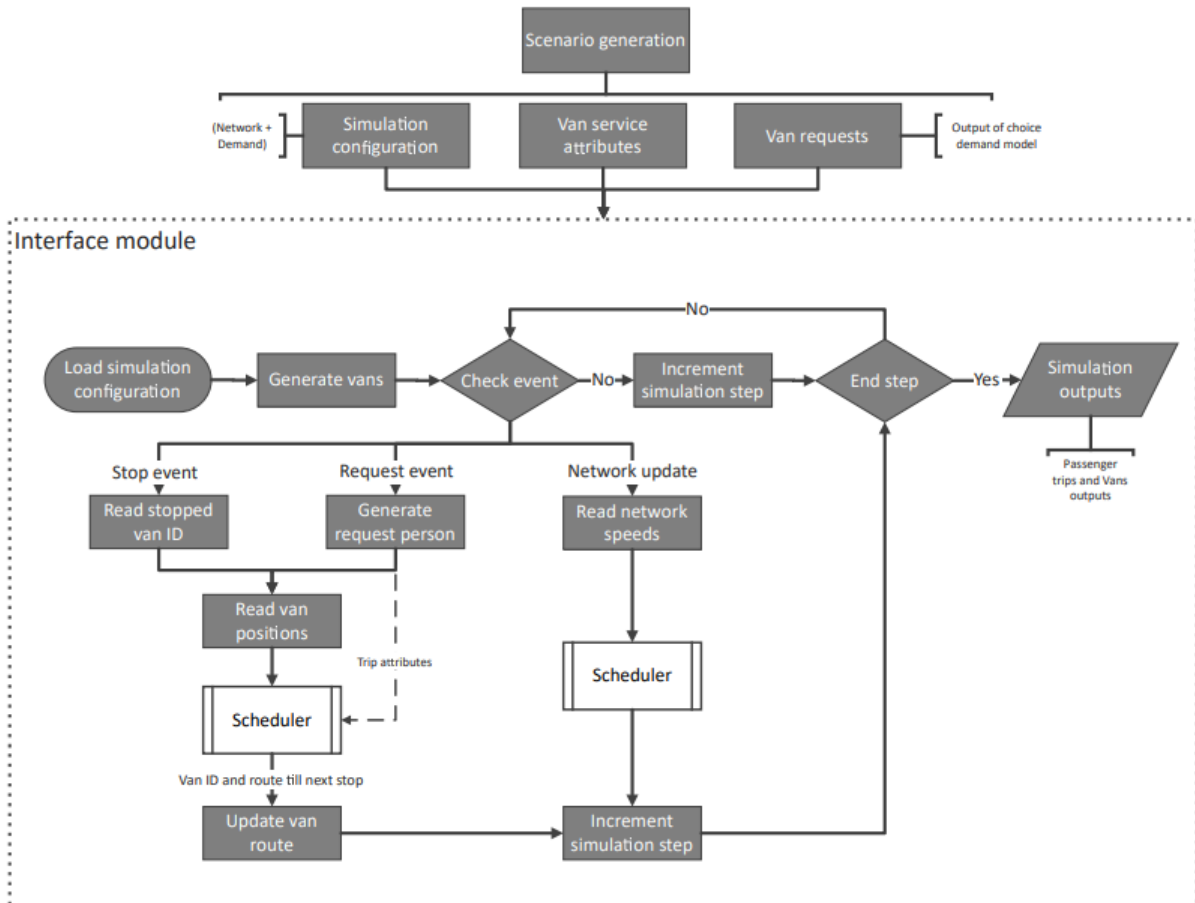


Figure 3.4: Scheduler Integration with Simulator (Qurashi et al., 2020)

However in this study we propose to use the ME model as the demand module of the simulation platform and is used to perform the mode choice of the ride sharing service. The present process of performing mode choice in the demand module will be modified by using a calibrated market equilibrium model for ride sharing market. The calibrated model will provide the ride sharing demand for the given supply parameters and network demand. In order to use it as the default mode choice method, it needs to be integrated with the simulation platform and using the calibrated model parameters to perform the mode choice.

3.2 Ride Sharing Market Equilibrium Model

Market equilibrium can be described as the balance state of the system when demand and supply interaction reaches an equilibrium state under certain operation conditions. In other words, the mathematical balancing of supply-demand simultaneous equations is known as the equilibrium state (Lu, 2020). Recently due to advancements in urban mobility, there has been a rise in mobility on demand services. Ride-sharing services have emerged as an alternate to public transport where the user can experience a door-to-door service with fairly less fares as compared to taxis.

3.2.1 Demand Modelling

A market equilibrium model for the ride-sharing services has been developed for a multi-modal network. In his work, the application of a multinomial logit (MNL) model is used to incorporate the passenger preferences.

The market equilibrium model analytically calculates the ride sharing requests along with the network detour and waiting times of the multi-modal network for the ride-sharing service. In his study, (Lu, 2020) has developed a relation of Q with detour time t and waiting time w by introducing model parameters A & B . Equation 3.1 explains the relation of ride sharing requests Q with detour time t . Similarly equation 3.2 shows how waiting time w is estimated.

$$\tilde{t}_i = \frac{A_i \sum_j Q_j t_j^d}{N \sum_j Q_j} \quad (3.1)$$

$$w_i = \frac{BQ_i}{\sqrt{Nn_s - \sum_j Q_j t_j}} \quad (3.2)$$

The complete derivation of these equations has been discussed in detail in (Lu, 2020). As shown in figure 2.5, ride-sharing requests are generated based on the passenger preference module. The passenger preference data used is available through the stated preference survey conducted for the city of Munich in his study by (Tsiamasiotis, 2019). This multinomial logit model based module is used to perform mode choice for the ride-sharing service by taking in the exogenous variables of the system. The generated requests are then used to estimate the waiting time w and detour time t which are then again used as the inputs for this passenger preference module. The above mentioned iterative process is how the equilibrium state of the market is estimated.

3.2.2 Integration with Simulation Platform

The scheduler integrated simulation platform models the demand of the dynamic van-pools by creating requisite text files which contains all the data of individual passenger requests that the platform will serve. The request files are created from the network demand using

a mode choice method. Currently, the ride sharing demand is extracted from the network demand using the utility functions and an equilibrium state is achieved by running multiple simulations. In order to use ME module as default mode choice, it has to be integrated with simulation platform. The integrated ME model will be replaced between the network demand and output request files as a mode choice method.

The global inputs of simulation platform are translated to ME module to model the market equilibrium for the specific study network. ME module calculates the ride sharing requests Q for the equilibrium state of that network. The generated demand requests Q can be modelled on the simulation platform to get the simulation specific attributes.

In order to run ride-sharing requests Q from ME module on simulation platform a requisite request text file is generated. Request trip attributes are communicated with scheduler via the requests files. The request files includes the trip information such as the depart times, potential detour time, and trip fare. Along with the requests demand Q from ME module, additional requisite parameters of potential detour time and fare are also initialized to create these request files. These parameters are calculated by equations 3.3 and 3.4.

$$t_d = T_{ij}\delta \quad (3.3)$$

where:

t_d = potential detour time

T_{ij} = direct travel time between OD pair i and j

δ = detour coefficient expressed as delay per hour

$$P = D_{ij}p\alpha \quad (3.4)$$

where:

P = trip fare

D_{ij} = direct distance between OD pair i and j

p = unit price per kilometer

α = profit ratio

3.3 ME-based Mode Choice Method

3.3.1 General Methodology

The process for mode choice is not straight forward with services having such a dynamic nature. The perceived service attributes by the user are not static. They change based on the availability of the service on the network overtime. Therefore the mode choice also needs to be dynamic with a feedback loop which updates the demand of the ride-sharing service in the network. However, to perform this feedback loop over and over again, extensive simulation runs needs to be performed to achieve the equilibrium state. To simplify this process and bypass the extensive simulation runs to perform the mode choice, this study developed a mode choice method for ride sharing services where a calibrated market equilibrium performs the mode choice of ride sharing services for given supply parameters. This calibrated ME model can save extensive simulations required to perform a basic equilibrium-based mode choice of ride sharing services for any change in supply or demand parameters. The calibration of model parameters is achieved by comparing the simulation results with the analytical results and reducing the goodness of fit error to achieve a set of model parameters that can provide the same output through analytical equilibrium rather running simulation equilibrium.

3.3.2 Simulation Database

The first step is to develop a database of simulations which contains the service attributes detour time and waiting time for varying range of ride-sharing demand Q and fleet size N . The simulation platform is used to perform the various simulation runs and the output results of every simulation is stored for large database. Fig 3.5 illustrates the data collection process of the simulated network attributes using a nested for-loop. For every fleet size, a series of demand is simulated to generate a fairly large data-set having high significance of the values by performing multiple simulations for any single scenario.

Where q is the ride sharing demand and n is the fleet size. They both have increment parameters a and b respectively. However, one of the main step in setting up the simulation database is to define the range of n and q for which the simulation runs will be performed. First step is to select an approximated range of fleet that can be operated on the network of study. Note that this range of fleet size will only be used to calibrate the ME model. The calibrated ME model will however be valid for the fleet size outside this range as well.

Since market equilibrium model is based on analytical equations, for a fleet size n the analytical equations 3.1 3.2 have bounds of model parameters within which the equations can be solved. Parameter's bounds are specific to network which can easily be determined by standard hit and trial method. Once an approximated fleet range is selected, a maximum possible ride sharing demand q that can be output from ME model within those bounds can be determined. Hence, a range of ride sharing demand q can be defined, starting from as low as 1 ride sharing demand to the maximum demand output q from ME model. Once these ranges for fleet n and demand q are defined, a suitable increment parameters a and b can be selected depending on the level of detail required from the data collected.

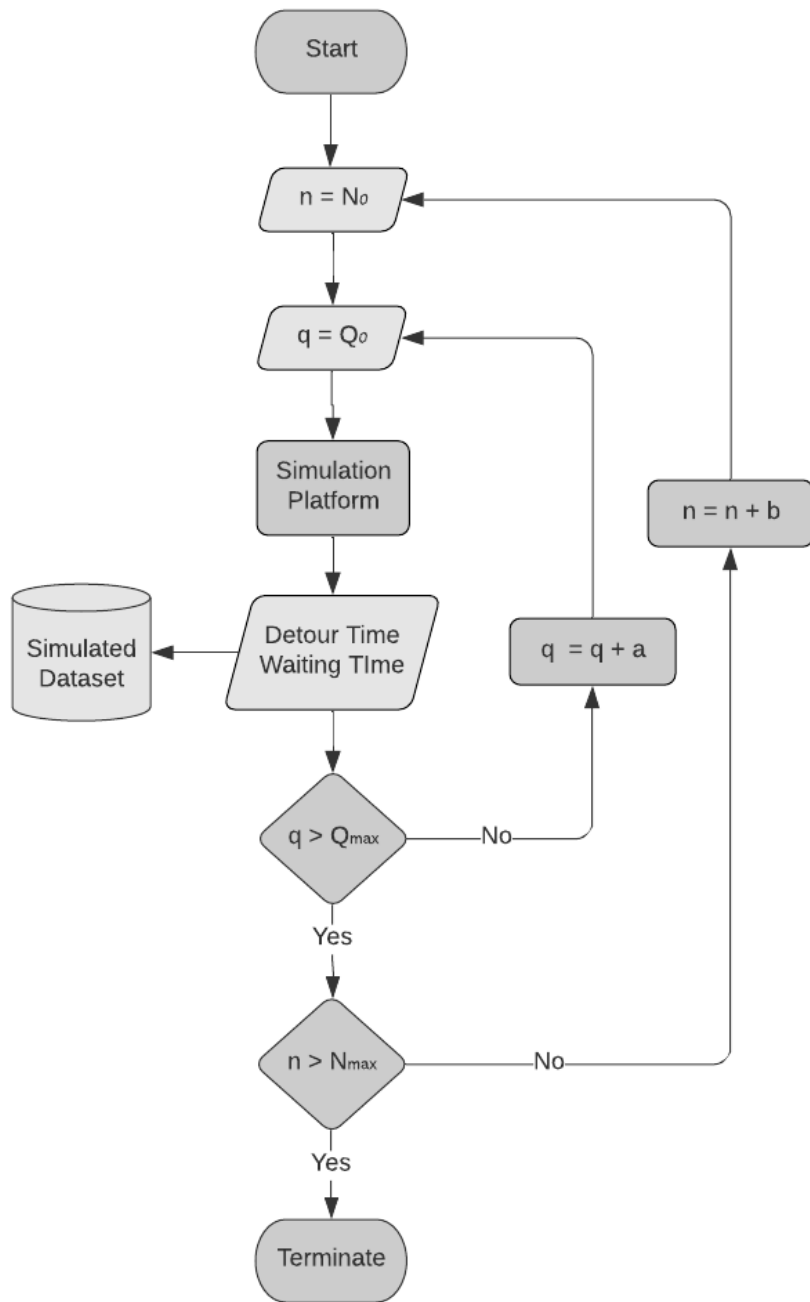


Figure 3.5: Data Collection from Simulation Platform using Nested For Loop

3.3.3 Calibration of ME Model

Upon creation of the required simulation database, the next step is to calibrate the ME model by comparing the outputs with the corresponding values from the database. The simulated database is considered as the observed data and the output from ME model are the predicted values. For a given inputs, the ME model returns a ride-sharing demand and the analytical networks attributes detour and waiting time for that particular demand. To calibrate the ME model parameters we develop an optimization objective function that uses the simulation database (observed values) to compare the ME model results(predicted values) for A and B model parameter's values and returns an error term as a measure of difference between the two data-sets.

