University of Zurich[UZH]

Identification of potential ride-sharing paths from GPS taxi trajectory data

GEO 511 Master's Thesis

Author
Christian Grass
14-710-552

Supervised by
Dr. Pengxiang Zhao (pezhao@ethz.ch)
Dominik Bucher (dobucher@ethz.ch)
Dr. Haosheng Huang

Faculty representative
Prof. Dr. Robert Weibel

30.09.2020
Department of Geography, University of Zurich

University of Zurich[UZH]
Department of Geography

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Identification of potential ride-sharing paths from GPS taxi trajectory data

GEO 511 Master thesis

Author
Christian Grass
14-710-552

Supervisors
Dr. Pengxiang Zhao
Dominik Bucher
Prof. Dr. Haosheng Huang

Faculty representative
Prof. Dr. Robert Weibel

Date of submission: 30.09.2020
Department of Geography, University of Zurich

Contact

Christian Grass
Halb Ger 1
CH-8908 Hedingen, Switzerland
chrigigrass1@hotmail.com

Dr. Pengxiang Zhao
Chair of Geoinformation Engineering
ETH Zurich
Stefano-Franscini-Platz 5
CH-8093 Zurich, Switzerland
Current affiliation:
Centre for Geographical Information Systems (GIS Centre)
Department of Geography and Ecosystem Science
Lund University
Sölvegatan 12
SE-22362 Lund, Sweden
pengxiang.zhao@nateko.lu.se

Dominik Bucher
Chair of Geoinformation Engineering
ETH Zurich
Stefano-Franscini-Platz 5
CH-8093 Zurich, Switzerland
dobucher@ethz.ch

Prof. Dr. Haosheng Huang
Geographic Information Systems
Department of Geography
University of Zurich - Irchel
Winterthurerstr. 190
CH-8057 Zurich, Switzerland
Current affiliation:
Research Group CartoGIS
Department of Geography
Ghent University
Krijgslaan 281
BE-9000 Gent, Belgium
haosheng.huang@ugent.be

Prof. Dr. Robert Weibel
Geographic Information Systems
Department of Geography
University of Zurich - Irchel
Winterthurerstr. 190
CH-8057 Zurich, Switzerland
robert.weibel@geo.uzh.ch

Abstract

As the world's population keeps on rising and more people tend to live in cities rather than in the countryside the road networks get congested, which leads to an increase in car accidents, environmental pollution, and fuel consumption. A solution to tackle this is given by ride-sharing systems that assist in sharing taxis, thus reducing the number of vehicles on a cities' road network. Considering information about traffic congestions and vehicle's speed in such systems did so far not receive much attention from the research community. To close the given research gap this work develops a framework on how to identify potential ride-sharing paths from GPS taxi trajectory data considering the traffic state information, and analyses how such information influences the identified shared paths and the overall results of a ride-sharing system. The considered traffic state information is estimated only based on information received from the GPS records and the road network dataset and directly included into the matching process by being used as the weight of the shortest respectively fastest path algorithm. By developing and implementing a new similarity measurement between taxi trips that potentially could be shared, the complexity of the matching process gets reduced and the system is made more efficiently. The proposed system is applied to real-world GPS data of the city centre of Chengdu, China: once considering the estimated traffic state information and once assuming an absence of traffic congestions. This way the influence of traffic state information on ride-sharing systems is analysed. By not considering traffic state information a matching rate of 70.57% results. 49.91% of the total travel time and 12.8% of the total travel distance are saved. This leads to a reduction of 1'705.3 kg CO_2. On average the second passenger must wait 2 min 6 s to get picked up. With the proposed method the taxi fleet is reduced by 27.59%. Considering traffic state information, a matching rate of 54.86% and savings in the total travel time and distance of 26.72% respectively 8.85% emerge. 1'179.1 kg CO_2 can be saved while the average waiting time amounts to 3 min 14 s. 21.15 % of the taxi fleet can be reduced. This analysis shows that traffic state information leads to more conservative (and thus likely more realistic) matching of trips, which shows itself in lower respectively worse values for all the calculated measures. Most affected are the travel time savings and the average waiting. This allows claiming that ride-sharing system not considering traffic state information distort their results as they are embellished. This can lead to a decrease in the user-friendliness of a system as unexpected different waiting times or delays can emerge. This work shows that including traffic state information can be a very important point to make a ride-sharing system more useful for real-world applications. Future research should analyse, based on the same data, how much computation time can be saved by this, due to the similarity measurement, simple, yet efficient ride-sharing approach and compare its results in absolute numbers to existing systems.

Keywords: ride-sharing system, taxi-sharing, traffic state estimation, GPS trajectory

Acknowledgments

I would like to express my gratitude to several people supporting me with their guidance, cooperation, and encouragement in the process of this work:

- First, I want to extend my sincere gratitude to my supervisors Dr. Pengxiang Zhao and Dominik Bucher from the Lund University and ETH Zurich and Prof. Dr. Haosheng Huang from the Gent University for the regular meetings during this process. These meetings were always very helpful, and it was interesting to discuss the related topics together. Furthermore, I would like to thank you for helping me getting familiar with Python and providing me with all the support in the programming part. I appreciated a lot working together with you at the ETHZ and the University of Zurich or later in the year online.
- Moreover, I would like to thank Prof. Dr. Robert Weibel form the University of Zurich for allowing me to conduct this thesis under your guidance. Your inputs during the milestone-meetings always helped me to develop this work. Additionally, I appreciated your immediate help to my problems aside the meetings.
- I also acknowledge my gratitude to Jörg Roth of the IT support of the Department of Geography at the University of Zurich for providing me with the infrastructure to render my computations. Your help with accessing the server, setting up the database and installing the correct version and modules of Python was very supporting, especially in the beginning of the process.
- Without the GPS dataset this work would not have been possible. Therefore, I would like to thank the company Didi Chuxing (https://outreach.didichuxing.com) for sharing this data for scientific purposes.
- My special thanks go to Michael Robert Smith and Roman Kirschner for proof-reading this thesis.
- Lastly, I would like to extend my gratitude to my family and friends who always supported me during this year.

Contents

List of figures

List of tables

Abbreviations

CO_2	**C**arbon **d**ioxide
CSV	**C**omma **S**eparated **V**alues
DBMS	**D**ata**b**ase **M**anagement **S**ystem
GIS	**G**eographical **I**nformation **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem
HMM	**H**idden **M**arkov **M**odel
IDE	**I**ntegrated **D**evelopment **E**nvironment
OSM	**O**pen**S**treet**M**ap
SMI	**S**imilarity **M**easurement **I**ndex

1. Introduction

1.1 Motivation

Since purchasing a vehicle has become possible for a big part of the world's society, our mobility rapidly increased and shaped our environment in many ways. Places that seemed to be unreachable have become relatively closer. Nowadays, most households in developed countries own a vehicle. From having an estimated total amount of 670 million vehicles worldwide in 1996, the number of vehicles on our road networks rapidly increased to approximately 1.32 billion in 2016. Since more economies have become wealthier, not only do developed countries from the global north affect this increase, but so do developing countries from regions like Asia or South America. In fact, the number of vehicles in developed countries stabilized in recent years, while a strong increase can be seen e.g. in Asia. (Petit, 2017)

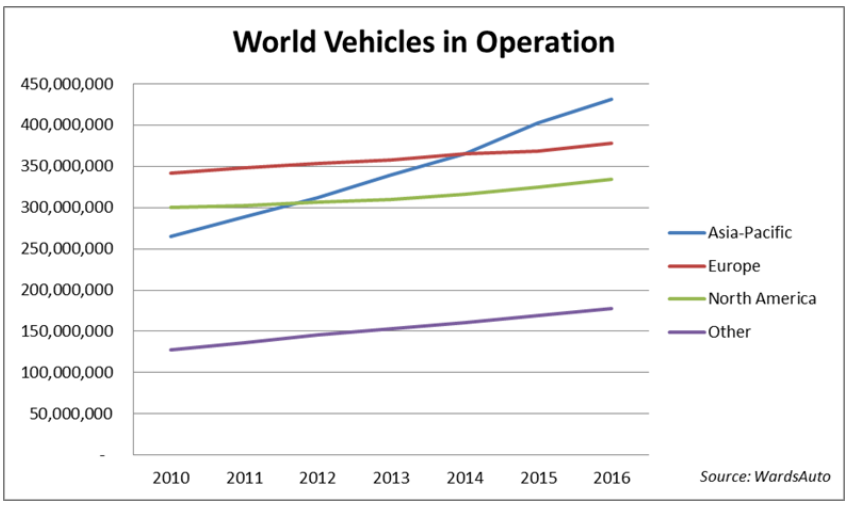


Figure 1: Development of the number of vehicles in operation worldwide between 2010 and 2016 divided into different regions (Petit, 2017).

