Shared Mobility-on-Demand Systems

Reducing service unreliability

E.L. Matthieu

Delft University of Technology

Shared Mobility-on-Demand Systems

Reducing service unreliability

by

Mees Matthieu

to obtain the degree of
Master of Science
at the Delft University of Technology.

Thesis committee:	Dr. J. Alonso-Mora,	TU Delft, supervisor
	Dr. A.S. Fielbaum Schnitzler,	TU Delft, daily supervisor
	Dr. B. Atasoy	TU Delft
Student number:	4436792	
Master:	Mechanical Engineering	
Track:	BioMechanical Design	
Specialization:	BioRobotics	

October 23, 2022

TU Delft
Delft
University of
Technology

Abstract

On-demand ridesharing might be an effective solution to reduce traffic congestion and, as a result, reduce emissions to mitigate environmental issues. However, travel times are more uncertain in ridesharing services compared to non-shared transit services due to their shared nature. For example, at one moment, the users are informed that the travel time takes five minutes, a moment later, a request is added to the trip and the users are now confronted with a travel time of ten minutes instead. This uncertainty in travel time is called unreliability. Unreliability is perceived as inconvenience by the users and it has a negative impact on the users' travel habits regarding ridesharing service.

Therefore, this research provides a novel solution to deal with this problem while mitigating the negative effects on the other quality-of-service indices, e.g., waiting time, rejection rate, and delays. This is achieved by providing two indicators to each new user: (1) the possibility that an additional request will be added to the trip that negatively influences the trip duration of the user; and (2) a prediction of the magnitude of a possible delay for whenever the case occurs that a new request will be added to the trip. These indicators are based on predictions, created by shareability shadows which determine the number of requests that might be added to the trip, and implementing a simple time-series demand forecasting method named exponential smoothing. The shareability shadow makes predictions by confining the regions of ridesharing opportunities by analysing the travel constraints of the involved users. The solution is validated using a state-of-the-art routing and assignment method and a test case of Manhattan, New York City. With the formulated measure, on average, 58% of the requests receive a correct prediction about the possibility of a negative change, 17% receive an indecisive prediction, i.e., a prediction stating that the possibility of a negative change is "medium", and 25% of the requests receive an incorrect prediction about the possibility of a negative change. Moreover, the measure successfully provides a predicted magnitude of the travel time increase with a 95% confidence interval.

Index Terms – on-demand ridesharing, ridesplitting, uncertainty & minimize/reduce unreliability

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. J. Alonso-Mora and Dr. A.S. Fielbaum Schnitzler for their invaluable expertise and advice during this research. Thank you Dr. A.S. Fielbaum Schnitzler for providing all the help I needed to succeed in this research. Your insightfulness and guidance was invaluable.

Additionally, I would like to thank Minne. I am grateful for all the support I received from you. It was a long journey, but you helped me to get through it.

Finally, I would like to express my gratitude to my friends, especially Michael, and family.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AR	Autoregressive
ARIMA	Autoregressive integrated moving average
ARMA	Autoregressive moving average
DARP	Dial-a-ride problem
DO	Drop-off
FV	Fewer Vehicles
IVTT	In-vehicle travel time
MA	Moving average
NC	Completed requests that face no change in travel time
O-D	Origin-destination
PU	Pick-up
PUDO	Pick-up and drop-off
RA	Vehicles for which Rescheduling is Allowed
SMoD	Shared mobility-on-demand
SV	Smaller Vehicles
TNC	Transportation network company
VMT	Vehicle Miles Travelled
W	Completed requests that both face a worse waiting time and in-vehicle time compared to their first-announced values
WIV	Completed requests that face a worse in-vehicle time compared to their first-announced values
WW	Completed requests that face a worse waiting time compared to their first-announced values

Symbols

$\#\mathcal{N}_{\Delta}^{tot}(z_D, T)$	Number of nodes in $\mathcal{N}_{\Delta}^{tot}(z_D, T)$
$\#\mathcal{N}_{\Delta}(n_D)$	Number of nodes in $\mathcal{N}_{\Delta}(n_D)$
$\#\mathcal{N}_{\Delta}^{tot}(t, T)$	Number of nodes in $\mathcal{N}_{\Delta}^{tot}(t, T)$
$\#\mathcal{N}_{\Omega}(n)$	Number of nodes in $\mathcal{N}_{\Omega}(n)$
$\#\mathcal{N}_{\Omega}(t)$	Number of nodes in $\mathcal{N}_{\Omega}(t)$
$\#\mathcal{N}_{C}$	Number of nodes in the spatial domain of the city
$\#\mathcal{N}_{\Delta}(z_D, n_D)$	Number of nodes in $\mathcal{N}_{\Delta}(z_D, n_D)$
$\#\mathcal{N}_{\Omega}(z_O, n)$	Number of nodes in $\mathcal{N}_{\Omega}(z_O, n)$
$\#\mathcal{N}_{z_D}$	Number of distinct nodes in zone z_D
$\#\mathcal{N}_{z_O}$	Number of distinct nodes in zone z_O
α	Exponential smoothing constant
χ_k	Binary variable defining whether request r_k has been assigned or rejected
$\Delta D(r)$	Increase in delay for request r since its first announcement
ΔL	Extra length vehicle v has to travel to serve the requests
$\Delta t_d(r)$	Increase in detour time for request r since its first announcement
$\Delta t_v(r)$	Increase in in-vehicle time for request r since its first announcement
$\Delta t_w(r)$	Increase in waiting time for request r since its first announcement
Δ	Maximum delay
Δ_v	Tolerable ridesharing delay of the users assigned to vehicle v
$\epsilon_{i,j}$	Binary variable of assigning vehicle v_j to trip T_i
$\hat{Y}_t$	Predicted demand at time t
$\hat{Y}_{z_O, z_D}(t)$	Predicted demand at time t with an origin in zone z_O and a destination in zone z_D
λ	Average number of trips per hour
$\lambda(t)$	Number of emerging requests at time t
$\lambda(t_n)$	Number of emerging requests at time t_n
$\lambda_r(t)$	Number of possible emerging requests r' that individually satisfy the trip constraints and negatively influence the trip duration of r, determined at time t

$\lambda_r^w(t)$ Number of possible emerging requests r' that individually satisfy the trip constraints and negatively influence the waiting time of r, determined at time t

$\lambda_r^{iv}(t)$ Number of possible emerging requests r' that individually satisfy the trip constraints and negatively influence the in-vehicle time of r, determined at time t

$\lambda_{z_O,z_D}(t_n)$ Number of emerging requests at time t_n with their origin in zone z_O and their destination in zone z_D

$\mathcal{A}_v$ Set of already assigned requests to vehicle v

$\mathcal{C}$ Cost of the assignment

$\mathcal{E}_{TV}$ Set of $\{i,j\}$ indexes for which an edge $e(T_i,v_j)$ exists in the RTV-graph

$\mathcal{N}(T,z_D)$ All nodes in the itinerary of trip T located in zone z_D

$\mathcal{N}(T,D_r)$ All nodes in the itinerary of trip T until node D_r

$\mathcal{N}(t,T)$ All nodes in the itinerary of trip T at time t

$\mathcal{N}_\Delta^{tot}(z_D,T)$ Set of nodes in zone z_D that can be reached from any of the remaining nodes of the trip's itinerary that satisfy Ω_v and Δ_v

$\mathcal{N}_\Delta(n_D)$ Set of nodes that can be reached from node n_D that satisfy Ω_v and Δ_v

$\mathcal{N}_\Delta^{tot}(t,T)$ Set of nodes that can be reached from any of the remaining nodes of the trip's itinerary that satisfy Ω_v and Δ_v

$\mathcal{N}_\Omega(n)$ Set of nodes that can be reached from node n that satisfy Ω, Ω_v, and Δ_v

$\mathcal{N}_\Omega(t)$ Set of nodes that can be reached without exceeding Ω, Ω_v, and Δ_v

$\mathcal{N}_\Omega^{postO_r}(z_O,n)$ Set of nodes in $\mathcal{N}_\Omega(z_O,n)$ that can be reached posterior to pick-up node O_r

$\mathcal{N}_\Omega^{priorO_r}(z_O,n)$ Set of nodes in $\mathcal{N}_\Omega(z_O,n)$ that can be reached prior to pick-up node O_r

$\mathcal{N}_C$ Set of nodes in the spatial domain of the city

$\mathcal{N}_\Delta(z_D,n_D)$ Set of nodes in zone z_D that can be reached from node n_D that satisfy Ω_v and Δ_v

$\mathcal{N}_\Omega(z_O,n)$ Set of nodes in zone z_O that can be reached from node n that satisfy Ω, Ω_v, and Δ_v

$\mathcal{P}_v$ Set of picked-up requests by vehicle v

$\mathcal{R}$ Set of requests

$\mathcal{R}_{10}$ Set of completed requests over the last ten minutes

$\mathcal{R}_{cp}(T_h)$ Set of requests with a correct prediction of the possibility of a negative change given the threshold values T_h

$\mathcal{R}_{ip}(T_h)$ Set of requests with an incorrect prediction of the possibility of a negative given the threshold values T_h

$\mathcal{R}_{ko}$ Unassigned requests

$\mathcal{R}_{mp}(T_h)$ Set of requests with an indecisive prediction of the possibility of a negative given the threshold values T_h

$\mathcal{R}_{ok}$ Assigned requests

$\mathcal{V}$ Vehicle fleet

$\mathcal{Z}_\Delta(n,T)$ Set of reachable zones from any of the remaining nodes in trip T starting from node n that satisfy Ω_v and Δ_v

$\mathcal{Z}_\Omega(n)$ Set of reachable zones from node n that satisfy Ω, Ω_v, and Δ_v

ν Vehicle speed

Ω Maximum waiting time

Ω_v Tolerable ridesharing waiting time of the users assigned to vehicle v

C Area of the city

c_0 Operating costs per vehicle-time unit

c_a Cost of serving already assigned request r

c_o Operator cost to serve request r

c_p Cost of serving passenger p

c_r Cost of serving request r

c_v Maximum vehicle capacity

$c_{i,j}$ Cost of edge $e(T_i,v_j)$

c_{ko} Cost of rejecting a request

$cp_v(t)$ Current position of vehicle v at time t

$D(r)$ Travel delay of request r

D_r	Destination of request r
D_T	Destination of trip T
$D_{r'}$	Destination of request r'
$e(T, v)$	Feasible edge of the RTV-graph
L	Dimensionless quantity
O_r	Origin of request r
O_T	Origin of trip T
$O_{r'}$	Origin of request r'
p_v	Unitary cost of in-vehicle travel time for the requests
p_w	Unitary cost of waiting for the requests
r	Request
r'	New request that can be added to the trip and negatively influences the trip duration of request r
S	Shareability
s_{cp}	Score value of having a threshold value that resulted in a correct prediction
s_{ip}	Score value of having a threshold value that resulted in an incorrect prediction
s_{mp}	Score value of having a threshold value that resulted in an indecisive (medium) correct prediction
T	Trip
t	time
$t_d(r)$	Detour time of request r
t_n	Time at which the vehicle reaches node n
t_r	Emerging time of request r
t_r^d	Drop-off time of request r
t_r^p	Pick-up time of request r
$t_v(r)$	In-vehicle time of request r
$t_w(r)$	Waiting time of request r
$t_{assign}(r)$	Time at which request r has been assigned
T_h	Threshold values to determine the possibility of a negative change
t_n	Time at which node n will be reached given the vehicle's current itinerary
t_{ra}	Expected duration until a request appears
v	Vehicle

1 Introduction

1.1 Problem description

Over the years, an increase in emissions from the transportation sector has contributed to substantial hazards to the world's environment and human health. The rapid growth of motorized private car use has become a dominant factor in the air pollution of urban areas (National Research Council, 2002). Finite oil supplies, rising gas prices, traffic congestion, and lack of parking spaces are emerging concerns that urban cities have to deal with nowadays. Furthermore, in Europe, private vehicle occupancy rates (i.e., the number of passengers per vehicle) are relatively low, with an average of 1.1 passengers for commuter trips and 1.8 passengers for leisure trips. As a result, the reduced occupancy rate and the high transportation demand in urban areas contribute to an increase in traffic congestion (Agatz et al., 2012). In the United States of America in 2005, traffic congestion led to a waste of 2.9 billion gallons of fuel, 4.2 billion hours of extra travel time, and $78 billion of delay and fuel costs (Schrank & Lomax, 2007). Moreover, the transport conditions of private car use obstruct sustainable economic performance, environmental resilience, and social welfare (Redman et al., 2013). Managing congestion by expanding infrastructure is environmentally unsustainable and costly. Therefore, to aim for a sustainable environment, it is key to make more efficient use of the existing transportation infrastructure by minimizing private car use.

Public transit systems are a convenient way to utilize the infrastructure more efficiently, as they can carry multiple passengers simultaneously. These public transit systems reduce the vehicle miles traveled (VMT), resulting in a reduction in traffic congestion. However, these transit services operate on fixed routes and schedules, limiting their spatial and temporal coverage, stimulating users to use private cars instead. Conventional taxi services and mobility-on-demand taxi services can compensate for this limitation. On the contrary, these services increase traffic congestion as their occupancy rates are low (Agarwal et al., 2019; Tirachini & Gomez-Lobo, 2020; Henao & Marshall, 2020). Moreover, these services take over public transport users as they are more flexible, convenient, and often faster, resulting in an increase in congestion (Dai et al., 2021).

It can be concluded that fixed-line public transportation services and taxi services contain flaws in solving traffic congestion. Nevertheless, with the rapid development of communication technologies, centrally controlled on-demand ridesharing services, also known as Shared Mobility On-Demand (SMoD) services, offer a convenient solution. Ridesharing refers to a mobile transit service in which different users can share the same vehicle simultaneously if their paths are compatible. Ridesharing allows users to split travel costs with others at the expense of convenience due to detours to serve other users of the shared trip. Moreover, ridesharing effectively uses car seats and delivers an important opportunity to increase occupancy rates, reduce traffic congestion, and reduce environmental issues (Furuhata et al., 2013). H. Zheng et al. (2019) indicate in a real-world on-demand ridesharing data investigation of the urban area of Hangzhou, China, that ridesharing positively influences users' travel habits. This means that in the short-term, the number of vehicles on the road will reduce by 3.051 vehicles per day (which is 2.6‰ of the vehicle ownership in Hangzhou). In the long-term, the willingness to purchase a car will be reduced. Additionally, a real-world data study on ridesharing services by X. Liu et al. (2021) indicates that ridesharing decreases the VMT by 22% on average compared to non-ridesharing services.

Although dynamic on-demand ridesharing services can combine the flexibility and speed of private cars while demanding the same reduced costs of public transport services, it provides a reduced convenient service in terms of unreliability. In this research, unreliability implies that it is hard for users to know what quality-of-service they will experience during a trip, e.g., unreliable waiting times and unreliable in-vehicle travel times. This means that users may face discomfort when the realization of their trips differs from what was initially predicted. Changes in travel time can occur when a new user is added to the trip that changes the route.

Unreliability is an important feature in ridesharing, as deviations from the planned travel times will result in the loss of users. This is because users are not only concerned about trip duration, but also consider trip reliability, i.e., the probability of on-time arrival (C. Li et al., 2020). Moreover, if a match has been created between a user's request and the service, users are only willing to travel with the same service again if the rides take place as proposed; otherwise, they will leave the service and use traditional mobility options (Gargiulo et al., 2015). Alonso-González et al. (2020) studied the value of reliability, i.e., the willingness to pay to reduce travel time variability/uncertainty, of urban Dutch citizens in on-demand ridesharing services. They discovered that the value of reliability for the in-vehicle time appears to be 3.15–5.76 €/h, and the value of reliability regarding the waiting stage appears to be 2.24-8.58 €/h, depending on the magnitude of the waiting time and participant's income. Hence, operators should consider unreliability when designing their ridesharing strategy, as users value reliability. Moreover, operators should consider unreliability because travel time unreliability has a significant impact on the users' decision-making (Z. Li et al., 2010).

1.2 Thesis objective

Due to the discussed problem of unreliability in SMoD systems, the theoretical background of unreliability in SMoD systems is explored in Chapter 2. This chapter includes a section that aims to discover techniques to minimize unreliability. Consequently, it can be concluded that currently no suitable techniques exist to efficiently resolve the problem of unreliability without degrading the efficiency of the service drastically. The number of existing solutions is minor, and they are all unsuitable to implement in real-life applications due to their strong negative impact on the other quality-of-service indices. Therefore, the objective of research is to

> alter the state-of-the-art assignment procedure to provide more accurate travel times without increasing the rejection rate, average delays, and average waiting times too much.

1.3 Contribution

So far, the amount of research on unreliability in on-demand ridesharing is minor. Research that sticks out regarding unreliability in ridesharing is the work by Fielbaum and Alonso-Mora (2020), which measures unreliability over realistic scenarios to gain insights about the operation of on-demand ridesharing, arguing the relevance of a measure to control unreliability. The contribution of this research is twofold: Firstly, the background of unreliability in on-demand ridesharing is theoretically analyzed to gain an improved understanding of the problem, showing that no suitable solution exists. Secondly, a methodology is proposed to deal with the problem of unreliability, whereafter, the solution is validated using real-life test case simulations, illustrating that the solution leads to a reduced unreliable service. The solution is tested using the state-of-the-art routing and assignment ridesharing method of Alonso-Mora et al. (2017a) and real-life taxi trip data from Manhattan, New York City.

1.4 Thesis structure

This research continues in Chapter 2 with a discussion of the relevant background topics regarding unreliability in SMoD systems, including the causes of unreliability, techniques to assign vehicles to requests, and the available techniques to minimize unreliability. Additionally, the characteristics of the demand of mobility services are explored and mathematical demand forecasting models are analyzed because the formulated unreliability solution uses predictions about the demand. Next, Chapter 3 provides an overview of the model implementation, including the state-of-the-art routing and assignment method, the shareability shadow concept, and the demand prediction process. After that, the results of the implemented model are presented and discussed in Chapter 4. Finally, Chapter 5 provides a conclusion regarding the implemented model and the obtained results.

2 Theoretical background

This chapter provides the reader a theoretical basis regarding unreliability in on-demand ridesharing services, also known as Shared Mobility-On-Demand (SMoD) services, where operators centrally control the assignment process of users to vehicles. As stated in the introduction, unreliability refers to inconvenience faced by the users due to changing travel times as the result of changes in the trip schedule. Due to unreliability, users cannot know in advance how their trips will be in an accurate way in terms of waiting time at their pick-up point and in-vehicle travel time (IVTT). Therefore, the main focus of the theoretical background is the following question:

Which techniques can be identified to minimize unreliability in on-demand ridesharing?

To answer this question, three sub-questions are stated:

- What are the different causes of unreliability in ridesharing?
- How is unreliability perceived by the users?
- What are the different techniques used to match vehicles to requests?

Sections 2.1 and 2.2 discuss the causes of unreliability and how they are perceived by the users, respectively. Since the ride-matching technique influences the unreliability, Section 2.3 provides a theoretical background regarding the different techniques to assign vehicles to requests. Ultimately, Section 2.4 explores the different techniques in the literature that aim to minimize unreliability in ridesharing. By exploring the different techniques, it has been decided that a predictive method will be implemented to reduce unreliability. Therefore, Section 2.5 provides a theoretical background regarding demand prediction processes. Finally, Section 2.6, summarizes the findings of this chapter, including a conclusion regarding the discovered techniques of Section 2.4.

2.1 Causes of unreliability in ridesharing

In this section, the causes of unreliability are investigated to obtain an improved understanding of the unreliability topic in ridesharing. The objective of analyzing the theoretical background of unreliability in ridesharing is to discover techniques that minimize unreliability caused by the operator's rules. By investigating the causes of unreliability, it becomes clear which type of unreliability this research aims to minimize. The causes of unreliability in on-demand ridesharing can be divided into two main categories: unreliability causes present in all types of transport, and unreliability causes specific to on-demand ridesharing systems, meaning that the cause relates to the vehicles' shared assignments.

2.1.1 Unreliability causes in all types of transport

Not all causes of unreliability in ridesharing are specific to ridesharing services. A list of examples of the causes that are not explicitly related to ridesharing services is provided below.

- Traffic congestion
- Accidents
- Temporary traffic control
- Sudden natural changes
- Traffic signal failures
- Changes in travel demand
- Vehicle breakdowns
- Random factors, e.g., crossing pedestrians

Many of these factors affect the traffic flow and as a result, the trip duration is affected, making the transit service less reliable. Even if there are no accidents or congestions in traffic, there are still variations in traffic flow due to normal traffic conditions that affect the reliability of the trip. X. M. Chen et al. (2017) analyzed Beijing's network travel time reliability using real-world data from the largest on-demand taxi and ridesharing service platform in China (DiDi-Chuxing). Chen et al. indicate that road network reliability depends on time and region, i.e., the road network is more unreliable in AM/PM peaks and more reliable in the early morning. Additionally, central city regions have higher unreliability rates than non-central city regions.

Therefore, it is useful for SMoD services to consider traffic flow since variations in traffic flow affect the reliability of the service. Z. Liu et al. (2019) executed a study that utilizes collected travel time information to determine on-time arrival reliability. In this study, real-time information and historical information are examined to determine the traffic flow. To ensure the on-time arrival of the users, the authors introduce a path guidance method that manages the variations in the network conditions. The authors illustrate that the reliable paths, obtained with the path guidance method, have a significantly higher on-time arrival ratio than the paths obtained based on the shortest travel distance.

2.1.2 Causes of unreliability specific to on-demand ridesharing

Besides causes of unreliability that are present in all types of transport systems, some causes of unreliability are related to the specific characteristics of on-demand ridesharing services. For example, uncertainty regarding the route the vehicle will take is specific to on-demand ridesharing due to required detours to pick-up and drop-off other users of the trip. Furthermore, the behavior of the users may affect the unreliability of the service of other users in

SMoD systems. Overall, the causes of unreliability specific to ridesharing can be divided into two categories: unreliability induced by the users and unreliability induced by the operators (Fielbaum & Alonso-Mora, 2020).

Unreliability induced by the users
The causes of unreliability induced by the users which affect the trip unreliability of other shared trip users are described below.

- **Users are too late at their pick-up point**. When users arrive too late at their pick-up point, other users by which the trip is shared are also delayed. Additionally, future users might face longer waiting times. Assigned trips might get rejected by the system depending on the system's rules due to the increased vehicle delay. Besides, waiting users might cancel the trip due to the increasing waiting time, contributing to extra variation in trip duration of the other users of the trip.

  Kucharski et al. (2021) analyzed how users' non-punctual arrival at their pick-up point influences the system's performance. They discovered that when users arrive following a random distribution, each additional user is likely to increase the delay of other users of the trip. However, the marginal impact of each additional user decreases. Furthermore, Kucharski et al. state that even minimal lateness of users at their pick-up point is likely to shift out other users from high-degree trips. This hinders higher occupancy rates and negatively affects the efficiency of ridesharing service.

- **Users cancel an assigned trip when the vehicle is on its way**. When a user cancels a trip when the vehicle is on its way, other assigned users may get rejected by the system since the vehicle is no longer shared efficiently. These users will only be rejected if the system enforces certain rules to maintain efficiency. However, when the other users are not rejected, the waiting time and drive time of other users may decrease due to a decreasing detour time. In both situations, the trip unreliability of other users of the trip is affected.

  There are several reasons why users cancel an assigned trip. Bansal et al. (2020) studied the behavior of users in on-demand services. They state that the decision of the user to accept or cancel a trip depends on the provided waiting time and the difference between the provided time and the real pick-up time during previous trips. An overestimated displayed waiting time will result in increased cancellations by the users due to long waiting times. Conversely, an underestimated displayed waiting time increases the acceptance rate of the service. However, when an underestimated waiting time is displayed, the real pick-up time is delayed. As a result, the likelihood of choosing the service again is decreased due to unreliable provided waiting times. Furthermore, they illustrate that on-demand services can increase the user's trip acceptance rate by 10% if the services provide improved predicted waiting time information.

- **Users want to change their destination when they are in the vehicle**. Sometimes users spontaneously change their minds when they are in the vehicle and force the driver to change their destination. If the system accepts their new destination, the in-vehicle users will most likely face large detours. Furthermore, depending on the system's rules, other users may get rejected as they are not on the route anymore.

- **Users forget to sign up their fellow passengers for the trip**. Two situations are possible whenever a user forgets to sing-up fellow passengers:
  - Not all users can fit in due to insufficient vehicle capacity. Most probably, the users will cancel their trip. As stated before, cancellation induces unreliability on the other users of the shared trip.
  - All users can fit in but other waiting users have to be rejected by the system due to the insufficient amount available seats.

Unreliability induced by the operators

The operators decide how the system behaves whenever a new request is received. The rules of the system design strongly influence the unreliability of the service. When a new request is received, the system decides which vehicle is matched to the request and how all trips are grouped together. Moreover, new requests may result in reassigning a request to another vehicle, changing the waiting time of the request and influencing the travel time of other users. Each change in the vehicle's schedule affects the route. Depending on the system's rules, new requests can cause a rejection of previously assigned requests to further optimize the system's objective. Therefore, operators make decisions that influence the service unreliability by optimizing their objective (Fielbaum & Alonso-Mora, 2020). An example of unreliability induced by the operators is illustrated in Figure 1. Passenger 1 is on his way from his origin O_1 to his destination D. Before completing the trip of passenger 1, a new request emerges by user 2. The operators accept the request of user 2 and force passenger 1 to face a sudden detour to serve user 2 efficiently. This sudden detour is perceived as unreliability by passenger 1.

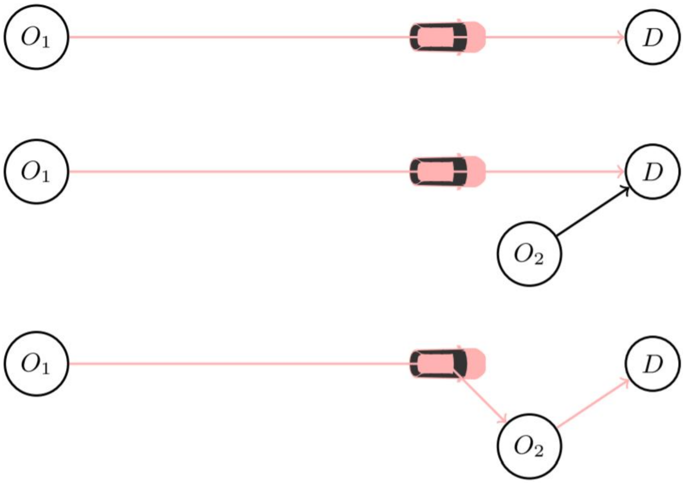


Figure 1: Operators induce unreliability for user 1 by accepting the request of user 2 (Fielbaum & Alonso-Mora, 2020).

Operators could decide to minimize the unreliability of the service. However, improved reliability may be achieved by implementing additional rules that affect the other quality-of-service indices (Fielbaum & Alonso-Mora, 2020). For example, the reliability of service could be improved by only adding new requests to a trip after the last request has been served. As a result, the unreliability of the trip is minimized, but fewer vehicles are available to serve the remaining requests. This strongly increases the waiting time of other users and the rejection rate significantly since the number of available vehicles is reduced. Hence, a trade-off exists between unreliability and other quality-of-service indices.

Even marginal changes in the system design greatly impact the matching process and the unreliability rate. A study that investigates the shareability rate of ridesharing systems, i.e., the amount of trips that could be shared, by analyzing the impact of service quality parameters is provided by Bilali et al. (2019). The analyzed service quality parameters in this study are the detour time, the time a user is willing to wait, and the boarding time. Results demonstrate that these parameters significantly influence the shareability rate. Hence, the authors argue that operators should use models that examine the shareability rate since different parameters result in different shareability rates. A similar study by Tachet et al. (2017) highlights the importance of different system parameters since the system parameters affect the fraction of trips that can be shared. As indicated by these studies, inferior changes in the system's design strongly affect the behavior of the SMoD service. Therefore, each operator's decision affects the service's unreliability.

This research is interested in minimizing unreliability induced by the operators' rules. Since it has been concluded that many causes of unreliability are related to the system design, it is useful to investigate the available techniques to match vehicles to requests. Investigating the matching techniques provides an improved understanding of how the system's behavior affects the unreliability of the service. An analysis of the different ride-matching techniques is provided in Section 2.3.

2.2 Unreliability as perceived by users

It is helpful to obtain an improved understanding of how the users perceive unreliability to create feasible solutions to solve the problem of unreliability. According to Fielbaum and Alonso-Mora (2020), two types of unreliability phenomena can be classified that describe how unreliability affects the user's perception of the quality of the SMoD service. The first type occurs within a single trip and is called *one-time unreliability.* The second type occurs between different realizations of the same trip and is called *daily unreliability.* These two unreliability types are discussed in the following two sections.

2.2.1 One-time unreliability

When a user is assigned to a trip, the real waiting time and in-vehicle travel time (IVTT) might differ from the first-announced values. The first-announced values are the travel time values a user receives from the service directly after a match has been made. These values estimate the waiting time and IVTT of the trip and might differ from the real-time values. As stated in Section 2.1, changes in waiting time and IVTT can be caused by different sources specific to or non-specific to SMoD systems. Therefore, within a single trip, changes can occur that induce unreliability in waiting and in-vehicle travel time of the user. These changes affect the short-term decisions of the user since there is less certainty that the user will be on time at the drop-off point (Fielbaum & Alonso-Mora, 2020). This research is interested in minimizing one-time unreliability.

2.2.2 Daily unreliability

Before a user decides to use a ridesharing service, the user has to consider how the ride's quality of service will be. Besides delays due to traffic conditions, the waiting time and delay time due to detours to share the trip with other users have to be considered before choosing a ridesharing service. Hence, daily unreliability is about variabilities in the duration of multiple executions of the same travel request. Therefore, this refers to unreliability between different realizations of the same trip.

A case study of the urban area of Chengdu, China, that investigated daily unreliability is executed by W. Li et al. (2019). The study investigates the travel time reliability of shared and non-shared rides with similar origin-destination pairs during the same time period (13:00-14:00). Results illustrate that the reliability of shared rides is inferior compared to non-shared rides due to ridesharing delays and detours. Additionally, Li et al. demonstrate that ridesharing forces users to plan for 37–61% extra time to ensure on-time arrival.

2.3 Techniques for assigning vehicles to requests

To understand which techniques can be adopted to minimize unreliability in ridesharing, it is useful to investigate which techniques can be applied to assign vehicles to requests. This section explores the different techniques for assigning vehicles to requests. First, the different algorithm design structures are discussed. Next, various objectives are identified. At last, different ridesharing constraints are explained.

2.3.1 Algorithm design

Fielbaum and Alonso-Mora (2020) investigate how different fixed fleet compositions and different assignment rules affect trip unreliability and illustrate that service unreliability can be improved at the cost of degrading other quality-of-service indices. Since it has been concluded in Section 2.1.2 that unreliability in ridesharing strongly depends on the implemented algorithm design, it is useful to obtain an improved understanding of the different ridesharing algorithm designs and structures. An overview of the different techniques for matching vehicles to requests, also known as ride-matching, assignment-matching, or vehicle dispatching, is provided in Figure 2. This section elaborates on the presented methods in Figure 2.