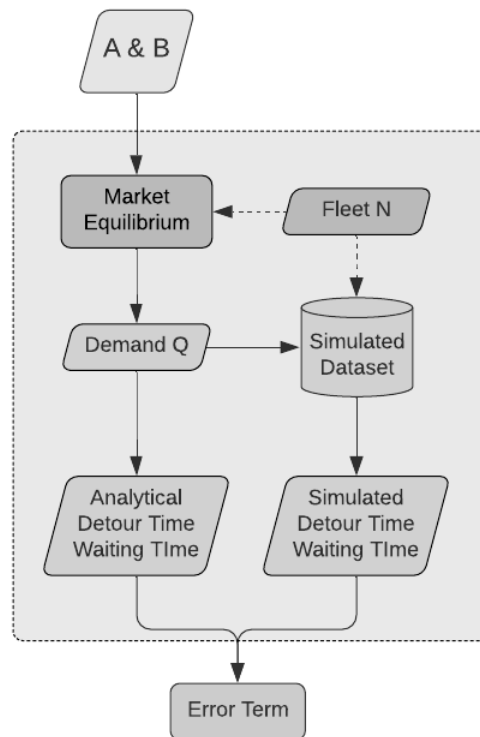


Figure 3.6: Objective Function Workflow

This becomes a simple optimization problem where the resulting error from the objective function can be minimized using one of the optimization algorithm. The optimization algorithm takes in some initial values of model parameters A and B and tries to minimize the error term returned by the objective function. The optimization algorithm performs different function evaluations to determine that in which direction does the model parameters reduces the error term and tries to converge to the slope minima. The optimization algorithm returns the set of A and B values after the convergence. The general framework of the function is

illustrated in figure 3.7 which reduces the error value until convergence. The converged values of parameters A and B calibrates the market equilibrium model.

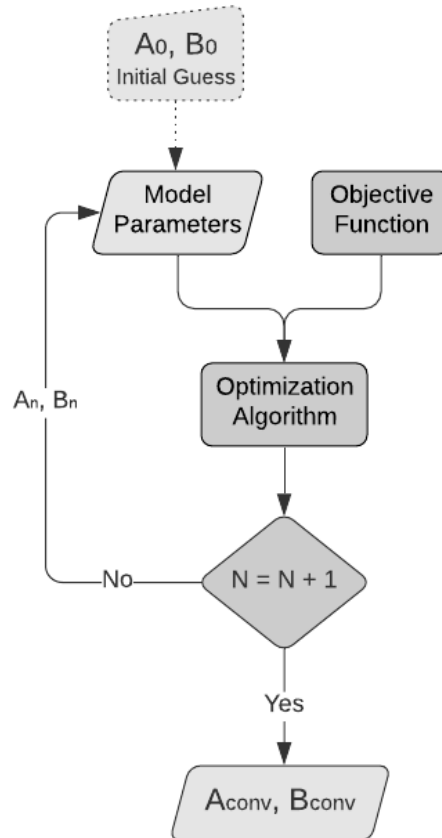


Figure 3.7: Calibration of ME Model Parameters

3.3.4 Discussion

As explained earlier, the conventional process of performing mode choice for ride sharing services includes a feedback loop based system where based on the network state, multiple simulation runs are required to achieve an equilibrium state. Note that the equilibrium state achieved is through a regress process and only valid for that particular supply parameters. For any change in supply, the process has to be performed again until the equilibrium is achieved. For larger networks, where a single simulation run takes several minutes to complete, and given the fact that to achieve equilibrium, multiple runs are needed, it can become computationally expensive to perform mode choice every time there is a change in supply.

The calibrated ME model however is viable against the varying service attributes and reduces the extensive computational time required in conventional simulation based mode choice method. For any change in supply, the calibrated ME model approximates the ride sharing demand through analytical equations and takes far less time to solve the equations compared to running multiple simulations. Another reason that gives ME model advantage over simulation runs is that the discrete level of a simulation run is the network links and nodes, where for ME model the discrete level are the OD pairs. In case of larger study areas, the OD pairs can be aggregated to reduce the number of equations that the ME model solve analytically however for simulations, the network links, individual vehicles will take up more time due to larger size of the network and the aggregation of network links is only possible to some extent.

4 Experimental Setup

This chapter describes the experimental setup used within this research. First section discusses the creation of simulation database, simulation platform and the standalone scheduler used. Section two discuss the steps involved in model calibration using the observed simulation data and predicted ME model data. In the last section, integration of the calibrated module with the simulation platform is explained.

4.1 Simulation Database

Simulated database acts as the observed values and are compared with predicted analytical model outputs of the ride-sharing market equilibrium (Lu, 2020). Simulation database is generated by running multiple simulations for different supply parameters as explained in section 3.3.2. For this study, ride sharing simulation platform designed by (Qurashi et al., 2020) was setup to perform simulations required, however during the execution process there were many errors due to some inconsistencies between the platform and the scheduler and despite spending quite a lot of time on platform setup it was not used to run simulations. To complete the study in time, a standalone version of the same scheduler was used to run the simulations without the interaction with the platform. This section describes both the original setup of the simulation platform and the use of standalone scheduler.

4.1.1 Ride Sharing Simulation Platform Setup

Simulation platform (Qurashi et al., 2020) designed for ride-sharing services is used to setup the database. It uses SUMO as its default simulation environment. The platform is itself written in python programming language and consists of different modules responsible for their specific tasks in modelling dynamic vans behaviour. Table 4.1 summarizes the general functionalities of the different modules of the simulation platform. In this study, these modules have been improved and where necessary additional modules were also written in python to achieve the desired functionality.

Table 4.1: Simulation platform modules summary

Module	Functionality
Network Integration	Methods definition to handle network file and create internal nodes
Operation	Definition of functions required by the platform during execution
Simulation	Interaction with Scheduler using DLL and SUMO using TraCi

Note: DLL:Dynamic Link Library, TraCi:Traffic Control Interface

Using the nested for loop 3.5 the platform is setup to perform various simulations on the case study networks and collect as much result possible for the AB estimation and thus model calibration. The AB estimation requires a large amount of data from the simulations runs that can be compared with the ME module outputs to calibrate the model. However the ME module as explained earlier outputs certain demand Q and its respective attributes for given number of vans. In order to completely calibrate we should have enough simulation data available for the various values of Q and number of vans N . For each case study network, the range for fleet and ride sharing demand is specified for which the data is collected.

Scheduler Interaction

As described earlier in section 3.1.1, the scheduler is responsible for the routing of the vans. It performs the routing of the vans based on the new requests generated after performing the optimization of the objective functions. However, scheduler is written in C++ and to perform the routing dynamically, platform needs to interact with the scheduler and call its methods in real-time. For this purpose, a dynamic link library is generated (DLL) which is used to interact with scheduler using python and make use of its functions. The simulator module of the python code is written for this real-time interaction with scheduler and also with the SUMO simulation. At an event of new requests generation, the simulator module first calls scheduler's functions to update the routes of the vans, and then uses TraCi module to update the new routes of the van on the network and continue with the simulation steps unless new request is generated.

Traffic Control Interface - TraCi

Traffic Control Interface or TraCi is one of the modules of SUMO which is used to interact in real-time with simulation environment and retrieve the values of the simulated objects and is also able to manipulate them. Traci can obtain various attributes of the simulated objects such as passenger info, passenger driving state, vehicles ids, their locations on the network etc. All these object attributes are needed by the simulator platform for requests handling. The simulation platform uses functionality of the TraCi module and models the behaviour of dynamic vans on the network. Vans can only be dynamic when they are able to route dynamically on the network. TraCi can only retrieve simulated object's values but cannot perform the dynamic routing on its own. For this purpose the simulation platform runs in

conjunction with the scheduler. On one hand the scheduler performs the dynamic routing of the vans and on the other, TraCi helps to model their behaviour on the network. Simulation platform manages the execution of these processes in a sequential and organized manner.

4.1.2 Issues/Errors

In order to create the simulation database, various simulation runs are required to collect sufficient amount of data to compare with market equilibrium outputs. However, the simulation platform had several errors during the execution process, mainly while interacting with the scheduler. Some errors created warnings during the execution where as some were very critical resulting in the abortion of simulation process.

U-Turn connections

Usually when the network is downloaded from OSM and then converted to XML format, all the road network is kept in its original state. Some junctions on the network might not allow a u-turn because of the physical limitations or simply because of the fact that a u-turn is not allowed. However, the scheduler treats a junction by default as if u-turn is possible which creates bugs and errors while simulating. The scheduler gives a certain route to van during simulation assuming that a u-turn is possible however in SUMO network that is not possible for van resulting in its crash. This was one of the problems with a larger network since there was more chance of occurrence of such scenarios. So at the time of crashing, the faulty edge had to be manually checked on the network using NETEDIT and the issue had to be dealt with.

Scheduler Incomplete Routing

During a normal simulation run, a request is read by the scheduler, then it is given the current positions of the vans and based on its internal optimization the scheduler assigns a route to one of the vans. The assigned route is communicated to the selected van on network using TraCi. However, sometimes during simulation runs, when scheduler selects a route to be assigned, few of the edges are absent in the route. Consequently when TraCi communicates the route to the vans on network, they are unable to identify the route because of the missing edges causing it to crash. Such errors were really hard to resolve and had very frequent occurrences.

Unserved Passengers

During an event of a new pickup request, a person is generated on the corresponding meeting point of that traffic assignment zone. The crashing of scheduler due to incomplete routes also resulted in vans not arriving at the pickup point and thus the passenger remains non-served during the whole simulation. Most of these errors were caused during the translation of data from scheduler to SUMO. Another reason is the dynamic link library, which was initially

generated, may also causes some methods or functions to crash in C++ which are responsible of routing the vans.

4.1.3 Standalone Scheduler

Some of the issues such as u-turn connections, were resolved in time of this study, however the other more time-consuming errors could not be resolved for larger networks because of the time-constraint of this research. Due to these reasons, a successful simulation run using simulation platform could not be made possible. Quite a lot of time of this study has been spent on debugging these errors for successful simulation runs but due to the time constraint of this study, an alternative approach was needed to continue the work and achieve the desired results. Nevertheless, to conclude the study in time the requests are run analytically on scheduler-only using a standalone scheduler executable file. It takes the same inputs as needed by the simulation platform to run multiple simulations. The only downside of this method was that it did not include the interaction of other traffic on road which could have impacted a little on detour or waiting time due to the effect of congestion but all in all the standalone scheduler still does provide the simulated network attributes as it does map the network waiting time and detour which are compared to the analytical results of ME module. Therefore the standalone scheduler is used to perform the simulation runs and the desired data is collected.