As the world's population keeps rising, especially in the latter regions, these numbers are expected to grow even higher in the coming years. Combining this with the fact that more people tend to live in cities rather than in the countryside, complications in a city’s road network are almost inevitable. The most common form of such complications is traffic congestion. Vehicle passengers can lose more than a hundred hours getting stuck in traffic congestions. The biggest time loss due to traffic congestion was recorded in 2019 in Bogotá, Colombia, with an average time loss of 191 hours per year (Reed, 2020). These traffic congestions do not only lead to time loss, but also to an increase in car accidents, air pollution, and fuel consumption (Shete et al., 2015). Facing these problems gets even more urgent once we take into account the ongoing global climate change. A first step to address these problems can be done by reducing the number of vehicles on a city's road network. To achieve this, people started to share cars on their way to work or school, so that the number of people per car increases and the total amount of vehicles on the network decreases. This does not only reduce traffic jams, car accidents, and environmental pollution, but also minimizes the costs of travelling. Sharing cars with people whose routes are similar is commonly known as carpooling or ride-sharing (Shete et al., 2015). At first, finding other people to share a ride, had to be done manually. Nowadays, with smartphone technology, applications based on algorithms automatically assist in finding the most suitable person to share a ride with.

1.2 Background

The fact that ride-sharing can save resources has been known already since around the second world war, where the US government organized a ride-sharing program by using bulletin boards at work to match people with similar destinations to conserve resources for the war effort. During an oil crisis in the 1970s, ride-sharing was used to save fuel by reducing the total fuel consumption of vehicles. Later, the main intention of sharing cars was to cope with problems of traffic congestions and air quality. Until the beginnings of the 2000s, the people would be matched together manually by organizations at work or telephone-based ride-sharing. With the rise of internet technology, though, the matching process could be transacted online. Today, this process has become ever-so dynamic through smartphone technology and Global Positioning Systems (GPS). (Chan & Shaheen, 2012)

Using these technologies, many different applications of ride-sharing systems have been developed by companies and start-ups. The applications differ in their target audience and the objective which they want to achieve. They can be divided mainly into two groups: ride-sharing systems, which have been developed to match private persons using their own cars, and ride-sharing systems, which match taxi or taxi-like requests of customers. The main difference between these two groups is that ride-sharing for private persons has no financial motivation, meaning that the price for a ride is only as high as the driver's costs. On the other hand, ride-sharing systems of taxis or taxi-like companies are profit-making (Chan & Shaheen, 2012). In recent years, companies like Carma (formerly known as Avego), Carticipate or Zimride entered the market of ride-sharing systems offering platforms that match private persons with similar start and end points to a driver (Agatz et al., 2012). This driver uses the capacity of his car to give a ride to people with trip destinations similar to his own. This has the benefit that the individual's travel costs get reduced and the number of vehicles on the road network decreases (Barann et al., 2017).

Sharing taxis has been a common transportation method in several developing countries (Hosni et al., 2014). In Colombia, for example, such taxi providers are called "colectivos". They usually have several fixed start and end points in a city and passengers can be dropped off or picked up on the way. This form of ride-sharing has also become common in developed countries in the last several years. The main difference is the use of smartphone technology and GPS to assist in matching suitable trip requests together. This allows the applications to find a match for people's requested trips even if their start points are not at the same location. This user-friendly system is what made them marketable. Therefore, ride-sharing services have become interesting for transportation companies. Such companies can be taxi providers or taxi-like transportation companies such as Uber and Lyft. The latter are examples of taxi-like companies which, besides their normal transportation service, additionally offer ride-sharing services for some years, commonly known as UberPool or Lyft Line (Schwieterman & Smith, 2018). Another famous example of such a company is Didi-Chuxing, the leading ride-sharing company in China (Stemler et al., 2019).

Their applications are based on algorithms which must compute (in a very short amount of time) which requested trips are similar and therefore suitable to be shared without generating prohibitive extra costs like an extended waiting time for the second passenger to be picked up or a bloated travel time in comparison to an individual's trip (Agatz et al., 2012). Depending on the characteristics of a user's trip request, the setup of the algorithm can differ. If a user can, besides his desired destination, define some points of interest (POI) which the taxi must visit during the ride, then the similarity of two trips is based on the whole trip. This means that only trips that are close in space (in sense of shape and distance) and time in their entirety are considered to be similar and therefore suitable for ride-sharing (Besse et al., 2016). The methods to measure this similarity can differ substantially between applications and have been analysed in many research studies. Based on them, clusters can be built to group similar trips together. Optimal shared paths will then only be computed inside each cluster individually.

On the other hand, if the application allows its users only to define a destination, then the similarity between the requested trips depends only on the similarity between each start and end point. Between similar trips, an optimal path can then directly be computed which visits the start and end point of both trips in the shortest possible way. Characteristics like total savings of travel time or delay time for the second user can be used to identify the most suitable trips to be shared. (Santi et al., 2014a)

In general, this means such algorithms assume that start and end points or even complete trips that are close to each other are less cost-intensive to share than ones far away from each other. Solving the ride-sharing problem this way presumes that the time to reach a target in space only depends on the distance. In the real world, however, the time to move on a road network depends a lot on the traffic state. Therefore, paths or start and end points that are close to each other in space do not always have to be less cost-intensive to share than others further away. Thus, to identify potential ride-sharing paths considering real-world circumstances, information on the traffic state should be included in the algorithms.

1.3 Aim and structure of the work

Each ride-sharing system is based on different algorithms, and even if they have received a lot of interest from the research community in the last years, these algorithms leave room for improvement (Hosni et al., 2014). One point of improvement could be considering retrieved information about traffic congestions, thus making the models more realistic. Therefore, this study focuses on creating a framework to identify potential ride-sharing paths more realistically by considering the traffic state of the underlying road network. The main goals of the work are to show how potential ride-sharing paths can be identified efficiently from a raw GPS taxi trajectory dataset and where the estimated traffic state information can be included in this process. By applying the framework to a dataset of GPS taxi trajectories in the Chinese megacity Chengdu, the influence of traffic state information on ride-sharing systems is analysed. As the first step, the data must be pre-processed for further analysis and the recorded GPS taxi trajectories map-matched to the underlying road network. Subsequently, the traffic state is computed based on the trajectory points and then included in the identification process of potential ride-sharing paths. To improve the performance of this identification process, a new similarity measurement is defined and implemented. By computing the analysis once using traffic state information and once assuming that the vehicle's speed is only limited by the maximum allowed speed of each road segment, it can be evaluated what differences in the results occur between assuming an absence of traffic congestions and considering the estimated real-world circumstances. We can therefore analyse the effect such information can have on ride-sharing systems.

This work is structured as follows. In Chapter 2 an overview of the related work is provided. Interesting papers in the field of ride-sharing are discussed and insights into studies about map-matching are provided, as this is a pre-processing step applied in this work. Additionally, research about traffic state estimation is summarized and interesting findings highlighted. Finally, the research gap is shown. In Chapter 3 the research questions and hypotheses for this study are presented. Chapter 4 provides an overview of the two datasets on which the framework is applied. The different methods and processes of this framework are then explained in detail in Chapter 5. Starting with the pre-processing of the data, methods about map-matching, traffic state estimation and the final identification of potential ride-sharing paths are illustrated. The results of these individual parts, as well as the final ride-sharing paths, are presented in Chapter 6. Chapter 7 puts the findings into perspective to the research questions and hypotheses by comparing the results between including traffic state information and assuming absence of traffic congestions and, furthermore, possible points for improvements are discussed. The most important results and findings are again highlighted in Chapter 8. This study is concluded by presenting ideas for future work in this field.