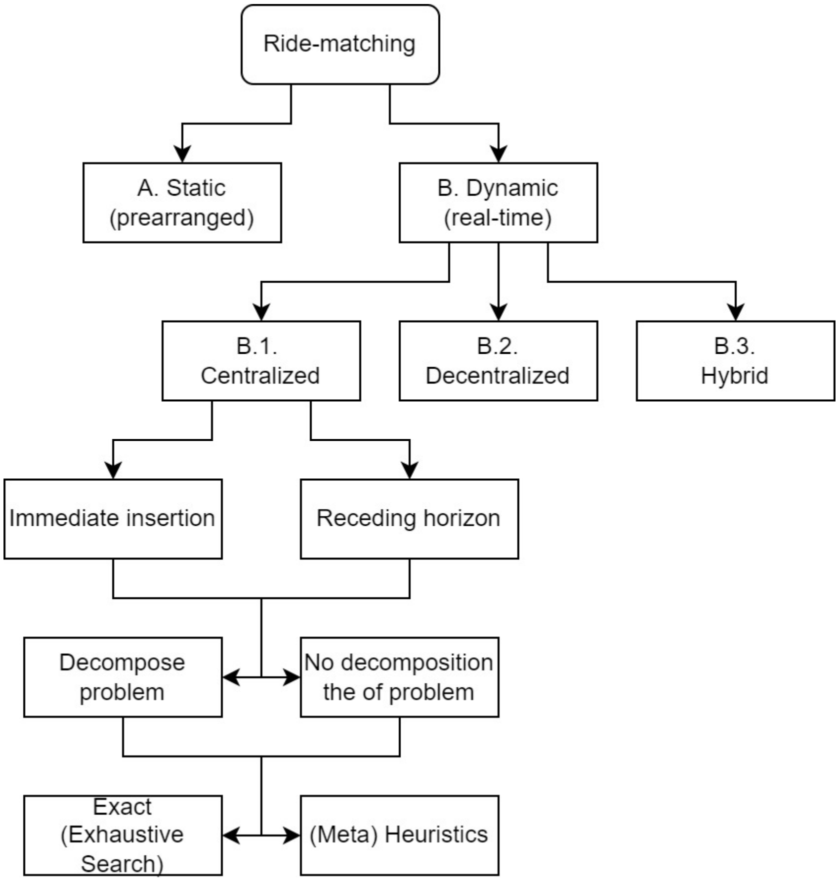


Figure 2: Overview of the Ride-matching techniques.

A. Static ridesharing

It has been observed that ride-matching techniques can be divided into static methods and dynamic methods. In static ridesharing, it is assumed that the number of user requests is known in advance prior to the execution of the ride-matching process. The origin and destination of every user are communicated in advance such that a feasible final route for all vehicles is decided. All trips are prearranged in static ridesharing. Static ridesharing systems are less difficult to develop than dynamic systems, however, static systems are not scalable and are only well suited for small instances. Moreover, static systems do not apply to real-time on-demand ridesharing systems (Silwal et al., 2019). Because this research is interested in dynamic ridesharing, no further subdivisions in static ridesharing techniques are provided.

B. Dynamic ridesharing

In dynamic ridesharing, the requests are not known in advance and the requests are received in real-time. New requests continuously enter the system and served users continuously leave the system. The incoming requests are handled directly or in short notice, ranging from a few seconds to a few minutes, depending on the system's rules. Due to the short response time, dynamic systems are more complex than static systems.

The implementation of dynamic ridesharing services has strongly increased due to the increase in mobile smartphones as they enable direct communication between users and ridesharing services (Agatz et al., 2011). Smartphones allow users to request trips whenever they want and wherever they are. Therefore, ridesharing services can operate in real-time and on-demand. In this research, the focus is on dynamic SMoD services.

Dynamic ridesharing systems can be divided into three different sub-methods: centralized, decentralized, and a hybrid version of centralized and decentralized systems. The fundamental differentiating features of the methods lie within the level of control (Silwal et al., 2019).

B.1. Centralized ridesharing

Centralized systems have a single system-wide objective function where all decisions are made centrally. Centralized systems can provide superior results due to their global optimization techniques. Conversely, it can be computationally heavy and slow for realistically sized instances of large metropolitan areas containing thousands of users, making it difficult to scale (Nourinejad & Roorda, 2016).

In this research, the emphasis lies on centralized systems for two reasons: (i) they provide more efficient solutions; and (ii) state-of-the-art solutions mainly consist of centralized systems. Centralized SMoD systems can be divided into Immediate insertion methods and Receding horizon methods. A detailed explanation of these two methods is provided in this section later on.

B.2. Decentralized ridesharing

Decentralized ridesharing systems are the opposite of centralized ridesharing systems. In decentralized systems, the decision-making does not rely on a central entity. It is composed of autonomous agents/drivers that optimize individual objectives using local information perceived from the system. Decentralized systems reduce computation time extensively. However, they are inferior regarding the solution quality compared to centralized systems

(Kümmel et al., 2017). Since decentralized systems are not the topic of interest, no further subdivisions in decentralized ridesharing are provided.

B.3. Hybrid ridesharing

Hybrid ridesharing, a combination of centralized and decentralized ridesharing systems, is introduced to achieve globally optimal real-time systems. The decision-making is distributed among all participating components, making it more scalable compared to centralized systems. The design of hybrid ridesharing systems is strongly dependent on the implementation (Silwal et al., 2019). No further subdivisions in hybrid systems are provided as hybrid systems are not the topic of interest in this research.

2.3.2 Dynamic centralized algorithm designs

This research is interested in dynamic centralized systems. Therefore, the algorithm designs for these systems are further investigated in this section. As illustrated in Figure 2, the design of dynamic centralized systems can be further narrowed down into *immediate insertion methods* and *receding horizon methods*. Thereafter, the assignment procedure can be divided into *decomposed* optimization problems and *not decomposed* optimization problems. At last, the assignment procedure can be solved with *exact searching methods* or *heuristics*. This section elaborates on these topics. Additionally, this section discusses the implementation of anticipatory methods. Anticipatory methods are discussed because the solution that will be implemented to reduce unreliability is based on demand predictions, and anticipatory methods deal with a similar problem of unknown future demands.

Immediate insertion

Immediate insertion methods are a subclass of centralized systems and handle the request directly as they arrive by inserting them into the schedule of the vehicles. Immediate insertion methods are event-based approaches since the algorithm initiates directly after receiving a request. This behavior constitutes a less computationally expensive approach since one request is handled simultaneously (L. Zheng et al., 2019). The downside of this behavior is that the system might be unable to keep up with the incoming stream of requests if the demand is high. Another disadvantage is that sub-optimal solutions might be created if no global optimization is executed that contains all requests (L. Zheng et al., 2019).

Receding horizon

Receding horizon methods or rolling horizon methods, a subclass of centralized systems, make plans utilizing all known information within a planning horizon. During each execution of the ride-matching process, the planning horizon is 'rolled' forward to include all currently received requests such that a batch of requests is handled at once. The rolling horizon method can be seen as a time-based approach. Key decisions of the rolling horizon approach include how frequently and when the algorithm is executed. Additionally, the duration of the rolling horizon is of huge importance. If the horizon is too short, it is more likely that sub-optimal solutions will be established. If the horizon is too long, requests might be served too late due to the increasing waiting time (S. Chen et al., 2020). The increasing waiting time stimulates users to cancel their requests. The rolling horizon method is computationally more expensive than the immediate insertion method due to the increasing number of requests handled at once. Conversely, due to the high amount of requests handled at once, solutions of higher quality are more likely to be achieved (Agatz et al., 2011).

Decompose problem vs not decomposing the problem

After it has been decided whether an immediate insertion approach or a rolling horizon approach is implemented, the decision is made whether the optimization problem is solved by decoupling the problem into smaller subsets or optimizing the requests over all available vehicles. Decomposing the problem is computationally less cumbersome, making it more suitable for large instances.

If the problem is not decomposed, it means that any vehicle can serve any group of requests. Lam et al. (2014) propose a scheduling algorithm that considers all vehicles during the matching process of a request. The authors formulate the scheduling problem as a mixed integer program. Modifications are allowed in the assigned trips that have not been served yet. The system's performance is studied with an artificial scenario containing 8 requests and 5 vehicles to realize a real-time execution of the scheduling approach. This illustrates that the optimization problem is hardly scalable when the problem is not decomposed. Therefore, the literature mainly focuses on algorithms that decompose the assignment problem and implement constraints on the time windows in which a request can be served.

Decoupling the assignment process into smaller problem sets can be executed by selecting a suitable set of vehicles that can satisfy the request(s) . The division of the problem can be executed based on geographical regions, distance-based information, or time constraints, i.e., the feasibility of a match. Alonso-Mora et al. (2017a) decompose the problem by computing all feasible trips and then assigning the trips to the vehicles. First, they create a request-vehicle shareability graph (RV-graph), representing which requests and vehicles might be pairwise-shared. Next, a graph of feasible trips and the vehicles that can serve the cliques of the RV-graph is computed (RTV-graph). Both graphs are created based on feasibility, i.e., the graphs have to satisfy a set of constraints (waiting time, delay, and capacity). Thereafter, to compute the best assignment of the vehicles to trips, an integer linear program (ILP) is solved.

When the problem is not decomposed, a more globally optimal solution might be obtained at the expense of computation time. Therefore, there is a trade-off between computation speed and the solution performance when deciding to divide the solution into smaller problem sets.

Heuristics vs exact

At last, the optimization problem of matching the requests to the vehicles is solved either with heuristics or with exact solving mechanisms. Heuristics are approximate solution techniques widely used in operations research to deal with difficult combinatorial optimization problems. Many combinatorial problems are NP-hard, making it impossible to solve them with exact and efficient procedures. As a result, heuristics are used to find a solution in finite time (Gendreau, 2003). Heuristics do not guarantee finding the optimal solution; however, most heuristics can sufficiently find an approximation of the optimal solution.

Exact methods or exhaustive search methods always find the optimal solution. However, these methods require a large amount of computation time and may not be able to find a solution in finite time for hard combinatorial problems. An example of an exhaustive search method is the Linear Programming technique. Linear Programming is a mathematical modeling technique to optimize a linear objective/cost function subjected to a set of linear constraints. The constraints represent the requirements that the objective function has to fulfill. The way that the cost function and the constraints are defined determines the type of optimization problem. This can either be a Linear Program, Integer Linear Program, or Mixed Integer Linear Program. If all variables are restricted to be integer values, the optimization problem is defined as an Integer Linear Program. When the optimization problem includes variables that are constrained to be integers, while other variables are allowed to be non-integers, the problem is defined as a Mixed Integer Linear Program.

After defining the cost function and the constraints, the cost function is most commonly solved with a Branch-and-Bound algorithm. The Branch-and-bound algorithm is used to find the optimal solution for combinatorial mathematical optimization problems by exploring the search space of possible solutions. It enumerates stepwise possible candidate solutions while exploring the search space to find the optimal solution.

Anticipatory methods

Although anticipatory methods are an independent type of method that does not fit in the ride-matching scheme of Figure 2, they provide valuable insights regarding the topic of unreliability. Therefore they are discussed in this section. Anticipatory methods are useful to explore as they try to acquire future demand data to deal with spatial demand imbalances. The solution this research will implement to deal with unreliability also requires knowledge about future demand.

A challenge that arises with the introduction of ridesharing services is the emergence of spatial imbalances in the demand-supply. Spatial demand-supply imbalances induce a mismatch between the origins of new requests and the position of the vehicle fleet. Therefore, anticipatory methods that operate at an earlier stage during the assignment and routing process can be implemented to prevent the occurrence of demand-supply imbalances. Different techniques exist that usually require knowledge about historical data or future demands to create efficient matches. An example of this method is splitting groups of passengers with a destination in a high-demand area into several different vehicles such that a greater amount of vehicles is transported to high-demand areas (Barth et al., 2004). Alonso-Mora et al. (2017b) insert artificial requests to a batch of to-be-served requests based on estimates of historical demand for different zones to prevent spatial demand imbalances. Fielbaum et al. (2021b) propose two anticipatory methods that influence how vehicles are assigned to users and in which order the requests are served to reach a more efficient state for future requests. These two proposed methods rely on currently received requests as future demand highly correlates with the current demand. Their first method modifies the objective function by introducing a reward for each vehicle whose final destination is in a high-demand area. Their second method adds artificially created future requests to the list of considered requests during the assignment process. The artificial requests are created by sampling from a probability distribution that mimics the observed request rate. As a result, the fleet moves towards high-demanding zones.

2.3.3 Objectives

The algorithm objective also has a strong influence on service unreliability. Different stakeholders can be identified for the on-demand ridesharing process, such as users, transportation network companies (TNCs), drivers, and the urban area of the ridesharing service. Each stakeholder has its objectives regarding the vehicle-request matching process. The stakeholders' objectives affect the matching process, and the stakeholder objectives may contradict other stakeholder objectives (Chakraborty et al., 2020). For example, to increase profit, the operators may decide to increase the amount users per shared trip. As a result, the operators' profit increases, but at the same time, users have to face more and larger detours, meaning an increase in unreliability.

- **User objectives** - The users' objective can be to minimize travel time, to have the lowest travel unreliability, to have the lowest travel cost in terms of money, or a combination of the three.

- **TNC objectives** - The main objective of the TNCs is to maximize their profit. Several methods can be applied to increase profit. This can be achieved by increasing the fleet utilization rate, i.e., increasing the number of matches by assigning more trips. Another method to increase profit is to minimize the travel cost or distance traveled by the fleet. As stated before, the objective of the TNC has a strong influence on the unreliability of the service.

- **Driver objectives** - The primary objective of the drivers is to maximize their earnings. Drivers may get incentives for each fulfilled trip and thus aim to complete as many trips as possible. This can be achieved by reaching each destination as soon as possible and by trying to minimize the users' drop-off time and boarding time. However, the drivers' earnings are fixed or the vehicles are automated in the centralized system that this research is concerned with. Therefore, the driver objectives are not relevant to this research.

- **City objectives** - The objective of the entire city is to minimize the negative impact of the TNC service on the city. This can for example be achieved by minimizing utilized city resources such as parking spots and traffic space. Vehicle Miles Traveled (VMT) is an indicator of road space that TNC services consume. Thus, efficient resource planning should be implemented.

Overall the different objectives for the ride-matching process can be classified into three groups, i.e., travel time-based, travel cost-based, and travel distance-based objectives (Z. Liu et al., 2019).

- **Travel time-based**
  An algorithm for computing optimal vehicle-trip combinations by minimizing the sum of the delays and penalizing ignored requests is provided by Alonso-Mora et al. (2017a). The concept of the assignment matching process consists of two main parts, i.e., determining the feasible trips and vehicles that can serve the requests, and solving an

ILP to compute the best assignment of the vehicles to trips. Additionally, Alonso-Mora et al. implement a rebalancing strategy considering idle vehicles to prevent fleet imbalances. From the experimental results, the authors quantify a trade-off between fleet size, capacity, waiting time, travel delay, and operational costs for different fleet capacities. The authors illustrate in their case study of Manhattan that 98% of the taxi rides, consisting of a fleet of 13.000 taxis, could be served by 3.000 ridesharing vehicles with a maximum capacity of four passengers.

Other studies that implemented an objective to minimize travel time are provided for example by Jung et al. (2016); M. Liu et al. (2015); Herbawi and Weber (2012).

- **Travel cost-based**
  The objective to maximize profit or minimize cost for on-demand ridesharing operators has been investigated by Jung et al. (2013). The authors implement this objective into three different algorithms. Their simulations indicate that the objective yielded higher profits for the ridesharing operators with higher demand for all three algorithms. Examples of other studies with a similar objective are Nanda et al. (2020); L. Zheng et al. (2018); Jung et al. (2016).

- **Travel distance-basted**
  Minimizing travel distance is an important objective from a social point of view since it reduces traffic congestion and pollution. Furthermore, minimizing travel distance is closely related to minimizing travel costs, as a reduction in travel distance decreases fuel consumption and vehicle wear. Luo et al. (2019) propose a ridesharing approach that minimizes the total travel distance while maximizing the service rate. Their large-scale experimental study indicates that the proposed algorithm has an improved efficiency and effectiveness compared to a state-of-the-art dynamic ridesharing method. Other studies that implement a distance-based approach are provided by Linares et al. (2016); Armant and Brown (2014); Lokhandwala and Cai (2018).

It can be concluded that several objectives exist which can be implemented by the ridesharing service. The different objectives strongly influence the unreliability of the service. However, it is unknown how the different objectives relate to each other regarding unreliability as no research exists that compares this.

2.3.4 Constraints

Several conditions are considered to determine the feasibility of a match between a vehicle and a request. These conditions are called constraints. These constraints are examined during the assignment procedure when the objective function is being optimized. Unfortunately, no constraints have been discovered that explicitly deal with the problem of unreliability. The three main constraints of a ridesharing system are listed below.

- **Vehicle capacity** defines the maximum amount of passengers per trip. This constraint is used to prevent the creation of shared tips that exceed the vehicle's capacity.

- **Maximum waiting time** refers to the maximum time a user has to wait before being picked-up by a vehicle. The waiting time starts counting directly after a request has been received. This constraint guarantees that each user has a minimum quality of service and prevents the users from canceling their trip.

- **Maximum delay** prevents the system from creating shared trips with unreasonable long detours which are unacceptable to the users. The delay is equal to the difference between the drop-off and the earliest possible time at which the destination can be reached if the vehicle travels directly with the shortest path from the origin to the destination of the user.

2.3.5 Conclusion ride-matching techniques

This section has discussed the different algorithm designs for the matching process between vehicles and requests. An overview of the ride-matching design structure is provided in Figure 2. This research is interested in dynamic centralized systems. Table 1 provides an overview of the advantages and disadvantages of the discussed dynamic centralized design methods. In conclusion, a trade-off exists between solution quality and computation time. Furthermore, the design structure strongly influences the unreliability of the service. Even without altering the algorithm design or parameters, different trip realizations can be obtained for the same request, resulting in different unreliability rates (Fielbaum & Alonso-Mora, 2020). Additionally, no research has been identified that examines the different algorithm structures regarding the unreliability rates. Therefore, no specific design structure can be identified as the optimal method to deal with unreliability.

Furthermore, different stakeholders have been identified, all with their own objectives regarding the SMoD service. Overall, three types of objectives exist: travel time-based, travel cost-based, and travel distance-based. An objective to explicitly minimize unreliability has not been discovered. The three main constraints in SMoD services are vehicle capacity constraints, maximum waiting time constraints, and maximum delay constraints. Unfortunately, no constraints have been discovered that explicitly focus on the control of unreliability.

Table 1: Advantages and disadvantages of dynamic centralized algorithm designs.

	Method	Advantage	Disadvantage
Algorithm initiation	Immediate insertion	• Computationally less expensive • Directly handle requests	• Not able to keep up at high demand • Lower solution quality
	Receding horizon	• Higher solution quality • Able to keep up with high demand	• Computationally more expensive • Increasing waiting time
Problem distribution	Decomposition	• Computationally less expensive	• Lower solution quality
	No decomposition	• Higher solution quality	• Computationally more expensive
Solving mechanism	Exact	• Higher solution quality	• Computationally more expensive
	Heuristics	• Computationally less expensive	• Lower solution quality

2.4 Techniques to minimize unreliability in ridesharing

The main objective of this chapter is to obtain insight into the available techniques that aim to minimize unreliability in ridesharing, as it provides insight in to how the assignment procedure can be efficiently controlled to reduce unreliability. Therefore, the existing techniques to (potentially) reduce unreliability are discussed in this section. The discussed techniques are: not allowing for any changes in the schedule, Section 2.4.1; only adding users if the drop-off constraints are satisfied, Section 2.4.2; objective functions which consider delays, Section 2.4.3, constraints that consider unreliability/delays, Section 2.4.4; increasing the demand-supply, Section 2.4.5; meeting points for pick-up and drop-off, Section 2.4.6; minimizing the number of stops, Section 2.4.7; and predictive methods, Section 2.4.8. Finally, a conclusion regarding the techniques to minimize unreliability is provided in Section 2.4.9.

2.4.1 Not allowing for changes in schedule

The user's discomfort due to unexpected changes in trip routes and trip companions affects the service unreliability. To reduce this unreliability, Y. Wang et al. (2018) propose a ridesharing framework that relies on a simple scheduling protocol. The scheduling protocol creates a complete trip plan containing a set of users in a predefined order before assigning the trip plan to a vehicle. Once a trip plan is created and assigned to a vehicle, no changes are allowed in the route or schedule during the trip. Consequently, the ridesharing strategy relieves the users from unreliability, i.e., inconvenience in changes of travel routes. However, according to the data study results of Y. Wang et al., this strategy results in a slightly longer waiting time compared to non-shared taxi trips. Unfortunately, the study of Y. Wang et al. does not compare the rejection rate, which will probably be significantly high. Additionally, the authors execute no performance comparison of their ridesharing framework to a state-of-the-art ridesharing framework. But, we can assume that the waiting times and rejection rate are significantly worse compared to ridesharing frameworks.

2.4.2 Only add users if drop-off constraints are satisfied

Closely related to the previously mentioned method of only adding users to a trip after trip completion is the proposed solution by Hörl and Zwick (2021). They propose a solution that does not allow for adjustments in a trip if the vehicle cannot satisfy the onboard users' drop-off times. Consequently, a vehicle is not allowed to engage in picking up new users until all users of the current trip are dropped off or until all drop-off times of the onboard users can be fulfilled again.

Similarly to this solution, Hörl and Zwick propose a less restricted solution by only adding new requests to a trip if the maximum value between the *updated* drop-off time and the *original* drop-off time is not violated. The currently assigned routes of all vehicles are examined during each matching process to calculate updated drop-off times for the users to efficiently add new requests to a trip. A new request is added to a trip if the maximum value between the updated drop-off time and the original drop-off time is not exceeded. However, a penalty is added for violating the drop-off constraints to favor the originally predicted travel times over the updated travel times. The same concept can analogously be applied to assigned users who are currently waiting at their pick-up point: additional users are only added to a trip if the (updated) pick-up times are not violated. However, a reassignment is applied if another vehicle can pick-up the waiting user at the original pick-up time instead of the updated pick-up time.

Hörl and Zwick present that their method leads to a trade-off between violating user constraints and rejecting requests after prior acceptance. If the travel times are underestimated, either assigned requests have to be rejected, or violations must be allowed in the provided pick-up and drop-off times. Therefore, the enforced constraints on on-time arrival result in increased rejection rates.

2.4.3 Objective function

The variations in travel time faced by the user can be considered as a disturbance to the user's travel time. Prorok et al. (2021) state that pre-operative approaches solve problems where a disturbance affects the system's performance. Pre-operative approaches determine the key variables that induce disturbances (in this case unreliability) in advance and provide resilience by design. For this research, this entails that the objective function could be optimized with respect to a decision variable related to the key variables of unreliability. Therefore, one of the approaches to minimize unreliability in ridesharing is to adopt an objective function that minimizes unreliability. Since no research exists on objective functions that consider unreliability, the objective functions that consider delays are explored.

Minimizing delays is closely related to minimizing unreliability. If a system aims to minimize delays, it is less likely that a user will face an increase in delay time and thus users have less travel unreliability. However, when a user faces a decrease in delay time, there is still variability in travel time, meaning unreliability. Besides, a user can still face fluctuations in travel time when delays are minimized. Furthermore, when the system aims to minimize the total delay time of all users, some users will face an increase in travel time to reduce the overall delay time of all users. Therefore, minimizing delays and minimizing unreliability are closely related, but they are not the same. Because no research exists on objective functions that consider unreliability, the next paragraphs discuss some of the existing methods that use an objective function that minimizes delays.

A study that implemented an objective that minimizes delays is provided by W. Zheng et al. (2021). They developed an on-demand ridesharing system that uses a heuristic approach to match vehicles to requests. They implemented two conflicting objectives which are minimizing overall cost and minimizing delay time, i.e., time spent waiting for a vehicle and vehicle detours. Case study results indicate a drastic reduction in delay time with the implantation of this ride-matching approach.

Manna and Prestwich (2014) modeled a ridesharing optimization planning approach that aims to maximize the shared trips while minimizing the expected travel delay for each trip by penalizing each travel delay in the objective function. The authors implement this approach to tackle the problem of reducing the number of shared trips when the delay time is optimized. Case study results illustrate that the resulting delay per user is, on average, negligible.

Since it is crucial to design optimal pick-up and drop-off orders to minimize total pick-up and drop-off delays in ridesharing, X. Chen et al. (2017) used an objective that minimizes the total mileage delay while serving as many requests as possible. Also, they discover a trade-off between the number of served passengers and the mileage delay. Therefore, an increase in served requests implies an increased mileage delay.

Y. Li and Chung (2020) proposed a robust optimization approach that considers uncertain travel time explicitly when matches and route decisions are made. However, in this study, uncertain travel time implies uncertainty due to traffic conditions, and not uncertainty due to schedule changes. In this study, a control parameter is implemented that simulates uncertain travel time in the road network. The authors state that it is requisite to consider time uncertainty as it significantly impacts the flexibility of time windows and detours, which are closely related to ride-matching. Therefore, their goal is to provide a robust route for each user under travel time uncertainty and to increase the matching rate of the requests. This is achieved by using an objective function that minimizes the travel cost and penalizes rejected requests while considering the maximum possible delays. By considering the maximum possible delays, the arcs of the road network that contain simulated uncertain travel times are avoided.

Optimizing the maximum expected mean probability of dropping off requests on-time by considering stochastic travel time information is another objective function that can be implemented. This method is proposed by C. Li et al. (2020), who study the role of travel time uncertainty in on-demand ride-sharing services. They consider stochastic travel time information to increase passengers' on-time arrival reliabilities. Furthermore, C. Li et al. incorporate the concept of late arrival compensation to increase profit.

It can be concluded that several studies have implemented an objective function that minimizes delays. However, they all discovered that minimizing delays is at the expense of other quality-of-service indices, such as rejection rate and average travel time. As stated in Section 2.1.2, the same holds for an objective function that penalizes unreliability. Hence, employing an objective function to minimize delays and unreliability is not the optimal solution.

2.4.4 Constraints

The objective function of the ridesharing model is subjected to constraints to determine the feasibility of created matches. Constraints greatly impact the matching procedure. Besides constraints that for example account for vehicle capacity, constraints that consider unreliability can be implemented. This section discuses constraints that have been implemented by other studies. The discussed constraints might not reduce unreliability, but they contain some interesting characteristics. Therefore, they could possibly be converted to constraints which might reduce unreliability.

Threshold for amount of changes

Several studies implement constraints to bound detours such that the detour times do not exceed a set threshold value (e.g., Daganzo et al., 2020). This concept can be extended by allowing users to define their own detour thresholds (e.g., Xia et al., 2019; Bian et al., 2020). The concept of bounding detours based on user preferences could be further exploited to reduce unreliability by allowing users to define their threshold for the number of changes per trip. Not allowing for any changes during a trip obstructs the efficient development of vehicle-request combinations, reducing the performance of the provided service. However, allowing a few changes per user based on user preferences might reduces the performance slightly while, on the other hand, slightly reducing unreliability.

Threshold for delays

As stated before, minimizing delays is closely related to unreliability. However, a decrease in delay time does not directly result in a decrease in unreliability. Delays can be restricted by assigning an upper bound value to the user's maximum delay time. Such a constraint is applied for example by C. Li et al. (2020) where each request's waiting time and travel delay are restricted to be lower than a set threshold value. Other studies that use such a constraint to minimize delays imposed on the users are provided by Gurumurthy and Kockelman (2018); Y. Wang et al. (2018); Ota et al. (2015).

Even though such constraints minimize the delay time of the users, they might not be sufficient enough to minimize unreliability since fluctuations in travel time are still possible within a single trip.

2.4.5 Increase demand-supply

The fleet size influences the unreliability of the service. Small fleets provide a higher possibility to match different users to the same trip, increasing the waiting time and detours. This is simply due to a fewer amount of available vehicles that can serve the requests, causing a reduction in the number of idle vehicles. Consequently, when a new request is received, only vehicles carrying passengers might be in proximity to serve the request, increasing the occupancy rate. On the contrary, large fleets provide the opposite situation, meaning a decrease in the number of shared users per trip (Beojone & Geroliminis, 2021). Therefore, to decrease unreliability, operators could increase the demand-supply to reduce the number of shared users per trip, leading to fewer variations in travel time.

The demand-supply can be increased by increasing the fleet size or by utilizing anticipatory methods. The concept of anticipatory methods is discussed in Section 2.3.1.

A disadvantage of increasing the demand-supply is that this is at the expense of the service's core principle of ridesharing: the shareability rate. Moreover, the operators' costs will increase by employing larger fleet sizes. Additionally, when the demand-supply is increased and all vehicles still operate at maximum capacity, the unreliability is not reduced. On the contrary, if the demand-supply is increased, it might also be possible that the service shifts to a non-shared service.

2.4.6 Meeting points for pick-up and drop-off

The introduction of meeting points for the pick-up and drop-off of the users decreases the detour time of other users (Fielbaum et al., 2021a). In such a system, users are requested to walk a short distance towards/from a pick-up/drop-off location. The increased flexibility of the users facilitates an increase in the number of possible matches and improves service efficiency (Stiglic et al., 2015).

The implementation of meeting points for the users' pick-up and drop-off might also affect the service's unreliability, as it might result in a reduced number of trip changes. This occurs for example when two different requests with different destinations are dropped-off at the same location, whereafter they walk a short distance to their final destination. As a result, the number of detours and the detour time are reduced. Nevertheless, the reduction of unreliability might only be minor, since the number of pick-ups and drop-offs that can be merged into one stop is minor.

2.4.7 Minimizing the number of stops for pick-up and drop-off

Reducing the number of performed stops for the pick-up and drop-off of the users can result in an unreliability reduction. If the system can pick-up/drop-off an additional user at the same origin/destination of another user, the unreliability imposed on the already assigned users is reduced.

Pimenta et al. (2017) analyzed the management of a Dial-a-Ride problem consisting of a closed industrial site to minimize the number of loading/unloading operations.A static problem is considered, i.e., all requests are known in advance before executing the optimization problem. The industrial site consists out of the main track, i.e., a circular circuit, and loading/unloading areas. Overtaking is forbidden on the main track and all vehicles must travel counterclockwise. The authors state that the steady flow of the vehicles correlates with the number of performed loading/unloading operations. Therefore, to minimize the number of stops, the vehicles are forced to stay as much as possible on the main track without deviating to the loading/unloading areas. Experimental results illustrate that the number of stops and the waiting time of the users are not convergent criteria. Minimizing the number of stops increases the number of driven laps while minimizing the waiting time results in an increase in the number of vehicles.

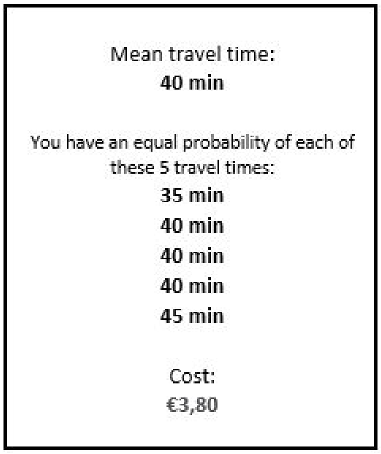
In conclusion, minimizing the number of stops for the pick-up and drop-off of the users is a promising concept for reducing unreliability. However, an efficient implementation still has to be discovered.

2.4.8 Predictive methods

By incorporating demand-side estimates in a simulation-based optimization, Bansal et al. (2020) discovered the optimal waiting time that mobility-on-demand services should display to attract more users. This was achieved by computing the travel time based on historical data and assuming that the travel time between every two locations follows a lognormal distribution. Their numerical experiments illustrated that mobility-on-demand services can increase their acceptance rate by around 10% by displaying appropriate wait time information.

The concept of providing improved travel time information can be implemented in SMoD services to provide improved reliable trip information such that the unreliability of the service is reduced. This can be achieved for example by analyzing historical demand data, the system's rules, and the current state of the fleet and the state of the traffic. Using this information, the SMoD service can provide estimated data on the likelihood that a user will face additional detours. Additionally, an estimate of the magnitude of these possible detours could be provided to reduce unreliability.

To execute stated preference experiments in SMoD systems, Alonso-González et al. (2020) analyzed different methods to present travel times that include unreliability in their predictions. These presentations can be in bar diagrams, histograms, circular arrangements, likelihood percentages of travel times, a verbal description of equally plausible travel times, and time variability with a single component, e.g., ± X minutes. By executing a stated preference study about the perception of travel time unreliability, Tseng et al. (2009) recommend adopting a verbal description to present travel time unreliability. Results indicated that participants with different educational levels emphasized that a verbal description is the most intuitive and easy-to-understand method. Figure 3 illustrates such a verbal representation. Such a representation could be adopted to provide estimated data on the likelihood and magnitude that a user will face additional detours such that the unreliability of the service is reduced.