4.1.4 Data Collection

To setup the simulation runs on standalone scheduler took very less time as the inputs were already generated in process of setting up the simulation platform. Therefore only an executable file with the paths to input files was required to run the simulations. The nested for-loop as illustrated in figure 3.5 is used to collect the data. To increase the significance of the data collected, the for-loop itself is repeated multiple times. The outputs of the scheduler contains the text files containing network attributes of all the simulations ran. The nested for-loop is setup in such a way that after the fleet size is incremented, the simulation results are categorized based on the number of vans. After the simulation run is complete for all vans and demand, all the resulting text files are read and the simulated ride-sharing demand Q is extracted along with simulated detour and waiting time. Since the for-loop itself is repeated several times, so for a given N and Q , there are multiple entries of data. The multiple entries are then aggregated and a mean value is taken for that particular N and Q entry. The extracted data is stored in a suitable format which can be easily processed and compared with the ME model outputs. Table 4.2 illustrates the format of simulation database.

Table 4.2: Simulation Database Format

Fleet Size (N)	Demand (Q)	Detour Time	Waiting Time
n_0	q_0	$t_{n_0q_0}$	$w_{n_0q_0}$
n_0	.	.	.
n_0	.	.	.
n_0	.	.	.
n_0	q_n	$t_{n_0q_n}$	$w_{n_0q_n}$
.	.	.	.
.	.	.	.
.	.	.	.
n_n	q_0	$t_{n_nq_0}$	$w_{n_nq_0}$
n_n	.	.	.
n_n	.	.	.
n_n	.	.	.
n_n	q_n	$t_{n_nq_n}$	$w_{n_nq_n}$

4.2 ME Model Calibration

4.2.1 ME Setup

ME model calibration requires observed and predicted values. ME outputs are the predicted values that are compared with the simulated database. ME module uses the available passenger preference data (Tsiamasiotis, 2019), the utility functions of other transport modes and the OD demand in form of a vector, it calculates ride sharing demand Q and the corresponding network attributes calculated analytically. In order to use market equilibrium module to get desired outputs for the case study network it requires certain network and supply specific inputs. Figure 4.1 illustrate the required inputs needed by the market equilibrium module.

ME module is also written in python and also requires the global inputs that are used during the simulations as of fleet size, van capacity etc. However, it additionally requires direct times and distances between all origin destination pairs. To obtain these values of direct times and distances for all OD pairs, simulations are required of the study network with complete OD network demand but without the ride sharing service. After running simulation runs vehicle routes XML file is generated as output. This vehicle route XML is also one of the outputs of SUMO which can be generated after a simulation. It contains all the vehicle routes information that were simulated. The route information are OD based and they are processed to extract the individual OD pair direct distance and direct time. These direct times and distances along with the supply specific inputs are then given to ME module to obtain predicted values.

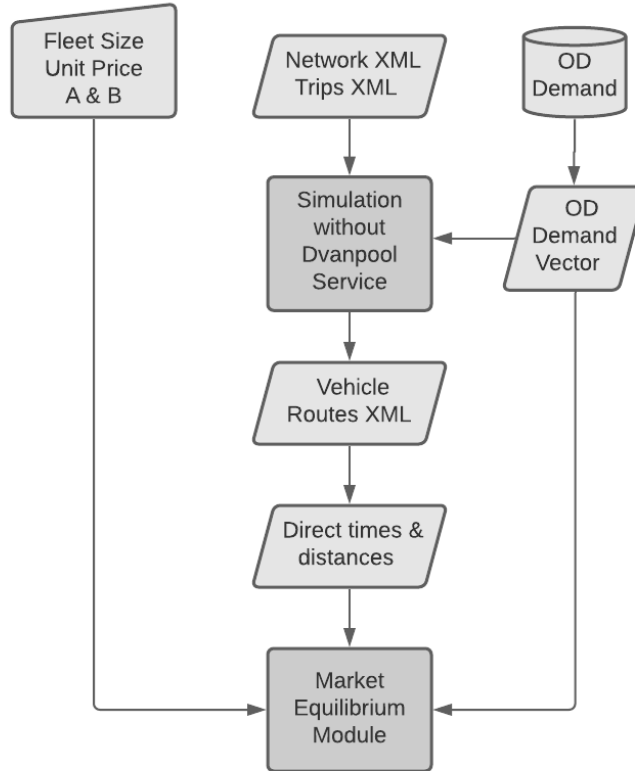


Figure 4.1: Requisite inputs for Market Equilibrium Module

4.2.2 ME Dataset

Once the ME module setup is completed, the next step is to run the market equilibrium model with initial guess values of parameters A and B and collect the data for the same range of fleet size as used in nested for-loop to collect the simulation data. However for getting ME, only single for-loop is required since the demand is one of the outputs of the ME module itself along with network attributes. This idea is better illustrated in the following figure 4.2

The resulting data from the ME module is generated in the format shown in table 4.3

Table 4.3: ME Dataset Format

Fleet Size (N)	Demand (Q)	Detour Time	Waiting Time
n_0	q_0	t_0	w_0
.	.	.	.
.	.	.	.
.	.	.	.
n_n	q_n	t_n	w_n

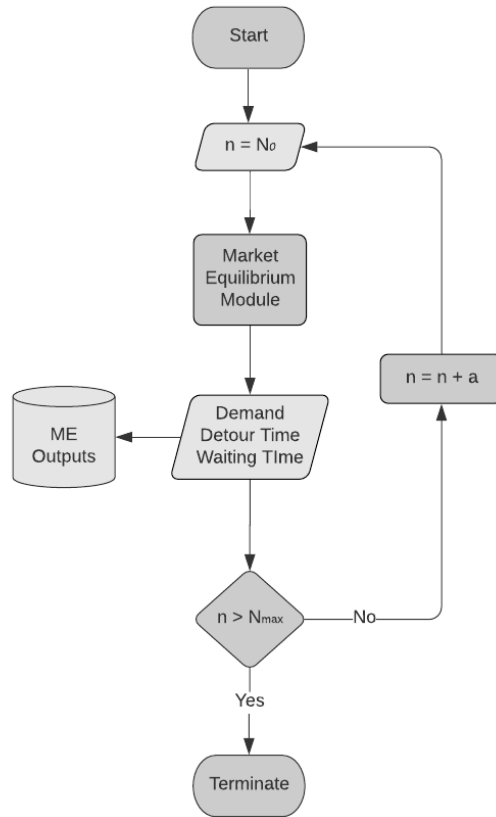


Figure 4.2: Data Collection from ME Module using for-loop

4.2.3 Data Comparison

For a given value of N , the ME outputs the ride sharing demand Q of the market, and the resulting network attributes. The next step is to join the corresponding simulated network attributes for the same fleet N and demand Q and compare the outputs. A code module is written in python again that performs this comparison process in a systematic way.

- Reading simulation data from output files and arranging it according to the fleet size in a dictionary of data frames in python. Where a single data frame consists of network attributes for varying range of Q for that particular N .
- Getting ME data outputs in a single data frame.
- The code reads the ME output line by line. First it determines the fleet size N , and then look for the dataframe of that particular N in the dictionary of simulation dataframes.
- Once the target dataframe is identified, it reads the Q from the ME output for the same line, and then tries to find the Q in target dataframe. However the ME outputs float

values for Q and the simulation dataset have integer values for Q .

- In the next step the function identifies the neighbouring values of the Q and then interpolates and gets a detour and waiting time value from simulation dataframe and attaches it with the ME dataset.

Once this code module is finished reading all the ME output lines and attaches all the corresponding results, the final comparison dataset is represented in table 4.4

Table 4.4: Comparison Dataset Format

N	Q	Detour _{me}	Waiting _{me}	Detour _s	Waiting _s
n_0	q_0	t_{me_0}	w_{me_0}	t_{s_0}	w_{s_0}
.
.
.
n_0	q_0	t_{me_n}	w_{me_n}	t_{s_n}	w_{s_n}

4.2.4 Goodness of Fit

The next step is to evaluate the difference between the values of both datasets. The goal is to measure how good the ME outputs fits with the simulation data. The goodness of fit can be measured through many indicators, in this study following errors have been used to measure the difference simultaneously on an initial guess values of model parameters and then the error with better convergence results is used for further evaluations in this study.

For error calculations, the simulation detour and waiting time will be treated as the actual data and the ME dataset will be the predicted values that has to be fit eventually.

Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - f_i)^2} \quad (4.1)$$

where:

- d_i = predicted values (ME)
- f_i = actual values (Simulation)
- n = range of fleet sizes simulated

Root Mean Square Normalized

$$RMSN = \frac{\sqrt{n \sum_{i=1}^n (d_i - f_i)^2}}{\sum_{i=1}^n f_i} \quad (4.2)$$

where:

- d_i = predicted values (ME)
- f_i = actual values (Simulation)
- n = range of fleet sizes simulated

Mean Absolute Percentage Error

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (4.3)$$

where:

- A_t = actual value (Simulated)
- F_t = forecast value (ME)
- n = number of observation values

4.2.5 Optimization

The initial value of any of the above mentioned error would be higher since the input model parameters are just the initial guess. They need to be optimized so that the model can be calibrated. The optimization workflow has been explained in detail in the figure 3.7. Python library *scipy* is used to for optimization of model parameters

The *Minimize()* function of the *scipy* library inputs a callable function and initial guess values of the *A* and *B* parameters. The callable function should take in the model parameters and return a float value of the error. The optimization algorithm will re-iterate by using different *A* and *B* inputs to the callable function and continue the process unless the error value is converged or maximum number of iterations have reached.

The convergence of the error value also depends on the initial guess values of the model parameters as well because the function converges to a local minima which is sometimes not the lowest value of the error. However, by running the minimize function few times with different initial guess, results in the convergence to the global minima. The minimize function after converging outputs results that contains the number of iterations needed to converge and the final values of model parameters that resulted in the lowest error value. The parameter values essentially calibrates the market equilibrium model to the extent where it can be used as an alternate to running simulations.

4.3 ME-based Simulation Mode Choice

4.3.1 SUMO Inputs

The simulation or modelling of vans happens in presence of the general origin destination demand of the network. So in order to incorporate the behavior of vans on an existing simulation network, some prerequisite input files are needed by SUMO to initialize a simulation and then use TraCi to interact with it. A standard SUMO simulation requires a configuration file `.sumocfg` which contains the necessary information regarding the simulation. For a simulation to run, the SUMO configuration file at least requires a path to network file, trips file containing the information of origin destination trips and an additional file, all in XML format.

Network File

The network files required by SUMO are in `.xml` format and they can be created by using the NETCONVERT feature of SUMO. Networks can be exported from Open Street Maps (OSM) in `.osm` format. Since the exported network from OSM also contains other information which is not necessary for the simulation such as buildings or footpaths, this extra information can be dropped using OSMFILTER. OSMFILTER is a commandline based program that keeps or drops the OSM tags from `.osm` files. After filtering the unnecessary data from `.osm` file, using NETCONVERT `.osm` network can be converted to SUMO readable `.xml` format.

Trip File

Another required input for the configuration file is the `trips.xml` file. This file usually contains the information of all the trips that will be simulated. Note that the trips here refer to the trips from the origin destination demand of the network zone. Usually the origin destination data is in matrix format commonly known as OD matrix. SUMO function OD2TRIPS is used to convert this origin destination data into `trips.xml` which can be given to configuration as input for trips. To run OD2TRIPS function, OD matrix is first converted in a vector format and then given as `.txt` input. In addition to that, a TAZ file is also needed by the function. TAZ file is also an XML file, which contains all the traffic assignment zones of the network.

4.3.2 Network Preparation

SUMO takes the network demand in traditional OD matrix format. The demand is represented as traffic assignment zones which have connectors and we need to represent the same for our network hence we create meeting points. The meeting points are various locations on the network that will act as the pickup or drop off locations for the dynamic van-pooling service. As explained earlier, these vans are modelled with the actual origin destination demand of the traffic assignment zones of the network. Due to the dynamic nature of their behavior, vans demand can not be written in form of a `trips.xml` and given to SUMO at the start of simulation. All the requests that are generated for vans is not always served if they do not

fulfill the optimization criteria of the scheduler, therefore meeting points are required on the network to create the demand during simulation and then the scheduler checks whether it can serve it or not.