2. Related work

The field of ride-sharing research, in which this work is embedded, is rapidly growing and the potential of such systems has become clearer in the last several years, especially in combination with problems related to global climate change. To show what has already been researched and to what extent this work can contribute new findings regarding the procedure of identifying potential ride-sharing paths, an insight into conducted studies of ride-sharing will be provided in this chapter. Studies of Agatz et al. (2012) and Furuhata et al. (2013) deliver a good overview of the different sub-categories of ride-sharing systems and corresponding research papers. The presented studies in this chapter mainly can be divided into static and dynamic ride-sharing systems and differ in their goal and applied methods. As these systems are usually based on GPS data, locating the recorded signals on the road network is another part of these studies. This procedure is known as map-matching and several different approaches exist to determine which road a GPS signal has been recorded on. Research on map-matching methods will therefore be discussed as the second part of this chapter. Information on traffic state will be used in this work and included in the process of identifying potential ride-sharing paths to solve the ride-sharing problem more realistically. Although methods on how such information can be derived represent its own field of research, they will be also presented here briefly. At the end of the chapter, considering the discussed studies, the research gap for this work will be presented and used in Chapter 3 to form the research questions.

2.1 Ride-sharing methods

The division of ride-sharing systems into a) systems focusing on matching private persons with other private persons, and b) systems focusing on matching trip requests of taxis or taxi-like companies, explained in the background section of this work, is also presented in the study of Furuhata et al. (2013). They divide the systems based on the type of service providers into the so-called service operators or matching agencies. Service operators are companies that provide their own vehicles to be used for ride-sharing systems. As mentioned in this study, a characteristic trait for such systems is that most of the decisions are made by the provider and the users only accept the proposed shared ride or refuse. This a typical situation of a ride-sharing system provided by a taxi or taxi-like company, where a user can request a ride, and the ride-sharing system computes an optimal ride-sharing path with another user's request. The only decision the user can make is to accept the provided ride-sharing path or to reject the offer. Matching agencies, on the other hand, are defined as ride-sharing systems that assist in the process of matching individual vehicle drivers and passengers. To better use the capacity of its vehicle and to share the travel expenses, a user can register his own vehicle for ride-sharing. In contrast to systems run by service operators, the driver himself is also seen as a ride-sharing participant who wants to reach a specific destination.

Irrespective of service operators or matching agencies, ride-sharing systems are normally constructed to reach at least one specific objective. This can be represented by an optimization problem. The most common objectives a ride-sharing system can follow are, as explained in the paper of Agatz et al. (2012), the following three: minimize the total travel distance, minimize the total travel time or maximize the number of participants of the ride-sharing system. The total travel distance is the sum of the number of driven kilometres of shared trips as well as of unshared individual trips. A ride-sharing system following this objective matches the user's trip requests where the difference in a sense of distance between two individual trips

and its shared trip is maximal. This then leads, in a global perspective, to the minimum total travel distance and additionally minimizes the emerging total travel costs. The total travel time represents how much time the participants have spent in the vehicles to reach their destinations. As a road network is highly complex, a vehicle cannot maintain the same speed on each road. Therefore, the travel time of a trip does not just depend on the driving distance, but also on the vehicle speed. Because of this, minimizing the total travel distance can result in different identified ride-sharing paths than minimizing the total travel time. The number of participants of a ride-sharing system is mainly incumbent on how many requested trips of users can be successfully matched. A good matching rate tells the user that there is a high chance to find a ride-sharing trip with this system. Consequently, the number of participants is higher in systems with good matching rates and this again attracts more potential participants as they rather register for frequently used rid-sharing systems than for uncommon ones. So, a ride-sharing system that follows the objective to maximize the number of participants matches the requested trips in a way so that it identifies as many ride-sharing trips as possible, regardless of how much travel distance and time is saved. (Agatz et al., 2012)

Other objectives of ride-sharing systems, which are affiliated to the ones defined by Agatz et al. (2012), can be to minimize the waiting time emerging for a user, if she or he is being picked up as the second person, what affects the user-friendliness and, therefore, the attraction for potential participants, or minimizing the total CO_2-emissions, which correlates with the total driving distance and the vehicle speed (Jung et al., 2013, Barann et al., 2017). The latter is shown in the study of Barann et al. (2017), where they compute how much kg of CO_2 could be saved by implementing their ride-sharing approach in the city of New York, USA. Resulting savings of around 532’000 kg of CO_2 emissions per week illustrate the potential of ride-sharing methods to contribute to the mitigation of global warming.

Besides distinguishing between studies about ride-sharing systems based on their service provider or followed objective, other characteristics like the dimensionality or the dynamics of the matching problem can be used to subdivide them. The dimensionality of the matching problem stands for the number of passengers involved in the matching process. A simple case of a matching problem is when the system only allows matching the requested trips of two users. For a ride-sharing system provided by a matching agency, this would mean, that only a driver and one passenger can be matched together. Service operators would limit their system to match only two participants to a taxi. Allowing the matching of multiple passengers in a ride-sharing system increases the complexity significantly. The maximum number of matched passengers is defined by the capacity of the used vehicle, in other words, the empty seats of the car. In this case, the system would have to match suitable trips of e.g. four persons, by still fulfilling the requirements of its objective. Considering the objective of minimizing the waiting time shows the increase in the complexity of a system matching more than two passengers, as then the overall waiting time for all passengers must be minimized and not only the waiting time of one person. (Furuhata et al., 2013)

The nomenclature for the different dynamics of ride-sharing systems varies over the research papers. Terms like static, dynamic, real-time, or on-demand systems are often being used but differently defined. Shen et al. (2015) e.g. use the terms static and dynamic ride-sharing to subdivide the systems. They define a static ride-sharing system as a system where both the origin and destination of the two participants are known in advance and the system matches the

requested trips before they have started. As soon as a trip has started, the matched ride-sharing path cannot be changed anymore. Dynamic ride-sharing systems refer to systems where an algorithm can match requested trips of users in real-time, regardless of whether a vehicle or a trip has started or not. Considering a situation where a system allows to match more than two persons, a dynamic system could match a third person to a ride-sharing trip of two users, which have been matched in advance and are already en route. While doing so, the systems must assure that the constraints of the first matched trips remain satisfied while at the same time considering the constraints of the additionally added trip. Agatz et al. (2011) do not use the term static ride-sharing but refer to the same by using the term dynamic ride-sharing. Dynamic real-time or on-demand systems are used to represent what was described by Shen et al. (2015) under the term dynamic systems. So, the underlying concept is the same but named differently. To avoid potential confusion, the nomenclature used in the study of Shen et al. (2015) will be applied in this work.

2.1.1 Static ride-sharing

In this section, interesting studies about static ride-sharing systems will be presented, discussed, and compared with each other. The models of the proposed systems are shown to later distinguish between them and the created framework of this work. Table 1 will give an overview of the discussed papers.

Armant & Brown (2014) present in their study a static ride-sharing system aiming to minimize the total travel distance. Their study represents the case of a system provided by a matching agency. There are three different types of participants in their model: drivers, riders, and the so-called shifters. Shifters are participants that can act as drivers or riders. A driver offers a trip, a rider requests a trip, and a shifter does both. They assume that each participant will complete their trip, either by a shared trip or individually. In the end, the total travel distance will be the driving distance of all the shared trips and the individual ones. Additionally, they do not set fixed start and end points to the trips. They create a set of standard locations, which are e.g. situated at the main junctions of the road network. Drivers and riders can negotiate over where to be picked up and dropped off. A rider can be matched to a driver if their defined start and end point are located in the right order on the path of the driver and they are requested in the same time window as the driver's trip. Only the riders who form a shared path that fulfils the objective of minimizing the total travel distance are matched. This ride-sharing system is then applied to a randomly generated dataset based on OpenStreetMap (OSM) data of Dublin, Ireland. The experiment is used to study the effect of changing input parameters of the system, like the number of participants or the number of different pick-up and drop off locations. The results show that an increase in participants leads to more saved kilometres and less unmatched users. Using more different locations leads to less saved kilometres and more unmatched users. This approach represents a simple ride-sharing system, where the flexibility of the user must be high, as they have to walk to the negotiated pick up location and matches are only made if the driver does not have to change his route.