Mean travel time:
40 min

You have an equal probability of each of
these 5 travel times:
35 min
40 min
40 min
40 min
45 min

Cost:
€3,80

Figure 3: A verbal presentation of travel time unreliability (Tseng et al., 2009).

Altshuler et al. (2019) proposed a dynamic travel network approach to predict the ridesharing utilization rate ahead of time. The authors demonstrate that the implemented approach can model the potential utilization rate of the ridesharing service based on the topological network properties. The utilization prediction is based on the construction of multiple network snapshots. These network snapshots are derived using a sliding-window-based aggregation of the ridesharing rides. Next, a correlation between the change in potential ridesharing utilization and the mean degree of the transportation network is created to obtain future utilization rates. The authors demonstrate that this method can forecast the potential utilization rate a few hours ahead.

In conclusion, demand predictions are an interesting concept as they can be used to reduce unreliability by providing improved travel time information. If the system can tell in advance that a user will face a detour with a specific duration, the user no longer faces unreliability as the detour is known in advance. Several studies exist that aim to predict demand. Therefore, it is interesting to analyze demand prediction methods. Additionally, anticipatory methods, discussed in Section 2.3.1, are also interesting to analyze, as they make predictions of the ridesharing utilization rate as well (e.g., X. Li et al., 2020; Al-Abbasi et al., 2019; Alonso-Mora et al., 2017b). A detailed analysis of demand prediction models is therefore provided in Section 2.5.

Table 2: Overview of the methods to minimize unreliability.

Method	Advantages	Disadvantages	Reference
Not allowing for changes in schedule 2.4.1	- No unreliability induced by operator rules - Simple method	- Long waiting times for the users - Increase in rejection rate - No changes possible after assignment	(Y. Wang et al., 2018)
Add users if time constraints are satisfied 2.4.2	- No unreliability induced by operator rules - Simple method	- Long wait times for the users - Increase in rejection rate - Violate user constraints vs reject requests	(Hörl & Zwick, 2021)
Objective function: penalize changes in trip duration 2.4.3	- Degree of unreliability decreased based on magnitude of penalty value	- Efficiency of the service might be reduced - Might result in increased rejection rate	N/A
Objective function: penalize delays 2.4.3	- Reduction in delay time	- Efficiency of the service might be reduced - Not clear if unreliability is reduced - Might result in increased rejection rate	(W. Zheng et al., 2021) (Manna & Prestwich, 2014) (X. Chen et al., 2017) (Y. Li & Chung, 2020) (C. Li et al., 2020)
Constraint: threshold for number of changes 2.4.4	- Unreliability rate can exactly be controlled - Simple method	- Efficiency of the service is reduced - Might result in increased rejection rate	N/A
Constraint: threshold for delays 2.4.4	- Reduction in delay time - Simple method	- Efficiency of the service might be reduced - Not clear if unreliability is reduced - Might result in increased rejection rate	(C. Li et al., 2020) (Gurumurthy & Kockelman, 2018) (Y. Wang et al., 2018) (Ota et al., 2015)
Increase demand-supply 2.4.5	- N/A	- Transforms system into non-shared service - Decrease in occupancy rates - Might result in profit decrease	N/A
Meeting points for pick-up and drop-off 2.4.6	- Increase efficiency of service - Decrease in delays	- Not clear if unreliability is reduced	(Fielbaum et al., 2021a) (Stiglic et al., 2015)
Minimizing the number of stops 2.4.7	- High potential to reduce unreliability	- No suitable practical solution exists	(Pimenta et al., 2017)
Predictive methods to provide improved travel time information 2.4.8	- No unreliability if the method works perfectly	- Hard to implement - Impossible to predict every single request	N/A

2.4.9 Discussion of the techniques to minimize unreliability

Several techniques have been discovered to (potentially) minimize unreliability in SMoD systems. An overview of the discovered techniques is provided in Table 2. Additionally, this table provides the advantages and disadvantages of the techniques. It can be concluded that no suitable technique exists that effectively reduces unreliability without strongly degrading other quality-of-service indices. Furthermore, it has been concluded that a trade-off exists between unreliability and the other quality-of-service indices such as rejection rate and shareability rate. The following paragraphs provide a conclusive analysis of the different methods of Table 2.

In the literature, two methods are discovered that are able to minimize unreliability. These two methods are: not allowing for changes in the schedule and only adding users if the drop-off times are satisfied, Sections 2.4.1 and 2.4.2, respectively. Nevertheless, these methods strongly increase the rejection rate and waiting time while degrading the shareability rate, i.e., the number of users per shared trip. Therefore, these methods are not suitable to implement in real-life applications. The same disadvantages occur for the conceptualized constraint method of setting a threshold on the number of trip changes a user can face (Section 2.4.4).

Closely related to minimizing unreliability is the concept of minimizing delays. Several methods are discovered to minimize delays: an objective function that penalizes delays, a constraint that restricts the allowable delay time of the users, and meeting points for the pick-up and drop-off of the users, Sections 2.4.3, 2.4.4, and 2.4.6, respectively. By reducing delays, the likelihood decreases that the trip duration increases, resulting in a reduction in unreliability. However, the system can still insert new users into a scheduled trip and apply changes to the trip which induces unreliability. Therefore, these methods are not able to effectively reduce unreliability.

Increasing the demand-supply, Section 2.4.5, will result in a reduced number of shared trips. Therefore, unreliability reduces as fewer users are added to a trip. On the contrary, this method is unsuitable since it transforms the ridesharing service into a non-shared service.

The remaining three methods are the most promising methods to minimize unreliability efficiently. However, these three methods have not been examined in the literature since they are conceptualized methods inspired by the already discussed methods. The first method is implementing a penalty factor in the objective function that penalizes variations in travel time during the assignment process (Section 2.4.3). This method can control the degree of unreliability by altering the magnitude of the penalty, making it a promising method to reduce unreliability. However, as the magnitude of the penalty value increases, the number of users per shared trip may decrease since an increase in users per shared trip results in an increase in unreliability. To counter this behavior, an additional parameter should be implemented in the objective function that values the efficiency of the service.

The following interesting conceptualized method is to minimize the number of stops for picking-up and dropping-off users during a trip (Section 2.4.7). In the ultimate case, this means that all passengers can be picked-up and dropped-off at a similar origin and destination. Nevertheless, such a method is hard to realize.

The last promising method is a predictive method to provide improved travel time information (Section 2.4.8). If the method works perfectly by forecasting all future demands, the system can provide the exact travel time information that includes yet unscheduled future detours, meaning that there is no unreliability at all. However, forecasting every single request appears to be impossible.

The conceptualized method of predicting trip changes to reduce unreliability has the least drawbacks in combination with its high potential to reduce unreliability. Therefore, this method will be implemented in this research. Hence, Section 2.5 provides a detailed analysis about demand prediction models. The concept of using an objective function that penalizes unreliability also has a high potential to reduce unreliability. However, this concept will strongly reduce the efficiency of the service, and therefore, it is not selected. The last conceptualized method of minimizing the number of stops to pick-up and drop-off users during a trip also has a high potential to reduce unreliability, but how this can be achieved is unknown.

2.5 Demand prediction

The method that will be implemented to reduce unreliability is based on the possibility that new users will be added to the trip. Therefore, it is crucial to make predictions about the possibility that new requests will appear. Hence, the theoretical background regarding demand prediction models is provided in this section, starting with an introduction in Section 2.5.1. Next, the mobility characteristics that influence the demand are explored in Section 2.5.2. After that, the mobility prediction dependencies are discussed in Section 2.5.3. Since most demand prediction methods use spatial clustering methods, different spatial clustering methods are discussed in Section 2.5.4. Finally, Section 2.5.5 elaborates on forecasting implementations and mathematical forecasting models.

2.5.1 Introduction

Urban infrastructures become increasingly reliant on network technologies as they enable the expansion and innovation of urban networks. The increasing availability of portable technologies stimulates the optimization of the urban transportation sector, pushing urban designs closer to the concept of "smart cities" (Caragliu et al., 2011). Due to digitalization, mobility demand behavior can extensively be analyzed as most human transport data can be traced digitally. These analyses deliver the opportunity to uncover human travel patterns (Froehlich et al., 2009). Several studies employ mobile phone data to analyze mobility demand patterns (Ahas et al., 2007; Gonzalez et al., 2008; Calabrese et al., 2013). In a large-scale data analysis, Gonzalez et al. (2008) illustrate that mobility patterns are far from random and that a single spatial probability distribution can efficiently describe these travel patterns. However, according to Gardner et al. (2013), travel demand is still inevitably stochastic in its nature. Since there exists some randomness in the travel demand, (Zhao et al., 2016) propose a taxi demand prediction approach while including a randomness factor in the taxi demand. This is achieved by governing the taxi demand by a randomness factor, e.g., unexpected events, and simultaneously adding a degree of regularity, e.g., weekly patterns. Although individual travel demand has the characteristics of personalization and randomness, exploring the spatio-temporal characteristics of the ridesharing demand facilitates to an improved understanding of the demand distribution and provides a basis for travel demand predictions (X. Li et al., 2020). The accurate prediction of the ridesharing demand entails several benefits:

- For the transpiration companies, the results of the demand predictions can be utilized to establish a proactive vehicle scheduling strategy that navigates vehicles to high-demanding zones to improve the fleet's utilization rate (Ke et al., 2017). Such a proactive method makes reactive rebalancing methods unnecessary and obsolete.

- For city administrators, information about demand predictions can be employed to regulate the urban traffic flow and to manage the traffic demand and supply, to contribute to more sustainable and healthy urban traffic (X. Li et al., 2020).

- For the users of the SMoD service, the unreliability of the service may be alleviated by obtaining improved travel time information based on demand predictions. In the ultimate scenario, the ridesharing system can predict every single request, such that the system can determine the exact travel times of each new user. As a result, users face no unreliability as the initially provided travel times portray the real travel times. However, in reality it is difficult, if not impossible, to precisely predict the demand. Still, the concept of demand predictions can be leveraged to reduce unreliability.

2.5.2 Mobility characteristics

By employing a predictive demand strategy, the unreliability of the service might be reduced. Therefore, a predictive strategy will be investigated to improve the unreliability of the SMoD service. The mobility characteristics of taxi-services and SMoD services are explored to obtain insights into the demand prediction process.

To obtain a comprehensive understanding of the taxi demand in New York City, Altshuler et al. (2019) demonstrate the distribution of taxi trips per day of the week and per hour of the day, Figure 4. It can be observed that the number of taxi-trips exhibits the very least a uniform time distribution. More specific, the number of trips is greater in the middle of the week compared to the other weekdays, including the weekend. Furthermore, a daily pattern exists with a peak in the morning and a peak in the afternoon around 7 PM.

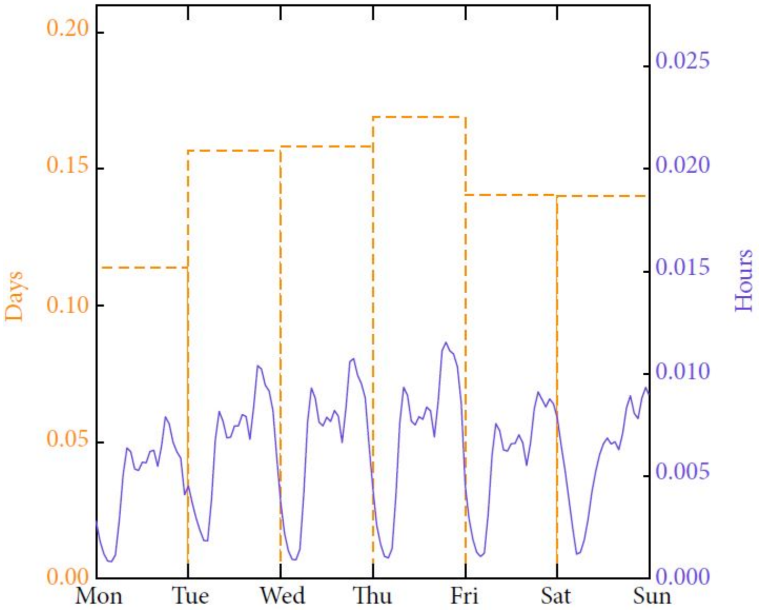
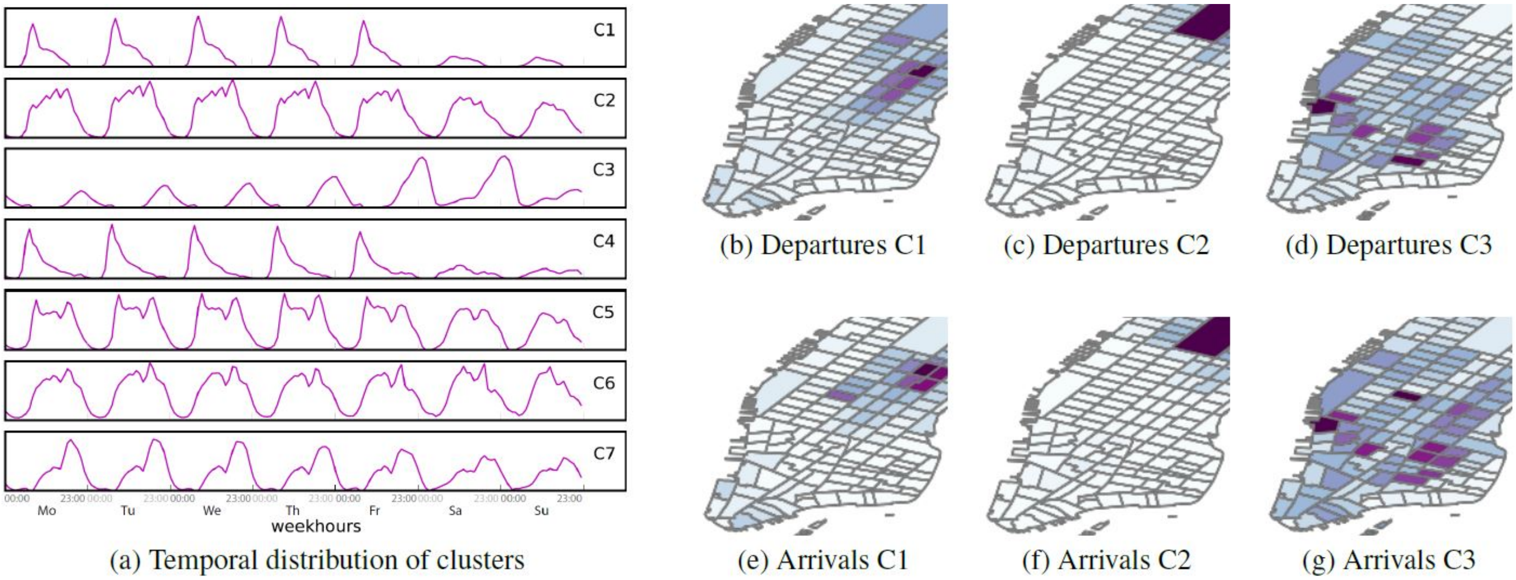


Figure 4: Probability Density Function (PDF) of the number of taxi-trips per day of the week/hour of the day in New York City, January 2013. Afternoon peaks are centered on average around 7 pm. Figure obtained from (Altshuler et al., 2019).

By analyzing the distribution of the number of taxi-trips between different squared regions of New York City, Altshuler et al. (2019) indicate that the taxi demand follows a power law distribution (Gonzalez et al., 2008), implying that most trips depart from the same zones and travel to the same zones.

Espín Noboa et al. (2016) characterize the taxi travel patterns of Manhattan by clustering the demand behavior based on spatio-temporal dimensions. The authors demonstrate that taxi demand is not one-dimensional but rather contains different facets of both time and space, urging that taxi demand cannot be identified in a collective analysis. For example, it has been discovered that on weekend nights, taxi rides are more likely to depart to locations with a high density of party venues. Figure 5 illustrates a selection of these discovered spatio-temporal mobility patterns of Manhattan.



(a) Temporal distribution of clusters

(b) Departures C1

(c) Departures C2

(d) Departures C3

(e) Arrivals C1

(f) Arrivals C2

(g) Arrivals C3

Figure 5: Spatio-Temporal travel behavior. (a) Each row represents a behavioral component with respect to the pick-up time (hour of the week). The maps on the right side show the areas of low and midtown Manhattan. The colors of each tact present how representative that location is in that cluster; the darker, the more dominant. (b,e) cluster C1 stands for taxi rides at 9 AM around the south-east of Manhattan; (c,f) Cluster C2 contains all evening short taxi rides at 6 PM around the center of Manhattan; and (d,g) Cluster C3 represents all weekend night taxi rides around the lower and middle part of Manhattan. Figure obtained from (Espín Noboa et al., 2016)

Foell et al. (2014) investigated personal mobility patterns, global mobility patterns, and geographic distance metrics to predict the mobility patterns of public transport users. Here, the personal mobility patterns, i.e., the trip history of a user, are leveraged to predict the destination of the user. It is likely that destinations that have been relevant in the past will also be equally important during future trips (Foell et al., 2014). The characteristics of mobility demand behavior can be divided into three aspects. These are global mobility, geographic mobility, and network mobility.

- **Global mobility -** Superimposing personal travel patterns give rise to the emergence of global travel patterns. Global travel patterns contain essential information about which destinations are most likely to be selected by the users. This results in high-demand zones and popular destination hot spots as these destinations are particularly attractive to the transport users. These high-demand zones result in high levels of transport activity in specific areas. High-demand zones arise due to a manifold of influences, such as urban centers of social activities (Foell et al., 2014).

- **Geographic mobility -** The probability of a user traveling to a destination decreases proportionally to the travel distance of the concerning trip (Gonzalez et al., 2008). Given this knowledge, the degree to which a specific zone is relevant to a user departing from a given zone can be estimated. A study that uses such an implementation is provided for example by (Foell et al., 2014).

- **Network mobility -** Besides the available mobility-on-demand services, public transport systems also participate in the network mobility infrastructure. Public transport systems influence the likelihood to which zone a request travels to. If zone a and zone b are well connected with public transport, it is less likely that a user departing from zone a will take a taxi to zone b as more convenient alternatives are available for this trip (F. Wang & Ross, 2019).

  The topology of a route network may significantly differ from the geographical relations in space. Although some zones are geographically close, the travel connections might not always be convenient or efficient. Therefore, a metric can be implemented to identify preferred travel destinations that considers the network mobility of a city (Foell et al., 2014).

2.5.3 Mobility dependencies

According to J. Zhang et al. (2017), the forecasting of the taxi demand flow in each city region is affected by three factors: spatial dependencies, temporal dependencies, and external influences. The selection of reasonable factors is the basis for an accurate demand prediction; therefore, (X. Li et al., 2020) mainly consider spatio-temporal, temporal, and spatial variables. Below, an elaboration of the different mobility dependencies is provided.

- **Time dependencies -** Passenger demand has a strong periodicity (Altshuler et al., 2019; Ke et al., 2017). This periodicity can for example be explained by the peak demands during the morning due to commute travel, and low demands during the night time. Therefore, the time dependencies include the day of the week and the time of the day.

- **Spatial dependencies -** It is well known that the characteristics of a specific zone, e.g., cultural occasions, work-offices, and residential areas, influence the mobility demand rate of that specific zone. However, the mobility demand of a specific zone does not solely depend on the characteristics of that zone but on the characteristics of all zones in the whole network (Yang et al., 2010). Nevertheless, the characteristics of nearby zones have a more powerful influence on the demand than the more distant zones (Ke et al., 2017). By analyzing the mobility of different cities, Noulas et al. (2012) demonstrate that mobility highly correlates with the distribution of the points of interest in a city.

  In an analysis of the ridesharing demand data of Haikou, China, X. Li et al. (2020) illustrate that there is a high spatial correlation between the demand in a certain zone and its neighboring zones, and the closer distance to the neighboring zone, the greater and more significant the correlation is.

- **Spatio-temporal dependencies -** The combination of time and space dependencies strongly influence the demand rate. Spatio-temporal dependencies include the historical demand rate of the concerning zone and its neighboring zones and the travel time rate, i.e., the travel time per unit travel distance. Historical demand is an important factor in demand forecasting (X. Li et al., 2020).

- **External dependencies -** Different external factors, such as weather conditions, events that take place, traffic events, and recent transport-related changes, e.g., new transit lines and urban projects, strongly influence the short-term mobility demand (Ke et al., 2017; J. Zhang et al., 2017; Fielbaum et al., 2021b).

2.5.4 Spatial clustering

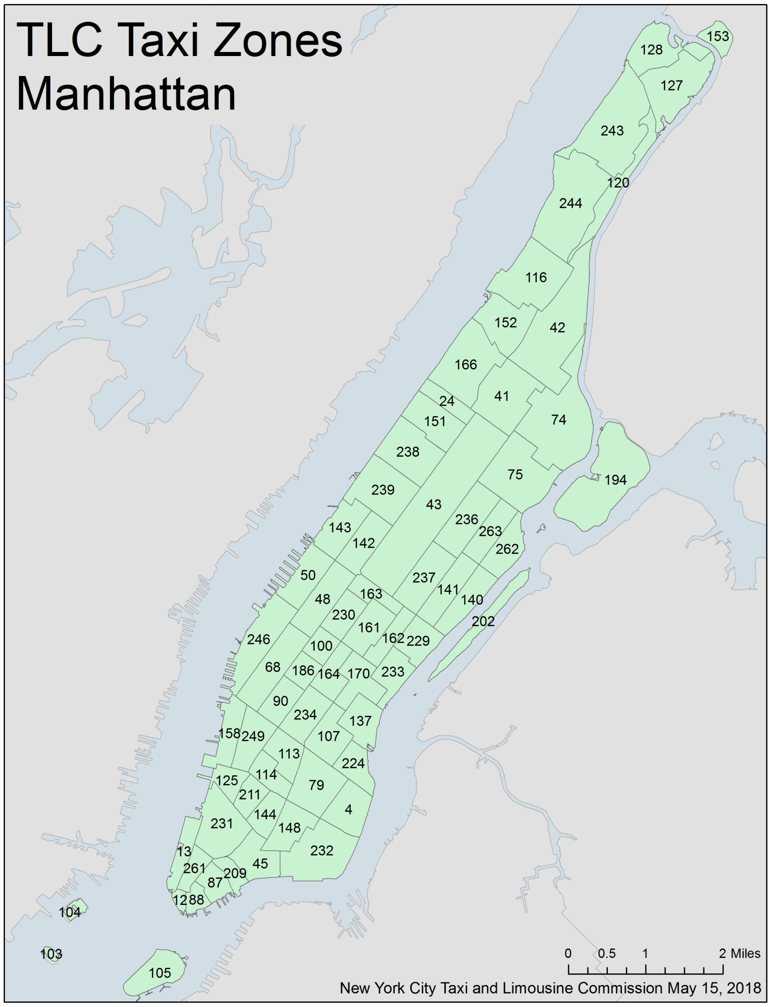
To analyze travel behavior, many studies use a spatial clustering method, also known as city partitioning, spatial clustering, or city discretization (e.g., Altshuler et al., 2019; X. Li et al., 2020; Ke et al., 2017; X. Li et al., 2012; Chang et al., 2010; Yuan et al., 2011). By applying spatial clustering methods, areas with a high density of emerging requests can be identified (Chang et al., 2010). This section discusses the different methods to cluster zones.

The most widely implemented method to cluster zones is the grid method which uses square areas to cluster the different zones. Several square area sizes are implemented in the literature.

To analyze mobility patterns, Espín Noboa et al. (2016) implement a clustering method that does not only rely on spatial dimensions, but also on temporal dimensions. The authors utilize non-negative tensor factorization (Cichocki et al., 2009; Hao et al., 2014) to identify the clusters.

Wallar et al. (2018) present a method to optimally cluster Manhattan into regions with corresponding region centers in order to rebalance idle vehicles. The operating area (Manhattan) is discretized into regions by considering the maximum travel time between any two nodes in the graph and the closest region center. The minimum number of region centers is determined by considering the maximum travel time between any two nodes. All nodes assigned to the same center form a zone. As a result, every node in the road network graph is reachable by at least one region center within a predefined maximum travel time.

The Taxi & Limousine Commission (TLC) of New York city has divided the city into 263 taxi zones. As a result, Manhattan has been divided into 64 zones. The geo-spatial dataset of Manhattan is presented in Figure 6. The zones are divided based on their geographic characteristics.



New York City Taxi and Limousine Commission May 15, 2018

Figure 6: Taxi zones of Manhattan by the Taxi & limousine commission (TLC).

2.5.5 Forecasting methods & mathematical models

From section 2.5.3 about Mobility dependencies, it can be concluded that multiple factors influence the demand. Therefore, different factors can be combined to obtain an adequate demand prediction. Several mathematical models exist that build predictions based on spatio-temporal demand data. This section gives an overview of the well-known demand prediction models.

Interesting research regarding taxi demand predictions is provided by (Moreira-Matias et al., 2013). To create predictions about the spatio-temporal distribution of the taxi demand using streaming data, the authors employ two well-know time-series forecasting techniques: the time-varying Poisson model (Ihler et al., 2006), and the and the autoregressive integrated moving average (ARIMA) (Box et al., 2015). The Poisson model calculates the probability that n requests emerge during a certain time period. The ARIMA technique is a well-known methodology to model and forecast univariate time-series data, e.g., traffic-flow data (Min

& Wynter, 2011) and taxi demand data (Moreira-Matias et al., 2013). As a result, Moreira-Matias et al. could predict the demand for 30-minute period intervals, with a total of 506873 correctly predicted demands with an aggregated error measurement lower than 26%.

Many researches have studied the development of short-term traffic demand prediction models in the field of transportation. Overall, these models can be divided into four categories: linear methods, non-linear methods, soft computing methods, e.g., neural networks, and hybrid methods (Y. Li et al., 2017). The most popular linear methods are moving average (MA) models, autoregressive integrated moving average (ARIMA) models, exponential smoothing, Kalman Filter/Fitters, and Poisson models (Ke et al., 2017; G. P. Zhang, 2003; Y. Li et al., 2017; Moreira-Matias et al., 2013). The following sections provide an explanation of the working principles of the most popular demand prediction models.

Naive methods

The most simple method to predict the demand is to set the prediction at time t equal to the demand the day before. Let $\mathcal{Z} = \{z_1, z_2, \ldots, z_N\}$ be the set of N zones in the spatial domain of the city, then the number of predicted requests $\hat{Y}$ at time t in zone k is as follows:

$$\hat{Y}_{k,t} = Y_{k,t-(24/s)}. \tag{1}$$

Here, s represents the length of the time segment in hours. For example, if s is set to 1 hour, then the demand prediction of time segment x is equal to the demand at time segment $x - 24$. Unfortunately, this prediction is only applicable if the demand of a specific zone k is highly periodic, meaning a low variance in the daily demand (X. Li et al., 2012).

Instead of solely considering the previous day, several weekdays in the past can be considered to predict the demand, assuming that there are marginal differences in the demand rate within the considered days and the predicted demand. Denoting $\mathcal{D}$ the set of days in the considered dataset, and $Y_{k,d,t}$ the number of requests emerging in zone k at time t on day d, the demand prediction rate $\hat{Y}$ can be described as follows:

$$\hat{Y}_{k,t} = \sum_{d \in \mathcal{D}} \frac{Y_{k,d,t}}{|\mathcal{D}|}. \tag{2}$$

A similar implementation of equation 2 is provided by (Fielbaum et al., 2021b).

Another simple implementation to define the demand rate of zone k is to consider the number of requests that just have been received in zone k, assuming that the number of current incoming requests is equal to the number of recently received requests. Denoting $\mathcal{R}_{w,t}$ the set of requests that are waiting for a match at time t, and O_r denoting the origin of request r, the predicted demand in zone k is then defined by:

$$\hat{Y}_{k,t} = |\{r \in \mathcal{R}_{w,t} : o_r \in k\}| \tag{3}$$

This equation is primarily applicable when the demand is highly random and no periodicity exists in the demand, insinuating that the demand is highly affected by external factors such as weather conditions and traffic accidents. Papers that consider the current demand to estimate future demands are (Sayarshad & Chow, 2017; Fielbaum et al., 2021b).

Exponential smoothing

Exponential smoothing (Hyndman et al., 2008) is a well-known forecasting method in time series analysis. Past observations are weighed unequally such that the weights decrease exponentially over time. Hence, recent observations have a greater influence on the prediction, while multiple past observations are still considered. The most simple form of exponential smoothing to predict the demand Y at time t is formulated as

$$\hat{Y}_t = \alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1} \tag{4}$$

where α represents the smoothing constant, with $\alpha \in [0, 1]$, and Y_{t-1} is the actual demand at time $t-1$. By direct substitution of equation 4 back into itself, the following exponential smoothing equation is obtained to determine the predicted demand $\hat{Y}$ at time t given the actual demands Y_t:

$$\hat{Y}_t = \alpha Y_{t-1} + \alpha(1-\alpha)Y_{t-2} + \alpha(1-\alpha)^2 Y_{t-3} + \ldots + \alpha(1-\alpha)^n Y_{t-n}. \tag{5}$$

The smoothing factor α determines the smoothing rate of past observations. As the value of α increases, the level of smoothing is reduced. If α is set to 1, the prediction at time t is equal to the actual demand at time $t-1$. Therefore, α values close to one produce a reduced smoothing effect and prioritize recent data values, while α values close to zero result in an increased smoothing effect with a less responsive behavior to current data values. No formal procedure exists to determine the value of α. Since the smoothing factor is a user-defined parameter, its value can be determined by analyzing the significance of the considered time stamps, or by applying a statistical technique, e.g., the least squares method (Levenberg, 1944), to optimize α (Hofmann, 2019).

Poisson prediction

As stated in section 2.5.2 about Mobility characteristics, the mobility demand exhibits a periodic pattern. This daily pattern can be employed in the time-varying Poisson model to predict the demand. A Poisson distribution counts the number of occurrences of an event in a given unit of time. The Poisson distribution assumes that events occur independently and that the probability that an event occurs in a given length of time does not change through time. Hence, the events occur randomly and independently (Haight, 1967). Considering that the probability $P(n)$ for n requests to emerge in a specific time period follows a Poisson distribution, the Poisson probability mass function is formulated as follows:

$$P(n) = \frac{\lambda^n e^{-\lambda}}{n!} \tag{6}$$

where λ represents the rate (the average expected demand) in a fixed time interval. However, the demand rate of mobility services is not constant but time variant. Therefore, Hyland et al. (2020); Ihler et al. (2006); Moreira-Matias et al. (2013) transform the variable λ to a time-dependent variable, i.e., $\lambda(t)$, such that the Poisson distribution transforms into a non-homogeneous Poission distribution, in which the degree of heterogeneity depends on the function $\lambda(t)$. As a result, $\lambda(t)$ is decomposed as

$$\lambda(t) = \lambda_0 \delta_{d(t)} \eta_{d(t),h(t)} \tag{7}$$

where λ_0 represents the average/expected rate over a full week, and $d(t)$ indicates the day on which time t falls, such that $d(t)$ takes on the values $\{1 \text{ (Monday)}, \ldots, 7 \text{ (Sunday)}\}$. $\delta_{d(t)}$ is the relative change for each day of the week, as different days have different demand rates. Similarly, $\eta_{d(t),h(t)}$ represents the relative change for each time period $h(t)$ at day $d(t)$. Equation 6 requires that

$$\sum_{i=1}^{7} \delta_i = 7 \quad \text{and} \quad \sum_{i=1}^{D} \eta_{d,i} = D \quad \forall d \tag{8}$$

where D represents the number of time intervals in a day. To predict the number of requests for each zone k in the city, each demand rate $\lambda(t)_k$ is obtained by taking the average of all previously measured demands during the same day $d(t)$ and time period t. So, for example, the expected demand $\lambda(t)_k$ on a Monday at 09:00h is equal to the average demand rate of each previously measured Monday at 09:00h.