The current working of the simulation platform includes creation of connectors or as we call them meeting points at random locations on the network. The drawback of random generation of these meeting points was that sometimes most of the meeting points were cluttered in a single vicinity resulting in all the pickup requests from a same area. The systematic way was to create these meeting points distributed over all traffic assignment zones. The additional set of code written solves this issue and creates distributed meeting points. The basic methodology adopted by the code is as follow:

- Using the Geopandas library of Python, the module reads the shapefile of the study area which consists of all the traffic assignment zones (TAZ).
- Module then reads all the edges and nodes of the study area from network XML file and identify the corresponding TAZ of each one of them from shape file.
- It then filters the links which have starting and ending node in a same TAZ.
- Among the filtered links, it finally selects few links of reasonable length from every TAZ which are then used as meeting points for the vans.

During initialization process, if the network XML is changed or modified externally, new meeting points are created every time and this is the default method to create them. During the process of matching TAZ with links, it also creates a TAZ file. TAZ file is one of the inputs for creating trips XML file. Using the XSD schema file available in SUMO installation folder, TAZ XML file is created in a SUMO readable format. This taz.xml is later used in generation of trips.xml file in the platform.

4.3.3 Demand Processing

Calibrated ME model replaces the conventional mode choice method. For the simulation platform to integrate the calibrated ME model as mode choice method, the outputs of ME model needs to be processed into request files required by the simulation platform. Request files are text files which basically contain the information regarding ride sharing demand trips, their estimated departure and arrival times. Since the market equilibrium module is a standalone package which performs analytical operations, the nomenclature of ME module and simulation platform is very different. Data processing and some additional definition of the variables is necessary to move forward.

The ride sharing requests Q generated for all OD pairs by the market equilibrium need to be processed in a way that eventually we have request files in .txt format which are given to scheduler later on. The request file however not only take Q as an input but it also requires few additional variables. Figure ?? briefly illustrates the process of creating request files once the results are obtained from market equilibrium.

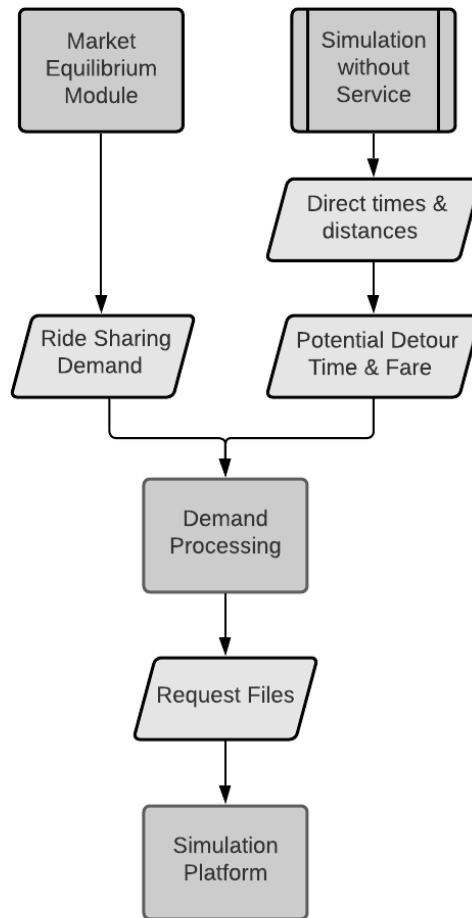


Figure 4.3: Ride Sharing Demand Processing

The ride sharing demand is generated in form of a vector for all OD pairs. Ride sharing demand output from market equilibrium model is processed into request files for simulation platform as shown in fig 4.3. Using OD2TRIPS function of SUMO, a trip file is created. Resulting trip XML file only contains the van trips that are to be served during the simulation. However this trip file is not used by SUMO as one of the input just like the OD demand trip file, rather this trip file is used to create passenger request file that the scheduler requires. The useful data such as trip departs, arrival, ID is extracted and is processed into the request files.

Since this request file is basically an input for the scheduler, other than standard trips info, it additionally requires the latest arrival, departure time and trip revenue which will be generated from individual trip. This information allows the scheduler to optimize during simulation that which of these requests would eventually be profitable and whether a certain user is willing to wait for their pickup or is acceptable with the fact that they might experience

extra detour during the trip. Equations 3.3 and 3.4 define these additional parameters using the available data of direct time and distance from earlier simulation runs without van service. These additional parameters of potential detour time and expected revenue along with ride sharing demand from market equilibrium are processed into request files.

Generation of these request files is the final step before the actual simulation starts. This new mode choice method incorporated a standalone market equilibrium module with the simulation platform. This integration allows us to simulate the ride sharing demand through a market equilibrium and then simulate those demand requests with actual OD demand. This enables us to include the stochastic nature of the actual traffic in the network and map the actual service attributes.

5 Case Studies

This chapter explains the setup of two different scenarios. In first section, a synthetic case study is developed for a small test network with synthetic demand and utility functions. Later on in section two the method is then implemented on a slightly bigger network with more traffic assignment zones and different applications of a calibrated model is explored.

5.1 Unterschleißheim - Synthetic

Unterschleißheim is a suburb of Munich City. It is a small neighbourhood with roughly over 30,000 inhabitants (Rathaus Unterschleissheim, 2020). The area has mostly residential roads and fairly well interconnected network suitable for our synthetic case study.

5.1.1 Network



Figure 5.1: Unterschleißheim Network

The network for this area is downloaded from (OpenStreetMap contributors, 2017). The downloaded .osm network file is converted to network XML file using NETCONVERT. Also additional features which are not needed are dropped using OSMFILTER. The resulting

network XML file contains only the useful road network which is needed essentially for the simulation. It also makes the XML file smaller in size which helps in loading the network faster in simulation platform later. The area is divided into 3 traffic assignment zones, making sure that all TAZ have equal distribution of the road network. A basic shape file is created in QGIS. Using an OSM standard layer at bottom, polygons are created to equally divide the area.

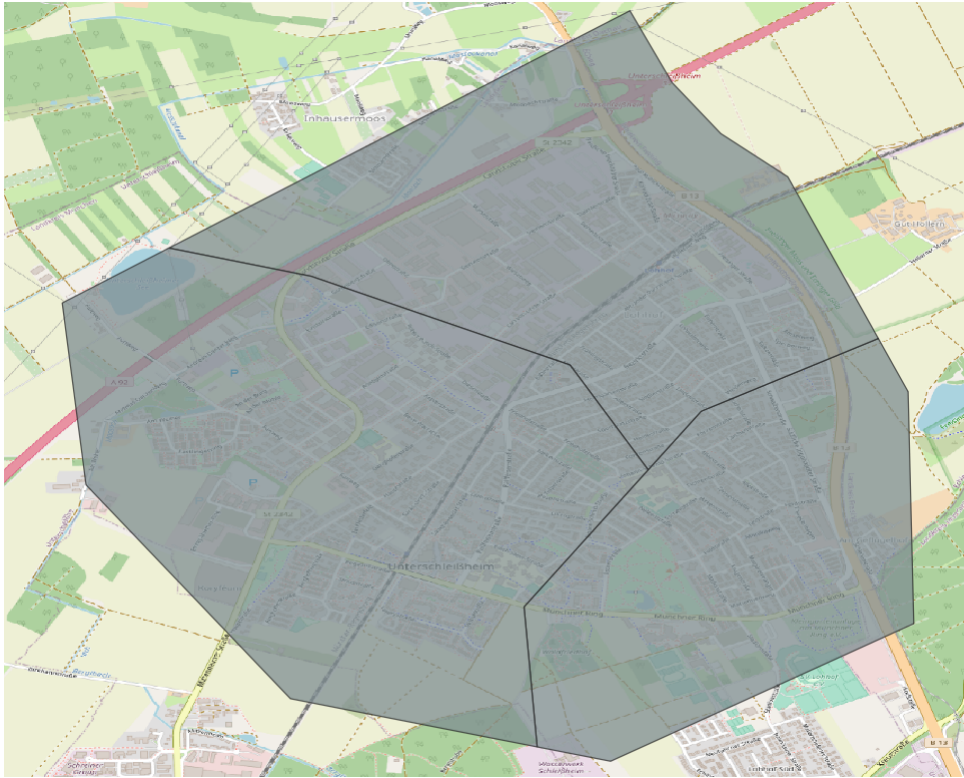


Figure 5.2: Unterschleißheim TAZs

A synthetic OD demand is created for the area of Unterschleißheim. Based on this synthetic OD demand, trips file, ride sharing demand and all other requisites required for the simulation platform are prepared.

5.1.2 Simulation Database

First step is to collect the base simulation dataset needed for model calibration. Simulation dataset for synthetic network is created using the nested for-loop. The dataset is aggregated based on the unique demand Q and fleet N values as described earlier in table 4.2. During the setup of nested for-loop, range for fleet size N and ride sharing demand Q is also initiated

where:

$$N \in \mathbb{Z} : 2 \leq N \leq 6$$

$$Q \in \mathbb{Z} : 1 \leq Q \leq 50$$

The simulation runs are initiated using the fixed parameters shown in table 5.1. Scenario runs here refer to the times the nested for-loop is repeated itself to get multiple data entries and then aggregating them at the end. This helps in removing the outliers during the multiple simulation runs.

Table 5.1: Simulation Setup Fixed Parameters

Parameters	Values
Seating Capacity	8
Unit Price	2.5
Scenario Runs	21
Profit Ratio	0.4
Detour Coefficient	1.5

5.1.3 Optimization

Once the simulation dataset is generated, next step is to start the process of the error minimization. For finding error term, initial parameters guess values are input in market equilibrium module. The ME module returns the initial output 5.3 for all the complete range of fleet size. The detour and waiting times are aggregated at the network level.

Table 5.2: ME Module Parameters - Initial Guess

Parameters	Values
A	6
B	0.06

Table 5.3: ME Output (Initial Guess)

N	Q	Detour Time	Waiting Time
2	5.526698	860.551500	108.448201
3	9.303947	567.199758	145.169679
4	11.662568	432.792975	155.017755
5	13.277706	346.311693	157.074269
6	14.885227	287.626189	160.027645

Once the ME output is available, the next step is to retrieve the corresponding detour and waiting time values for the given values of Q in table 5.3. The ME output dataset usually has demand Q in float format and the simulation dataset has integer values for the column Q , the codes are written in such a way that during comparison process it automatically returns the interpolated values between the two neighbouring integers for the given float value. The resulting output of the comparison dataset is shown in table 5.4

Table 5.4: Comparison Dataset (Initial Guess)

N	Q	Detour (ME)	Waiting (ME)	Detour (Sim)	Waiting (Sim)
2	5.526698	860.551500	108.448201	168.576500	186.046727
3	9.303947	567.199758	145.169679	236.467202	156.899426
4	11.662568	432.792975	155.017755	275.138632	147.523902
5	13.277706	346.311693	157.074269	279.561362	142.720219
6	14.885227	287.626189	160.027645	244.268669	151.724580

Once the comparison dataset is available, the next step is to evaluate the goodness of fit by finding error terms to measure the difference of both datasets. Error term is calculated individually for both detour and waiting times. Once the error is calculated for both attributes, a mean value is taken for error term. This is because *Minimize()* function of python library *Scipy* requires a scalar callable function that only returns a single float value.

Individual error minimization is performed for three selected evaluation criteria. While minimization is performed for one of the error, the change in error term for other two errors is also calculated based on the initial and converged parameter values.

The process is repeated for each error separately. Table 5.5 represent three sections. In each section, minimization algorithm is performed with respect to one error, and change in the values of other two errors is also calculated based on the initial and final parameters values. This allows us to determine that which minimization will result an overall better reduction of all three errors.

Table 5.5: GOF Evaluation - Initial Guess

Error	Initial	Final	Δ	% Change
Minimization w.r.t RMSE				
RMSE	193.622	53.189	140.433	72.53
RMSN	1.197	0.236	0.961	80.28
MAPE	83.199	19.476	63.723	76.59
Minimization w.r.t RMSN				
RMSE	193.622	53.709	139.913	72.26
RMSN	1.197	0.2042	0.993	82.94
MAPE	83.199	20.158	63.041	75.77
Minimization w.r.t MAPE				
RMSE	193.622	47.619	146.003	75.41
RMSN	1.197	0.2052	0.992	82.85
MAPE	83.199	16.959	66.240	79.62

The complete convergence results of the first optimization run with the initial guess values 5.2 are shown in table 5.6. As indicated in table 5.5 that minimization with respect to MAPE gives the best error reduction results therefore all convergence results hereafter are based on MAPE minimization.