Like the previous study, the work of Stiglic et al. (2015) proposes a static ride-sharing system that matches trip requests of private people using their own vehicle. Like in Armant & Brown (2014), Stiglic et al. (2015) allow the location of the users to be unfixed. This means they too assume that a user is willing to walk a certain distance to a meeting point to be picked up. Their study aims to analyse how using such meeting points in ride-sharing systems can lead to a

higher matching rate and bigger travel distance savings. The matching process is configured so that highest priority is given to match as many users as possible, and then the travel distance savings are maximized as a second main objective. In their study, they take a vehicle with three empty seats as the standard. Therefore, up to three riders plus the driver can be matched. Allowing to match multiple riders can have an influence on the user-friendliness of a system as a shared route with multiple riders has several stops, potentially causing inconvenience for certain users. To avoid this problem, they condition that only multiple riders per vehicle are allowed if they share the same meeting point in the sense of origin and destination. Each rider must define a start and end location and a certain amount of meeting places around these locations when using the system. The shared route can start either at the origin or at one of the surrounding meeting points. Additionally, a time for the earliest departure from the origin and the latest arrival at the destination must be known. The drivers define a maximum travel time they would accept. Drivers and multiple riders are then matched so that all the constraints are met, and the two objectives are followed in order. To study the influence of the meeting points, they apply their model on generated data from Atlanta, USA. To calculate the travel time of a driven route, they assume that the vehicle speed is 15 m/h (about 24 km/h) and remains constant for all roads and does not change in time. By changing the number of considered meeting places per rider between zero and four, the effect on measures, like percentage of matched users, total distance savings or average walking time of a user to its meeting point, is detected. They found out that allowing meeting places in ride-sharing systems and increasing the number of them leads to an increase in the percentage of matched users and the total distance savings. Nevertheless, their approach still assumes big flexibility of the users, which is not always given. The average walking time for a rider to its meeting places is around eight minutes. This is very high and might not be attractive for new users. Especially for use in cases of taxi ride-sharing system, this model would not fit very well as taxi customers usually are not that flexible.

The study of Agatz et al. (2011) is based on the same study area and a very similar dataset is used as in the presented work of Stiglic et al. (2015). The main difference is that in Agatz et al. (2011), no meeting points exist, and a higher but still constant vehicle speed is given. So, all the generated trips connect fixed start and end location of the user's requests. In their study, they propose a ride-sharing system that can be provided by a matching agency, which is interested in getting a revenue (a small percentage of the travel cost savings). Each user can be either a driver or a rider, which means that each user has a vehicle available. Different from the previous study, multiple matches are only allowed if the riders share the same start and end location and not just a meeting point. In the beginning, each user enters information about the earliest departure time and the latest arrival time into the system. The objective of the system is again to maximize the total distance savings. A match is only used if the resulting travel time is smaller than the sum of the travel time of the two individual requested trips. As new trip requests can enter the system at any time, the matching process must be performed repeatedly. Therefore, they define a time interval, and, always at the end of this interval, the matching process is performed again for the remaining and newly entered requested trips. They apply two different algorithms as they consider two cases. In the first case, each user can only be a driver or a rider, but not both. To find the optimal collocation of the matched trips, which minimizes the total travel distance, a maximum-weight bipartite matching model is applied. If it is not known whether a user is a driver or a rider, then a general graph matching model is used. These algorithms are very complex. In a simple case, an algorithm could always just create a match between the two trips with the biggest travel distance savings, remove them from the candidate

list, and search for the next best match. This would be repeated until no trips are left. The results show, that for the used dataset, the complex algorithms can match about 74% of all the requested trips, which leads to a total distance saving of 26%. So, they clearly outrun the simple algorithm (28% matched trips and 12% distance savings). The proposed ride-sharing system is more realistic than the previously presented ones as the start and end locations are fixed, and the flexibility of a user does not need to be very high. Nevertheless, the approach was only applied to generated data and was not tested on a real-world GPS dataset.

The structure of the ride-sharing system proposed in the study of Cai et al. (2019) is to some degree similar to the three previously discussed papers. However, it is built for a different intent as this approach is an example of a ride-sharing system provided by a service operator (taxi company). Cai et al. (2019) create a taxi ride-sharing system based on real-world GPS data. They analyse the environmental benefits of their method applied to a historical trajectory dataset of taxis in the city of Beijing, China. These trajectories are recorded by a GPS device in the taxis. Each trip comes with information about its start and end point, start and end time, total travel time of the trip, total travel distance, and the average speed value of the vehicle. Different from the previous studies, only matched trips of a maximum of two users are allowed. This means a taxi can serve a maximum of two people. The ride-sharing system aims to maximize the total travel distance savings. Their matching process can be divided into two phases. First, all shareable trips of the dataset are identified and then the shared trips that create a global maximum of the travel distance savings are selected. How they decided if two trips are shareable is explained in Figure 2.

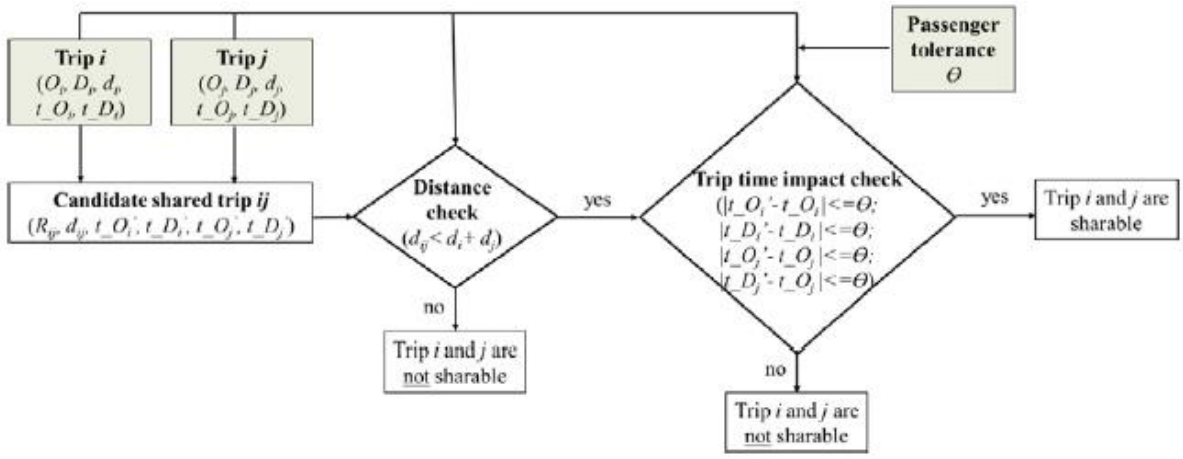


Figure 2: Framework of shareable trip identification proposed by Cai et al. (2019). O stands for origin, D for destination, d for distance and e.g. t_0 for the departure time respectively t_0' for the departure delay.

A pair of two trips must pass two tests to get marked as shareable. First, the total distance of the shared path between a pair of two trips must be smaller than the sum of the distance of the two individual trips. If this is not the case, combining these two trips will not save travel distance and is therefore not useful. Second, the shared path of a pair of two trips must fulfil defined constraints concerning the departure and arrival time. A value of 10 minutes is set as a threshold. If the difference between the departure time of the individual trip and the shared trip is bigger than 10 minutes, sharing these two trips will not be considered as it would be inconvenient for the users. The same applies to the arrival time of the trips. In the end, only trips that save travel distance and do not lead to big differences in departure and arrival time are marked as shareable.

From all the identified shareable trips, in a second step, the ride-sharing paths are selected, which lead to a global maximum in the sense of travel distance savings. A final identified shared path is a combination of two trips that visits each start and end point in the shortest distance. In what order these points are visited is not predefined. Therefore, for each combination of two trips, they define four possible paths. Setting the start points of two trips i and j to O_i and O_j, respectively the end points to D_i and D_j, the following collocations are possible: O_i-O_j-D_i-D_j, O_i-O_j-D_j-D_i, O_j-O_i-D_i-D_j, O_j-O_i-D_j-D_i. The collocation that results in the shortest distance is selected as the shared path. Cai et al. (2019) estimate the shortest distance connecting two points based on a linear relation to the Manhattan distance between them. The final distance of the shared path is, therefore, only an estimation and not an exact measurement. They apply the described method on the taxi GPS dataset of Beijing and detected that 77% of all trips can be shared. This means 33% of the total travel distance can be saved by implementing their approach. To further illuminate the potential environmental benefits of their method, they assume a linear correlation between emissions and driven kilometres. Doing so, they calculate annual savings of 28.3 million gallons of gasoline and a reduction of 2'392 tons of CO_2 emissions. Compared to the previous studies, the main difference of the proposed ride-sharing system of Cai et al. (2019) is, besides the different providers, that it is more realistic as it is based on real-world GPS data. Additionally, the approach is more user-friendly as no walking distance or multiple stops are assumed. Nevertheless, using an estimation method based on a linear relation to the Manhattan distance to calculate the distance of the shared path can lead to inaccuracies. Therefore, working with the shortest path algorithm instead based on the road network distances could have helped.