However, the demand rate can be seasonal; hence, the more recent data values are increasingly important to determine the expected demand rate $\lambda(t)$. Therefore, the relevance of the more recent values should be increased to obtain improved $\lambda(t)$ values (Moreira-Matias et al., 2013). This can for example be accomplished by implementing the discussed Exponential smoothing method.

ARIMA based prediction

The autoregressive integrated moving average (ARIMA) model (Box et al., 2015) is a widely-recognized benchmark model to forecast short-term traffic flow (J. Liu et al., 2017). It is a time series method that fits a specific statistical model to past data to forecast future data. The ARIMA model exists out of three parts, i.e., the Auto Regressive (AR) part, the Integrated (I) part, and the Moving Average (MA) part. To define the order of the ARIMA model, i.e., the number of evaluated time instances, the following notation is adopted:

$$\text{ARIMA}(p, d, q). \tag{9}$$

Here, p represents the order of the autoregressive part, d indicates the order of the integrated part, and q represents the order of the moving average part (Tsay, 2005). The autoregressive (AR) part indicates that the variable of interest regresses on its own lagged, i.e., prior, values. To predict the number of demands Y at zone k at time period t, the AR part can be described with the following function:

$$\hat{Y}_{k,t} = \phi_0 + \phi_1 Y_{k,t-1} + \phi_2 Y_{k,t-2} + \cdots + \phi_p Y_{k,t-p}. \tag{10}$$

The variable $\phi_l (l = 1, 2, \ldots, p)$ is the AR model parameter, and Y_{t-1} is the actual number of demands during the previous time period.

The moving average (MA) part forecasts the variable of interest based solely on past error values of the forecasted variable of interest. Therefore, the MA part considers the error between the prediction value of the previous time periods and the real value of these time periods. This error is formulated as ε. The MA part is defined as follows to determine the predicted demand $\hat{Y}$ at zone k at time period t:

$$\hat{Y}_{k,t} = \theta_0 + \varepsilon_{k,t} - \theta_1 \varepsilon_{k,t-1} - \theta_2 \varepsilon_{k,t-2} - \cdots - \theta_q \varepsilon_{k,t-q} \tag{11}$$

Here, $\theta_m (m = 0, 1, 2, \ldots, q)$ is the MA model parameter, and $\{\varepsilon_{k,t}, \varepsilon_{k,t-1}, \varepsilon_{k,t-2}, \ldots\}$ are error terms observed in the past signals.

The AR and MA components can be combined into an autoregressive moving average (ARMA) model to predict the demand Y at time t in zone k as

$$\begin{aligned} \hat{Y}_{k,t} = \text{constant} + \phi_1 Y_{k,t-1} + \phi_2 Y_{k,t-2} + \cdots + \phi_p Y_{k,t-p} \\ + \varepsilon_{k,t} - \theta_1 \varepsilon_{k,t-1} - \theta_2 \varepsilon_{k,t-2} - \cdots - \theta_q \varepsilon_{k,t-q} \end{aligned} \tag{12}$$

The ARMA model can be described in words as

$$\begin{gathered} \text{Predicted } Y_t = \text{Constant} + \text{Linear combination Lags of } Y \text{ (upto } p \text{ lags)} \\ + \text{ Linear Combination of Lagged forecast errors (upto } q \text{ lags).} \end{gathered} \tag{13}$$

The ARMA model assumes that the time-series data is stationary, i.e., the mean and variance of the time-series data are constant for the whole series, no matter what time period is selected. Hence, the time-series data cannot have an upward or downward trend. However, this does not hold for the mobility demand since the demand has a clear daily periodicity, as indicated in section 2.5.2. Therefore, the integrator (I) part is introduced in the ARIMA model to transform the non-stationary data into stationary data. The trend in the data is removed by differencing. As formulated in equation 10, the d parameter indicates the order of the integrated part, i.e., the number of differences needed to achieve stationary. The first difference can be presented as

$$d = 1: \quad \Delta Y_{k,t} = Y_{k,t} - Y_{k,t-1}. \tag{14}$$

Likewise, the second difference is formulated as

$$d = 2: \quad \Delta Y_{k,t} = (Y_{k,t} - Y_{k,t-1}) - (Y_{k,t-1} - Y_{k,t-2}) = Y_{k,t} - 2Y_{k,t-1} + Y_{k,t-2}. \tag{15}$$

2.6 Theoretical background summary

The theoretical background of unreliability in on-demand ridesharing systems is explored in this chapter to ultimately discover techniques to minimize unreliability. In this research, unreliability indicates that it is hard for the users to know what quality of service they will receive in terms of time management provided by the service. Users may face discomfort when the realization of their trip differs from what was initially predicted, affecting the users' short-term decisions.

From Section 2.1, it can be concluded that the causes of unreliability can be divided into two categories: causes of unreliability present in all types of transport and causes of unreliability specific to on-demand ridesharing systems, meaning that the cause is related to the vehicles' shared assignments. The second category can be further divided into unreliability induced by the users and unreliability induced by the operators. This research is interested in minimizing unreliability induced by the operators' rules during the assignment procedure.

An analysis of the users' unreliability perception, Section 2.2, illustrates that two types of unreliability phenomena can be classified to describe how unreliability affects the user's perception of the SMoD's quality-of-service. The first type, *one-time unreliability*, occurs within a single trip: when a user is assigned to a trip, the real waiting time and IVTT might differ from the originally predicted values. These changes affect the short-term decisions of the user. The second type of unreliability, *daily unreliability*, occurs between different realizations of the same trip. Users have to take into consideration how the quality of service will be before they decide if a SMoD service will be used. Since the travel time is uncertain, users might want to plan extra time to ensure on-time arrival. The objective of this research is to discover techniques that aim to minimize one-time unreliability.

Next, Section 2.3 analyzed the different assignment processes to match vehicles to requests. It can be concluded that each technique influences the unreliability of the service. Even inferior changes in the system parameters affect the unreliability of the service.

Finally, Section 2.4 provides an overview of the discovered techniques that aim to minimize unreliability. Table 2 summarises the discovered techniques. The discoveries illustrate that there are currently no suitable techniques to efficiently resolve the unreliability problem in SMoD systems without degrading other quality-of-service indices. The few discovered techniques that aim to minimize unreliability in SMoD systems degrade the efficiency of the service drastically, making them unsuitable to implement in real-life applications. Based on the discovered techniques, several conceptualized methods are proposed in Section 2.4 to reduce unreliability. The method of providing improved travel time information by making predictions about future changes will be implemented in this research. This method has the least drawbacks in combination with its high potential to reduce unreliability. Section 2.5 provides an overview of the demand prediction models to facilitate the implementation of the unreliability measure.

In conclusion, the management of unreliability in SMoD systems has emerged as a novel key issue. However, empirical research on how operators can efficiently minimize unreliability is limited. Therefore, a gap in the literature exists of efficiently dealing with the problem of unreliability. This gives the occasion for the establishment of this research that aims to solve this knowledge gap.

3 Methodology

This chapter describes the solution implementation to reduce unreliability in ridesharing. The implemented solution is tested using a state-of-the-art routing and assignment method. Therefore, a description of the implementation of the state-of-the-art routing and assignment method is provided in Section 3.1. Next, the formulated solution is elaborately described in Section 3.2. At last, the simulation setup is provided in Section 3.3.

3.1 State-of-the-art ridesharing implementation

The state-of-the-art ridesharing routing and assignment method of Alonso-Mora et al. (2017a) is implemented to assign requests to vehicles. For the assignment process, a request is defined by its origin, destination, number of passengers, and the emerging time at which the request has been received. A fleet $\mathcal{V}$ of m vehicles with capacity c_v is considered. During each iteration, an assignment Σ of the set of n requests $\mathcal{R}$ to m vehicles $\mathcal{V}$ is optimized to minimize cost $\mathcal{C}$ of the assignment. The cost $\mathcal{C}$ of the assignment considers the cost of assigning the requests while simultaneously considering the set of rejected requests by penalizing them. The assignment process is executed iteratively every minute, such that $\mathcal{R}$ contains the requests received during the last minute and the requests that can be reassigned. A request can be reassigned if it has been assigned during previous iterations and has not been picked-up yet. The cost function of Fielbaum and Alonso-Mora (2020), which is based on the cost function of Alonso-Mora et al. (2017a), is implemented in this research, and is formulated as:

$$\mathcal{C}(\Sigma) = \sum_{v \in \mathcal{V}} \sum_{p \in \mathcal{P}_v} c_p + \sum_{r \in \mathcal{R}_{ok}} c_r + \sum_{r \in \mathcal{R}_{ko}} c_{ko} + \sum_{r \in \mathcal{R}_{ok}} c_o. \tag{16}$$

In this equation, $\mathcal{P}_v$ describes the set of picked-up requests (also known as passengers) by vehicle $v \in \mathcal{V}$ which are now traveling to their destination, $\mathcal{R}_{ok}$ is the set of requests that have been assigned to some vehicle, and $\mathcal{R}_{ko}$ is the set of unassigned requests. Furthermore, c_p is the cost of serving passenger p, c_r is the cost of serving request r, c_{ko} is a constant to penalize each rejected request, and c_o is the operator cost to serve request r. Equation 16 is subjected to the following constraints:

- Each request r cannot exceed a maximum waiting time Ω. The waiting time $t_w(r)$ of request r is equal to the difference between the pick-up time t_r^p and the request's emerging time t_r.
- Each request r cannot exceed a maximum travel delay Δ. The travel delay $D(r)$ of request r is equal to the difference between the drop-off time t_r^d and the earliest possible time at which the destination can be reached if the vehicle would travel directly with the shortest path from the origin O_r to the destination D_r of request r.
- Each vehicle v cannot exceed its maximum capacity c_v.

Since solving this optimization problem is highly computationally expensive, Alonso-Mora et al. split the optimization problem into four consecutive steps to reduce computation time while maintaining high solution quality. These steps are: computing a pairwise request-vehicle shareability graph (RV-graph); computing a request-trip-vehicle-graph (RTV-graph) to explore the cliques (complete sub-graphs) of the RV-graph; solving an Integer Linear Program (ILP) to compute the best assignment of vehicles to trips; and rebalancing idle vehicles. These steps are also implemented in this research; a detailed explanation of the steps is provided in the following sections.

After completing these steps, the rejected requests $\mathcal{R}_{ko}$ are removed such that they cannot be assigned during the subsequent iterations. Next, the system starts accumulating newly received requests such that they can be assigned during the next iteration. Besides, the already assigned requests which have not been picked-up yet are added to the batch of requests that will be considered during the successive assignment procedure.

3.1.1 RV-graph

A request-vehicle graph (RV-graph) is created to determine which requests can be pairwise-shared and which requests and vehicles can be pairwise-shared. Therefore, two requests r_1 and r_2 are feasible to share a trip if a virtual vehicle starting at the origin of one of them can serve both requests without violating their constraints. Similarly, a combination of a request r and a vehicle v is feasible if the vehicle can serve the request without violating the constraints of the assigned users of the vehicle. All feasible matches create the RV-graph.

3.1.2 RTV-graph

A request-trip-vehicle-graph (RTV-graph) is created to determine which sets of requests can be served by which vehicle. This is determined by exploring the RV-graph's cliques, i.e., the feasible request-vehicle combinations and request-request combinations. The RTV-graph is created by exploring the feasible trips for each vehicle incrementally in trip size. Here, a trip is defined as the set of n_T requests. A trip is feasible when its subtrips, i.e., RV-graphs, are also feasible, reducing the number of potential trips. Therefore, a trip is feasible if all trip requests can be picked-up and dropped-off by a vehicle without violating the constraints of the trip requests and without exceeding the constraints of the already assigned users of the vehicle. Each feasible trip T is described by an edge $e(T, v)$, containing the set of n_T requests in trip T that can be served by vehicle v and the associated cost of serving this trip. It might be possible that a request is part of multiple trips of different sizes. Likewise, multiple vehicles might be able to serve the same trip.

3.1.3 ILP assignment

The problem is solved with an Integer Linear Program (ILP) to compute the best assignment of vehicles to trips. The ILP is solved by selecting the edges of the RTV-graph such that the cost function, Equation 16, is minimized. To solve the ILP, Alonso-Mora et al. introduce a binary variable $\epsilon_{i,j} \in \{0, 1\}$ for each edge $e(T_i, v_j)$ of the RTV-graph. $\epsilon_{i,j} = 1$ indicates that vehicle v_j has been assigned to trip T_i. The set of $\{i, j\}$ indices for which a feasible edge $e(T_i, v_j)$ exists in the RTV-graph is denoted by $\mathcal{E}_{TV}$. Furthermore, a binary variable $\chi_k \in \{0, 1\}$ is introduced for each request $r_k \in \mathcal{R}$, defining whether request r_k has been assigned ($\chi_k = 0$) or rejected ($\chi_k = 1$). Edge $e(T_i, v_j)$ has a cost $c_{i,j}$, such that the following assignment is being optimized:

$$\Sigma_{\text{optim}} = \arg\min_{\mathcal{X}} \sum_{i,j \in \mathcal{E}_{TV}} c_{i,j}\epsilon_{i,j} + \sum_{k \in \{1,\ldots,n\}} c_{ko}\chi_k. \tag{17}$$

Equation 17 is obtained from Alonso-Mora et al. (2017a). To solve this optimization, two constraints are added to ensure that each request is assigned to at most one vehicle and each vehicle is assigned to at most one trip.

Edge $e(T_i, v_j)$ has a cost $c_{i,j}$. In this research, the cost of an edge is equal to the formulated edge cost function by Fielbaum and Alonso-Mora (2020), where the cost $c_{i,j}$ depends on the waiting and in-vehicle time of each request of trip T_i, on the extra time induced to the current requests of vehicle v_j, and on the extra travel distance that v_j has to travel (called operators' cost). The cost $c_{i,j}$ of serving trip T_i with vehicle v_j is expressed as:

$$\begin{aligned} c(i,j) = &\sum_{r \in T_i} [p_w t_w(r,i,j) + p_v t_v(r,i,j)] \\ &+ \sum_{r \in v_j} [p_w \Delta t_w(r,i,j) + p_v \Delta t_v(r,i,j)] \\ &+ c_0 \Delta L(i,j). \end{aligned} \tag{18}$$

The first term describes the cost of the users of trip T_i, the second term represents the cost of the requests that were already being served by v_j, and the third term describes the operators' cost. The unitary cost of waiting and in-vehicle travel time for the requests is denoted by p_w and p_v, respectively, and the waiting time and in-vehicle travel time of a request are denoted by t_w and t_v, respectively. The operators' cost is described by the extra length ΔL vehicle v_j has to travel to serve the requests of trip T_i and the respective unitary operator cost c_0.

Such a cost function is implemented as operators are interested in providing high-quality service to the users and minimizing their own costs (Fielbaum & Alonso-Mora, 2020). The numeric values of the implemented parameters are similar to those of Alonso-Mora et al. (2017a) and Fielbaum and Alonso-Mora (2020) and are provided in table 3.

Table 3: Parameters used for the assignment process. The maximum waiting time and delay are equal to the values of Alonso-Mora et al. (2017a). The remaining parameter values are equal to the values of Fielbaum and Alonso-Mora (2020).

Parameter	**Value**
Max. admissible waiting time Ω	5 [min]
Max. admissible delay Δ	10 [min]
Weight of one in-vehicle minute for users p_v	1
Weight of one minute waiting for users p_w	2
Weight of rejecting a request c_{ko}	40
Weight of one minute with a vehicle in-motion for operators c_0	1.5

3.1.4 Rebalancing

A rebalancing step is performed to move idle vehicles, i.e., vehicles with no requests assigned, to regions of higher demand. Rebalancing is achieved by matching the idle vehicles to the rejected requests $\mathcal{R}_{ko}$. First, the travel time of each individual vehicle to each individual rejected request's origin is computed. Next, the optimization problem is solved with an ILP to reduce the total travel time. If there are more vehicles than rejected requests, some vehicles remain idle. If there are more rejected requests than idle vehicles, each vehicle is assigned to exactly one request. The rejected requests are not actually picked-up. They are only used to move idle vehicles to regions where the demand cannot be satisfied.

3.1.5 Not allowing for reassignments

This research aims to minimize unreliability by providing information about the possibility and magnitude that a request will face extra travel time. This information is obtained by predicting the possibility that a new request will be added to the trip, which is determined with the shareability shadow concept, discussed in Section 3.2.1. An advantage of the shareability shadow is that it is able to obtain an adequate prediction without being computationally expensive and without knowing the exact origins and destinations of new requests. However, a disadvantage of this method is that it cannot consider rescheduled requests, i.e., requests that were first assigned to a specific vehicle and the next iteration are reassigned to another vehicle. This results in a reduced performance of the unreliability predictions.

Therefore, simulations are also performed for which rescheduling requests is not allowed. When rescheduling is not allowed, the set of n requests $\mathcal{R}$ considered during the optimization process does not contain the set of requests which are already assigned to a vehicle during previous iterations. The same cost function of Equation 16 can be used for the assignment problem, except that the set of already assigned requests to vehicle v, denoted by $\mathcal{A}_v$, is considered for the first term instead of $\mathcal{P}_v$, and the cost of serving the already assigned requests, denoted by c_a, is introduced. This gives the following equation for the cost of the assignment:

$$\mathcal{C}(\Sigma) = \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{A}_v} c_a + \sum_{r \in \mathcal{R}_{ok}} c_r + \sum_{r \in \mathcal{R}_{ko}} c_{ko} + \sum_{r \in \mathcal{R}_{ok}} c_o. \tag{19}$$

3.2 Reliability model implementation

A description of the measure to reduce unreliability is provided in this section. The measure provides each user the prediction of the possibility and magnitude of a possible negative change in trip duration. These indices are determined using the shareability shadow concept, described in Section 3.2.1, and a demand forecasting method, discussed in Section 3.2.2. Finally, the construction of the unreliability measure is provided in Section 3.2.3.

The measure aims to reduce the unreliability of each user. Therefore, we consider a user who poses a request r. The following definitions are formulated for this request:

- Waiting time $t_w(r)$, which is equal to the difference between the pick-up time t_r^p and the request's emerging time t_r.
- In-vehicle time $t_v(r)$, which is the difference between the drop-off time t_r^d and the pick-up time t_r^p.
- Detour time $t_d(r)$, which is equal to the in-vehicle time minus the travel time of the shortest origin-destination path of the request.
- Travel delay $D(r)$, which is equal to the difference between the drop-off time t_r^d and the earliest possible time at which the destination can be reached if the vehicle would travel directly with the shortest path from the origin O_r to the destination D_r of request r.

3.2.1 Shareability shadow

A well-known notion in the world of ridesharing is shareability. Shareability defines the number or fraction of individual trips that can be shared. Tachet et al. (2017); Bilali et al. (2019) examine the shareability by specifying different quality-of-service indices, such as delay time and waiting time, by implementing a *shareability shadow* model that does not require computationally expensive simulations or expensive real-life pilots. Their presented model can accurately determine the shareability rate. Hence, the shareability shadow model is relevant for operators and city planners to explore ridesharing characteristics. The shareability shadow approach is implemented in this research because it requires less computation time, while still providing an accurate prediction of the shareability rate.

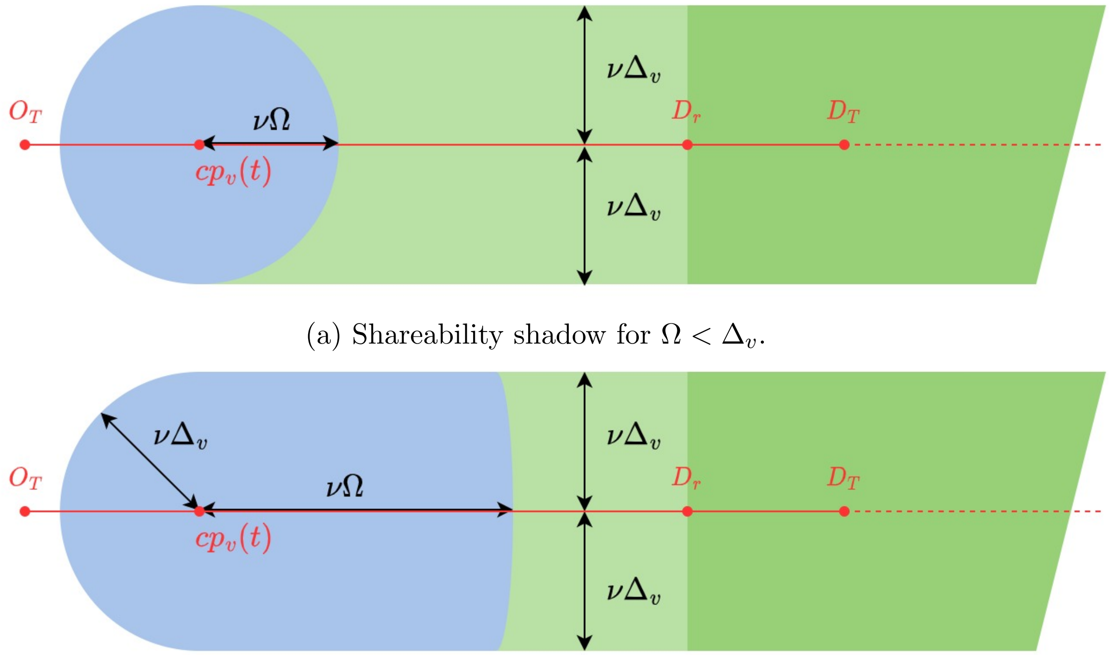
Shareability shadow theoretical background

The shareability shadow is an existing concept in the world of ridesharing. As stated by Tachet et al. (2017); Bilali et al. (2019), shareability is modeled by considering a trip T and estimates the probability that there exists at least one request r' that can be added to T. The notion of the shareability shadow confines the region of ridesharing opportunities and is created by geometric shapes representing different constraints. The shareability shadow, illustrated in Figure 7, is based on the shareability shadows by Tachet et al. (2017); Bilali et al. (2019), and is defined by the tolerable ridesharing delay of the in-vehicle users, denoted by Δ_v, the admissible waiting time Ω of r', and the average speed ν of the vehicle v. A new request r' is shareable with T if their trajectories (partially) overlap in space and time, and the deviation is not larger than the tolerable ridesharing delay of the in-vehicle users Δ_v. The ridesharing delay Δ_v depends on the accumulated delay of the vehicle's assigned requests. Note that Δ_v is different from the implemented maximum delay time Δ, which is equal to ten minutes for all requests (See Table 3). The red line in Figure 7 presents the trajectory of trip T which has an origin O_T and destination D_T. The trajectory is presented as a horizontal straight line and forms the trajectory in space and time of trip T. At time t, the vehicle of the trip is positioned at its current position $cp_v(t)$. The tolerable ridesharing delay time Δ_v of the in-vehicle users cannot be exceeded when adding request r' to trip T. Hence, the deviation from the trajectory of T (red line) cannot exceed $\nu\Delta_v$. Here, $\nu\Delta_v$ presents the maximum distance a vehicle can deviate, considering the tolerable ridesharing delay time Δ_v of the in-vehicle users and vehicle speed ν. Therefore, the delay constraint is presented in the shareability shadow as the green area with height $2\nu\Delta_v$.

Besides analyzing the constraints of the in-vehicle users, the constraints of the new request r' have to be analyzed as well to determine whether request r' can be shared with Trip T. Hence, the origin $O_{r'}$ of request r' should be within the blue area surrounding $cp_v(t)$ such that the maximum waiting time Ω is not exceeded. Note that the implemented maximum waiting time Ω is equal to five minutes for all requests (See Table 3), while the maximum ridesharing delay Δ_v can take different values, depending on the accumulated delay of the in-vehicle users. This results in two different shareability shadows:

- If the maximum waiting time Ω is *smaller* than the maximum ridesharing delay time Δ_v of vehicle v's assigned users, the shareability shadow in Figure 7a is established.
- On the other hand, if the maximum waiting time Ω is *larger* than the maximum ridesharing delay time Δ_v of vehicle v's assigned users, the shareability shadow in Figure 7b is established.

This research aims to reduce unreliability for each request r that has been added to trip T by providing improved travel time information. Thus, the shareability of T is analyzed when r has been added to T. Hence, when analyzing the shareability of request r, with $r \in T$, the destination $D_{r'}$ of the new request r' should either be before the drop-off D_r of requests r, i.e., the light green area, or after D_r, i.e., the dark green area. Additionally, this fulfills the constraint that T and r' should have (partially) overlapping trajectories.



(a) Shareability shadow for $\Omega < \Delta_v$.

(b) Shareability shadow for $\Omega > \Delta_v$.

Figure 7: Shareability shadows representing the different constraints on sharing trip T, traveling in a straight trajectory (red line) from origin O_T to destination D_T, with a new request r' which has a maximum waiting time Ω. The vehicle of trip T travels with speed ν, and users assigned to vehicle v of trip T have a tolerable ridesharing delay time Δ_v. At time t, the vehicle of trip T is positioned at its current position $cp_v(t)$. Trip T can be shared with a new request r', if the origin of r' is in the blue region while the destination of r' is in the green region. This shareability shadow is based on the shareability shadows of Tachet et al. (2017); Bilali et al. (2019)

The discussed shareability shadow can be described mathematically. Equations 20 up to 24 provide a mathematical description of the discussed shareability shadow. Nevertheless, these equations are not implemented in the final solution to reduce unreliability. They are discussed to provide a basis for establishing the final equation (Equation 35), which will be used to reduce unreliability.

The discussed equations are based on the mathematical models of Tachet et al. (2017); Bilali et al. (2019). However, Bilali et al. (2019) also consider the vehicle boarding and disembarking time, which is neglected for simplicity in this research. According to Tachet et al. (2017), the shareability S of a trip is defined by the dimensionless quantity L, describing the ratio of the tolerable ridesharing delay time Δ_v to the expected duration until a request appears, denoted by t_{ra}, such that

$$L = \frac{\Delta_v}{t_{ra}}. \tag{20}$$

The expected duration until a request appears t_{ra} describes how long it takes until a new request is generated in the trip's vicinity. To determine t_{ra}, Tachet et al. (2017) define two parameters; the operating area of the city, denoted in this research by C, and the average number of trips per hour with both endpoints in C, denoted by λ. The characteristic linear scale of a vicinity is defined by $\nu\Delta_v$, describing the distance a vehicle can travel with velocity ν when the vehicle's assigned users have a maximum tolerable ridesharing delay time Δ_v. The total amount of vicinities is

$$\#\text{ vicinities} = \frac{C}{(\nu\Delta_v)^2}, \tag{21}$$

where each vicinity has an area of $(\nu\Delta_v)^2$. By assuming that the generated requests follow a uniform distribution in space, the expected duration time until a request is generated in the trip's vicinity is defined as:

$$t_{ra} = \frac{1}{\lambda} \times \frac{C}{(\nu\Delta_v)^2}. \tag{22}$$

Substituting Eq. 22 into Eq. 20 gives

$$L = \frac{\lambda\Delta_v^3\nu^2}{C}. \tag{23}$$

Finally, Tachet et al. (2017) state that the overall shareability S of a trip can be determined by considering the probability that a trip can be shared and considering the probability that requests will emerge at a specific point in time and space. Next, the authors integrate these probabilities over the corresponding time and space. By executing complex integrations and performing some simplifications, Tachet et al. (2017) define the final equation for the shareability of a trip as:

$$S = 1 - \frac{1}{2L^3}\left(1 - e^{-L}\right)\left(1 - (1 + 2L)e^{-2L}\right). \tag{24}$$

As stated before, Equations 20 up to 24 are discussed as they provide a basis for the establishment of the final equation (Equation 35) that will be used in this research to reduce unreliability.

Shareability shadow as function of the node travel times

The previously discussed shareability model determines the shareability of a trip by analyzing the geographical regions a vehicle can travel through without violating the constraints of the considered users. Tachet et al. (2017); Bilali et al. (2019) determine these regions by adopting a constant vehicle speed ν. However, in this research, the vehicle speed is not assumed to be constant, but it can take different speeds between different nodes. Therefore, the shareability shadow regions will be a function of the node travel times instead of a constant vehicle speed.

As a result, a new shareability shadow is presented in Figure 8 which depends on the node travel times. Again, the figure illustrates the shareability shadow of trip T at time t when request r has been assigned to T. Each dot in the figure represents a node, and the travel time between the nodes does not scale linearly with the distance between the nodes. The trajectory of T is represented by a red line in space in time. The trajectory is no longer represented as a straight line but as the path T follows by traveling through the different nodes of the trip. Some of these red nodes are the scheduled stops, others are nodes the vehicle has to travel through to reach its stops. The current position of vehicle v of trip T

is at $cp_v(t)$ (node 1). The final destination of T is at D_T (node 14) while the drop-off node of r is at D_r (node 8). The green area defines the set of nodes that can be reached without exceeding the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v.

As stated before, the objective of this research is to reduce unreliability for each request r that has been assigned to trip T by providing improved travel time information to r. Thus, the shareability is analyzed when r has been assigned to T. This shareability provides information about the possibility that the travel time of r will increase. Therefore, the green area of Figure 8 is divided into a dark green and light green area. The dark green area presents the set of nodes that can be reached posterior to the drop-off node of r, and the light green area presents the set of nodes that can be reached prior to the drop-off node of r.

The blue area presents the set of nodes that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v. Each node in this blue area can be reached within a travel time smaller than the maximum waiting time Ω. Hence, a new request r' can be added to T if its origin is located in the blue area while its destination is in the green area.

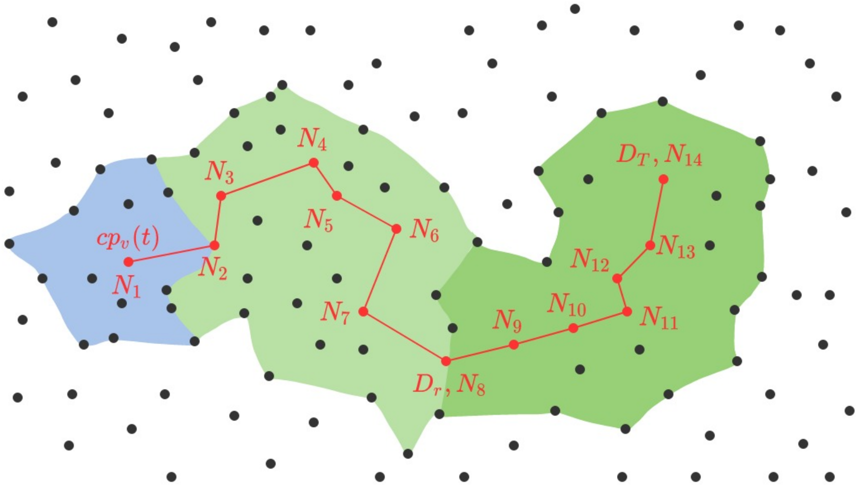


Figure 8: Shareability shadow of the pick-up and drop-off constraints to share trip T of request r with another request r'. Each dot represents a node. The red line defines the trajectory of T in space and time. The current position of the vehicle of the trip is at $cp_v(t)$ (Node 1). The final destination of T is at node 14 and the drop-off node of r is at D_r, node 8. The blue area represents the set of nodes that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v. The green area defines the drop-off region of r', as these nodes can be reached without exceeding the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v. Light green indicates a drop-off of r' prior to D_r, dark green indicates a drop-off of r' posterior to D_r.