Table 5.6: Convergence Results - Initial Guess

Attributes	Values
A_{conv}	2.8055
B_{conv}	0.0607
Mean Absolute Percentage Error	16.959
Number of Iterations	51
Objective Function Evaluations	126

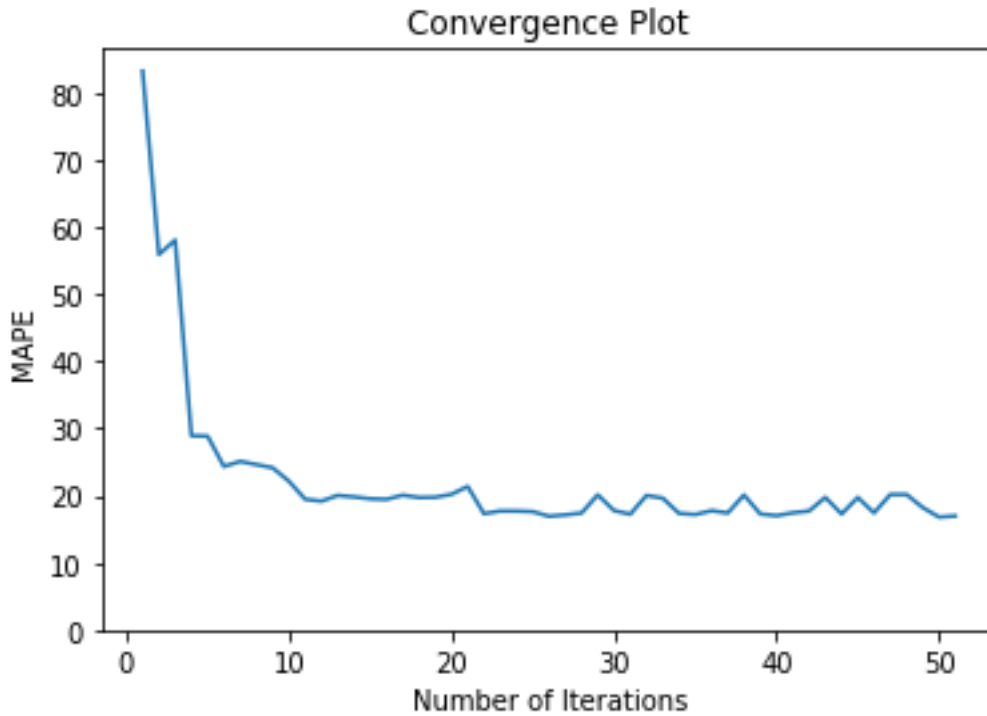


Figure 5.3: Convergence Plot - Initial Guess

To check whether the optimization function converged to a local minima or global minima, a series of optimizations runs are performed with varying starting A and B values.

Table 5.7: Summary of Optimization Runs

#	A_0	B_0	A_{conv}	B_{conv}	MAPE
1	6	0.06	2.8055	0.0607	16.959
2	6	0.05	2.8506	0.0609	16.796
3	4	0.05	3.557	0.0625	20.985
4	2	0.035	2.4533	0.056	17.183
5	5	0.04	2.357	0.057	17.749

Among the performed various optimization runs, 2nd optimization results are the ones with the lowest error term.

Table 5.8: Comparison Results - Best Optimization

N	Q	Detour _{me}	Waiting _{me}	Detour _s	Waiting _s
2	10.815798	410.178564	213.596544	264.665725	220.222070
3	13.898782	267.661554	218.247680	225.447341	210.168037
4	15.818252	208.723097	212.591986	216.724377	207.950440
5	17.169498	163.723947	204.704164	240.168799	183.208322
6	18.215285	136.536325	197.583651	242.944651	194.648312

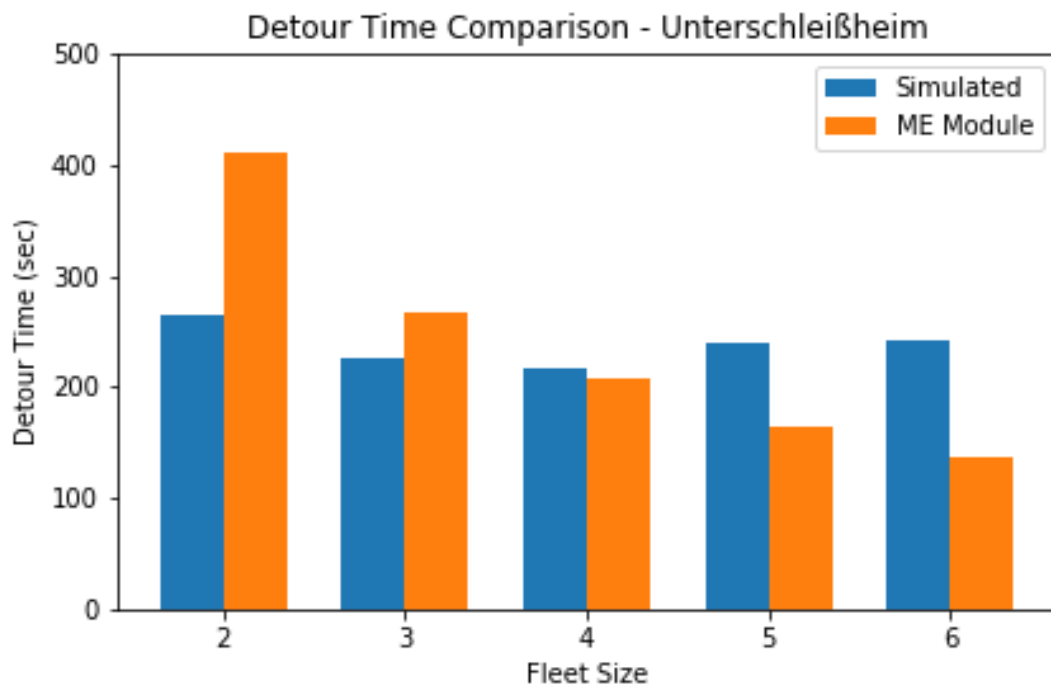


Figure 5.4: Detour Time Comparison

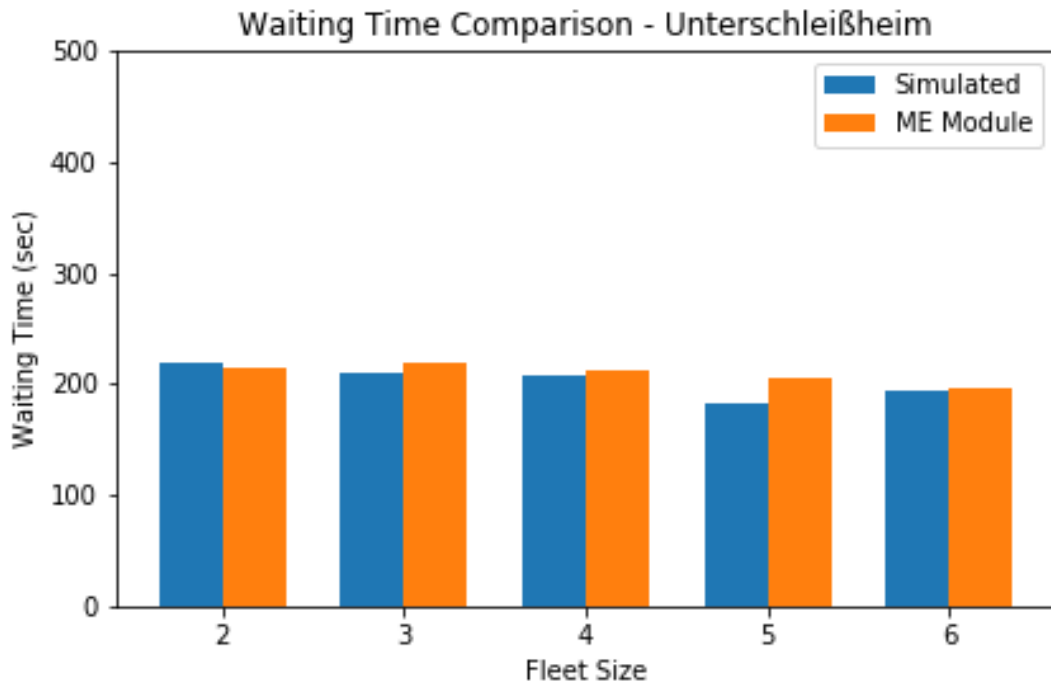


Figure 5.5: Waiting Time Comparison

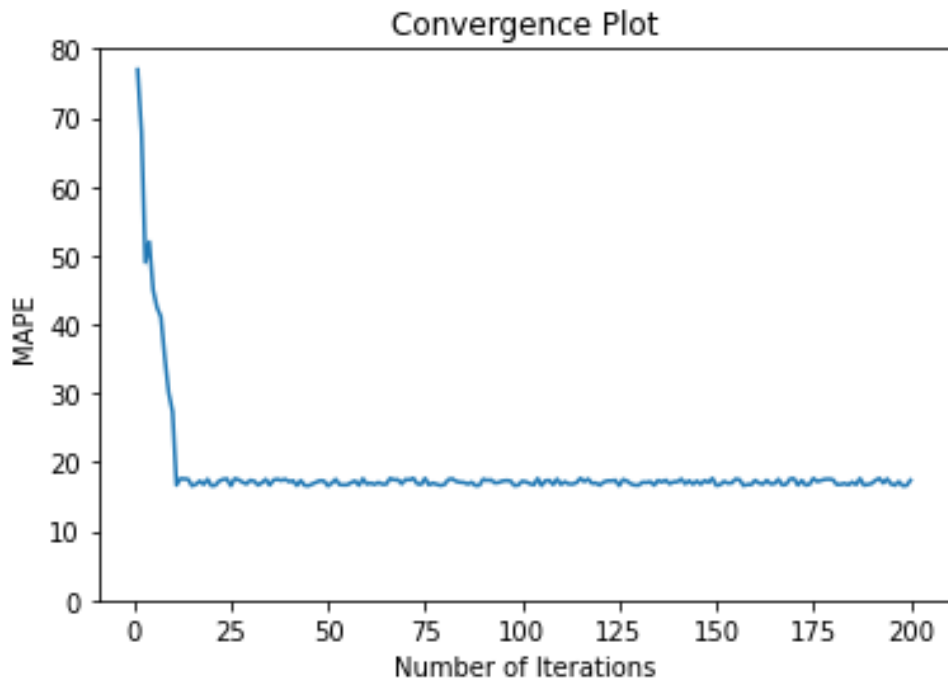


Figure 5.6: Convergence Plot - Best Optimization

Table 5.9: Convergence Results - Best Optimization

Attributes	Values
A_0	6
B_0	0.05
A_{conv}	2.8506
B_{conv}	0.0609
MAPE	16.796
RMSE	46.741
RMSN	0.1998
Number of Iterations	200
Objective Function Evaluations	526

5.2 Munich

5.2.1 Network

The main area selected for this study are neighbourhoods of Munich, namely Maxvorstadt and Schwabing. This network has roughly 1300 edges and 500 nodes. It is large enough to represent a real world scenario which can run a reasonable amount of vans. The study area is divided into 12 traffic assignment zones. Figure 5.8 shows the shapefile consisting of TAZ over the selected study area. Note that there might be some additional zones included in the original shapefile but the road module only selects the zones that are actually linked to the edges in network XML file.