This disadvantage is solved in the work of Wang et al. (2018). They propose a rather simple yet accurate taxi ride-sharing system, where they include the road network distance and additionally the travel time into the shortest path algorithm. Their system aims to match taxi users in a way that the total travel costs get minimized. Here, static means, that each user gets the shared path proposed before the ride starts and the path will not be changed anymore during the trip. The travel costs can either be the total driving distance or the total travel time. In their system, each user defines the following parameters when entering a trip request: maximal waiting time (here the waiting time relates to the time it takes for the system to find a shared ride), maximal acceptable departure and arrival delay, number of people requesting the trip, and minimal taxi fare reduction (percentage of taxi fare reduction due to ride-sharing). Additionally, Wang et al. (2018) define a percentage of the taxi fare a taxi driver must earn extra when serving a shared ride. For a set of two trips to be accepted as possible ride-sharing paths, all the above-mentioned constraints must be met. The requested trips are then ordered by their request time and each of them gets analysed separately. Their system creates a shortest path (either based on the road network distance or the travel time) between the analysed trip and each trip that has not been matched so far, and that fulfils the constraints. The shared path that minimizes the travel costs is selected and gets assigned to a taxi driver, rendering it not a part of the system anymore. If no shared path is found for the available trips, the analysed trip will be stored to be eventually matched to a future trip request. If the time a trip is stored exceeds its departure delay, the trip gets served individually and leaves the system as well. This optimization strategy results in a local optimum, not in a global optimum as applied in the study of Cai et al. (2019). This means, that a served shared trip is a combination of two trip requests that minimizes the travel costs at this moment. From a global perspective, this does not mean, that this combination was the best possible choice, but it nevertheless highly reduces the complexity of the system.

This study represents an alternative approach to the global optimum model proposed in the paper of Cai et al. (2019) and shows that a local optimum can highly reduce the complexity of a system without suffering a significant loss of accuracy. Moreover, interesting constraints about the price setting of ride-sharing systems are included, which is not the case in the previously discussed papers.

The proposed ride-sharing system in the study of Santi et al. (2014a) connects the mentioned characteristics of the systems presented by Cai et al. (2019) and Wang et al. (2018). Their system is based on a global optimum model and the shortest respectively fastest paths are computed in the matching process. This results in an accurate and efficient method but leads to time-intensive computations. With their approach, they assist in matching taxi requests so that either the number of identified shared trips is maximal or the total travel time is minimal. Depending on which objective is pursued, different results are obtained. Similar as in the previous studies, the system of Santi et al. (2014a) identifies a set of two trips as shareable provided there exists a path that connects each start and end point in the right order so that for each trip the start point is served before its end point and some specified constraints are met. Excluded is the collocation where one trip is being served before the other has started, as this does not represent a shared ride. The built shared path must not lead to a delay bigger than a threshold set by the user and must result in a total travel distance shorter than the sum of the two individual trips. For every single path, the trips that have been requested during a time window, more specifically in a certain amount of time before and after the analysed trip, serve as candidate trips for ride-sharing. Between each set of two trips, the fastest path connecting all the origins and destinations is built. As in the study of Cai et al. (2019), four possibilities to build the fastest path are given. As opposed to the shortest path, the edges used in the algorithm are weighted by the estimated travel time and not by the distance. This travel time represents the time a vehicle needs to travel on a specific road segment. Through a heuristic approach, the travel time gets estimated based on the information of the origin and destination of each trip. For all the identified fastest paths in a time window that fulfil the constraints, a global optimum is found that follows one of the two defined objectives. This procedure can be seen in Figure 3.

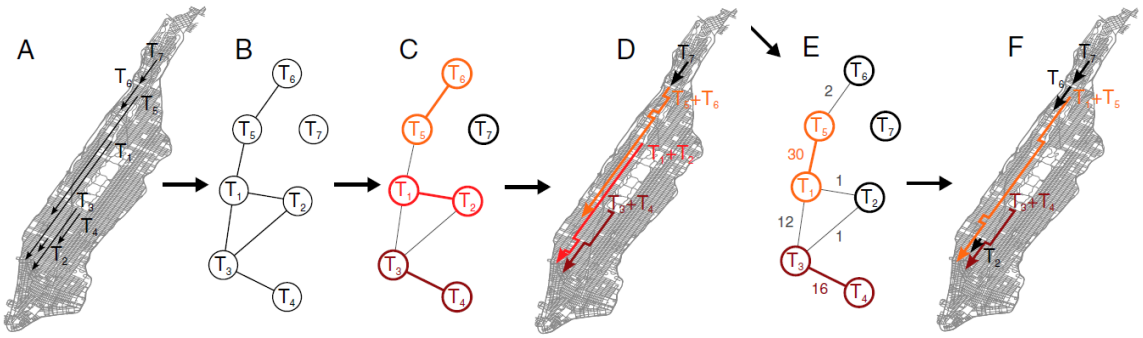


Figure 3: Visualisation of the process of finding a global optimum for the potential ride-sharing paths. (A) shows all candidate trips in a time window. In (B) the fastest paths that fulfil the constraints are displayed. (C) represents the global optimum for maximizing the number of shared trips and is visualised in (D). The global optimum in (E) and (F) minimizes the total travel time. (Santi et al., 2014a)

They apply the presented approach on real-world GPS data from the city of New York. The threshold for the maximum delay is set to five minutes and the analysed time window to one minute. This means trips that started one minute before and after the analysed trip serve as candidates. For the case where maximally two trip requests can be matched and the objective of minimizing the travel time is followed, 93% of the trips can be shared. This results in travel time savings of 32%. Changing the objective to maximize the number of shared trips leads to a matching rate of nearly 100% but to a decrease in the travel time savings. Additionally, they analyse how the number of users per shared ride affects the results. They conclude that the improvement in the results by allowing multiple users to be matched is too small to legitimize the significant increase in the computation complexity. (Santi et al., 2014a)

Besides the presented static ride-sharing studies, other similar works exist as e.g. the study of Sun et al. (2020). They build a non-profit peer-to-peer ride-sharing model provided by a matching agency aiming to maximize the total cost saving. The path between two trips that maximizes these total cost saving represents its' shared path. By a column generation based heuristic approach, the optimization problem gets solved. Ota et al. (2015) propose a taxi ride-sharing system with the objective of minimizing the total driving distance. The matching process is based on the Dijkstra's shortest path algorithm and they tested their approach on real-world data of the city of New York, USA. Similar in principle but still uniquely different is the ride-sharing study of Barran et al. (2017). They conduct a taxi sharing study based on GPS taxi data from New York, USA. They do not follow a specific objective and therefore do not try to optimize their approach. They simply define several constraints which must be met by the shared path. All trips in a time window are analysed in order of their request time. As soon as a set of two trips fulfils the constraints, the identified shared path is taken as the final ride-sharing path. Thus, their approach is based on the principle of first-come-first-served. Summarizing this section, all the discussed studies are listed again in Table 1.

Study of:	**Service provider**	**Objective(s)**	**Matching process**
Agatz et al. (2011)	Matching agency	Minimize total travel distance	Fastest path based on constant vehicle speed
Armant & Brown (2014)	Matching agency	Minimize total travel distance	Matching if negotiated pick up / drop off points of user are located on drivers' path
Barran et al. (2017)	Service operator (taxi sharing)	No objective (first-come-first-served principle)	Shortest path
Cai et al. (2019)	Service operator (taxi sharing)	Minimize total travel distance	Shortest path based on Manhattan distance between origin / destination
Ota et al. (2015)	Service operator (taxi sharing)	Minimize total travel distance	Shortest path
Santi et al. (2014a)	Service operator (taxi sharing)	Maximize matching rate or minimize total travel time	Fastest path based on by heuristic approach estimate travel time
Stiglic et al. (2015)	Matching agency	Maximize matching rate and minimize total travel distance	Shortest path between drivers' and users' origin and destination or meeting points
Sun et al. (2020)	Matching agency	Maximizing the total cost saving	Shortest path where the weights of the edges are the cost saving
Wang et al. (2018)	Service operator (taxi sharing)	Minimize total travel distance or minimize total travel time	Shortest path / fastest path

Table 1: Overview of the discussed static ride-sharing studies.