The new equation for the expected duration t_{ra} until a request r' appears that satisfies the trip constraints, which is a function of node travel times instead of a constant vehicle speed, can be described mathematically in a similar manner as Equation 22. It is assumed that the origins and destinations of the ridesharing demand are uniformly distributed in space, such that the demand rate can be rescaled to obtain the expected duration t_{ra} until a new request r' is received that can be added to trip T. Let $\mathcal{N}_C = \{n_1, n_2, \ldots, n_N\}$ be the complete set of M nodes in the spatial domain of the city. The duration t_{ra} until a single request r' has been received that can be added to trip T is defined as:

$$t_{ra} = \frac{1}{\lambda} \times \underbrace{\frac{\#\mathcal{N}_C}{\#\mathcal{N}_\Omega(t)}}_{\substack{\text{ratio reachable}\\ \text{pick-up nodes}}} \times \underbrace{\frac{\#\mathcal{N}_C}{\#\mathcal{N}_\Delta^{tot}(t,T)}}_{\substack{\text{ratio reachable}\\ \text{drop-off nodes}}}, \tag{25}$$

with

$$\#\mathcal{N}_\Delta^{tot}(t,T) = \sum_{n_D \in \mathcal{N}(t,T)} \#\mathcal{N}_\Delta(n_D), \tag{26}$$

and

$$\forall n_1, n_2 \in \mathcal{N}_\Delta^{tot}(t,T) \rightarrow n_1 \neq n_2. \tag{27}$$

λ presents the average number of emerging requests per hour with both endpoints in the spatial domain of the city. To obtain the duration t_{ra} until a request r' appears that satisfies the trip constraints, $1/\lambda$ is multiplied by the ratio of nodes that satisfy the constraints to pick-up any request r' and the ratio of nodes that satisfy the constraints to drop-off r'. Hence, $\mathcal{N}_\Omega(t)$ describes the set of distinct nodes that can be reached without exceeding the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v. Similarly, $\mathcal{N}_\Delta^{tot}(t,T)$ describes the complete set of distinct nodes that can be reached without exceeding the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v. Thus, $\#\mathcal{N}_\Omega(t)$ is the number of nodes in $\mathcal{N}_\Omega(t)$, $\#\mathcal{N}_\Delta^{tot}(t,T)$ is the number of nodes in $\mathcal{N}_\Delta^{tot}(t,T)$, and $\#\mathcal{N}_C$ is the total number of distinct nodes in the spatial domain of the city where the service operates.

The equation of $\#\mathcal{N}_\Delta^{tot}(t,T)$ is presented in Equation 26. In this equation, $\mathcal{N}(t,T)$ presents the set of remaining nodes that are part of the trip itinerary at time t. For each node $n_D \in \mathcal{N}(t,T)$, the set of reachable nodes is determined that satisfy the constraints of the users assigned to vehicle v, denoted by $\mathcal{N}_\Delta(n_D)$. This gives the total number of distinct nodes that satisfy the constraints of the vehicle's assigned users, denoted by $\#\mathcal{N}_\Delta^{tot}(t,T)$. Note that $\mathcal{N}_\Delta^{tot}(t,T)$ only contains distinct nodes, which is illustrated by Equation 27. Thus, when for example $\mathcal{N}_\Delta(n_{D_1})$ and $\mathcal{N}_\Delta(n_{D_2})$ have a non-empty intersection, every node in this intersection is counted once in $\#\mathcal{N}_\Delta^{tot}(t,T)$.

In summary, the duration t_{ra} until a request r' appears that satisfies all constraints is computed by multiplying the time required until one request has been received $\frac{1}{\lambda(t)}$ with the fraction of reachable nodes that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to the vehicle $\frac{\#\mathcal{N}_C}{\#\mathcal{N}_\Omega(t)}$ and multiplying with the fraction of reachable nodes that satisfy the constraints of the vehicle's assigned users $\frac{\#\mathcal{N}_C}{\#\mathcal{N}_\Delta^{tot}(t,T)}$.

Parameter to reduce unreliability

A new parameter is introduced, denoted by $\lambda_r(t)$, to reduce the unreliability of request r. Section 3.2.3 describes how $\lambda_r(t)$ is utilized to reduce unreliability. The parameter $\lambda_r(t)$ defines the number of possible emerging requests that individually satisfy the shareability constraints of the trip of request r. Moreover, these requests must have a negative influence on the travel time of request r; else, they are not included in $\lambda_r(t)$. Furthermore, for simplicity, $\lambda_r(t)$ assumes that at most one request will be added to the trip to determine whether the emerging requests individually satisfy the trip constraints. At time t the value of $\lambda_r(t)$ is calculated.

Additionally, the parameter λ is refined to $\lambda(t)$ to improve the accuracy of the shareability equations. The parameter $\lambda(t)$ describes the number of emerging requests during time period t. As stated in Section 2.5.3, the number of emerging requests is highly time-dependent; therefore, taking the average number of requests per hour λ is highly inaccurate.

The parameter $\lambda_r(t)$, considers the value of $\lambda(t)$. The mathematical description of $\lambda_r(t)$ is derived from Equation 25 and is defined as:

$$\lambda_r(t) = \lambda(t) \times \underbrace{\frac{\#\mathcal{N}_\Omega(t)}{\#\mathcal{N}_C}}_{\substack{\text{ratio reachable}\\ \text{pick-up nodes}}} \times \underbrace{\frac{\#\mathcal{N}_\Delta^{tot}(t,T)}{\#\mathcal{N}_C}}_{\substack{\text{ratio reachable}\\ \text{drop-off nodes}}}, \tag{28}$$

with

$$\#\mathcal{N}_\Delta^{tot}(t,T) = \sum_{n_D \in \mathcal{N}(t,T)} \#\mathcal{N}_\Delta(n_D), \tag{29}$$

and

$$\forall n_1, n_2 \in \mathcal{N}_\Delta^{tot}(t,T) \to n_1 \neq n_2. \tag{30}$$

In words this can be described as:

$$\begin{aligned} \lambda_r(t) = &\text{ number of emerging requests at time } t \\ &\times \text{ ratio of the reachable pick-up nodes} \\ &\times \text{ ratio of the reachable drop-off nodes.} \end{aligned} \tag{31}$$

Period shareability shadow

Equation 28 determines for a specific time t the number of emerging requests that individually satisfy the trip constraints. However, this research needs to determine the number of emerging requests that satisfy the trip constraints *over a time period*, as this provides insight into the probability that new requests will be added during the whole trip of request r. This new quantity is called **period shareability**. Therefore, the notion *shareability* is different from the notion *period shareability*.

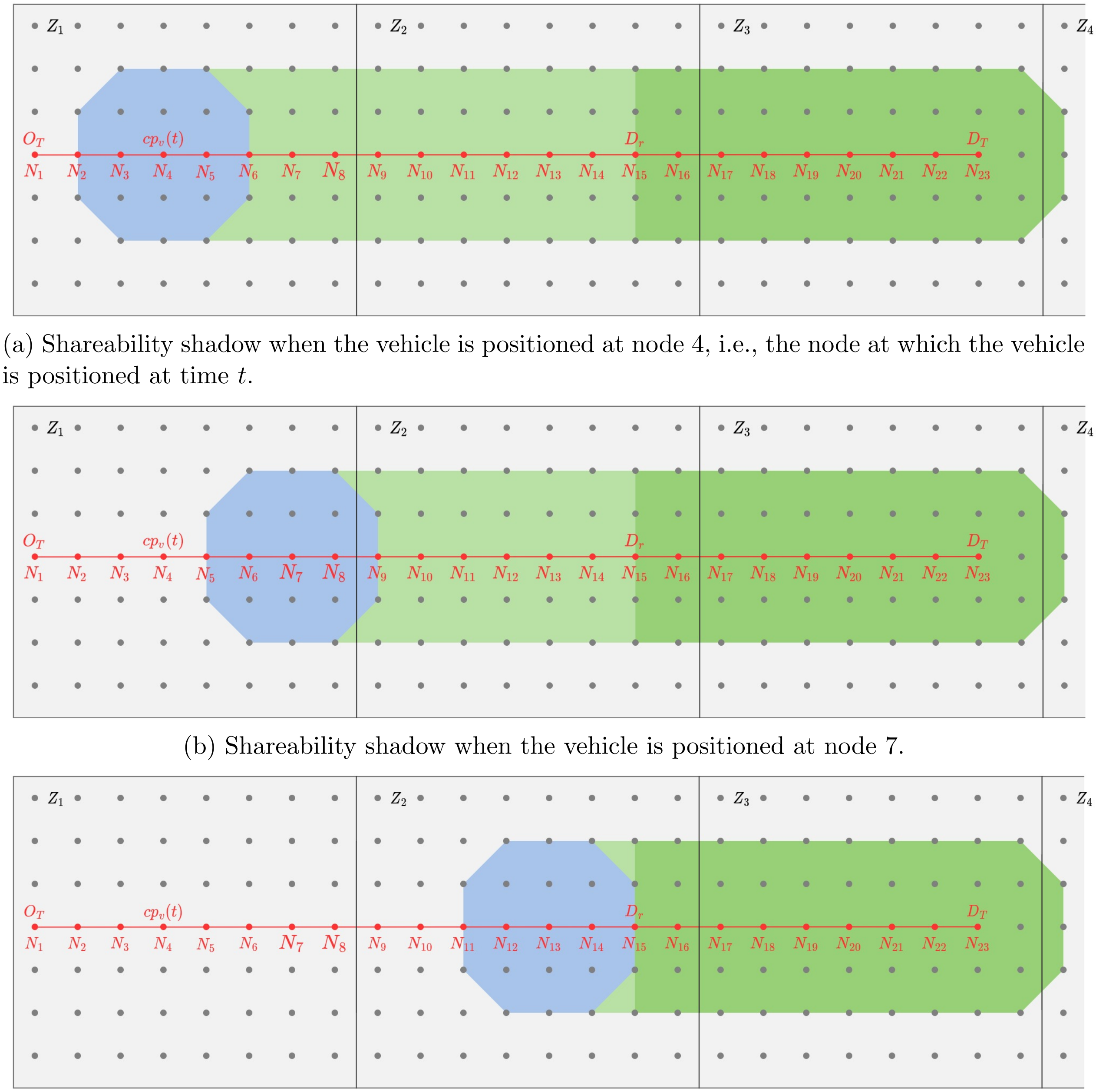
The period shareability of request r is determined when request r is assigned to the trip because this research is interested in reducing the unreliability of request r. The unreliability measures consider the possibility that new requests will be added during the trip of request r. Therefore, the period shareability for request r is determined over a time period, starting at $t = t_{assign}(r)$, with $t_{assign}(r)$ presenting the time at which r has been assigned, and ending at $t = t_r^d(r)$, with $t_r^d(r)$ presenting the drop-off time of r according to the current itinerary.

The drop-off time t_r^d is determined given the original travel times of the itinerary at the moment that r has been assigned and assumes that the vehicle does not change its itinerary. An end time of $t_r^d(r)$ is considered as adding new requests to trip T after the drop-off time of request r will not influence the trip duration of r. In this section, the period shareability shadow model is presented, and subsequently, the mathematical description of the period shareability shadow is provided.

The period shareability is modeled by considering trip T to which request r has been assigned and estimates the number of possible emerging requests r' that can individually be added to T during the evaluated time period. The period shareability shadow presents the region of nodes that can be reached without violating the shareability constraints of the trip and the waiting time constraint of any new request r'.

Figure 9 illustrates three subfigures of the period shareability shadow of trip T. The three subfigures represent the different constraints on sharing trip T, traveling in a straight trajectory (red line) from origin O_T to destination D_T, with a new request r'. The red line defines the trajectory of T in space and time. The subfigures of the period shareability are presented for request r which has its drop-off at D_r, i.e., node 15. The presented subfigures represent the shareability shadows when the vehicle reaches node 4, node 7, and node 13, respectively. To obtain the complete period shareability shadow for this specific trip, the shareability shadows of node 4 up to node 15 must be combined.

For simplicity and clarity, the nodes in Figure 9 are uniformly distributed in space, and the travel time between all adjacent nodes is set to be equal. At time t, the vehicle of trip T is positioned at its current position $cp_v(t)$. Trip T can be shared with a new request r', if the origin of r' is in the blue region while the destination of r' is in the green region. The blue region presents the nodes that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v. The green region presents the nodes that satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v. Light green indicates that the drop-off of r' is prior to the drop-off of r, and the dark green region indicates that the drop-off of r' is posterior to the drop-off of r. Combining all shareability shadows of node 4 up to node 15 yields the period shareability for request r traveling with trip T. This complete period shareability shadow is provided in Figure 10.


Z_1
Z_2
Z_3
Z_4
O_T
$cp_v(t)$
D_r
D_T
N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} N_{17} N_{18} N_{19} N_{20} N_{21} N_{22} N_{23}

(a) Shareability shadow when the vehicle is positioned at node 4, i.e., the node at which the vehicle is positioned at time t.

(b) Shareability shadow when the vehicle is positioned at node 7.

(c) Shareability shadow when the vehicle is positioned at node 13.

Figure 9: Subfigures of the period shareability shadow representing the different constraints to share trip T, traveling in a straight trajectory (red line) from origin O_T to destination D_T, with a new request r'. The red line defines the trajectory of T in space and time. The subfigures of the period shareability are presented for request r, which has its drop-off at D_r, i.e., node 15. The presented subfigures represent the shareability shadows when the vehicle is positioned at node 4, node 7, and node 13, respectively. At time t, the vehicle of trip T is positioned at its current position $cp_v(t)$. Trip T can be shared with a new request r', if the origin of r' is in the blue region, while the destination of r' is in the green region. The blue region presents the nodes that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v. The green region presents the nodes that satisfy the constraints of the users assigned to vehicle v. Light green indicates that the drop-off of r' is prior to the drop-off of r, and the dark green region indicates that the drop-off of r' is posterior to the drop-off of r.

As stated in the previous paragraph, the complete period shareability shadow for the discussed request r of Figure 9 can be obtained by combining all shareability shadows of all relevant nodes. Therefore, the complete period shareability shadow of request r is presented in Figure 10. The period shareability shadow presents the shareability constraints of the trip of request r in space and time. This implies that Figure 10 represents where and when a new request r' can appear such that it could be assigned to the trip of request r. Specifically, the blue region presents the region where a new request r' can be picked-up, which negatively influences the trip duration of request r. Similarly, the green region presents the region where a new request r' can be dropped-off. The mixed blue-and-green region presents the region where request r' can both be picked-up and dropped-off such that it negatively influences the trip duration of request r.

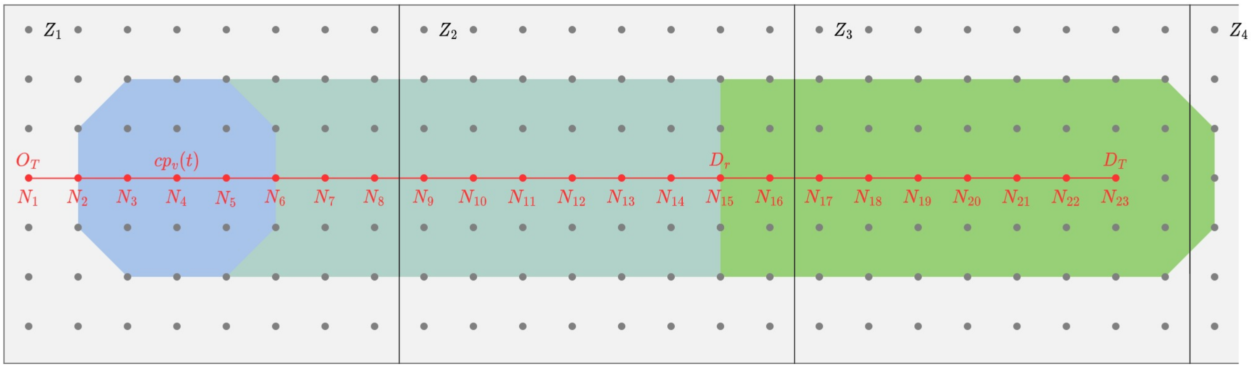


Figure 10: Period shareability shadow of the pick-up and drop-off constraints to share trip T of request r with another request r'. The red line defines the trajectory of T in space and time. The period shareability shadow in this figure is obtained by combining the shareability shadows, i.a. Figure 9, for each node in the trip into one figure. The blue region presents the region where a new request r' can be picked-up, which negatively influences the trip duration of request r. Similarly, the green region presents the region where a new request r' can be dropped-off. The mixed blue-and-green region presents the region where request r' can both be picked-up and dropped-off such that it negatively influences the trip duration of request r.

For clarification, the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v take on different values depending on the vehicle's location. Therefore, the period shareability shadow of Figure 10 may not correctly represent the values of Ω_v and Δ_v. Hence, a more realistic representation of the period shareability shadow is presented in Figure 11. In this Figure, the vehicle has to make a stop at nodes 7, 12, 15, and 23 to serve the users of the trip.

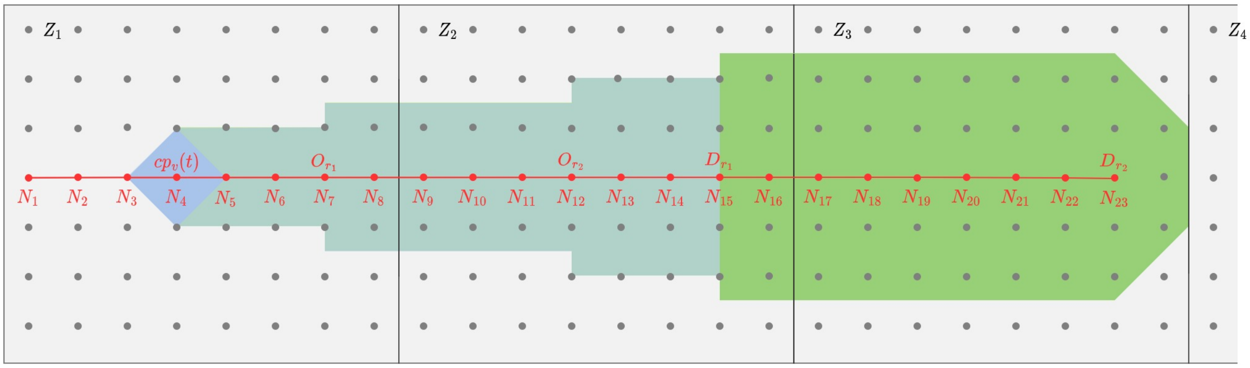


Figure 11: A more realistic presentation of the period shareability shadow presented in Figure 10. The vehicle has to make a stop at nodes 7, 12, 15, and 23 to serve the users of the trip. Therefore, the values Ω_v and Δ_v take different values depending on the location of the vehicle.

The period shareability shadow is used to compute the total number of possible emerging requests that individually satisfy the trip's shareability constraints and which negatively influence the travel time of request r. Hence, Equation 28 is extended by taking the sum over all nodes n that T travels through up to drop-off node D_r. Let $\mathcal{N}(T, D_r) = \{n_1, n_2, \ldots, n_L\}$ be the set of L nodes that trip T travels through up to drop-off node D_r, and time t_n describes the time at which the vehicle reaches node $n \in \mathcal{N}(T, D_r)$, then the number of possible emerging requests that individually satisfy the constraints of the trip of request r, denoted by $\lambda_r(t)$, calculated at time t, is described as:

$$\lambda_r(t) = \sum_{n \in \mathcal{N}(T,D_r)} \lambda(t_n) \times \underbrace{\frac{\#\mathcal{N}_\Omega(n)}{\#\mathcal{N}_C}}_{\substack{\text{ratio reachable} \\ \text{pick-up nodes}}} \times \underbrace{\frac{\#\mathcal{N}^{tot}_\Delta(n, T)}{\#\mathcal{N}_C}}_{\substack{\text{ratio reachable} \\ \text{drop-off nodes}}}, \tag{32}$$

with

$$\#\mathcal{N}^{tot}_\Delta(n, T) = \sum_{n_D \in \mathcal{N}(n,T)} \#\mathcal{N}_\Delta(n_D), \tag{33}$$

and

$$\forall n_1, n_2 \in \mathcal{N}^{tot}_\Delta(t, T) \to n_1 \neq n_2. \tag{34}$$

Note that $\lambda_r(t)$ only considers the number of possible emerging requests that have an influence on the travel time of r because Equation 32 considers the sub-shareability shadows for all nodes n in the trip up to the drop-off node D_r of request r. Furthermore, $\lambda(t_n)$ describes the total number of emerging requests in the spatial domain of the city at time t_n, where t_n represents the time at which the vehicle will reach node n given its current itinerary.

Equation 32 assumes that the emerging requests are uniformly distributed over the spatial domain of the city. However, as stated in section 2.5.2, this distribution is far from uniform. Therefore, demand predictions should be performed over each zone instead of the entire geographical space of the city to improve the value of $\lambda_r(t)$. This impacts Equation 32, which can be rewritten by considering the zones trip T travels through and assuming that the distribution of requests over each zone is uniform. As a result, the final equation for the number of possible requests that individually satisfy the trip shareability constraints and negatively influence the trip duration of request r, determined at time t, is formulated as:

$$\lambda_r(t) = \sum_{n \in \mathcal{N}(T,D_r)} \sum_{z_O \in \mathcal{Z}_\Omega(n)} \sum_{z_D \in \mathcal{Z}_\Delta(n,T)} \lambda_{z_O,z_D}(t_n) \times \underbrace{\frac{\#\mathcal{N}_\Omega(z_O,n)}{\#\mathcal{N}_{z_O}}}_{\substack{\text{ratio reachable} \\ \text{pick-up nodes}}} \times \underbrace{\frac{\#\mathcal{N}^{tot}_\Delta(z_D,T)}{\#\mathcal{N}_{z_D}}}_{\substack{\text{ratio reachable} \\ \text{drop-off nodes}}}, \tag{35}$$

with

$$\#\mathcal{N}^{tot}_\Delta(z_D,T) = \sum_{n_D \in \mathcal{N}(T,z_D)} \#\mathcal{N}_\Delta(n_D), \tag{36}$$

and

$$\forall n_1, n_2 \in \mathcal{N}^{tot}_\Delta(z_D,T) \to n_1 \neq n_2. \tag{37}$$

Equation 35 is the outcome of a fairly complex path. This path is formed by the previously discussed shareability shadow equations and started by reformulating Equation 22, obtained from Tachet et al. (2017). Equation 35 contains many parameters; therefore, the description of the different parameters is provided in Table 4. Furthermore, for simplicity, Equation 35 assumes that at most one request will be added to the trip to determine whether an emerging request r' individually satisfies the trip constraints. Otherwise, the period shareability shadow will become narrower each time a new request r' is added to the trip. This assumption prevents the equation from becoming too complex. Equation 35 can be presented in words in a simplified manner as:

$$\begin{aligned} \lambda_r(t) = \text{Shareability score} = \text{predicted demand between every two zones of the trip} \\ \times \text{ ratio of reachable pick-up nodes } zone_1 \times \text{ ratio of reachable drop-off nodes } zone_2. \end{aligned} \tag{38}$$

Table 4: Parameter definitions of the period shareability equation. Equation 35.

Symbol	**Meaning**
$\lambda_r(t)$	Number of possible emerging requests that satisfy the trip constraints of request r and which influence the travel time of r, according to the established period shareability shadow at time t
r	Request for which the period shareability is analyzed, which is part of trip T
D_r	Destination node of r
r'	Additional request that can be added to trip T
$\mathcal{N}(T, D_r)$	All nodes in the itinerary of trip T until node D_r
$\mathcal{Z}_\Omega(n)$	Set of reachable zones from node n that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v
$\mathcal{Z}_\Delta(n, T)$	Set of reachable zones from any of the remaining nodes in trip T starting from node n that satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v
$\lambda_{z_O,z_D}(t_n)$	Number of requests at time t_n with their origin in zone z_O and their destination in zone z_D
t_n	Time at which node n will be reached given the vehicle's current itinerary
$\mathcal{N}_\Omega(z_O, n)$	Set of nodes in zone z_O that can be reached from node n that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v
$\mathcal{N}_\Delta(z_D, n_D)$	Set of nodes in zone z_D that can be reached from node n_D that satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v
$\mathcal{N}_\Delta^{tot}(z_D, T)$	Set of nodes in zone z_D that can be reached from any of the remaining nodes of the trip's itinerary that satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v
$\#\mathcal{N}_\Omega(z_O, n)$	Number of nodes in zone z_O that can be reached from node n that satisfy the maximum waiting time Ω of a new request r' and simultaneously satisfy the constraints of the users assigned to vehicle v
$\#\mathcal{N}_\Delta(z_D, n_D)$	Number of nodes in zone z_D that can be reached from node n_D that satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v
$\#\mathcal{N}_\Delta^{tot}(z_D, T)$	Number of nodes in zone z_D that can be reached from any of the remaining nodes the in trip itinerary that satisfy the tolerable ridesharing waiting time Ω_v and delay time Δ_v of the users assigned to vehicle v
$\#\mathcal{N}_{z_O}$	Number of distinct nodes in zone z_O
$\#\mathcal{N}_{z_D}$	Number of distinct nodes in zone z_D
$\mathcal{N}(T, z_D)$	All nodes in the itinerary of trip T located in zone z_D

The number of possible emerging requests that satisfy the trip constraints and which negatively influence the travel time of r, denoted by $\lambda_r(t)$, will be used to reduce the unreliability of request r. This is achieved by providing predictions about possible changes in waiting time and in-vehicle time (which will be explained in Section 3.2.3). Therefore, $\lambda_r(t)$ is divided into two parameters: (1) the number of possible emerging requests which have a negative influence on the *waiting time* of r, denoted by $\lambda_r^w(t)$; and (2) the number of possible emerging requests which have a negative influence on the *in-vehicle time* of request r, denoted by $\lambda_r^{iv}(t)$. These two parameters are determined by dividing the set of nodes $\mathcal{N}_\Omega$ into two different groups: (1) the set of nodes that can be reached *prior to* pick-up node O_r, denoted by $\mathcal{N}_\Omega^{priorO_r}$; (2) and the set of nodes that can be reached *posterior to* pick-up node O_r, denoted by $\mathcal{N}_\Omega^{postO_r}$. Figure 12 provides an illustration of which nodes will be assigned to $\mathcal{N}_\Omega^{priorO_r}$ and $\mathcal{N}_\Omega^{postO_r}$ for a request r with an origin O_r at node 7 and a destination D_r at node 15. All nodes in the light blue region are assigned to $\mathcal{N}_\Omega^{priorO_r}$ because they can be reached prior to O_r. Likewise, all nodes in the dark blue region are assigned to $\mathcal{N}_\Omega^{postO_r}$ because they will be reached posterior to O_r.

The following two equations are formulated to determine the number of possible emerging requests which have a negative influence on the waiting time and in-vehicle time of request r, respectively:

$$\lambda_r^w(t) = \sum_{n \in \mathcal{N}(T,D_r)} \sum_{z_O \in \mathcal{Z}_\Omega(n)} \sum_{z_D \in \mathcal{Z}_\Delta(n,T)} \lambda_{z_O,z_D}(t_n) \times \underbrace{\frac{\#\mathcal{N}_\Omega^{priorO_r}(z_O,n)}{\#\mathcal{N}_{z_O}}}_{\substack{\text{ratio reachable}\\ \text{pick-up nodes}}} \times \underbrace{\frac{\#\mathcal{N}_\Delta^{tot}(z_D,T)}{\#\mathcal{N}_{z_D}}}_{\substack{\text{ratio reachable}\\ \text{drop-off nodes}}}, \tag{39}$$

and

$$\lambda_r^{iv}(t) = \sum_{n \in \mathcal{N}(T,D_r)} \sum_{z_O \in \mathcal{Z}_\Omega(n)} \sum_{z_D \in \mathcal{Z}_\Delta(n,T)} \lambda_{z_O,z_D}(t_n) \times \underbrace{\frac{\#\mathcal{N}_\Omega^{postO_r}(z_O,n)}{\#\mathcal{N}_{z_O}}}_{\substack{\text{ratio reachable}\\ \text{pick-up nodes}}} \times \underbrace{\frac{\#\mathcal{N}_\Delta^{tot}(z_D,T)}{\#\mathcal{N}_{z_D}}}_{\substack{\text{ratio reachable}\\ \text{drop-off nodes}}}, \tag{40}$$

with

$$\#\mathcal{N}_\Delta^{tot}(z_D,T) = \sum_{n_D \in \mathcal{N}(T,z_D)} \#\mathcal{N}_\Delta(n_D), \tag{41}$$

and

$$\forall n_1, n_2 \in \mathcal{N}_\Delta^{tot}(z_D,T) \rightarrow n_1 \neq n_2. \tag{42}$$

Here, $\#\mathcal{N}_\Omega^{priorO_r}(z_O,n)$ presents the number of nodes in zone z_O that satisfy the pick-up constraints of the trip and which can be reached prior to pick-up node O_r from node n. Similarly, $\#\mathcal{N}_\Omega^{postO_r}(z_O,n)$ presents the number of nodes in zone z_O that satisfy the pick-up constraints of the trip and which can be reached posterior to pick-up node O_r from node n.

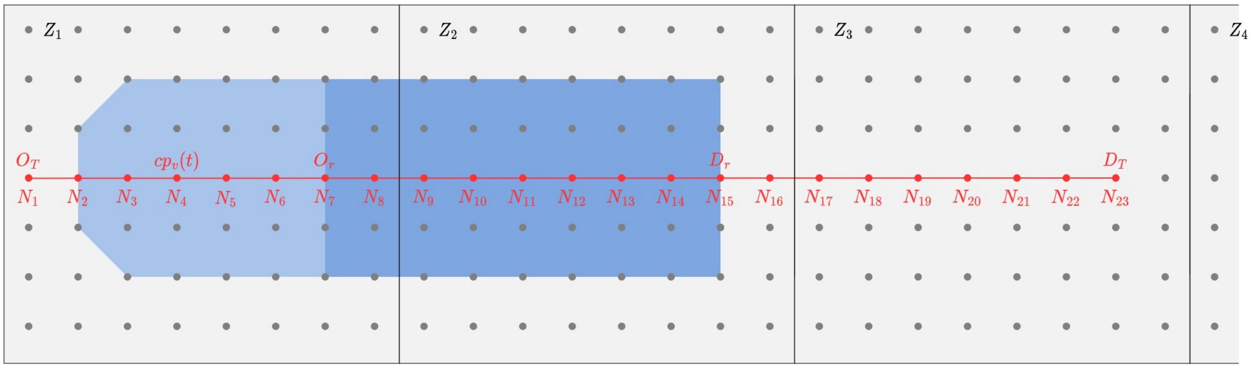


Figure 12: Distribution of the reachable origin nodes of a new request r'. All nodes in the blue region of Figure 10, are divided into two different groups to determine $\mathcal{N}_{\Omega}^{priorO_r}$ and $\mathcal{N}_{\Omega}^{postO_r}$ for request r which has its origin O_r at node 7. All nodes in the light blue area can be reached prior to O_r and are assigned to $\mathcal{N}_{\Omega}^{priorO_r}$. All nodes in the dark blue area can be reached posterior to O_r and are assigned to $\mathcal{N}_{\Omega}^{postO_r}$.

3.2.2 Demand prediction

A prediction about the ridesharing demand is used to provide a more reliable service. Therefore, the exponential smoothing (ES) model is implemented in this research to predict the demand. Exponential smoothing is a well-known forecasting method in time-series analysis, and a detailed description of exponential smoothing is provided in Section 2.5.5 about Forecasting methods & mathematical models. The exponential smoothing forecasting method is selected because of its simplicity while being able to provide sufficient predictions. A simple algorithm is selected because demand forecasting is not the main objective of this research. Thus, it is not necessary to implement complex/superior forecasting methods. As stated in Section 2.5.5, the most simple form of exponential smoothing to predict the demand Y at time t is formulated as

$$\hat{Y}_t = \alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1}, \tag{43}$$

where α represents the smoothing constant, with $\alpha \in [0,1]$, and Y_{t-1} represents the actual demand at time $t-1$.

The implemented measure to reduce unreliability uses the number of emerging requests at time t with their origin in zone z_O and their destination in zone z_D, denoted by $\lambda_{z_O,z_D}(t)$. The value of $\lambda_{z_O,z_D}(t)$ is predicted using the exponential smoothing method. Hence, parameter $\hat{Y}_{z_O,z_D}(t)$ is introduced, defining the predicted demand at time t with their origin in zone z_O and their destination in zone z_D, which is formulated as:

$$\hat{Y}_{z_O,z_D}(t) = \alpha Y_{z_O,z_D}(t-1) + (1-\alpha)\hat{Y}_{z_O,z_D}(t-1). \tag{44}$$

The demand highly fluctuates per minute; therefore, the system predicts the demand at time t over a time span of ten minutes. Furthermore, ES considers N timestamps for its predictions; hence, the same day of the week for the past ten weeks at time t are analyzed, such that $t-1$ is equal to the same time period a week before, $t-2$ is equal to the time period two weeks before, and so on. Thus, when for example the demand is predicted between 09:00 AM and 09:10 AM on Wednesday the 14th, then $t-1$ is equal to the time period from 09:00 AM to 09:10 AM on Wednesday the 7th.