Figure 5.7: Munich Network

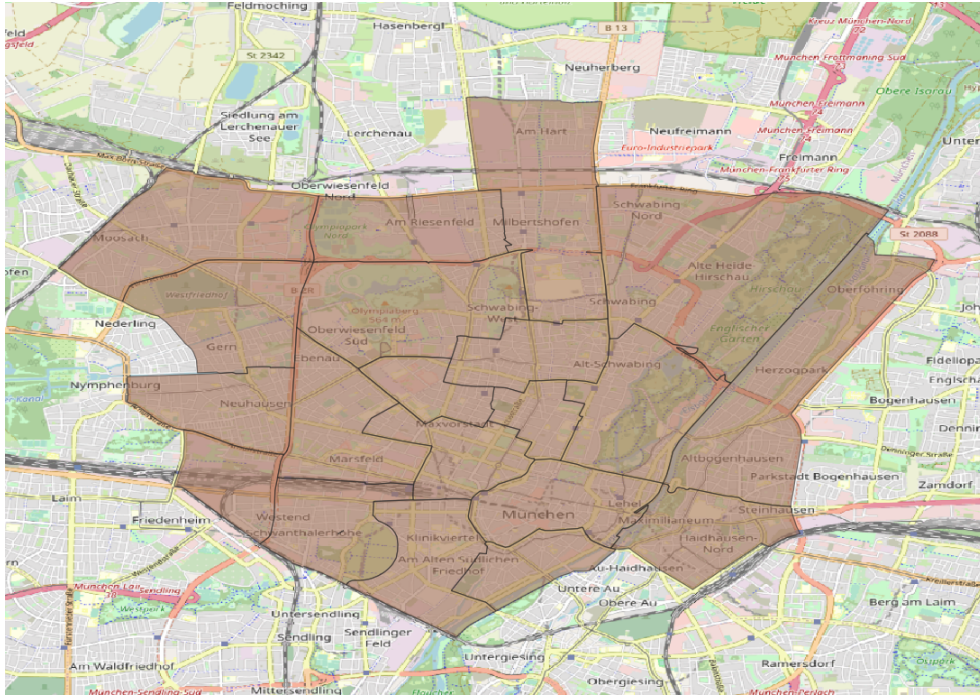


Figure 5.8: Munich TAZs

5.2.2 Simulation Database

Using the same nested for-loop as used for developing the simulation database for Unterschleißheim, data is collected for Munich network as well. However, for Munich, the fleet size ranges from 8-12 vans as the network size is suitable for this much number of vans. Using the standalone scheduler executable, the simulation runs are performed. The simulation setup is with the same fixed parameters as described in table 5.1. The ranges for fleet size N and Q are also defined where:

$$N \in \mathbb{Z} : 8 \leq N \leq 12$$

$$Q \in \mathbb{Z} : 10 \leq N \leq 180$$

5.2.3 Optimization

For the Munich area, the goodness of fit is evaluated by mean absolute percentage error since minimization w.r.t to MAPE has the better convergence results 5.5. The process adopted for synthetic study is repeated similarly for Munich area as well. Number of different optimization runs are performed with various initial guess of the ME model parameters. 5.10 summarizes the results of different optimization runs. 3rd Optimization run has the best convergence results and appeared to be the lowest MAPE value that was converged over multiple runs hence is considered as the global minima.

Table 5.10: Summary of Optimization Runs

#	A_0	B_0	A_{conv}	B_{conv}	MAPE
1	13	0.1	8.769	0.136	13.94
2	14	0.15	12.444	0.1583	39.253
3	15	0.1	8.295	0.126	13.767
4	15	0.2	12.963	0.228	52.596
5	12.5	0.15	10.5	0.18	28.914

Following are the results for the best optimization runs i.e. No.3 from the summary table 5.10.

Table 5.11: Comparison Table

N	Q	Detour _{me}	Waiting _{me}	Detour _s	Waiting _s
8	124.992465	208.325390	338.658645	153.433134	299.438695
9	131.785147	184.866072	330.890171	136.933852	296.804585
10	137.573444	166.421769	323.781784	148.943827	306.092624
11	142.741945	151.682394	316.626252	199.920478	305.013470
12	147.428403	138.902176	310.256527	156.678991	304.610151

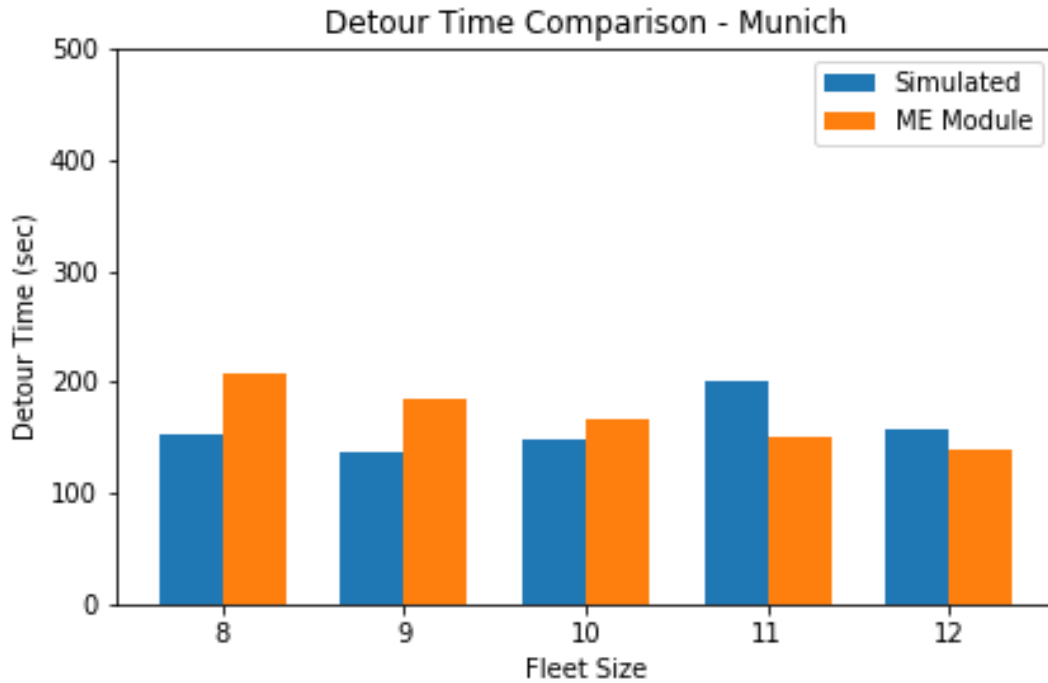


Figure 5.9: Detour Time Comparison

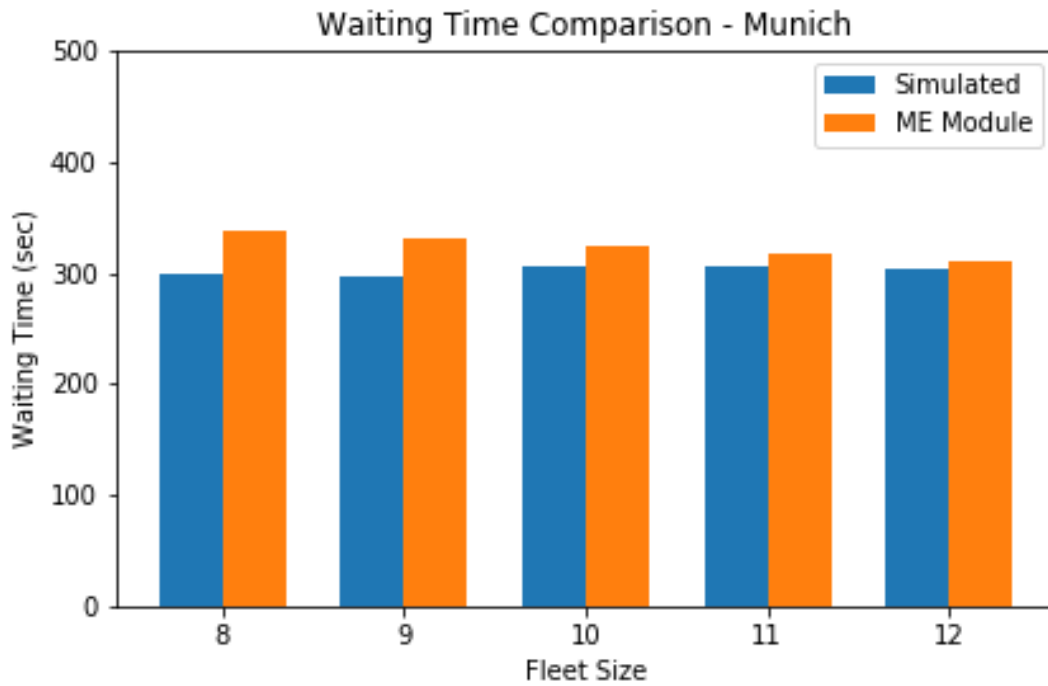


Figure 5.10: Waiting Time Comparison

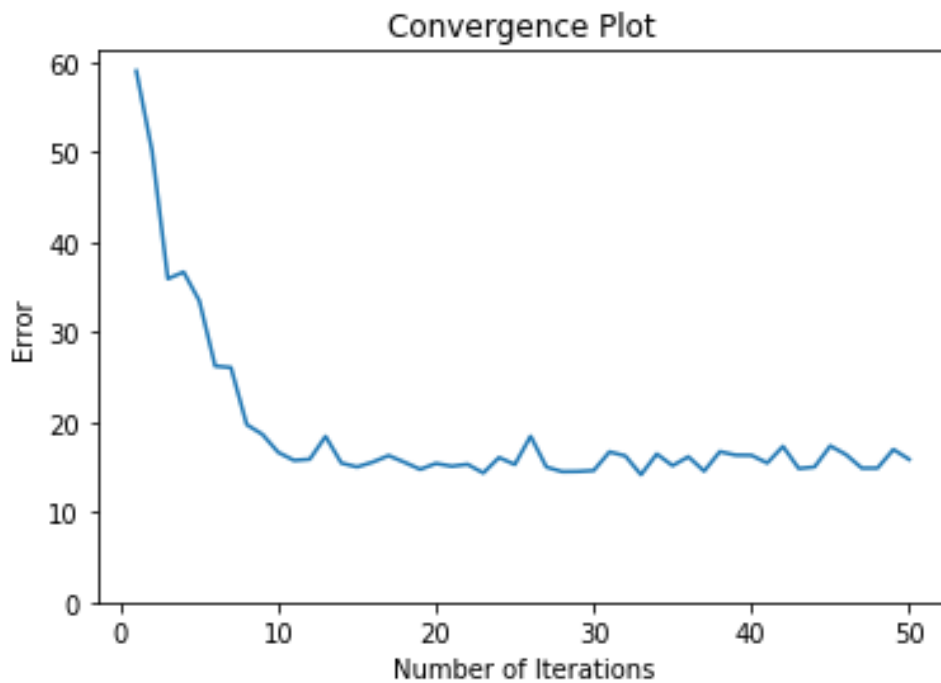


Figure 5.11: Convergence Plot - Best Optimization

Table 5.12: Convergence Results - Best Optimization

Attributes	Values
A_0	15
B_0	0.1
A_{conv}	8.295
B_{conv}	0.126
MAPE	13.767
RMSE	43.120
RMSN	0.157
Number of Iterations	51
Objective Function Evaluations	118

Figure 5.9 and 5.10 illustrates the final predicted values compared to the observed data. The optimization algorithm has fitted the ME model closest to the observed data. The goodness of fit is also reflected through the mean absolute percentage error in table 5.12. Figure 5.11 shows the convergence of the error term. After first 10 - 15 iteration the error term stays more or less in the same region and does not converge further below than that. The convergence plot shows that there is still some level of noise available in the predicted output of the ME model. The main reason for this noise is the discrete level on which the predicted and observed data is compared. Aggregation level of the data is on network. This cannot allow the ME model to fit perfectly. This is the reason that after few iterations, it just stays at certain level.

Waiting time in figure 5.10 has been mapped effectively with the observed data. However, for detour time figure 5.9, ME outputs are decreasing with the increase in fleet size. This is because ME calculates analytically and it follows a certain slope, where as observed data has some fluctuations in detour time even with the increase of fleet size. As the fleet increases, the ride sharing demand also increases and with the presence of network stochasticity the detour time increased rather than decreasing if compared to the ME output.

5.2.4 Mode Choice Methods Comparison

Calibrated ME model has computational advantage over the conventional simulation-based equilibrium mode choice method. The run-time comparison for Munich network for ME model and a single simulation run is shown in table 5.13.

Table 5.13: Run Times Comparison - Munich

Mode Choice Method	Run Time (secs)
ME Model	10-20
Single Simulation Run	115-170

In order to achieve a simulation-based equilibrium, approximately 5 - 10 simulation runs are required (Y. Liu et al., 2019). For calibration of ME model, approximately 10 iterations were needed as shown in convergence plot 5.11. Once ME model is calibrated, it only requires single run (10-20 secs run-time) to perform mode choice.