2.1.2 Dynamic ride-sharing

As in the previous section, in this one dynamic ride-sharing studies will be presented, discussed, and related to each other. It will be shown why these studies are categorized as dynamic and not static ride-sharing studies. Table 2 provides an overview of the discussed studies in this section.

In the study of He et al. (2014) a ride-sharing system that dynamically creates ride-sharing routes based on GPS trajectories is proposed. Their system represents a case of a matching agency where multiple passengers can be matched together. A driver can also become a rider if this optimizes the followed objective of minimizing the total travel distance. Similar to some of the previously presented studies, they assume that a user is willing to walk a specified maximum distance to a connection point to be picked up. Figure 4 shows the architecture of their system. In the first step, frequent routes of users are detected by mining their complete GPS trajectories. So, this system uses the trajectory in their entirety to compare it with the other ones and not just the start and end points of a trip. This is, as described in the background section of this work, often used in ride-sharing systems provided by matching agencies. In the second step, the stored frequent routes are used to form shared routes. Qualified riders for a route are selected and ranked by a defined service cost where the best-ranked rider and her or his route are matched to the original route. The defined service cost is the sum of the following parameters: the travel cost, which is proportional to the travel distance, the distance a user would have to walk to a pick-up location, the detour distance, the time a user has to wait to start the ride and the social distance, which is defined as the distances between the start and end points of two trips. Between the original route and the route that creates minimal service costs, a shared route is generated if some additional constraints are met. This procedure is repeated until there are no empty seats left. This study is categorised as a dynamic ride-sharing system because if a rider leaves the car, in other words is dropped off, a new rider can be matched to the route, which can in turn change the shared route again while still meeting the constraints set by the other passengers. (He et al., 2014)

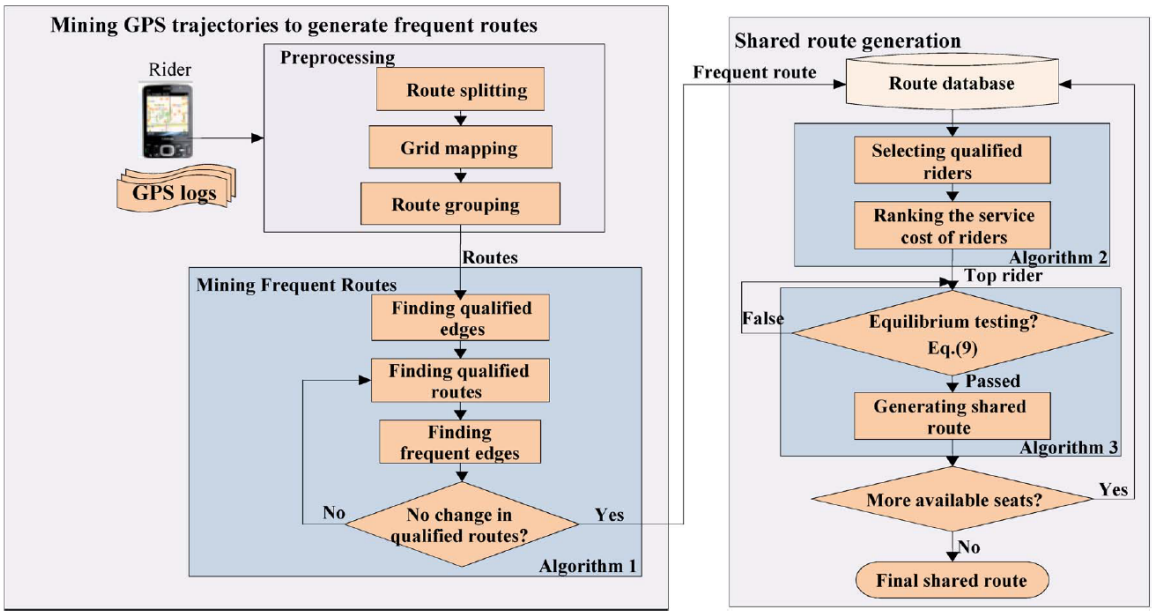


Figure 4: Model of the dynamic ride-sharing system proposed by He et al. (2014).

Similar base characteristics are given in the work of Haddad et al. (2013). They also propose a dynamic ride-sharing system for multiple passengers representing a matching agency case where it is assumed that the users are willing to walk to a meeting place. The matching process is based on the principle of the longest common path. Since the shared route which maximizes this longest common path is selected as the final ride-sharing path of two or more users, the objective of their system is to minimize the total travel distance. When a new user registers for the application, either a request as a driver or a request as a rider can be entered. The driver enters her or his origin, destination, desired departure time, maximum acceptable deviation, and some additional information about his vehicle. The system then computes a set of possible routes for its request based on the Google Maps API, and the driver must select one of them, which is then stored in a database. The rider, on the other hand, must define its origin, destination, and desired departure time. Again, routes for the entered information are derived by the Google Maps API. The system then selects all routes of drivers (empty vehicle or already serving a shared ride) that are suitable to be matched to the rider and sorts them based on the longest common path. Finally, the rider selects the driver of its choice (does not necessarily have to be the one with the longest common path) and gets matched to this trip. Automatically, the capacity of empty seats of the driver's vehicle decreases by one. A driver's trip is available for sharing until there are no empty seats left in the vehicle. This shows the dynamic part of their system and therefore the location of every driver must always be updated. This approach again differs from the case of a taxi sharing system as not only start and end points are considered for the shared path, but also the path in its entirety. Unfortunately, there exists no experiment of their approach based on real-world data and, therefore, no information on the performance of their system is given. (Haddad et al., 2013)

Both presented studies already demonstrate the increasing complexity of dynamic ride-sharing systems in comparison to static ones due to permanent location updates and having to meet multiple constraints. An even higher degree of complexity is given in the system proposed by Tian et al. (2013). Different from the previous works, they create a dynamic ride-sharing system to match taxi requests of multiple users. They design a model named Noah and apply it to a GPS dataset of the city of Shanghai, China. Their matching process is based on two user-defined constraints: the maximum acceptable waiting time to be picked up and the maximum percentage of a detour to the shortest path between its origin and destination. Its algorithm must identify all the taxi requests satisfying these constraints while building the shared path. In other words, for each analysed trip request, the shortest path is generated that connects the origin and destination point of itself and of each entered and not yet separately finished trip. Only the computed shared paths that fulfil the constraints are kept. The objective of their system is to minimize the waiting time occurring for the new user, and therefore the shared path that leads to the shortest distance between a candidate trip and the pick-up location of the analysed one is selected as the final shared path. Because the proposed system allows matching multiple users even en route, the matching process gets much more complex than just described. A taxi that is already occupied by two users that share a ride and is en route still serves as a possible matching partner for a new entering request, provided both the old constraints and the newly added ones are met. Therefore, in this system, a lot of shortest paths must be computed to select the best fitting one. To tackle the problem of increasing time consumption of this computation, Tian et al. (2013) included a caching process in their model so that the same shortest path does not get computed twice. Additionally, and very much like the previous works, the location of a taxi must be updated all the time to properly compute shortest paths to new requested trips.

Tian et al. (2013) then test their approach on the mentioned dataset of Shanghai. An average waiting time of 4 minutes and an average detour of 12% result. Unfortunately, no findings of travel time or travel distance savings, matching rate, nor reduction of the taxi fleet are given. Moreover, characteristics of the underlying road network such as travel speed are ignored in their approach. Such a dynamic system might be useful as it is flexible and can handle real-time requests even if taxis are in motion, but as described, the complex architecture leads to a very time-consuming model.