As stated in Section 2.5.5, the smoothing factor α determines the smoothing rate of past observations. Values of α close to one produce a reduced smoothing effect and prioritize recent data values, while α values close to zero result in an increased smoothing effect with a less responsive behavior to current data values. No formal procedure exists to determine the value of α. Therefore, α is determined by trial-and-error while evaluating the mean squared error (MSE) of the predicted demand. As a result, an α value of 0.8 is obtained. Table 5 provides an overview of the implemented values of the exponential smoothing forecasting process.

Table 5: Implemented parameters for the exponential smoothing forecasting process.

Parameter	**Value**
Number of evaluated timestamps N	10
Exponential smoothing constant α	0.8
Analyzed time span [minutes]	10

3.2.3 Reduce unreliability

The demand prediction and the period shareability shadow are utilized to provide two parameters to each new user to reduce unreliability. These two parameters are the possibility of a negative change and the size of a negative change.

Possibility of a negative change
A request r receives travel time information directly after a match has been created between a vehicle and the request. This information is based on the current status of the trip and is called *the first announced travel times.* The first announced travel times are achieved if the vehicle does not change its itinerary from the moment request r has been assigned until the drop-off of request r. However, both waiting time and in-vehicle time of request r can change due to the addition of a new request r' into the trip, resulting in unreliability faced by request r. Hence, the possibility is provided whether a new request r' will be added to the trip that negatively influences the waiting time and in-vehicle time of request r. For both the waiting time and in-vehicle time, either one of the three following indication words: "low possibility", "medium possibility", or "high possibility" is provided to indicate the possibility that a new request r' will be added that negatively influences the waiting time and in-vehicle time. These indication words are provided instead of a percentage of the possibility as the theoretical background of unreliability in ridesharing, Section 2.4.8, illustrates that this is the most intelligible method for users to interpret. Moreover, these indication words are easy to analyze, which is useful.

The possibility of a negative change in waiting time and in-vehicle time is determined by comparing the shareability score to certain threshold values. The shareability score, denoted by $\lambda_r(t)$, defines the number of possible emerging requests that individually satisfy the PUDO constraints of the trip of request r. This shareability score $\lambda_r(t)$ is divided into two different parameters: (1) the shareability score of the trip during the waiting time of request r, denoted by $\lambda_r^w(t)$ (see Equation 39); and (2) the shareability score of the trip during the in-vehicle time of request r, denoted by $\lambda_r^{iv}(t)$ (see Equation 40). By comparing $\lambda_r^w(t)$ and $\lambda_r^{iv}(t)$ to set threshold values, the possibility of a negative change is determined. For example, if $\lambda_r^{iv}(t)$ is less than 1, the system informs the request that there is a low possibility that the in-vehicle time will increase.

The set threshold values are configured every 10 minutes to maintain an adequate prediction of the possibility of a negative change. This is done since the shareability score $\lambda_r(t)$ depends on the number of emerging requests, which strongly fluctuates over time. Therefore, every 10 minutes, the shareability scores of the completed requests over the last 10 minutes are analyzed to configure new threshold values.

The threshold values T_h are obtained by maximizing the score $\mathcal{S}(\mathcal{R}_{10}, T_h)$ of the threshold values T_h. The score function $\mathcal{S}(\mathcal{R}_{10}, T_h)$ considers the set of completed requests over the last ten minutes, denoted by $\mathcal{R}_{10}$, and the score of the threshold values T_h. The completed requests of the last ten minutes $\mathcal{R}_{10}$ are divided into three groups: the set of requests with a correct prediction $\mathcal{R}_{cp}(T_h)$ given the threshold values T_h, the set of requests with an incorrect prediction $\mathcal{R}_{ip}(T_h)$ given the threshold values T_h, and the set of requests with an indecisive prediction $\mathcal{R}_{mp}(T_h)$ given the threshold values T_h. For clarification, the set of requests with an indecisive prediction present the requests that received a *medium* possibility that the travel time will increase given the threshold values T_h. Given the set of completed requests over the last ten minutes $\mathcal{R}_{10}$ and the threshold values T_h, the score function is formulated as:

$$\mathcal{S}(\mathcal{R}_{10}, T_h) = \sum_{r \in \mathcal{R}_{cp}(T_h)} s_{cp} + \sum_{r \in \mathcal{R}_{ip}(T_h)} s_{ip} + \sum_{r \in \mathcal{R}_{mp}(T_h)} s_{mp}. \tag{45}$$

In this equation, s_{cp}, s_{ip}, and s_{mp} represent the score values of having a threshold value that resulted in a correct prediction, an incorrect prediction, and an indecisive (medium) prediction, respectively. The implemented score values are presented in Table 6.

Table 6: Implemented score values to compute the threshold values for the predicted possibility of a negative change.

Parameter	**Value**
Score of having a correct prediction s_{cp}	1
Score of having an incorrect prediction s_{ip}	-1
Score of having an indecisive prediction s_{mp}	0.1

Magnitude of a possible negative change

For each new request r, the magnitude of the extra waiting time and in-vehicle time is provided for whenever the case occurs that a new request r' will be added to the trip. Although the predicted possibility of a negative change might states that this possibility is low or medium, the system still provides each new request r a prediction of the possible waiting time and in-vehicle time increase. The provided prediction of the travel time increase contains an average, minimum, and maximum value. Moreover, the minimum and maximum values are set to follow a 95% confidence interval. To obtain a full range for all possible magnitudes of the waiting time increase, the assumption is made that the in-vehicle time will not increase. Similarly, to obtain the full range for all possible magnitudes of the in-vehicle time increase, the assumption is made that the waiting time will not increase. As a result, a 95% confidence interval for the waiting time and in-vehicle increase is obtained.

The travel time increase is determined by analyzing two scenarios: (1) only the pick-up of request r' is prior to the drop-off of request r; (2) both pick-up and drop-off of request r' are prior to the drop-off of request r. Both cases are assumed to be equally probable. For case 1, the travel time increase is determined by calculating the detour time for every reachable pick-up detour node of the period shareability shadow (i.e., all nodes in the blue regions of the period shareability shadow). For case 2, the travel time increase is determined by calculating the detour time between every two pairs of pick-up and drop-off detour nodes which are prior to the drop-off of r (i.e., all nodes in the blue and green regions, except the dark green region, of the period shareability shadow).

As a result, a batch of multiple detour times is obtained. From the batch of detour times, an average value and a standard deviation are obtained. Next, the minimum value and maximum value, which follow a 95% confidence interval, are calculated as follows:

$$[\min, \max] = [\mu - 2\sigma, \mu + 2\sigma], \tag{46}$$

where μ presents the population mean, and σ describes the standard deviation.

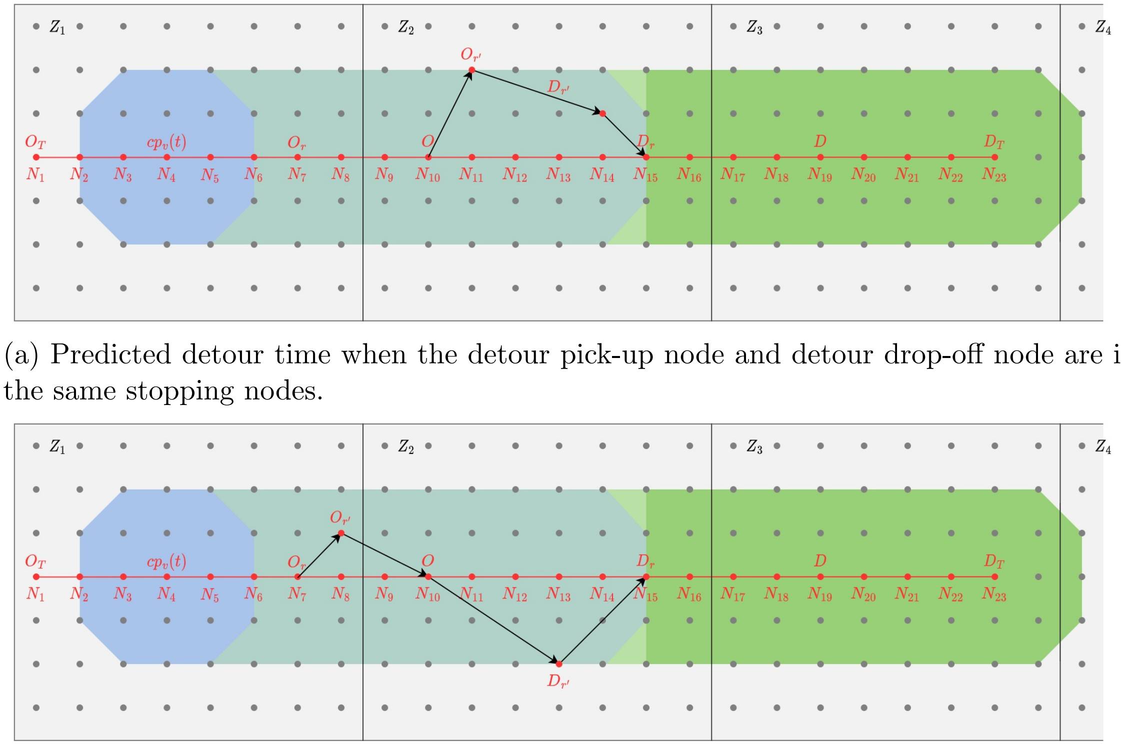
For each detour node of the period shareability graph is checked whether it will be reached prior to or posterior to the pick-up of request r. This check is made since a minimum, maximum, and average value of the travel time increase has to be determined for both the waiting time and the in-vehicle. If the detour node is reached prior to the pick-up node O_r of request r, the detour time is added to the array of the waiting time increase. If the detour node is reached posterior to the pick-up node O_r of request r, the detour time is added to the array of the in-vehicle time increase.

Figure 23 illustrates two examples of how the detour time is determined in the case that a request r' has its origin at $O_{r'}$ and its destination at $D_{r'}$. For Subfigure 13a, the detour time is equal to

$$\begin{aligned}\text{detour time} = {} & \text{travelTime}(N_{10}, O_{r'}) + \text{travelTime}(O_{r'}, D_{r'}) \\ & + \text{travelTime}(D_{r'}, N_{15}) - \text{travelTime}(N_{10}, N_{15}).\end{aligned} \tag{47}$$

For Subfigure 13b, the detour time is equal to

$$\begin{aligned}\text{detour time} = {} & \text{travelTime}(N_7, O_{r'}) + \text{travelTime}(O_{r'}, N_{10}) \\ & - \text{travelTime}(N_7, N_{10}) \\ & + \text{travelTime}(N_{10}, D_{r'}) + \text{travelTime}(D_{r'}, N_{15}) \\ & - \text{travelTime}(N_{10}, N_{15}).\end{aligned} \tag{48}$$



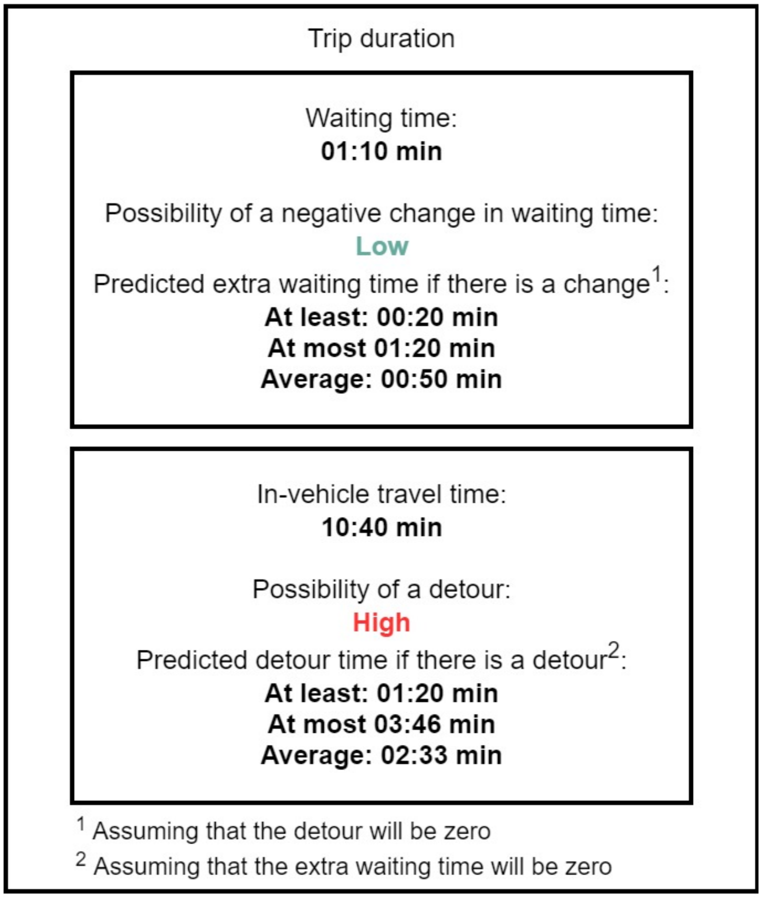
(a) Predicted detour time when the detour pick-up node and detour drop-off node are in between the same stopping nodes.

(b) Predicted detour time when the detour pick-up node and detour drop-off node are not in between the same stopping nodes.

Figure 13: Illustrations of the detour for a trip traveling in a straight line from its origin O_T to its final destination D_T, which has its stopping nodes (nodes at which the trip has a PU or DO) at N_7, N_{10}, N_{15}, N_{19}, and N_{23}. At time t the vehicle's current position is at $cp_v(t)$. For both subfigures, the detour is illustrated in the case that a request r' has its origin at $O_{r'}$ and its destination at $D_{r'}$.

Example of unreliability measure

To clarify how the discussed unreliability reduction measures are provided to the users, Figure 14 provides an example of how this is communicated to a user. The first square presents the waiting time and the possibility and magnitude of a negative change in waiting time. The second square presents the in-vehicle travel time and the possibility and magnitude of a possible detour. The travel times are communicated with a verbal presentation because this is the most intuitive and easy-to-understand method to communicate travel times (Tseng et al., 2009).



Trip duration

Waiting time:
01:10 min

Possibility of a negative change in waiting time:
Low
Predicted extra waiting time if there is a change[1]:
At least: 00:20 min
At most 01:20 min
Average: 00:50 min

In-vehicle travel time:
10:40 min

Possibility of a detour:
High
Predicted detour time if there is a detour[2]:
At least: 01:20 min
At most 03:46 min
Average: 02:33 min

[1] Assuming that the detour will be zero
[2] Assuming that the extra waiting time will be zero

Figure 14: Example of the provided unreliability measures.

3.2.4 Reliability baseline method

A simple baseline method is implemented to test the validity of the unreliability measures. Currently, no measures exist to reduce unreliability by predicting the possibility and magnitude of a possible change. Therefore, the implementation of a straightforward baseline method is provided in this section.

Each user is informed about the possibility of a negative change by randomly generating a “low”, “medium”, or “high” possibility for the indication of a possible waiting time and in-vehicle time increase. Additionally, the users are informed about the magnitude of a possible change. This is achieved in two steps. First, a range between zero and a maximum value is randomly generated. For the magnitude of the waiting time increase, this maximum value is equal to the maximum waiting time Ω of five minutes. For the magnitude of the in-vehicle time increase, this maximum value equals the maximum delay time Δ of ten minutes. Second, a mean is randomly generated such that the range falls within zero and the maximum value. As a result, a minimum, a maximum, and an average value of how much the waiting time and in-vehicle time will increase are obtained.

3.3 Experimental setup

The discussed models of Sections 3.1 and 3.2 are tested on a real dataset of requests using the public dataset of taxi trips of Manhattan, New York City, obtained from the *TLC Trip Record Data* (n.d.). Manhattan is one of the most densely populated areas in the world and there is a huge amount of governmental and private travel data publicly available on the Web, making it an interesting region to investigate. The assignment problem is solved over a subset of Manhattan, consisting of 4091 nodes and 9453 edges. Some of the requests might not start from a node in real-life but somewhere within an edge. For simplicity, it is assumed that each request starts from the closest node. This assumption is made as it is not worth it for this research to increase the level of detail regarding the implemented set of nodes. The same subset of nodes is also implemented by Alonso-Mora et al. (2017a); Fielbaum and Alonso-Mora (2020) because this dataset provides enough detail to run simulations (Fielbaum & Alonso-Mora, 2020). An edge defines whether a vehicle can travel from one node to the other; thus, it tells if one node is connected to the other. For each edge, it is assumed that each vehicle takes a specific time to travel over this edge. The edge travel times used by Alonso-Mora et al. (2017a); Fielbaum and Alonso-Mora (2020) are also implemented here. Using the Dijkstra Algorithm (an algorithm that finds the shortest path between a given node and all other nodes in a graph using the weights of the edges) the travel time between any pair of nodes is obtained.

The implemented method is analyzed over a time period from 8 AM to 11 AM on 31/05/2016. This time period is selected as the demand over this period is nonstationary, i.e., the mean and variance of the dataset depend on time, resulting in a trend in demand. Given a trend in demand, the implemented method is tested on its boundaries. When the method achieves sufficient performance, it is likely that a sufficient performance will also be achieved for a stationary demand period.

Four different scenarios are considered for testing the performance of the implemented method and for analyzing whether the results are not specific to one particular instance. The basic scenario consists of 2000 vehicles with a maximum capacity of four passengers, with the constraint that rescheduling requests is not allowed. As stated in Section 3.1.5, this constraint is implemented since allowing for reassignments of requests results to reduced performance of the unreliability predictions. The four alternative scenarios are: fewer vehicles (FV, 1000 vehicles of capacity 4 and no rescheduling allowed), smaller vehicles (SV, 2000 vehicles of capacity 3 and no rescheduling allowed), and vehicles for which rescheduling is allowed (RA, 2000 vehicles of capacity 4). The SV and FV scenarios are analyzed to validate the robustness of the basic scenario, while the RA scenario is analyzed for benchmark testing. Additionally, a baseline method with the same configurations of the basic scenario, i.e., 2000 vehicles of capacity 4 and no rescheduling, is considered to test the validity of the unreliability measure.

Given a vehicle capacity of four passengers, removing all requests with more than four users results in a total of 47482 requests over the three-hour period. Every request r contains an origin node O_r, a destination node D_r, an emerging time t_r, and the number of passengers. An overview of the implemented user parameter values is provided in Table 3 of Section 3.1.3. Furthermore, the start positions of all vehicles are generated randomly over Manhattan.

3.3.1 Zones

The implemented period shareability shadow and demand prediction model divide the city area into zones to make computations. Therefore, the clustered zones provided by the Taxi & Limousine Commission (TLC) of New York city are implemented, see Figure 6 of Section 2.5.4. An advantage of implementing the provided zones by the TLC is that the used dataset of taxi trips, which the TLC also provides, already contains information about the TLC-zones, meaning that each request contains an origin zone and a destination zone. Furthermore, the provided zones by the TLC are clustered based on geographic characteristics, which is useful for the implemented methods. Moreover, utilizing TLC zones does reduce computation time because the geographical coordinates do not have to be converted to zones. This is handy since the exponential smoothing forecasting method uses ten days of the past ten weeks of historical data to make predictions. For February, March, April, and May of 2016, over 48 million different requests exist. Converting all of the geographic coordinates to nodes takes the program several days.

4 Results & discussion

The results of the measurements proposed in Chapter 3 are provided and discussed in this chapter. As stated in Section 3.3, the four analyzed scenarios are: a basic scenario, a fewer vehicles (FV) scenario, a smaller vehicles (SV) scenario, and a scenario for which rescheduling is allowed (RA). For clarification, rescheduling means that requests can be reassigned to other vehicles after being assigned to a vehicle.

First, the quality-of-service indices not related to unreliability are analyzed in Section 4.1. Next, the quality-of-service indices related to unreliability are measured and discussed in Section 4.2. Finally, Section 4.3 discusses the results of the implemented solution to reduce unreliability, whereafter, an extensive analysis of the basic scenario is provided.

4.1 Quality-of-service indices not related to unreliability

Before analyzing the performance of the implemented method, the performance of the quality-of-service indices not related to unreliability are analyzed to understand how the different scenarios behave. Table 7 presents the performance of the quality-of-service indices not related to unreliability for the four different scenarios. The basic scenario has a rejection rate of 16.4%. As expected, the scenario with fewer vehicles (FV) has a significantly higher rejection rate than the basic scenario since fewer vehicles are available to serve the requests. Likewise, the smaller vehicles (SV) scenario has a slightly higher rejection rate than the basic scenario, as fewer vehicles can serve the requests due to their reduced capacity. The scenario for which rescheduling is allowed has a slightly lower rejection rate. This is also as expected since rescheduling requests enables the system to create more matches between vehicles and requests.

In terms of travel times perform the different scenarios also as expected. Compared to the basic scenario, the fewer vehicles (FV) and the smaller vehicles (SV) scenarios have fewer available vehicles in vicinity of a new request, resulting in longer average waiting times. Furthermore, the fewer vehicles (FV) scenario has slightly longer average in-vehicle times, average detours, and average delays as fewer vehicles are available, resulting in less efficient matches between vehicles and requests. Contrarily, the smaller vehicle (SV) scenario has slightly shorter average in-vehicle times, average detours, and average delays since fewer requests can be added to a trip, resulting in fewer detours. When rescheduling is allowed (RA), the average in-vehicle times, average detours, and average delays slightly improve as rescheduling enables the system to create more efficient matches.

In conclusion, the different scenarios behave as expected regarding rejection rates and travel times. Increasing the vehicle fleet size results in fewer rejections and shorter average trip durations, and the opposite happens when decreasing the vehicle fleet size. When rescheduling is not allowed, the same behavior occurs when the fleet size is increased. The quality-of-service indices only slightly deteriorate when reassignments are not allowed, which justifies the prohibition of allowing for reassignments. Furthermore, increasing the vehicle capacity reduces the rejection rate and average waiting time while it increases the average in-vehicle time, detour size, and delay time.

Table 7: Quality-of-service indices not related to unreliability for the different scenarios.

Measure	**Basic scenario**	**FV**	**SV**	**RA**
Rejection rate	16.4%	35.2%	19.2%	13.8%
Avg. Waiting time [min]	3.16	3.26	3.31	3.14
Avg. In-vehicle time [min]	13.1	13.2	12.7	13.0
Avg. Detour [min]	2.88	3.06	2.55	2.83
Avg. Delay [min]	6.04	6.32	5.86	5.97

The average travel times are longer compared to the provided values by Fielbaum and Alonso-Mora (2020) because the average number of requests per hour in this simulation (15827 requests/hour) is significantly higher than the average number of requests per hour (9805 requests/hour) used in the simulations of Fielbaum and Alonso-Mora. Consequently, more matches are feasible between vehicles and requests, resulting in longer travel times per trip.

4.2 Quality-of-service indices related to unreliability

Fielbaum and Alonso-Mora (2020) introduce several quantitative indices to measure unreliability in on-demand ridesharing systems. These quantitative indices are also used in this research to analyze unreliability, and an overview of these indices, formulated by Fielbaum and Alonso-Mora, is provided in Table 8.

Table 8: Notations used in this research to analyze quality-of-service indices related to unreliability. Obtained from the work by Fielbaum and Alonso-Mora (2020).

Symbol	**Meaning**
t_r	Time when request r is received/posed
O_r	Origin of request r
D_r	Destination of request r
$t_w(r)$	Real waiting time faced by request r
$t_w^1(r)$	First-announced waiting time for request
$t_v(r)$	Real in-vehicle time faced by request r
$t_v^1(r)$	First-announced in-vehicle time for request r
$t_d(r)$	Real detour faced by request r
$t_d^1(r)$	First-announced detour for request r
$D(r)$	Real delay faced by request r
$D_w^1(r)$	First-announced delay for request r
$\Delta t_w(r)$	Increase in waiting time for request r since its first announcement
$\Delta t_v(r)$	Increase in in-vehicle time for request r since its first announcement
$\Delta t_d(r)$	Increase in detour time for request r since its first announcement
$\Delta D(r)$	Increase in delay for request r since its first announcement

Table 9 provides an overview of the behavior of the different scenarios related to unreliability. The first row presents the possibility of facing any change during the trip. In the basic scenario, 44.6% of the completed requests face a change in their trip duration. Accordingly, almost half of the users face unreliability during their trip, strengthening the urgency of a method to reduce unreliability. As expected, the possibility of facing a change is even higher for the fewer vehicles (FV) scenario and the scenario for which rescheduling is not allowed (RA). By contrast, the requests of the smaller vehicles (SV) scenario have, as expected, a slightly lower possibility of facing a change.

Furthermore, every next four rows of Table 9 provide the same four measures to analyze unreliability in waiting time, detours, and delays, respectively. The first row of the measures about the waiting time presents the possibility of facing an increased waiting time ($P(\Delta t_w > 0)$). The second row illustrates the average extra waiting time of the whole system if a user faced a waiting time increase with respect to their first announced waiting time ($E(\Delta t_w | \Delta t_w > 0)$). Next, the average number of changes in waiting time is presented (Avg.

NC_{t_w}). At last, the maximum number of changes in waiting time faced by a single user is provided (Max. NC_{t_w}).

When analyzing Table 9, a remarkable conclusion emerges: the possibility of a negative change in waiting time is extremely small when rescheduling is not allowed. This is because of three reasons: (1) users spend a longer time in the vehicle than waiting for it, resulting in more detour changes than waiting time changes (Fielbaum & Alonso-Mora, 2020); (2) it is harder to insert a request before the pick-up of another request. Inserting a stop prior to the pick-up/drop-off of already assigned users has a higher negative influence on the score function that computes the optimal assignment than inserting a stop posterior to the pick-up/drop-off of already assigned users. Thus, new users have a higher likelihood of being inserted at the end of the trip schedule, resulting in fewer changes in waiting time; (3) when rescheduling is allowed, requests can be swapped between trips, affecting the waiting times of already assigned requests. This phenomenon does not occur when rescheduling is not allowed, resulting in fewer changes in waiting time.

When the possibility of a negative change in waiting time is minimal, the possibility and magnitude of facing an increased delay is nearly equal to the possibility of facing an increased detour. This can be observed in the scenarios when rescheduling is not allowed.

As stated by Fielbaum and Alonso-Mora (2020), increasing the number of vehicles results in more reliability, and the opposite happens with respect to the capacity of the vehicles. Hence, fleet size and unreliability are related because increasing the vehicle fleet increases the chance of assigning idle vehicles to new requests, keeping the same conditions for previously assigned requests of other vehicles. Furthermore, vehicle capacity and unreliability are related since the passengers of vehicles that reach their maximum capacity will face no new changes, which is more likely the happen with small vehicles (Fielbaum & Alonso-Mora, 2020). This behavior can also be observed in Table 9.

At last, it can be observed in Table 9 that the system becomes more unreliable when rescheduling is allowed. This is logical because reassigning requests results in more changes and, consequently, more significant variations in trip durations.

Table 9: Quality-of-service indices related to unreliability for the different scenarios.

Measure	**Basic scenario**	**FV**	**SV**	**RA**
Completed requests that face changes	44.6%	45.7%	40.2%	61.6%
$P(\Delta t_w > 0)$	0.58%	0.54%	0.41%	12.9%
$E(\Delta t_w \vert \Delta t_w > 0)$ [min]	0.74	0.70	0.70	1.02
Avg. NC_{t_w}	0.0058	0.0055	0.0041	0.33
Max. NC_{t_w}	2	2	1	4
$P(\Delta t_d > 0)$	44.4%	45.5%	40.0%	45.9%
$E(\Delta t_d \vert \Delta t_w > 0)$ [min]	2.25	2.14	2.30	2.43
Avg. NC_{t_d}	0.60	0.62	0.51	1.10
Max. NC_{t_d}	6	5	4	11
$P(\Delta D > 0)$	44.6%	45.7%	40.2%	48.2%
$E(\Delta D \vert \Delta D > 0)$ [min]	2.25	2.13	2.29	2.35
Avg. NC_D	0.60	0.62	0.51	1.18
Max. NC_D	6	5	4	11

4.3 Performance implemented method

This section provides and discusses the results of the implemented method to reduce unreliability. First, the performance of the four different scenarios is compared. After that, a detailed analysis of the basic scenario is provided.

4.3.1 Comparison different scenarios

Each user receives two indices to reduce unreliability. These indices are:

1. The predicted possibility that the waiting time and in-vehicle time will increase. This is achieved by providing each user a “high”, “medium”, or “low” possibility for a waiting time and in-vehicle time increase.

2. The magnitude of the travel time increase in the case that a new request will be added to the trip. This is indicated by providing a minimum, maximum, and average value of how much the waiting time and in-vehicle time increase. Furthermore, the minimum and maximum values are set to follow a 95% confidence interval.

The performance of the provided indices to reduce unreliability for the different scenarios is displayed in Table 10.

Table 10: Performance of the unreliability reduction measures for the different scenarios. The first three rows present the percentage of requests that received a correct, indecisive, and incorrect prediction for the possibility of a negative change in waiting time, respectively. Note that “indecisive” means that a request received a “medium” possibility for a negative change. Similarly, the following three rows present the percentage of requests that received a correct, indecisive, and incorrect prediction for the possibility of a negative change in in-vehicle time, respectively. The last row illustrates the percentage of requests that have their real travel time increase in range of the provided 95% confidence interval.

Measure	**Basic scenario**	**FV**	**SV**	**RA**
Possibility neg. change waiting time correct	97.8%	97.7%	97.8%	85.7%
Possibility neg. change waiting time indecisive	0.39%	0.47%	0.36%	0.35%
Possibility neg. change waiting time incorrect	1.80%	1.87%	1.81%	14.0%
Possibility neg. change in-vehicle time correct	60.4%	63.2%	53.4%	54.7%
Possibility neg. change in-vehicle time indecisive	15.4%	13.1%	21.8%	20.3%
Possibility neg. change in-vehicle time incorrect	24.2%	23.7%	24.8%	25.0%
Actual extra travel time in range of 95% CI	94.9%	95.2%	94.6%	69.2%

What directly stands out in Table 10 is that for the scenarios for which rescheduling is not allowed, nearly 100% of the requests receive a correctly predicted possibility for a negative change in waiting time. This is because the number of requests that face a longer waiting time (their real waiting time is longer than their first-announced value) is insignificant, see

Table 7. As a result, the method for optimizing the threshold values (Equation 45), used to determine the possibility of a negative change, raises the threshold values to such an extent that nearly all requests receive a low possibility for a negative change in waiting time.

Besides, it can be observed in Table 10 that the “smaller vehicles” (SV) scenario has an inferior performance in the predicted possibility of a negative change in in-vehicle time compared to the basic scenario. This is because fewer requests can be added to the trip in the SV scenario. Consequently, shorter trips are created, resulting in smaller differences in shareability score values $\lambda_r^{iv}(t)$ between the requests that face no change and the requests that face a negative change in in-vehicle time. This smaller difference in $\lambda_r^{iv}(t)$ is because of how $\lambda_r^{iv}(t)$ is formulated: it computes the number of possible emerging requests that could be added individually to the trip by considering trip duration, among other things. Due to the shorter average trip durations in the SV scenario, there is a higher overlap in the shareability score values $\lambda_r^{iv}(t)$ between the different groups of requests. A detailed analysis of the shareability score values is provided in Section 4.3.2. Moreover, Section 4.3.2 provides a detailed description of how this prediction can be improved.

Furthermore, it can be observed that the scenario for which rescheduling is allowed (RA) has an inferior performance in all measures compared to the basic scenario. This is as expected because the shareability shadow cannot include rescheduled requests. As a result, trips are created that deviate in route and duration compared to what has been forecasted with the period shareability shadow, resulting in less accurate predictions.