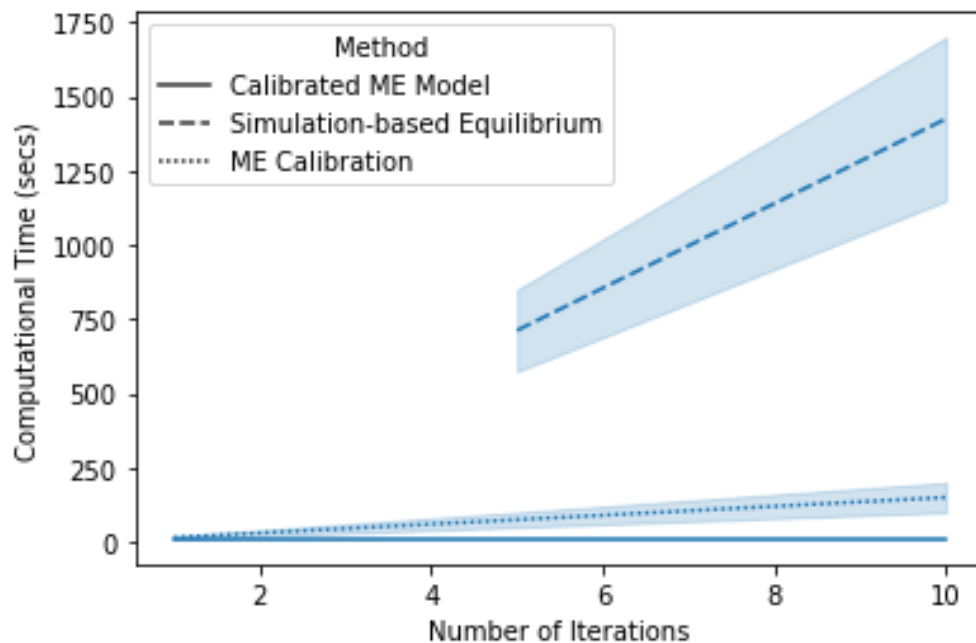


Figure 5.12: Computational Time Comparison

Figure 5.12 illustrates the computational time comparison of the two methods. The total time needed for calibrating an ME model after running 10 iterations and then using it to perform mode choice is significantly less than running simulation runs from 5 - 10. Even for the minimum of 5 iterations, the run time is very high in relation to using calibrated ME model.

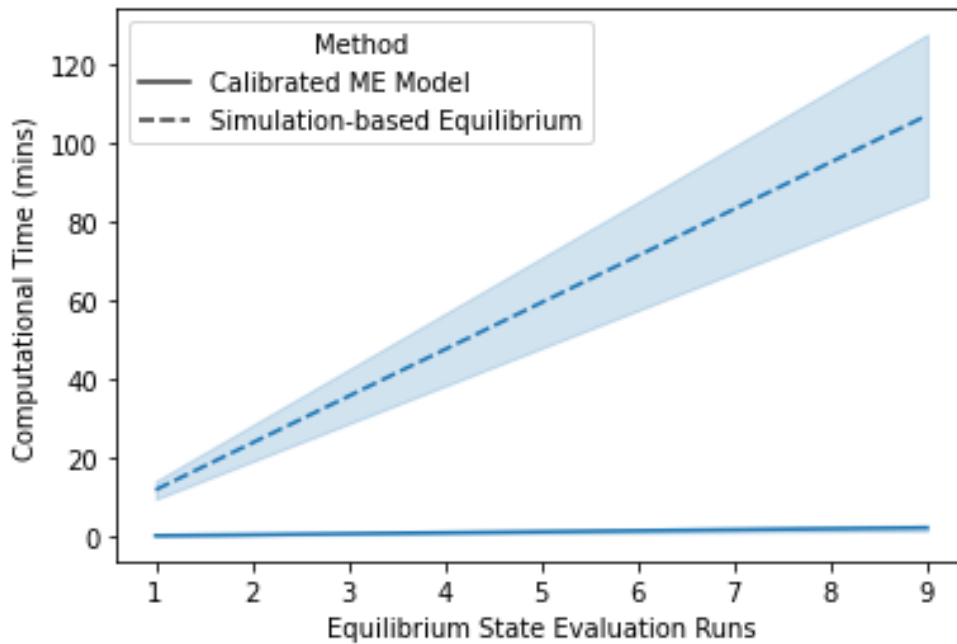


Figure 5.13: Computational Time Comparison - Impact Assessment Analysis

Figure 5.13 shows an example of the computational time comparison where both methods are used in performing an impact assessment analysis. Multiple equilibrium state evaluation runs are needed to perform the impact assessment analysis. For this example it is considered that a simulation-based mode choice method requires 5 iterations to achieve the equilibrium state after every change in supply in comparison to a calibrated ME model that requires its basic run-time of 10-20 secs to perform mode choice. This example shows the advantage of using a calibrated ME model over a regress and extensive process of multiple simulation run, which is computationally expensive. Note that the computational time of running simulations will further increase as the network size increases. Therefore, the utility of using a calibrated ME model increases multi-fold as the network size increases.

5.2.5 Model Applications

The optimization algorithm results includes the converged values of the parameters A and B which makes the analytical market equilibrium model calibrated for this study network. The calibrated ME model as proposed, can now be used to perform mode choice for varying fleet size and reduce expensive computations. For any change, the analytical outputs will output the ride sharing demand, which can be simulated for this network and various useful simulation related attributes can be extracted. Figure 5.14 shows the mode share of the ride sharing demand for a large range of fleet size. Similarly figure 5.15 shows the change in probabilities of modes as the fleet size increases. Such application of calibrated analytical model is very useful to get a quick overview how the change in supply will effect the ride sharing demand.

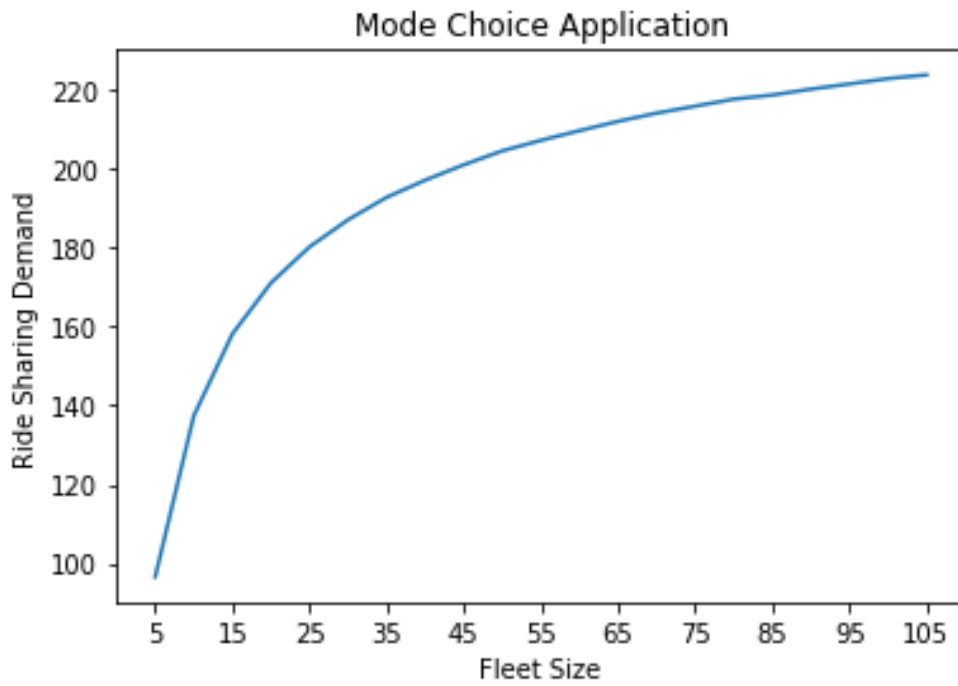


Figure 5.14: Mode Choice Application - Mode Share

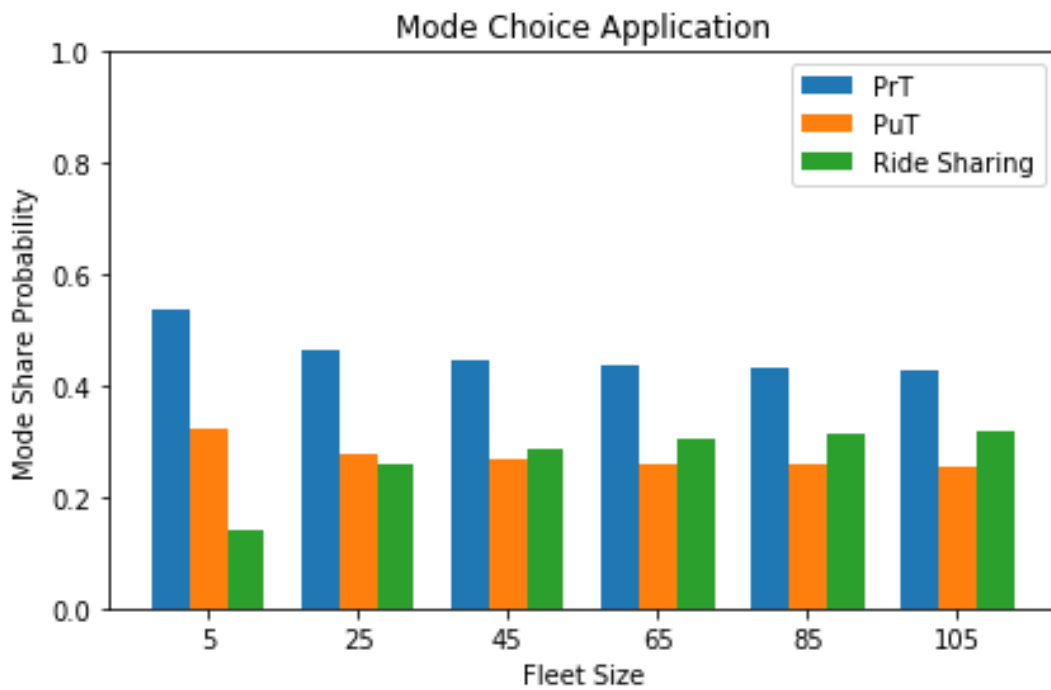


Figure 5.15: Mode Choice Application - Mode Probabilities

Mode share initially increases significantly as the fleet increases, however after certain fleet size, the change in demand is very less. The trend shows that no matter how much fleet size will increase, the demand will remain constant. This information can help decide service providers to find the optimum fleet size to serve the network demand. It can allow them to choose optimum fleet for which maximum revenue can be generated.

The optimum fleet's ride sharing demand can also be modelled using any simulation platform to extract simulated attributes. Apart from the default mode choice method of the simulation platform, one of the additional utility of the calibrated ME model is the analytical service attributes that are calculated along with the ride sharing demand. Case scenarios where simulations are not necessary, the calibrated ME model can be used in place to retrieve service attributes for any change in supply.

6 Conclusion

6.1 Summary

Mode choice for ride sharing services is an iterative process where the current state of the network needs to be evaluated to achieve an equilibrium state and thus the ride sharing demand. However, in order to access the current state of the network, service attributes such as detour time and waiting time of the ride sharing service is required and this can be obtained by running simulations. Simulations however becomes computationally expensive as the network size increases. Also any equilibrium calculated is valid for the given supply and demand parameters and has to be calculated again for any change in these parameters. This increases the computational time many folds.

In this study a simulation based mode choice method has been developed for the ride sharing services. An analytical ride sharing market equilibrium model is calibrated to be used instead of simulation based rigorous method to perform the mode choice. The calibration process of the market equilibrium model includes a database of simulated data of the ride sharing services for varying fleet size and ride sharing demand. The simulation database is setup using a ride sharing scheduler and simulation platform. This simulation database is taken as the observed data for model calibration in this study. The analytical outputs of the market equilibrium model are the predicted values. The market equilibrium model has parameters A and B which are a function of analytical detour and waiting time respectively. These parameters can be calibrated in a way that the model outputs are matched with the observed data available.