As written in Bathla et al. (2018), such computation-intensive models can lead to performance problems if they have a centralized architecture. Centralized means that all the trip requests are handled on a central server. With the increasing complexity of the systems, these servers can slow down the performance. To avoid a potential performance loss in a dynamic ride-sharing system, Bathla et al. (2018) propose an alternative, so-called distributed taxi ride-sharing solution to the problem. They apply a messaging system based on wireless transmission that allows the system to handle the requests locally. Their approach differs from the so far presented architectures as not all trip requests in a time window are considered in the matching process. Only trip requests within a specified radius are considered. A trip request is then sent by a user to all the taxis that are located in this specified radius. Each taxi that receives the message stores it in a temporary schedule. If a shared trip is found, the trip request leaves the temporary schedule and the shared path is stored in a permanent schedule. Thus, let us assume that a taxi is en route serving a shared trip and is located inside the defined radius of the new user when a new trip gets requested. The request is then received by this taxi and the shortest path between the origins and destinations of the new request and the already shared path will be computed if there are empty seats available in the moving taxi. To check if the computed shortest path is a potential ride-sharing path, two constraints must be met. The waiting time for the user to be picked up must not surpass a defined value, and the delay time must be less than a certain threshold. Additionally, the constraints of the two users that are already en route must still be met. After each taxi that received the request has evaluated it, the user receives an answer from them. The proposed shared trips are then sorted by their costs (how much an individual must pay for the ride), the number of free seats in the taxi (objective of maximizing the occupancy of the taxi), and the minimum time it takes to reach its destination. The best one is selected as the final shared path and the corresponding taxi receives a confirmation from the user. The trip request then leaves the temporary schedule and the adjusted shared path is restored in the permanent schedule. All other taxis that were not selected, remove the trip request from their temporary schedule. This procedure goes on as long as there are empty seats available in a taxi. If a user is dropped off, a new person can be matched to this ride.

The last point shows why this approach is categorized as a dynamic system. Each taxi must always compute the shortest path to a newly received request, even if it is already half full. This again leads to a time-consuming computation, but in comparison to centralized systems, to a much less complex one as only taxis inside the defined radius of a user are considered. The presented system is then applied to a GPS dataset of the city of Shanghai. Based on the set constraints the distributed model achieves a matching rate of 3.5% and a total distance saving of 2%. In comparison to the results of previous studies, these numbers are very small. This is due to the limited number of candidate trips respectively taxis inside the short radius. So, although this alternative approach might be less risky regarding performance issues, it still leads to a heavy loss of effectiveness. (Bathla et al., 2018)

There are several other rather interesting studies of dynamic ride-sharing system in the literature. Yu et al. (2020) e.g. develop a distributed taxi-sharing system based on the model of Bathla et al. (2018). There are three main differences in their approach, which make the model even more dynamic. While waiting on the confirmation of a passenger for a shared ride, in Bathla et al. (2018), the taxi is blocked for handling other requests. In Yu et al. (2020), the taxi still receives new requests and processes them. After receiving the confirmation from the passenger, the taxi must check the shared path again on the constraints to see if it still is suitable to be shared. Only if this is the case, the passenger gets a second confirmation message from the taxi to fix the shared ride. Therefore, different from the previous study, the taxis can, in a parallel fashion, handle multiple incoming requests. If sharing the ride request of a passenger with a taxi is not suitable, the request does not just get disclaimed, but it is transmitted to the neighbour taxi instead. Through these three changes, the system of Yu et al. (2020) reduces the average waiting time and increases the matching rate. Another example is the study of Gökay et al. (2019) which represents a dynamic ride-sharing system provided by a matching agency. Like in previously presented studies, it is assumed that the users are willing to walk a certain distance to a pick-up location. Trip requests that have similar start and end points at the same time window are grouped and a new pick-up and drop-off location for that group is generated. These requests are then handled as one trip and matched to a driver. If there are empty seats available in the vehicle, additional individual or grouped trips can be matched. With their approach, they try to better utilise the resources. They argue that the small decrease in customer convenience due to walking can be recouped by offering cheaper rides. Additionally, they show that the total vehicle costs decrease and the matching rate increases. Aydin et al. (2020) propose a dynamic ride-sharing system to again match private person with private drivers. They analyse that allowing the driver to be matched to more than one passenger's request in real-time, which is what represents the dynamic aspect, leads to an increase in the matching rate by 33%. Additionally, a different approach compared to the previous works is chosen. They implement a social compatibility score JSS, which consists of parameters like age, gender, employment, and the degree of willingness to meet new people. If the maximizing of this JSS score is chosen as the systems' objective instead of maximizing the distance savings, only a small decrease in the distance savings emerges; at the same time, many qualitative matches are found. So, different from other studies, besides trying to maximize the distance savings, they focus their approach on optimizing the social component of the ride-sharing problem as well.

Study of:	**Service provider**	**Objective(s)**	**Matching process**
Aydin et al. (2020)	Matching agency	Maximize distance savings or maximizing the JSS score	Shortest path
Bathla et al. (2018)	Service operator (taxi sharing)	Minimize costs and maximize vehicle's occupancy	Distributed and not centralized solution using shortest paths
Gökay et al. (2019)	Matching agency	Minimizing the vehicle costs	Shortest path after grouping similar user requests
Haddad et al. (2013)	Matching agency	Minimize total travel distance	Longest common path
He et al. 2014	Matching agency	Minimize total travel distance	Based on a defined service cost function that is minimized
Tian et al. (2013)	Service operator (taxi sharing)	Minimize waiting time	Shortest path in combination with a caching model
Yu et al. (2020)	Service operator (taxi sharing)	Minimize costs and maximize vehicle's occupancy	Distributed and not centralized solution using shortest paths

Table 2: Overview of discussed dynamic ride-sharing studies.

2.2 Map-matching

Most of the presented ride-sharing systems have been applied or tested on real-world GPS trajectory data. As defined by van Kreveld & Luo (2007): “The trajectory of a moving object is a continuous function τ(t) of time t such that given a time instant t, it returns the position of the moving object. In reality, the moving object trajectory is recorded by a finite set of observations at discrete time stamps t_1, t_2, ..., t_n.”. Each observation of a trajectory represents a trajectory point. The position of these trajectory points is recorded by a GPS signal. This means, that a driven taxi trip is represented through a trajectory consisting of several GPS signals. These signals were recorded by a GPS device in the corresponding vehicle. High-frequency GPS devices can record the location, given through the x- and y-coordinates, the exact timestamp, and sometimes the speed and direction of the vehicle e.g. every second (Greenfeld, 2002 & He et al., 2019). But the recorded coordinates do not represent the exact location where the vehicle was located at the recorded timestamp as GPS devices in urban environments normally only have accuracy of about 10 meters (Aly et al., 2016). This means, by visualising each GPS signal and connecting them, one does not automatically receive the exact driven vehicle path on the underlying road network as especially in urban road networks, more than one road can be located inside this error radius of around 10 meters.

To use GPS trajectory data in ride-sharing systems, it is important to know on which road segment the vehicle truly was during a GPS record. Using information from position systems like GPS together with road network data to determine on which road segment and where on this road segment a vehicle was located is called map-matching (Quddus et al., 2007).

As map-matching GPS trajectory data is not only used for ride-sharing systems, many other studies have been published. With more and more different methods emerging as a product of these numerous studies, map-matching forms its own field of research. In the following, this field is shortly presented, and some different approaches are discussed. This is used in the method section to explain why the chosen map-matching approach was selected.

A good example of why the map-matching process can be very complex where one cannot just match each GPS signal to the closest road segment in the road network, is provided in Figure 5. Each black dot shows the coordinates of a GPS signal of a moving vehicle. Due to the accuracy error, it is not clearly visible which road segment a GPS signal belongs to if each dot is analysed separately. Simply matching each GPS signal to the closest road segment would lead to an incorrect vehicle path as in this case location 2 and 3 would be matched to the wrong road. By considering the 3 locations together, it is clear which road segment the GPS signals belong to.

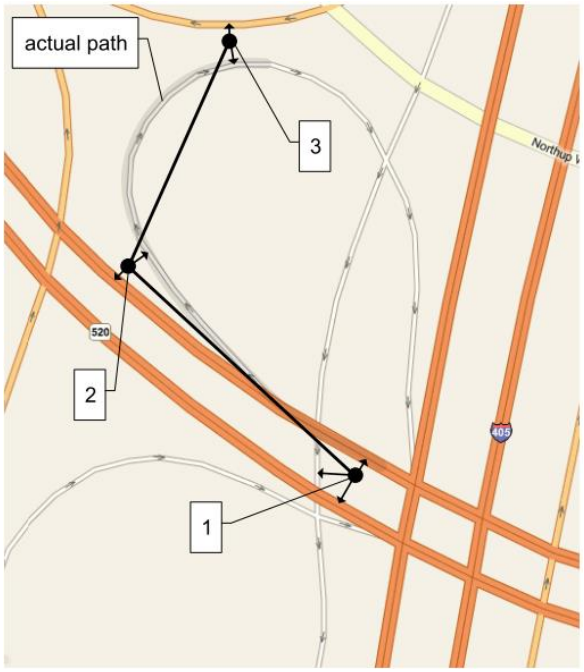


Figure 5: Visualisation of the map-matching problem by Newson & Krumm (2009). The black dots represent the GPS signals and the light grey curve shows the vehicle's actual path. Each dot could be assigned to more than one road segment.