The last row of Table 10 presents the performance of the predicted travel time increase. Analyzing the scenarios for which rescheduling is not allowed, when aiming to provide a 95% confidence interval of the travel time increase, also roughly 95% of the requests have their real travel time increase in range of the provided interval. Therefore, the implemented method can sufficiently predict the travel time increase by providing a 95% confidence interval. On the other hand, this method cannot provide a 95% confidence interval of the travel time increase for the scenario when rescheduling is allowed (RA). For the RA scenario, only 69.2% of the requests have their real travel time increase in the range of the predicted interval. As expected, the method has an inferior performance in the RA scenario as the shareability shadow cannot include rescheduled requests. Consequently, the travel time increase might differ from the predicted values, resulting in a worse performance of the predicted travel time increase.

Because the quality-of-service indices not related to unreliability only slightly reduce when rescheduling is not allowed, it is worth not allowing for rescheduling if the operators aim to improve unreliability, as not allowing for reschedules achieves a higher performance in terms of unreliability predictions.

4.3.2 Detailed results and discussion of the basic scenario

The different scenarios for which rescheduling is not allowed obtained very similar results in terms of unreliability predictions (see Table 10). Therefore, a detailed analysis of the performance of the basic scenario is provided in this section. Figure 15 illustrates the different groups of requests for this scenario. The first bar presents the number of rejected requests, and the second bar presents the number of requests that face no change during their trip, meaning that their real travel time was equal to their first-announced values. Additionally, 16443 requests face a worse travel time, meaning that their real travel time is longer than their first-announced values. At last, a bar is included to present the assigned requests that are not yet completed when the simulation ended. These uncompleted requests exist since the simulation ended exactly after three ours in simulation time and not any moment later when all assigned requests would be completed. An end time of three hours is maintained as ending the simulation at any moment later would prevent the concerning requests from being exposed to the unreliability characteristics of the service because no new requests are added to the trips. This could affect the performance analysis; therefore, these requests are excluded from further analysis. As illustrated in Tables 7 and 9, 44.6% of the completed requests faced a negative change in travel time, and 16.4% of the requests got rejected.

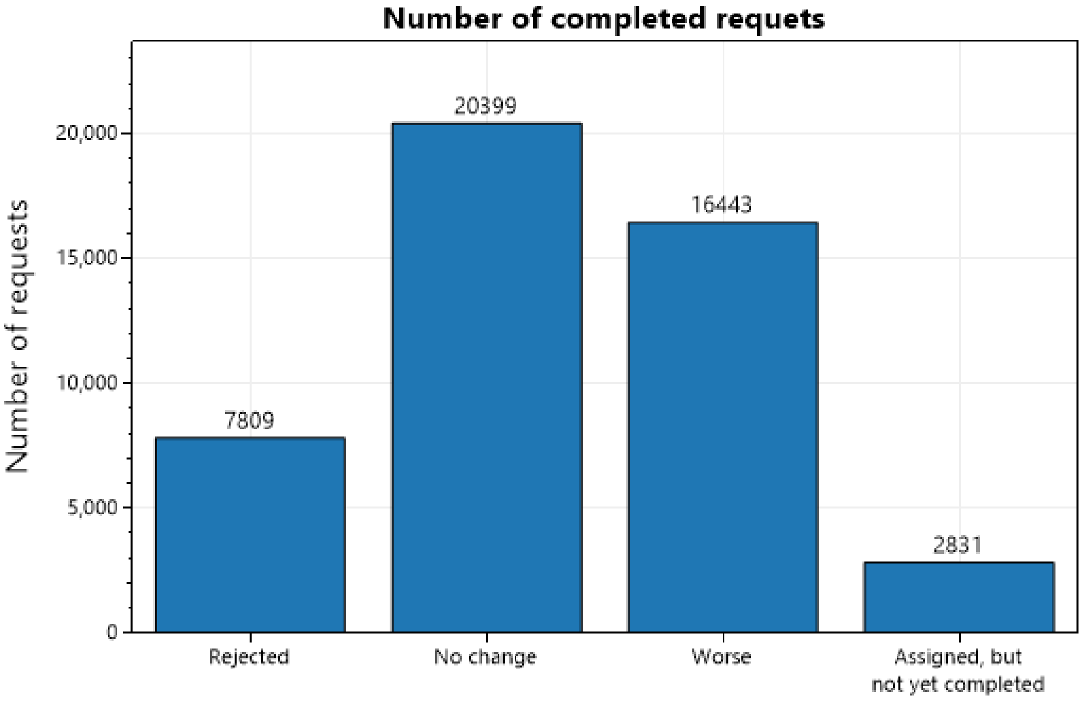


Figure 15: Distribution of the requests. No change: real travel time is the same as the first announced travel time. Better: real travel time is better than the first announced travel time. Worse: real travel time is worse than the first announced travel time.

This research is interested in requests that face a negative change in travel time compared to their first-announced travel times to reduce their unreliability. Therefore, Figure 16 illustrates the distribution of requests with such a negative change. The completed requests can be divided into four different groups:

- requests that face no change in travel time, called NC;
- requests that both face a worse waiting time and in-vehicle time compared to their first-announced values, called W;
- requests that only face a worse waiting time compared to their first-announced values, called WW;
- requests that only face a worse in-vehicle time compared to their first-announced values, called WIV.

As discussed in Section 4.2, it is most likely when rescheduling is not allowed that a request faces a negative change in in-vehicle time if it faces any change. Now that it is clear how the different groups of requests are distributed during this simulation, an analysis is provided in the following sections of the performance of the implemented unreliability measures.

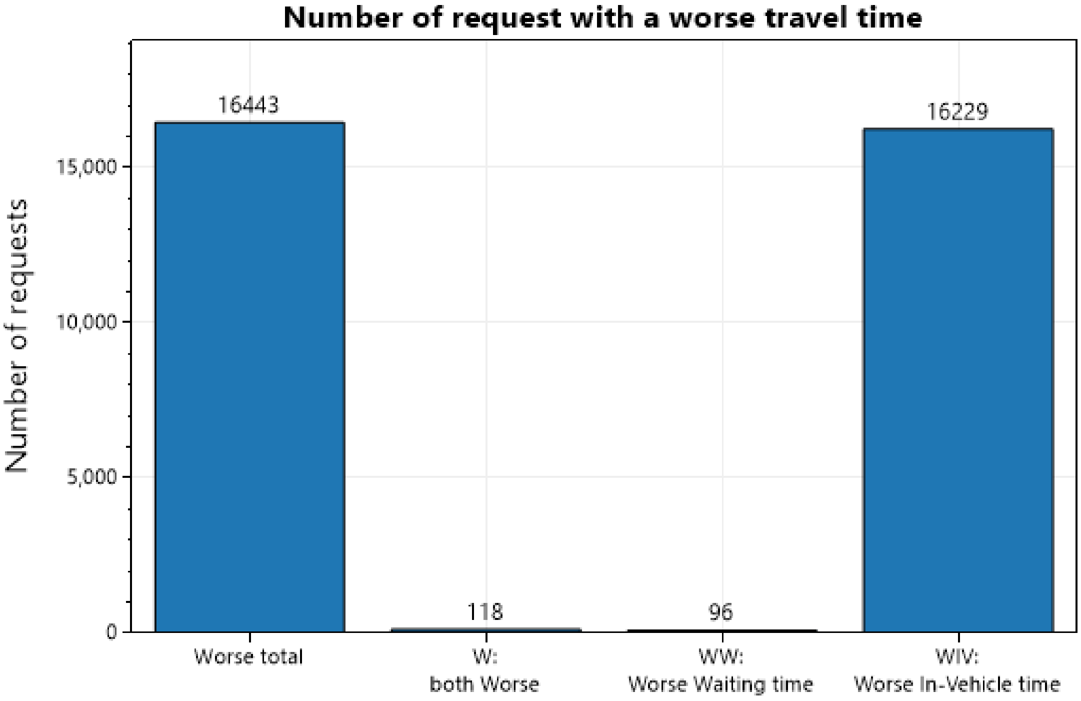


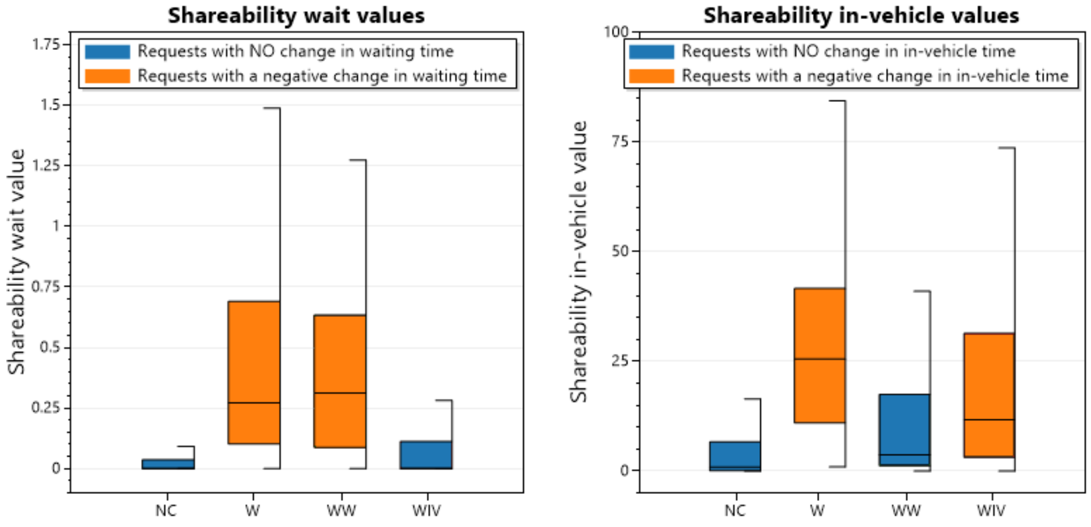
Figure 16: Distribution of the requests which face a worse travel time, meaning their real travel time is longer than the first announced value. W: set of requests that both face a worse waiting time and in-vehicle time compared to their first announced values. WW: set of requests that only face a worse waiting time compared to their first announced values. WIV: set of requests that only face a worse in-vehicle time compared to their first announced values.

Possibility of a negative change

The first provided parameter to each new user to reduce unreliability is the predicted possibility of a negative change in waiting time and in-vehicle time. These possibilities are determined by comparing the shareability score $\lambda_r(t)$ to the threshold values. Before analyzing the performance of the predictions, the values of $\lambda_r(t)$ are analyzed. Figure 17 presents the box-plots of the shareability scores of the waiting time and in-vehicle time for the different groups of completed requests. Additionally, the box-plots of the shareability scores of the waiting time and in-vehicle time are grouped in Figure 18 such that the shareability scores of the requests that face a negative change (orange) can be compared to the requests that do not face a negative change (blue).

The box-plots of the shareability wait scores $\lambda_r^w(t)$ are presented in the left figures, Figures 17a and 18a. The predicted possibility of a negative change in waiting time is perfect if the $\lambda_r^w(t)$ values of the NC and WIV groups do not overlap with the values of the W and WW groups. This is because the NC and WIV groups do not face a change in waiting time, while the W and WW groups do face a change in waiting time. Furthermore, it can be observed that the average values of the requests that face a negative change in waiting time (orange) are higher than the average values of the requests that do not face a change (blue). However, the lower quartiles of the W and WW groups (orange) overlap with the box-plots of the NC and WIV groups (blue). This overlap will result in false-negatives and false-positives of the predicted waiting time changes. An explanation for this overlap is provided later on.

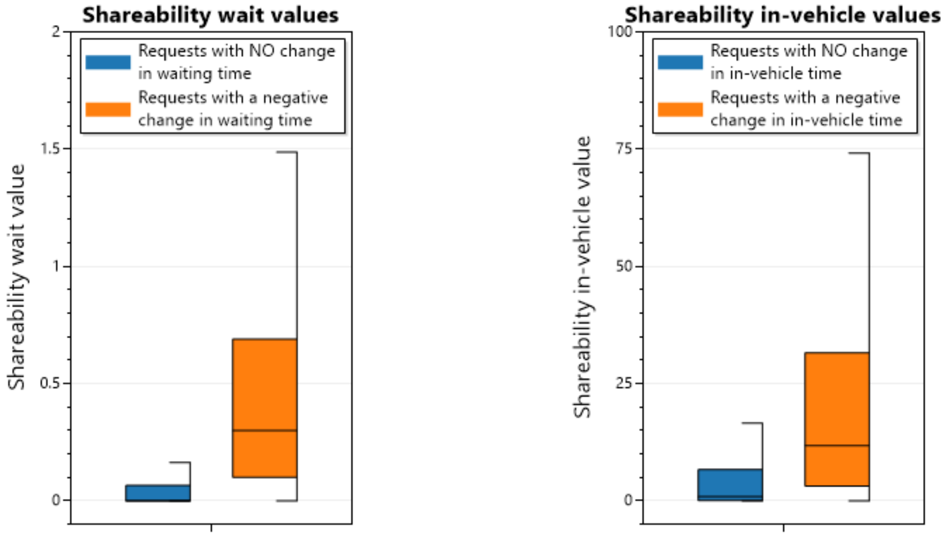
The box-plots of the shareability in-vehicle scores $\lambda_r^{iv}(t)$ are presented in Figures 17b and 18b. The predicted possibility of a negative change in in-vehicle time is perfect if the $\lambda_r^{iv}(t)$ values of the NC and WW groups do not overlap with the values of the W and WIV groups, as the NC and WW groups do not face a change in in-vehicle time, while the W and WIV groups do face a change in in-vehicle time. As desired, the average values of the requests that face a negative change in in-vehicle time (orange) are higher than the average values of the requests that do not face a change (blue). However, there is an undesired overlap between the two. Unfortunately, this overlap is even more significant than the undesired overlap in the box-plots about the waiting time. Luckily, an adequate prediction of the possibility of a negative change in in-vehicle time can still be obtained, which will be illustrated in the next sections.



(a) Shareability wait scores $\lambda_r^w(t)$.

(b) Shareability in-vehicle scores $\lambda_r^{iv}(t)$.

Figure 17: Box-plots of the shareability wait scores $\lambda_r^w(t)$ (left) and shareability in-vehicle scores $\lambda_r^{iv}(t)$ (right) for the different groups of completed requests. In the ultimate scenario, the box-plots of the shareability wait scores (left figure) of the NC and WIV groups do not overlap with the box-plots of the W and WW groups. For the shareability in vehicle scores (right figure), in the ultimate scenario, the box-plots of the NC and WW groups do not overlap with the box-plots of the W and WIV groups.



(a) Shareability wait scores $\lambda_r^w(t)$.

(b) Shareability in-vehicle scores $\lambda_r^{iv}(t)$.

Figure 18: Box-plots of the shareability wait scores $\lambda_r^w(t)$ (left) and shareability in-vehicle scores $\lambda_r^{iv}(t)$ (right) for the completed requests. In the ultimate scenario, there is no overlap between the orange and blue box-plots.

An undesired overlap in the $\lambda_r^w(t)$ and $\lambda_r^{iv}(t)$ values exists between the requests that face no change and the requests that do face a change because the mathematical equation to compute these values is not perfect for making this distinction between facing a change and not facing a change. This can be explained by several reasons:

1. At the moment, the shareability score $\lambda_r(t)$ predicts the number of possible emerging requests r' that individually satisfy the trip constraints and negatively influence the trip duration of r. However, an improved measure is to predict the number of requests that will be added to the trip and which simultaneously negatively influence on the trip duration of request r. To predict the number of requests that will be added to the trip, several indices have to be included, for example, the number of vehicles in vicinity of the trip should be included because when more vehicles are in vicinity, it is less likely that a new request r' will be added. Additionally, the vehicle's maximum capacity could be included because a higher capacity results in a higher possibility of adding new requests.

2. The shareability score $\lambda_r(t)$ considers the prediction of the number of emerging requests per zone the trip travels through. The implemented zones may not have the optimal shapes and sizes to cluster the requests of each zone. This has an impact on the prediction.

3. The predicted demand in each zone assumes that the demand is distributed uniformly over the space in the zone, which is not the case in reality. Hence, the characteristics of each zone should be evaluated to obtain improved predictions about the demand.

4. The demand in each zone is predicted with a simple time-series forecasting method named exponential smoothing. Improved forecasting methods such as ARIMA models or deep learning approaches could be implemented to improve demand predictions. Moreover, the characteristics of the city should be considered because they strongly influence the demand, as stated in Section 2.5.2.

5. Unavoidable randomness exists in the ridesharing process. Even when all previously motioned measures are implemented, the predictions will still not be perfect because of the involved randomness, e.g., randomness in demand and randomness in the assignments when multiple assignments exist with the same cost.

In conclusion, several measures can be applied to improve the prediction of whether a new request r' will be added to the trip, leaving room for future research.

The performance of the predicted possibility of a negative change in waiting and in-vehicle time for the different groups of completed requests is presented in Figures 19 and 21, respectively. In these figures, three colors are utilized to present the predicted possibility of a negative change. Green implies that a low predicted possibility of a negative change in travel time is provided to the requests, yellow implies a medium possibility, and orange implies that a high possibility of a negative change in travel time is provided.

The predicted possibility of a negative change in waiting time, Figure 19, is perfect if the NC and WIV groups are completely green, since these groups do not face a negative change in waiting time. Likewise, in the perfect scenario, the W and WW groups are completely orange, as these groups do face a negative change in waiting time. The function that optimizes the threshold values, Equation 45, sets the thresholds for a negative change in waiting time significantly high because the groups that face no change in waiting time (NC and WIV) are significantly larger than the groups that do face a negative change in waiting time (W and WW). Consequently, nearly all requests are informed that there is a low possibility of a negative change in waiting time, which is displayed by a green color, resulting in the best the overall performance of the predictions.

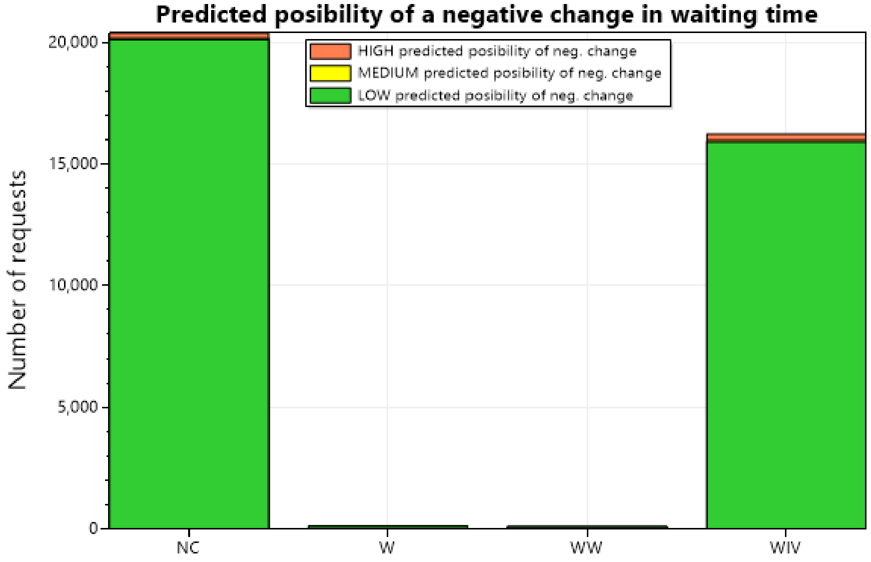


Figure 19: The predicted possibility of a negative change in waiting time for the different groups of completed requests. NC: set of requests that face no change. W: set of requests that both face a worse waiting time and in-vehicle time. WW: set of requests that face a worse waiting time. WIV: set of requests that face a worse in-vehicle time. Green implies that a low predicted possibility of a negative change in waiting time is provided to the requests, yellow implies that a medium possibility is provided, and orange implies that a high predicted possibility of a negative in waiting time is provided to the requests. If the method works perfectly, the NC and WIV groups are completely green, while the W and WW groups are completely orange.

Although the W and WW groups are insignificantly small and have an insignificant impact on the computation of the threshold values used to determine the possibilities of a negative change, Figure 20 presents the predicted possibilities of a negative change in waiting time for the W and WW groups. It can be observed that nearly all requests in these groups are informed that there is a low possibility of a negative change in waiting time (green area). By contrast, they should be informed that there is a high possibility of a negative change in waiting time as they will face a negative change in waiting time. As stated before, this is because the system optimizes the overall performance of the threshold values. Since these two groups are insignificantly small, they have a small influence on these computations.

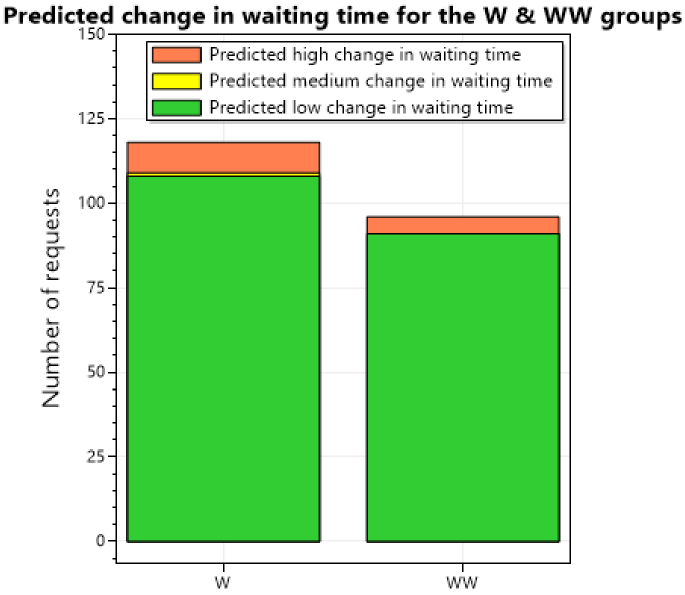


Figure 20: Predicted possibility of a negative change in waiting time for the W and WW groups. W: set of requests that both face a worse waiting time and in-vehicle time. WW: set of requests that only face a worse waiting time. Green implies that a low predicted possibility of a negative change in waiting time is provided to the requests, yellow implies that a medium possibility is provided, and orange implies that a high predicted possibility of a negative in waiting time is provided to the requests. The W and WW groups are completely orange if the method works perfectly.

The solution also informs the users about the possibility of a negative change in in-vehicle time. The predicted possibilities of a negative change in in-vehicle time are presented in Figure 21. The predicted possibility of a negative change in in-vehicle time is perfect when the NC and WW groups are completely green while the W and WIV groups are completely orange. As indicated in Table 10, 60.4% of the requests receive a correct prediction about the possibility of a negative change in in-vehicle time, 15.4 % receive an indecisive prediction (yellow areas in Figure 21), and 24.2% receive an incorrect prediction. Because there is an undesired partial overlap in the shareability in-vehicle score values $\lambda_r^{iv}(t)$ between the NC and WIV groups, illustrated in Figure 17b, the performance of the predicted possibility of a negative change in in-vehicle time is suboptimal. This undesired overlap is even larger than the undesired overlap in the shareability wait score values, Figure 17a, resulting in inferior predictions of a change in in-vehicle time compared to predictions of a change in waiting time. Still, the method obtains respectable predictions to reduce unreliability as less than a fourth of the requests receive an incorrect prediction about the possibility of a negative change in in-vehicle time.

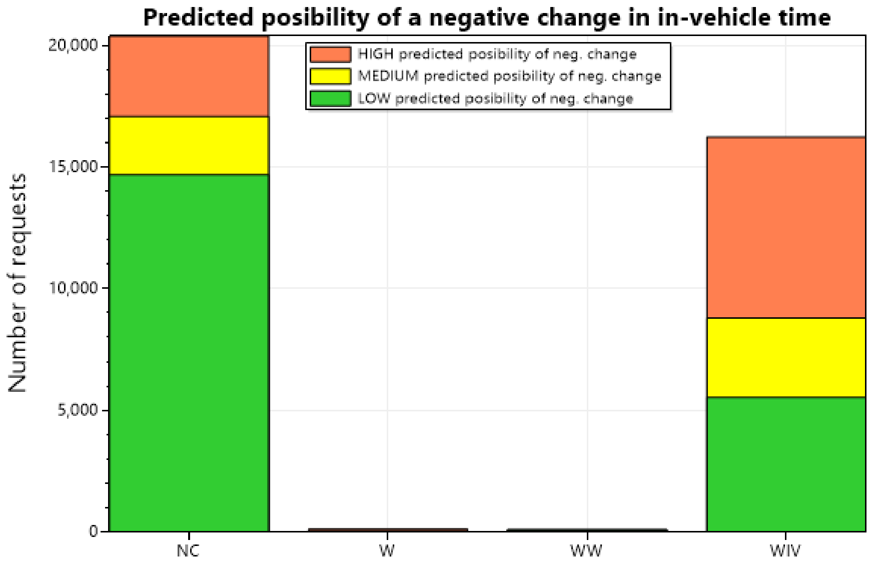


Figure 21: Predicted possibility of a negative change in in-vehicle time for the different groups of completed requests. NC: set of requests that face no change. W: set of requests that both face a worse waiting time and in-vehicle time. WW: set of requests that face a worse waiting time. WIV: set of requests that face a worse in-vehicle time. Green implies that a low predicted possibility of a negative change in in-vehicle time is provided to the requests, yellow implies that a medium possibility is provided, and orange implies that a high predicted possibility of a negative in in-vehicle time is provided to the requests. If the method works perfectly, the NC and WW groups should be completely green, while the W and WIV groups should be completely orange.

The sizes of the W and WW groups are insignificant compared to the NC and WIV groups. Still, an enlarged graph of the predictions of a negative change in in-vehicle time is presented in Figure 22 for the W and WW groups. The W group faces a negative change in in-vehicle time, and the WW group does not. Therefore, in the ultimate scenario, the W group is completely orange, and the WW group is completely green. It can be observed in Figure 22 that the predictions for these groups perform quite well. Only 16.1% of the requests of the W group receive an incorrect prediction (green area), and 28.1% of the requests of the WW group receive an incorrect prediction (orange area).

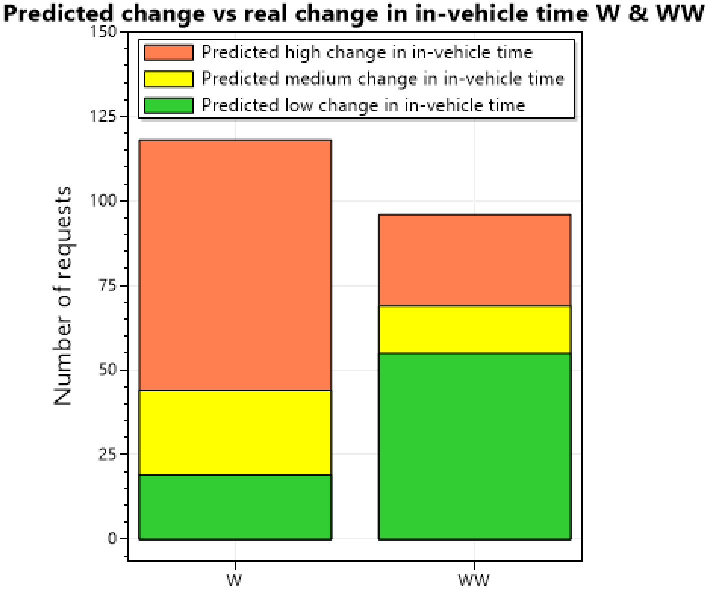


Figure 22: Predicted possibility of a negative change in in-vehicle time for the W and WW groups. W: set of requests that both face a worse waiting time and in-vehicle time. WW: set of requests that only face a worse waiting time. Green implies that a low predicted possibility of a negative change in waiting time is provided to the requests, yellow implies that a medium possibility is provided, and orange implies that a high predicted possibility of a negative in waiting time is provided to the requests. If the method works perfectly, the W group is completely orange and the WW group is completely green.

Overall, it can be concluded that the method obtains respectable predictions regarding the possibility of a negative change in travel time. Therefore, the users obtain an improved understanding of what they can expect from their trip, resulting in a reduced unreliability experience.

Predicted size of a negative change

The second provided parameter to each new user to reduce unreliability is the predicted magnitude of a negative change for whenever the case occurs that a new request will be added to the trip. Hence, a prediction of the magnitude of a possible waiting time increase and in-vehicle time increase is provided to each new user. The increase is indicated by providing a minimum value, a maximum value, and an average value of the travel time increase. Moreover, the minimum and maximum values are set to follow a 95% confidence interval. As indicated in Table 10, 94.9% of the requests of the basic scenario face a real travel time increase with its magnitude in range of the provided interval. Therefore, the method behaves as desired and is very well capable of predicting the magnitude of the travel time increase within a 95% confidence interval.

Figure 23 illustrates the average values of the real waiting and in-vehicle time increase for the different groups of requests that faced a negative change and compares it to the average values of the respective predicted magnitudes. It can be observed that when only a negative change in waiting time (WW) or in-vehicle time (WIV) occurs that the average values of the predicted magnitudes are nearly equal to the average values of the real increase. Besides, for the group of requests that faced a negative change in both waiting and in-vehicle time (W) it can be observed that the predicted magnitudes of a negative change in in-vehicle time overestimate, on average, the real magnitudes. This overestimation occurs because the implemented method assumes that the waiting time increase will be zero to obtain a full range for the minimum and maximum value of the magnitude of the in-vehicle time increase. Because these requests also face a negative change in waiting time due to a detour, their in-vehicle time increase will be less since there is less space for extra delays, resulting in an overestimation of the extra in-vehicle time. Additionally, the overestimation of the waiting time for the W group is significantly smaller than the in-vehicle time overestimation. This is because the ridesharing tolerance in the waiting time is considerably smaller, resulting in smaller overestimations.

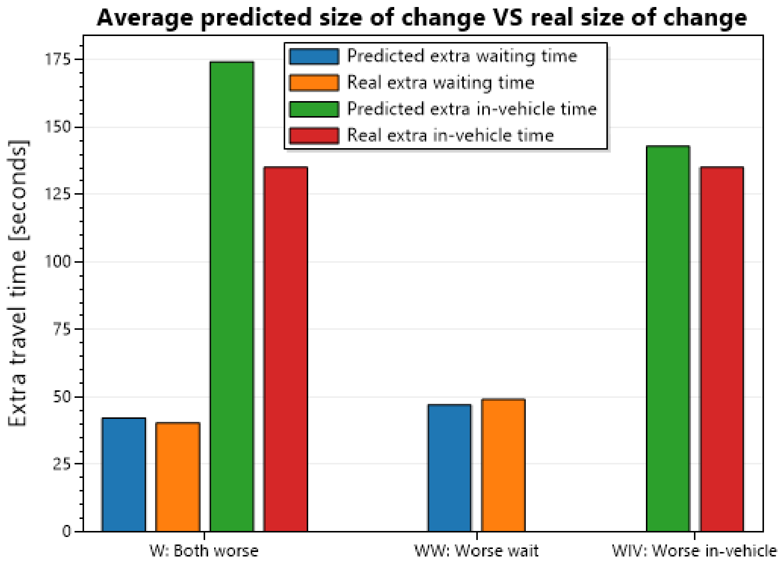


Figure 23: Average values of the predicted magnitudes of a negative change VS average values of the real magnitudes of a negative change.

It is clear that the method obtains correct predictions (it provides a 95% confidence interval of the travel time increase). However, it is crucial to analyze whether the method provides a reasonable width of the provided ranges. For clarification, the width of a range presents the difference between the minimum and maximum value of the predicted magnitude of the travel time increase. Furthermore, the minimum value of the provided range might not be zero. To obtain an improved understanding of the widths of the ranges, Figure 24 presents the widths of the ranges of the predicted waiting time increase for the requests that face a negative change in waiting time. By examining the predicted ranges of the waiting time increase, it is most likely that the requests receive a width of less than a minute. Receiving a width of one minute between the minimum and maximum value of the waiting time increase is a respectable indication for the users to reduce their unreliability regarding their waiting time. The widths of the provided ranges of the waiting time increase have a mean of 43 seconds and a standard deviation of 41 seconds.