Model calibration essentially becomes an optimization problem where the ME outputs are compared with observed data and the model parameters are changed using an optimization algorithm from python libraries to better fit. The goodness of fit is evaluated using the mean absolute percentage error. The optimization algorithm inputs some initial guess values of the parameters and returns the converged values. The converged values can converge to a local minima therefore optimization runs are performed on different sets of initial values to obtain the global minima convergence. The final converged parameters calibrates the ME model and makes it viable for any change in the supply parameters.

The calibrated ME model is then integrated with the ride sharing simulation platform as the default mode choice method. Once calibrated for a certain network, ME model performs mode choice for any change in supply in few seconds where as an simulation equilibrium require multiple runs and the time of these simulations increases with the increase in size of network. Also the simulation equilibrium has to be repeated again if the supply changes.

Considering the benefits of lesser computational time over regress and extensive process of running multiple simulations, it is concluded that calibrated ME model has the potential to

be an effective mode choice method for ride sharing services. The error convergence results obtained from the case study also shows promising results and makes the calibrated ME model as a viable alternative to conventional simulation driven mode choice equilibrium.

6.2 Limitations and Future Scope

For this study, the model calibration has been performed for the service aggregated values at network level. However, this resulted in a constant noise in the aggregated data which could not be converge below a certain value. To better map the analytical market equilibrium model on the simulation database, the above calibration can be performed at a more dis-aggregate level i.e., OD pair level. The ME model inputs the parameters A and B which calculates the aggregated network attributes, it can also generate the same results for each OD pair level. Similarly, simulation database can also be compiled at the OD pair level rather network level. So for every single value of N , we can have demand Q for all the OD pairs individually. This will increase the sensitivity of the observed data and the ME predicted outputs will improve and fit better with the observed data.

One of the aim of this study was to include the network stochasticity by simulating dynamic vans with actual traffic demand. However due to the errors arised between the scheduler and simulation platform, and the time constraint of this study, the simulations were performed on a standalone scheduler. The standalone scheduler mapped the detour and waiting time of the network however the interaction of dynamic vans with the traffic was not included. During the setup of simulation database, the simulation platform can also include the network traffic demand which was originally planned for this study as well. The interaction with network traffic demand will generate more realistic results and reduce the noise in the data which can be seen in convergence results.

Calibrated ME model replaces the default mode choice for the ride sharing services in a simulation platform. For given demand and supply parameters, it can generate the demand which can then be simulated on a network and various impact assessment scenarios related to the ride sharing services can be performed. A calibrated ME model will come in handy especially in any kind of impact assessment scenario where the supply and demand parameters will be continuously changing. Also the calibrated ME model itself returns the approximated network detour and waiting time directly for which the simulation runs are not necessary.

Bibliography

- Abou-Zeid, M., & Ben-Akiva, M. (2010). A model of travel happiness and mode switching. *Choice modelling: The state-of-the-art and the state-of-practice*. Emerald Group Publishing Limited.
- 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-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3), 462–467.
- Araldo, A., Akkinepally, A. P., Basak, K., Seshadri, R., Biran, B. H. N., Deshmukh, N., Kumar, N., Azevedo, C. L., & Ben-Akiva, M. (2018). Implementation & policy applications of amod in multi-modal activity-driven agent-based urban simulator simmobility.
- Arbib, J., & Seba, T. (2017). Rethinking transportation 2020-2030. *RethinkX*, May, 143, 144.
- Ashok, K., Dillon, W. R., & Yuan, S. (2002). Extending discrete choice models to incorporate attitudinal and other latent variables. *Journal of marketing research*, 39(1), 31–46.
- Bischoff, J., Maciejewski, M., & Nagel, K. (2017). City-wide shared taxis: A simulation study in berlin. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 275–280.
- Bolduc, M. B.-A. D., & Ben-AkiWand, D. (1996). *Multinomial probit with a logit kernel and a general parametric specification of the covariate structure* (tech. rep.). MIT Working Paper.
- Boutueil, V. (2018). New mobility services. *Chapter*, 2, 39–78.
- Cai, H., Wang, X., Adriaens, P., & Xu, M. (2019). Environmental benefits of taxi ride sharing in beijing. *Energy*, 174, 503–508.
- Cardell, N. S., & Dunbar, F. C. (1980). Measuring the societal impacts of automobile downsizing. *Transportation Research Part A: General*, 14(5-6), 423–434.
- Cardell, N., & Reddy, B. (1977). A multinomial logit model which permits variations in tastes across individuals. *Charles Rivers Associates Inc.*
- Colvile, R., Hutchinson, E. J., Mindell, J., & Warren, R. (2001). The transport sector as a source of air pollution. *Atmospheric environment*, 35(9), 1537–1565.
- Cools, M., Moons, E., Janssens, B., & Wets, G. (2009). Shifting towards environment-friendly modes: Profiling travelers using q-methodology. *Transportation*, 36(4), 437–453.
- Daly, A., & Zachary, S. (1978). Improved multiple choice models. *Determinants of travel choice*, 335, 357.
- de Dios Ortúzar, J., & Willumsen, L. G. (2011). *Modelling transport*. John wiley & sons.
- Dia, H., & Javanshour, F. (2017). Autonomous shared mobility-on-demand: Melbourne pilot simulation study. *Transportation Research Procedia*, 22, 285–296.

- Djavadian, S., & Chow, J. Y. (2017a). An agent-based day-to-day adjustment process for modeling 'mobility as a service' with a two-sided flexible transport market. *Transportation research part B: methodological*, 104, 36–57.
- Djavadian, S., & Chow, J. Y. (2017b). Agent-based day-to-day adjustment process to evaluate dynamic flexible transport service policies. *Transportmetrica B: Transport Dynamics*, 5(3), 281–306.
- Domencich, T. A., Kraft, G., & Valette, J. P. (1968). *Estimation of urban passenger travel behavior: An economic demand model*. Charles River Associates.
- Domencich, T. A., & McFadden, D. (1975). *Urban travel demand—a behavioral analysis* (tech. rep.).
- 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(1), 143–158.
- Fiedler, D., Čáp, M., & Čertick, M. (2017). Impact of mobility-on-demand on traffic congestion: Simulation-based study. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 1–6.
- Gaudry, M., & Quinet, E. (2012). Shannon's measure of information, path averages and the origins of random utility models in transport itinerary or mode choice analysis.
- Goodwin, P. B. (1977). Habit and hysteresis in mode choice. *Urban studies*, 14(1), 95–98.
- Greenblatt, J. B., & Shaheen, S. (2015). Automated vehicles, on-demand mobility, and environmental impacts. *Current sustainable/renewable energy reports*, 2(3), 74–81.
- Hartley, T., & Ortuzar, J. d. D. (1980). Aggregate modal split models: Is current uk practice warranted? *Traffic Engineering & Control*, 21(1).
- Hörl, S. (2017). Agent-based simulation of autonomous taxi services with dynamic demand responses. *Procedia Computer Science*, 109, 899–904.
- Horowitz, J. L. (1981). Sources of error and uncertainty in behavioral travel demand models. *New horizons in travel-behavior research*.
- 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.
- Kraft, G. (1968). Demand for intercity passenger travel in the washington-boston corridor, north-east corridor project report systems analysis and research corporation.
- Kuhnimhof, T. (2017). Car ownership and usage trends in germany—response to the commission on travel demand's call for evidence: Understanding travel demand.
- Langdon, M. (1976). *Modal split models for more than two modes* (tech. rep.).
- 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. (2019). The value of prepositioning in smartphone-based vanpool services under stochastic requests and time-dependent travel times. *Transportation Research Record*, 2673(2), 26–37.

- Linda, S. (2003). Can public transport compete with the private car? *IATSS research*, 27(2), 27–35.
- Liu, J., Kockelman, K. M., Boesch, P. M., & Ciari, F. (2017). Tracking a system of shared autonomous vehicles across the austin, texas network using agent-based simulation. *Transportation*, 44(6), 1261–1278.
- 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.
- Local, T. (2018). *Munich named traffic jam capital of germany (again)*. <https://www.thelocal.de/20180206/munich-traffic-jam-capital> (accessed: 12.01.2021)
- 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.
- Lu, Q.-L. (2020). *Developing dynamic pricing methods for ride-sharing services* (Masterarbeit). Technical University of Munich.
- Ma, S., Zheng, Y., & Wolfson, O. (2013). T-share: A large-scale dynamic taxi ridesharing service. *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 410–421.
- 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 [Connected and Automated Vehicles: Effects on Traffic, Mobility and Urban Design]. *International Journal of Transportation Science and Technology*, 6(1), 13–27. <https://doi.org/https://doi.org/10.1016/j.ijst.2017.05.005>
- McFadden, D., & Train, K. (2000). Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5), 447–470.
- Morikawa, T., & SASAKI, K. (1998). Discrete choice models with latent variables using subjective data (chapter 24 of travel behaviour research: Updating the state of play). *Publication of: Elsevier Science, Limited*.
- Oi, W. Y., & Shuldiner, P. W. (1962). An analysis of urban travel demands.
- OpenStreetMap contributors. (2017). Planet dump retrieved from <https://planet.osm.org>.
- Ortuzar, J. d. D. (1980a). Mixed-mode travel demand forecasting techniques. *Transportation Planning and Technology*, 6(2), 81–95.
- Ortuzar, J. d. D. (1980b). Modelling park'n ride and kiss'n ride as submodal choices. *Transportation*, 9(3), 287–291.
- Qian, X., & Ukkusuri, S. V. (2017). Taxi market equilibrium with third-party hailing service. *Transportation Research Part B: Methodological*, 100, 43–63. <https://doi.org/https://doi.org/10.1016/j.trb.2017.01.012>
- Qurashi, M., Jiang, H., & Antoniou, C. (2020). Modeling autonomous dynamic vanpooling services in sumo by integrating the dynamic routing scheduler. *SUMO User Conference 2020*.
- Rathaus Unterschleissheim. (2020). Unterschleissheim's statistics.
- Shaheen, S., Sperling, D., & Wagner, C. (1998). Carsharing in europe and north american: Past, present, and future.

- Sobel, K. L. (1980). *Travel demand forecasting by using the nested multinomial logit model*.
- Solli-Sæther, H., & Gottschalk, P. (2010). The modeling process for stage models. *Journal of Organizational Computing and Electronic Commerce*, 20(3), 279–293.
- Spear, B. D. (1977). *Applications of new travel demand forecasting techniques to transportation planning. a study of individual choice models* (tech. rep.).
- Tsiamasiotis, K. (2019). *Stated preference survey design to identify factors for using dynamic vanpooling* (tech. rep.). Supervisors: C. Antoniou, E. Chaniotakis.
- Van Can, V. (2013). Estimation of travel mode choice for domestic tourists to nha trang using the multinomial probit model. *Transportation Research Part A: Policy and Practice*, 49, 149–159.
- Warner, S. L. (1962). Stochastic choice of mode in urban travel: A study in binary choice. *METROPOLITAN TRANSPORTATION SERIES*.
- Williams, H. (1979). *Travel demand forecasting: An overview of theoretical developments*. School of Geography, University of Leeds.
- Williams, H. C. (1977). On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and planning A*, 9(3), 285–344.
- Wilson, A. G. (1969). The use of entropy maximising models, in the theory of trip distribution, mode split and route split. *Journal of transport economics and policy*, 108–126.
- 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(9), 819–842.
- Wong, K., Wong, S. C., Yang, H., & Tong, C. (2002). A sensitivity-based solution algorithm for the network model of urban taxi services. *Transportation and traffic theory in the 21st century*. Emerald Group Publishing Limited.
- Wong, S. C., & Yang, H. (1998). Network model of urban taxi services: Improved algorithm. *Transportation Research Record*, 1623(1), 27–30. <https://doi.org/10.3141/1623-04>
- Zhang, W., Guhathakurta, S., Fang, J., & Zhang, G. (2015a). Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. *Sustainable Cities and Society*, 19, 34–45.
- Zhang, W., Guhathakurta, S., Fang, J., & Zhang, G. (2015b). The performance and benefits of a shared autonomous vehicles based dynamic ridesharing system: An agent-based simulation approach. *Transportation Research Board 94th Annual Meeting*, (15-2919).