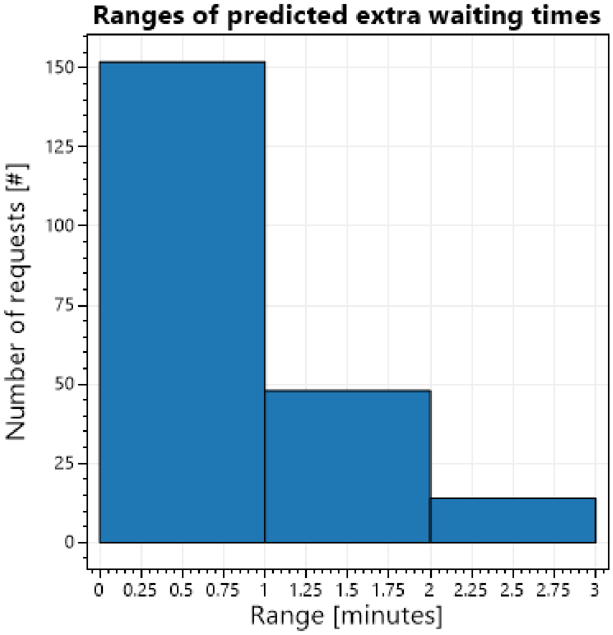


Figure 24: Widths of the provided ranges of the predicted magnitude for a negative change in waiting time for the requests which face a worse waiting time. The provided ranges follow a 95% confidence interval. Range = $[\mu - 2\sigma, \mu + 2\sigma]$.

The widths of the provided ranges for the predicted in-vehicle time increase for the requests that faced a negative change in in-vehicle time are presented in Figure 25. Note that the predicted in-vehicle time increase assumes that there will be no increase in waiting time to obtain a full range of the predicted in-vehicle time increase. In Figure 25, it can be observed that 74.2% of the requests receive a width of less than five minutes for the in-vehicle time increase. Furthermore, 4.6% of the requests receive a width of more than seven minutes. The widths of the provided ranges of the in-vehicle time increase have a mean of 3.7 minutes and a standard deviation of 1.9 minutes. Unfortunately, a small number of requests (0.11%) receive a width of more than nine minutes. Providing such a long width leaves the concerning requests still in uncertainty as they do not know precisely how much their travel time will increase. Luckily, the number of requests that receive such a long width is inferior. Most requests receive such long widths because they are individual requests in a trip or because they have a short travel time from origin to destination. As a result, their trip can face large detours without violating their delay constraints. Therefore, some of these requests do also face a real detour of nine minutes.

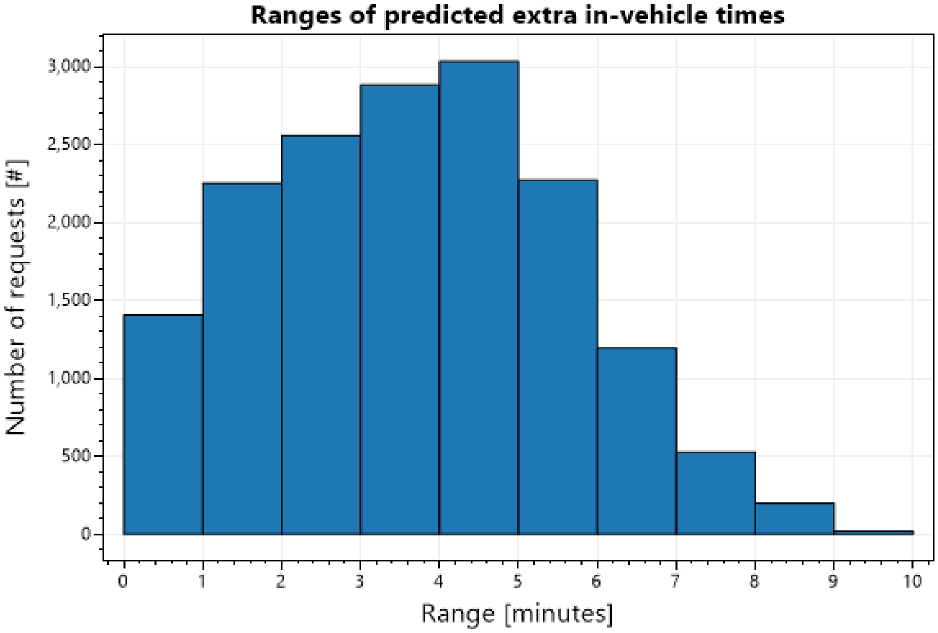


Figure 25: Widths of the provided ranges of the predicted size of a negative change in in-vehicle travel time for the requests which face a worse in-vehicle travel time. The provided ranges follow a 95% confidence interval. Range = $[\mu - 2\sigma, \mu + 2\sigma]$.

Because 4.6% of the requests receive a width of more than seven minutes for a possible detour, it is interesting to compare the widths to the total travel times of the requests. Note that "width" implies the difference between the minimum and maximum value of the provided range. Figure 26 compares the trip durations to the width of the ranges for the in-vehicle time increase. Because there are too many requests (16347) to display individually, the requests are grouped by trip durations with a range of one minute. The x-axis presents the trip durations with a range of one minute, and the y-axis presents the ratio between the trip duration and the average value of the widths of the ranges for the predicted in-vehicle time increase. The ratio of the width of the ranges to the trip durations has a mean value of 0.21 and a standard deviation of 0.16. A ratio of 0.21 implies that a request with a trip duration of 10 minutes receives a range with a width of 2.1 minutes for the predicted detour time.

It can be observed that passengers of shorter trips receive, on average, slightly longer ranges for the predicted in-vehicle time increase than longer trips. This seems counter-intuitive as longer trips are expected to have a higher possibility of receiving new users. This is also true, but that does not imply that users of longer trips are more likely to receive longer ranges for the predicted travel time increase. On average, users of shorter trips receive a longer range for the predicted travel time increase because shorter trips contain, on average, less requests. Users of trips with less requests face smaller delays because less detours are made to serve other requests. Consequently, shorter trips have more room to make detours. Therefore, the system is allowed to make long detours without violating the maximum delay of these users.

Additionally, shorter trips are more likely to occur in the center of the network. From the network center, the proportion of far-away nodes that are reachable is larger than on the boundaries of the network. As a result, the unreliability solution provides a prediction for the magnitude of the in-vehicle time increase with a large deviation between the minimum value and the maximum value as detours of various durations are possible. Therefore, it can be concluded that the implemented method behaves properly.

Furthermore, the total number of completed requests with a trip duration of less than six minutes is minor (5.1%), which can be observed in Figure 27. Moreover, the number of requests that receive long ranges of more than seven minutes is only 4.6%. Thus, receiving such a long range has a very low possibility.

In conclusion, the method obtains respectable predictions regarding the possibility of a negative change and successfully provides a 95% confidence interval for the magnitude of a possible delay. Regarding the magnitude of a possible delay, 74.2% of the requests receive a range of less than five minutes between the minimum and maximum value. Therefore, the users obtain an improved understanding of what they can expect from their trip, resulting in a reduced unreliability experience.

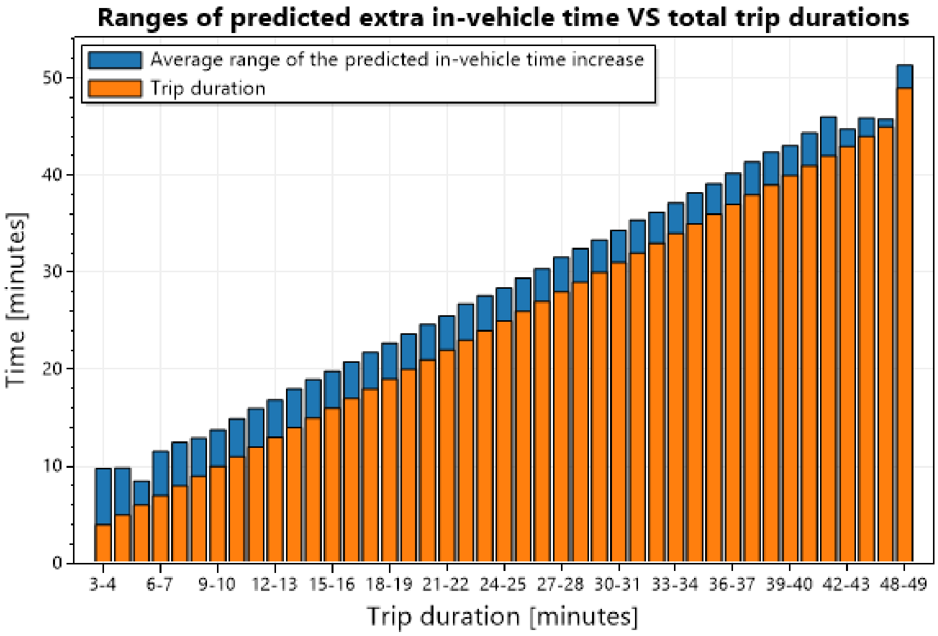


Figure 26: Comparison of the predicted ranges of the in-vehicle time increase to the total trip durations. Range = [min. value, max. value].

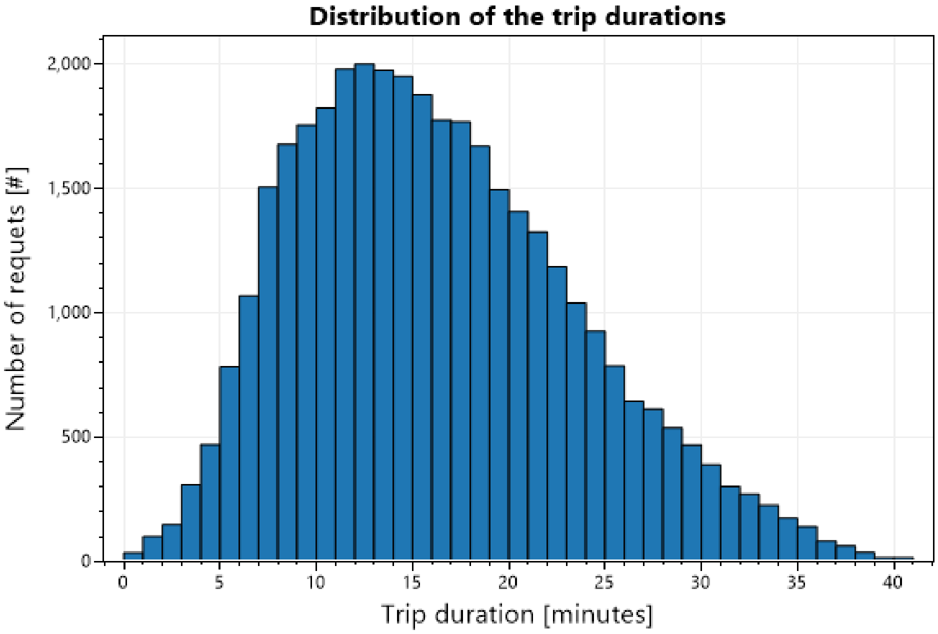


Figure 27: Distribution of the trip durations of the completed requests. Trip duration = waiting time + in-vehicle time.

4.3.3 Comparison basic scenario to baseline method

A baseline method has been implemented to test the validity of the unreliability measure. An overview of the performance of the formulated unreliability measure and the baseline method for the basic scenario is provided in Table 11. As expected, a third of the requests receive a correct prediction about the possibility of a negative change with the implemented baseline method. The same number of requests receive an incorrect and indecisive prediction for the possibility of a negative change. This is because the baseline method informs each request randomly about the possibility of a negative change. Furthermore, with the baseline method, it can be observed that less than half of the requests (44.7%) have a real travel time increase that falls in range of the provided interval. Therefore, the formulated unreliability measure does clearly outperform the simple baseline method.

Table 11: Performance of the formulated unreliability measure and the baseline method for the basic scenario. The first three rows present the percentage of requests that received a correct, indecisive, and incorrect prediction for the possibility of a negative change in waiting time, respectively. Note that “indecisive” means that a request received a “medium” possibility for a negative change. Similarly, the following three rows present the percentage of requests that received a correct, indecisive, and incorrect prediction for the possibility of a negative change in in-vehicle time, respectively. The last row illustrates the percentage of requests that have their real travel time increase in range of the provided interval.

Measure	**Formulated method**	**Baseline method**
Possibility neg. change waiting time correct	97.8%	33.3%
Possibility neg. change waiting time indecisive	0.39%	33.3%
Possibility neg. change waiting time incorrect	1.80%	33.3%
Possibility neg. change in-vehicle time correct	60.4%	33.3%
Possibility neg. change in-vehicle time indecisive	15.4%	33.3%
Possibility neg. change in-vehicle time incorrect	24.2%	33.3%
Actual extra travel time in range of provided interval	94.9%	44.7%

5 Conclusion

From theoretical research regarding unreliability in on-demand ridesharing systems, it has been concluded that a gap in the literature exists of efficiently dealing with the problem of unreliability. Users are confronted with unreliability when their real travel times deviate from the first-announced values, i.e., the received travel time information directly when the users are assigned. Moreover, case study results show that almost half of the served users face unreliability when rescheduling requests is not allowed, and nearly two-thirds of the served users face unreliability when rescheduling is allowed, strengthening the urgency of a method to reduce unreliability (rescheduling requests implies that already assigned requests can be reassigned to other vehicles). Therefore, this research formulates a novel measure to reduce unreliability in on-demand ridesharing systems. This measure aims to reduce "one-time unreliability" (Fielbaum & Alonso-Mora, 2020) induced by the operators' rules. One-time unreliability appears when new users are added to the trip, increasing the waiting time and/or detour time of the already assigned users.

The proposed measure reduces unreliability by providing two indicators to each new user: (1) the possibility that an additional request r' will be added to the trip of user r, which negatively affects the trip duration of user r; and (2) a prediction of the magnitude of the change for whenever the case occurs that a new request r' is added to the trip of user r. The first measure indicates for both the waiting time and in-vehicle time of user r the possibility of a negative change by providing a "low", "medium", or "high" possibility. The second measure provides a minimum, maximum, and average value of how much the waiting and in-vehicle time will increase. The values are determined using a simple time-series forecasting method, named exponential smoothing, and the shareability shadow concept, confining the region of ridesharing opportunities.

The measure is analyzed by simulating a state-of-the-art routing and assignment ridesharing method over a time period of three hours using a real dataset of taxi requests in Manhattan and implementing the proposed unreliability measures. Four different scenarios are analyzed to test the validity of the measure. From the simulations, it can be concluded that the method has a respectable performance to minimize unreliability. Additionally, it has the highest performance when reassignments (also known as rescheduling) of requests are not allowed. On average, for the scenarios where rescheduling is not allowed, 58% of the requests receive a correct prediction about the possibility of any negative change, 17% receive an indecisive prediction, i.e., a prediction stating that the possibility of any negative change is "medium", and 25% of the requests receive an incorrect prediction about the possibility of any negative change. Furthermore, the measure provides a 95% confidence interval of the magnitude of the travel time increase when rescheduling requests is not allowed. Not allowing for reassignments of requests only slightly reduces the system's performance in terms of rejection rate, average waiting times, and average delays. Therefore, it is worth not allowing for reassignments when implementing this measure when operators aim to minimize their service unreliability.

Still, the method is suboptimal as, on average, 25% of the requests receive an incorrect prediction about the possibility of a negative change during their trip. Several measures have been proposed to improve the performance. Hence, there is room for future research.

References

Agarwal, S., Mani, D., & Telang, R. (2019). The impact of ride-hailing services on congestion: Evidence from indian cities. *Indian School of Business*.

Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, *223(2)*, 295-303.

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

Ahas, R., Aasa, A., Silm, S., & Tiru, M. (2007). Mobile positioning data in tourism studies and monitoring: case study in tartu, estonia. In *Enter* (p. 119-128).

Al-Abbasi, A. O., Ghosh, A., & Aggarwal, V. (2019). Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, *20(12)*, 4714-4727.

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

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017a). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, *114(3)*, 462-467.

Alonso-Mora, J., Wallar, A., & Rus, D. (2017b). Predictive routing for autonomous mobility-on-demand systems with ride-sharing. *In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *(pp. 3583-3590)*, IEEE.

Altshuler, T., Altshuler, Y., Katoshevski, R., & Shiftan, Y. (2019). Modeling and prediction of ride-sharing utilization dynamics. *Journal of Advanced Transportation, 2019*.

Armant, V., & Brown, K. N. (2014). Minimizing the driving distance in ride sharing systems. *In 2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, *(pp. 568-575)*, IEEE.

Bansal, P., Liu, Y., Daziano, R., & Samaranayake, S. (2020). Impact of discerning reliability preferences of riders on the demand for mobility-on-demand services. *Transportation Letters*, *12(10)*, 677-681.

Barth, M., Todd, M., & Xue, L. (2004). User-based vehicle relocation techniques for multiple-station shared-use vehicle systems.

Beojone, C. V., & Geroliminis, N. (2021). On the inefficiency of ride-sourcing services towards urban congestion. *Transportation research part C: emerging technologies*, *124*, 102890.

Bian, Z., Liu, X., & Bai, Y. (2020). Mechanism design for on-demand first-mile ridesharing. *Transportation research part B: methodological*, *138*, 77-117.

Bilali, A., Dandl, F., Fastenrath, U., & Bogenberger, K. (2019). Impact of service quality factors on ride sharing in urban areas. *In 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, *(pp. 1-8)*, IEEE.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control.* John Wiley Sons.

Calabrese, F., Diao, M., Di Lorenzo, G., Ferreira Jr, J., & Ratti, C. (2013). Understanding individual mobility patterns from urban sensing data: A mobile phone trace example. *Transportation research part C: emerging technologies*, *26*, 301–313.

Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). Smart cities in europe. *Journal of urban technology*, *18*(2), 65–82.

Chakraborty, J., Pandit, D., Chan, F., & Xia, J. (2020). A review of ride-matching strategies for ridesourcing and other similar services. *Transport Reviews*, 1-22.

Chang, H. W., Tai, Y. C., & Hsu, J. Y. J. (2010). Context-aware taxi demand hotspots prediction. *International Journal of Business Intelligence and Data Mining*, *5(1)*, 3-18.

Chen, S., Wang, H., & Meng, Q. (2020). Solving the first-mile ridesharing problem using autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *35(1)*, 45-60.

Chen, X., Miao, F., Pappas, G. J., & Preciado, V. (2017). Hierarchical data-driven vehicle dispatch and ride-sharing. *In 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, *(pp. 4458-4463)*, IEEE.

Chen, X. M., Chen, X., Zheng, H., & Chen, C. (2017). Understanding network travel time reliability with on-demand ride service dat. *Frontiers of Engineering Management*, *4(4)*, 388-398.

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. I. (2009). Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. *John Wiley & Sons.*.

Daganzo, C. F., Ouyang, Y., & Yang, H. (2020). Analysis of ride-sharing with service time and detour guarantees. *Transportation Research Part B: Methodological*, *140*, 130-150.

Dai, J., Li, R., Liu, Z., & Lin, S. (2021). Impacts of the introduction of autonomous taxi on travel behaviors of the experienced user: Evidence from a one-year paid taxi service in guangzhou, china. *Transportation Research Part C: Emerging Technologies*, *130*, 103311.

Espín Noboa, L., Lemmerich, F., Singer, P., & Strohmaier, M. (2016). Discovering and characterizing mobility patterns in urban spaces: A study of manhattan taxi data. *In Proceedings of the 25th International Conference Companion on World Wide Web*, *(pp. 537-542)*.

Fielbaum, A., & Alonso-Mora, J. (2020). Unreliability in ridesharing systems: Measuring changes in users' times due to new requests. *Transportation Research Part C: Emerging Technologies*, *121*, 102831.

Fielbaum, A., Bai, X., & Alonso-Mora, J. (2021a). On-demand ridesharing with optimized pick-up and drop-off walking locations. *Transportation research part C: emerging technologies*, *126*, 103061.

Fielbaum, A., Kronmueller, M., & Alonso-Mora, J. (2021b). Anticipatory routing methods for an on-demand ridepooling mobility system. *Transportation*, 1-42.

Foell, S., Phithakkitnukoon, S., Kortuem, G., Veloso, M., & Bento, C. (2014). Catch me if you can: Predicting mobility patterns of public transport users. *In 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, *(pp. 1995-2002)*, IEEE.

Froehlich, J., Neumann, J., & Oliver, N. (2009). Sensing and predicting the pulse of the city through shared bicycling. *In IJCAI*, *9*, 1420-1426.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M. E., Wang, X., & Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part*

B: Methodological, *57*, 28-46.

Gardner, L. M., Duell, M., & Waller, S. T. (2013). A framework for evaluating the role of electric vehicles in transportation network infrastructure under travel demand variability. *Transportation Research Part A: Policy and Practice*, *49*, 76–90.

Gargiulo, E., Giannantonio, R., Guercio, E., Borean, C., & Zenezini, G. (2015). Dynamic ride sharing service: are users ready to adopt it? *Procedia Manufacturing*, *3*, 777-784.

Gendreau, M. (2003). An introduction to tabu search. *In Handbook of metaheuristics*, *(pp. 37-54)*, Springer, Boston, MA.

Gonzalez, M. C., Hidalgo, C. A., & Barabasi, A. L. (2008). Understanding individual human mobility patterns. *Nature publishing group*, *453(7196)*, 779-782.

Gurumurthy, K. M., & Kockelman, K. M. (2018). Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from orlando, florida. *Computers, Environment and Urban Systems*, *71*, 177-185.

Haight, F. A. (1967). *Handbook of the poisson distribution.* New York: Wiley.

Hao, N., Horesh, L., & Kilmer, M. E. (2014). Nonnegative tensor decomposition. *In Compressed sensing sparse filtering*, *(pp. 123-148)*, Springer, Berlin, Heidelberg.

Henao, A., & Marshall, W. E. (2020). The impact of ride-hailing on vehicle miles traveled. *Transportation*, *46(6)*, 2173-2194.

Herbawi, W. M., & Weber, M. (2012). A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. *In Proceedings of the 14th annual conference on Genetic and evolutionary computation*, (pp. 385-392).

Hofmann, M. (2019). Developing a streaming-based architecture for demand prediction of taxi trips in the presence of concept drift.

Hyland, M., Dandl, F., Bogenberger, K., & Mahmassani, H. (2020). Integrating demand forecasts into the operational strategies of shared automated vehicle mobility services: spatial resolution impacts. *Transportation Letters*, *12*(10), 671–676.

Hyndman, R., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach.* Springer Science & Business Media.

Hörl, S., & Zwick, F. (2021). Traffic uncertainty in on-demand high-capacity ride-pooling. *In 101st Annual Meeting of the Transportation Research Board (TRB).*

Ihler, A., Hutchins, J., & Smyth, P. (2006). Adaptive event detection with time-varying poisson processes. In *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining* (pp. 207–216).

Jung, J., Jayakrishnan, R., & Park, J. Y. (2013). Design and modeling of real-time shared-taxi dispatch algorithms. *In TRB Annual Meeting*, *8*, 1-20).

Jung, J., Jayakrishnan, R., & Park, J. Y. (2016). Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. *Computer-Aided Civil and Infrastructure Engineering*, *31(4)*, 275-291.

Ke, J., Zheng, H., Yang, H., & Chen, X. M. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, *85*, 591-608.

Kucharski, R., Fielbaum, A., Alonso-Mora, J., & Cats, O. (2021). If you are late, everyone is late: late passenger arrival and ride-pooling systems' performance. *Transportmetrica A: Transport Science*, *17(4)*, 1077-1100.

Kümmel, M., Busch, F., & Wang, D. Z. (2017). Framework for automated taxi operation: The family model. *Transportation research procedia*, *22*, 529-540.

Lam, A. Y., Leung, Y. W., & Chu, X. (2014). Autonomous vehicle public transportation system. *International Conference on Connected Vehicles and Expo (ICCVE)*, *(pp. 571-576)*, IEEE.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, *2*(2), 164–168.

Li, C., Parker, D., & Hao, Q. (2020). Vehicle dispatch in on-demand ride-sharing with stochastic travel times. *Vehicle Dispatch in On-Demand Ride-Sharing with Stochastic Travel Times*, *(IROS'21)*.

Li, W., Pu, Z., Li, Y., & Ban, X. J. (2019). Characterization of ridesplitting based on observed data: A case study of chengdu, china. *Transportation Research Part C: Emerging Technologies*, *100*, 330-353.

Li, X., Pan, G., Wu, Z., Qi, G., Li, S., Zhang, D., & Wang, Z. (2012). Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, *6(1)*, 111-121.

Li, X., Zhang, Y., Du, M., & Yang, J. (2020). The forecasting of passenger demand under hybrid ridesharing service modes: A combined model based on wt-fcbf-lstm. *Sustainable Cities and Society*, *62*, 102419.

Li, Y., & Chung, S. H. (2020). Ride-sharing under travel time uncertainty: Robust optimization and clustering approaches. *Computers Industrial Engineering*, *149*, 106601.

Li, Y., Lu, J., Zhang, L., & Zhao, Y. (2017). Taxi booking mobile app order demand prediction based on short-term traffic forecasting. *Transportation Research Record*, *2634*(1), 57–68.

Li, Z., Hensher, D. A., & Rose, J. M. (2010). Willingness to pay for travel time reliability in passenger transport: A review and some new empirical evidence. *Transportation research part E: logistics and transportation review*, *46(3)*, 384-403.

Linares, M. P., Montero, L., Barceló, J., & Carmona, C. (2016). A simulation framework for real-time assessment of dynamic ride sharing demand responsive transportation models. *In 2016 Winter Simulation Conference (WSC)*, *(pp. 2216-2227)*, IEEE.

Liu, J., Cui, E., Hu, H., Chen, X., Chen, X., & Chen, F. (2017). Short-term forecasting of emerging on-demand ride services. In *2017 4th international conference on transportation information and safety (ictis)* (p. 489-495).

Liu, M., Luo, Z., & Lim, A. (2015). A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, *81*, 267-288.

Liu, X., Li, W., Li, Y., Fan, J., & Shen, Z. (2021). Quantifying environmental benefits of ridesplitting based on observed data from ridesourcing services. *Transportation Research Record*, *03611981211997827*.

Liu, Z., Miwa, T., Zeng, W., Bell, M. G., & Morikawa, T. (2019). Dynamic shared autonomous taxi system considering on-time arrival reliability. *Transportation Research Part C: Emerging Technologies*, *103*, 281-297.

Lokhandwala, M., & Cai, H. (2018). ynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of nyc. *Transportation Research Part C: Emerging Technologies*, *97*, 45-60.

Luo, H., Bao, Z., Choudhury, F., & Culpepper, S. (2019). Dynamic ridesharing in peak travel periods. *IEEE Transactions on Knowledge and Data Engineering*.

Manna, C., & Prestwich, S. (2014). Online stochastic planning for taxi and ridesharing. *In 2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, *(pp. 906-913)*, (pp. 906-913).

Min, W., & Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, *19(4)*, 606-616.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, *14(3)*, 1393-1402.

Nanda, V., Xu, P., Sankararaman, K. A., Dickerson, J., & Srinivasan, A. (2020). Balancing the tradeoff between profit and fairness in rideshare platforms during high-demand hours. *In Proceedings of the AAAI Conference on Artificial Intelligence*, *34(2)*, 2210-2217.

National Research Council. (2002). The congestion mitigation and air quality improvement program: Assessing 10 years of experience. *Transportation Research Board*, *Vol. 264*.

Noulas, A., Scellato, S., Lambiotte, R., Pontil, M., & Mascolo, C. (2012). A tale of many cities: universal patterns in human urban mobility. *PloS one*, *7*(5), e37027.

Nourinejad, M., & Roorda, M. J. (2016). Agent based model for dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*, *64*, 117-132.

Ota, M., Vo, H., Silva, C., & Freire, J. (2015). A scalable approach for data-driven taxi ride-sharing simulation. *In 2015 IEEE International Conference on Big Data (Big Data)*, *(pp. 888-897)*, IEEE.

Pimenta, V., Quilliot, A., Toussaint, H., & Vigo, D. (2017). Models and algorithms for reliability-oriented dial-a-ride with autonomous electric vehicles. *European Journal of Operational Research*, *257(2)*, 601-613.

Prorok, A., Malencia, M., Carlone, L., Sukhatme, G. S., Sadler, B. M., & Kumar, V. (2021). Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems. *arXiv preprint*, *arXiv:2109.12343*.

Redman, L., Friman, M., Gärling, T., & Hartig, T. (2013). Quality attributes of public transport that attract car users: A research review. *Transport policy*, *25*, 119-127.

Sayarshad, H. R., & Chow, J. Y. (2017). Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. *Transportation Research Part E: Logistics and Transportation Review*, *106*, 60–77.

Schrank, D. L., & Lomax, T. J. (2007). The 2007 urban mobility report (no. ntis-pb2008103659). *Texas Transportation Institute*.

Silwal, S., Gani, M. O., & Raychoudhury, V. (2019). A survey of taxi ride sharing system architectures. *In 2019 IEEE International Conference on Smart Computing (SMART-COMP)*, *(pp. 144-149)*, IEEE.

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological*, *82*, 36-53.

Tachet, R., Sagarra, O., Santi, P., Resta, G., Szell, M., Strogatz, S. H., & Ratti, C. (2017). Scaling law of urban ride sharing. *Scientific reports*, *7(1)*, 1-6.

Tirachini, A., & Gomez-Lobo, A. (2020). Does ride-hailing increase or decrease vehicle kilometers traveled (vkt)? a simulation approach for santiago de chile. *International journal of sustainable transportation*, *14(3)*, 187-204.

Tlc trip record data. (n.d.). https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

Tsay, R. S. (2005). *Analysis of financial time series.* John wiley & sons.

Tseng, Y. Y., Verhoef, E., de Jong, G., Kouwenhoven, M., & van der Hoorn, T. (2009). A pilot study into the perception of unreliability of travel times using in-depth interviews. *Journal of Choice Modelling*, *2(1)*, 8-28.

Wallar, A., Van Der Zee, M., Alonso-Mora, J., & Rus, D. (2018). Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In *2018 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 4539–4546).

Wang, F., & Ross, C. L. (2019). New potential for multimodal connection: Exploring the relationship between taxi and transit in new york city (nyc). *Transportation*, *46*(3), 1051–1072.

Wang, Y., Zheng, B., & Lim, E. P. (2018). Understanding the effects of taxi ride-sharing—a case study of singapore. *Computers, Environment and Urban Systems*, *69*, 124-132.

Xia, J., Curtin, K. M., Huang, J., Wu, D., Xiu, W., & Huang, Z. (2019). A carpool matching model with both social and route networks. *Computers, Environment and Urban Systems*, *75*, 90-102.

Yang, H., Leung, C. W., Wong, S. C., & Bell, M. G. (2010). Equilibria of bilateral taxi–customer searching and meeting on networks. *Transportation Research Part B: Methodological*, *44(8-9)*, 1067-1083.

Yuan, J., Zheng, Y., Zhang, L., Xie, X., & Sun, G. (2011). Where to find my next passenger? *In Proceedings of the 13th international conference on Ubiquitous computing*, (pp. 109-118).

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159-175.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. *In Thirty-first AAAI conference on artificial intelligence*.

Zhao, K., Khryashchev, D., Freire, J., Silva, C., & Vo, H. (2016). Predicting taxi demand at high spatial resolution: Approaching the limit of predictability. In *2016 ieee international conference on big data (big data)* (pp. 833–842).

Zheng, H., Chen, X., & Chen, X. M. (2019). How does on-demand ridesplitting influence vehicle use and purchase willingness? a case study in hangzhou, china. *IEEE Intelligent Transportation Systems Magazine*, *11(3)*, 143-157.

Zheng, L., Chen, L., & Ye, J. (2018). Order dispatch in price-aware ridesharing. *Proceedings of the VLDB Endowment*, *11(8)*, 853-865.

Zheng, L., Cheng, P., & Chen, L. (2019). Auction-based order dispatch and pricing in ridesharing. *In 2019 IEEE 35th International Conference on Data Engineering (ICDE)*, *(pp. 1034-1045)*, IEEE.

Zheng, W., Zhuang, X., Liao, Z., Li, M., & Lin, Z. (2021). All roads lead to the places of your interest: An on-demand, ride-sharing visitor transport service. *International Journal of Tourism Research*.