However, because it is not possible to visually analyse each GPS trajectory of a dataset containing several thousand of them, different methods that automatically map-match the trajectory points have been created. Quddus et al. (2007) offer a good overview of these different map matching methods and studies. They categorise the different approaches mainly into four groups: geometric, topological, probabilistic, and advanced techniques.

Geometric methods represent the naive approach of just considering the geometry of the road network. This means that the connection and therefore the topology of the road segments is not considered. Only the shape of them is used to determine the exact location of the vehicle. Such methods are e.g. point-to-point matching, point-to-curve matching, or curve-to-curve matching. (Quddus et al., 2007)

As described by Bernstein & Kornhauser (1996), in a point-to-point matching approach, the closest point on the road network for each GPS signal is searched. Each road segment consists of at least two nodes (start and end node) and an edge connecting these nodes. Complex road segments can additionally have some shape points (vertices between the start and end nodes). For each node or shape point in a reasonable radius, the Euclidean distance to the GPS signal is calculated, and the signal is then matched to the node or shape point with the shortest distance. The more shape points a road segment has, the better the result. A point-to-curve approach is similar, but rather than the closest point, the closest edge is searched. Therefore, the distance between the GPS signal and each edge must be computed. In curve-to-curve approaches, not only one GPS signal is used, but also several others that form a curve or a line. The goal is then to compute the distance between this curve and the surrounding edges in order to find the shortest one. As already the publication date of the mentioned study tells, such naive geometric approaches are very old and not up to date anymore. Especially in dense urban road networks, such map-matching methods are not used any longer as they are very erroneous.

Topological map-matching methods are algorithms that also make use of the topology of the road network to define the exact location of a GPS signal and, therefore, consider the information of connectivity and contiguity of the edges (Quddus et al., 2007). A prime example of this method is the enhanced weight-based topological map-matching algorithm proposed by Yang et al. (2013). Based on the study of Velaga et al. (2009), they define four weights related to topological information and calculate them for each candidate road segment. Additionally, they define weight coefficients depending on the density of the road network, meaning that e.g. some weights are more important in dense road networks than in sparse areas. Based on the total weight score, the best candidate road segment is selected for each GPS signal. The four weights are proximity, heading, edge connectivity, and turn restriction. The proximity is the distance between the GPS signal and each candidate edge and is calculated like in the geometric point-to-curve method. The heading depends on the angle between the moving direction of the vehicle (some GPS devices also record this information) and the direction of the candidate edge. If an edge is connected to a junction, then the edge connectivity and at best information about turn restriction is considered as well. This already gives significantly more accurate results as not only the closest point or edge is searched. Yang et al. (2013) test the presented approach on real-world GPS data and show that in dense urban areas, around 97% of the trajectory points are correctly map-matched by their method.

Probabilistic map-matching methods are based on topological inputs but use probabilistic measurements rather than absolute weight scores to determine which road segment a vehicle was located on (Quddus et al., 2007). Therefore, with such methods, for each candidate trip, a probability is measured that this road segment is the one where the vehicle was driving at the given timestamp. Ochieng et al. (2004) propose such a probabilistic map-matching method in their study. They define an elliptical error region based on the error variances of the GPS device and for all candidate road segments inside of this error region, the probability of being the correct road segment for a GPS signal is computed based on information like heading, connectivity, or closeness. In the study of Bierlaire et al. (2013), they go even further by computing the possibility for a whole trajectory instead of individual trajectory points only. They created multiple hypothetical paths made up of connected road segments and computed the probability that such a path would lead to the measured GPS signals. Assuming a normal distribution of the GPS error, they computed, for each trajectory point of this hypothetical path, the probability that the distance between this point and the measured point is smaller than the estimated error. Building an integral over all the probabilities of the trajectory points gives a probability value for each hypothetical path. The most probable one is then selected as the map-matched path.

The fourth category of map-matching methods consists of advanced techniques which are methods that are based on more sophisticated and complex algorithms (Quddus et al., 2007). Most of them follow the principle of probabilistic approaches but in a more complicated way. They too result in probabilities for each candidate road segment, and the estimated location of the GPS signal is selected by these values. Examples of such complex algorithms are the Extended Kalman Filter used in the study of Obradovic et al. (2006), the Dempster-Shafer theory of evidence in Yang et al. (2003), a fuzzy logic-based approach by Syed & Cannon (2004), or a Hidden Markov model presented in studies of Ren & Karimi (2009) and Newson & Krumm (2009). The Hidden Markov model is based on the optimization of the product of emission and transition probabilities of the different candidate road segments. The emission

probability of a road segment stands for the likelihood that a GPS signal would be recorded if the vehicle is driving on that segment. The transition probability represents the chance that the vehicle is driving on a certain path given the connectivity of the road segments. The Viterbi algorithm is then used to find the optimal path that maximizes the product of the two probabilities, and therefore is most likely the truly driven path of the vehicle.

Hashemi & Karimi (2014) compared in their work the accuracies of some of the presented map-matching methods in this chapter. Most of the advanced map-matching methods lead to an accuracy higher than 90% correct identified road segments. By comparing the methods of the different categories, topological, probabilistic, and advanced techniques most of the time outrun simple geometric approaches. This is illustrated in Figure 6, where simple algorithms stand for geometric approaches, weight-based algorithms for topological methods, and probabilistic and advanced techniques are represented by the advanced algorithms. When using map-matching methods, it is important to consider the data in the sense of GPS accuracy, frequency of the records, and density of the road network. The more accurate the methods get, the more complex and time-consuming the computations will be. Therefore, using complex map-matching methods in sparse rural areas makes less sense than in dense urban regions as in these cases, simpler algorithms can lead to satisfactory results as well. Considering this, the accuracy values of such methods must always be regarded with suspicion as each model only works that accurate for specific conditions. In the end, selecting the right map-matching method is a trade-off between accuracy and performance regulated by the type of data.

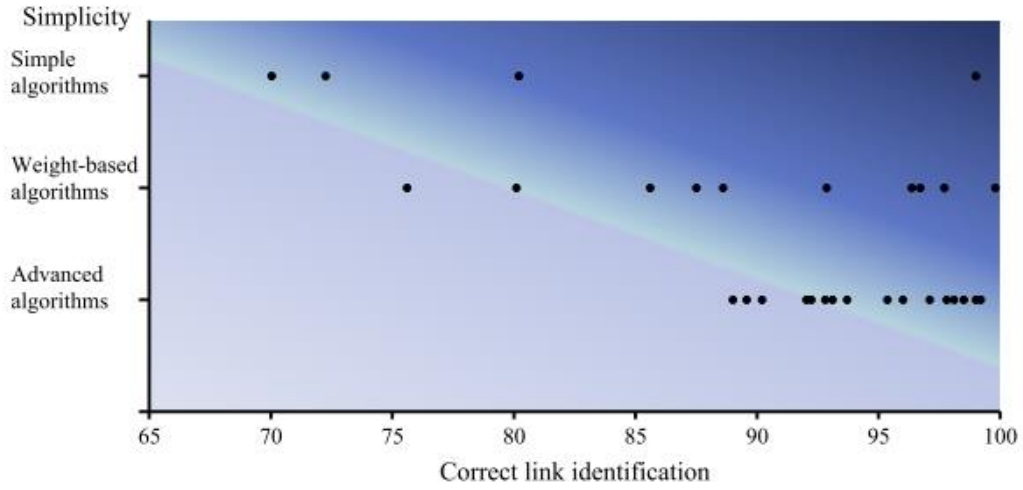


Figure 6: Accuracy of the different categories of map-matching methods analysed and illustrated in the work of Hashemi & Karimi (2014).

2.3 Traffic state estimation

Information on traffic state in a city's road network can be very useful for any kind of mobility analyzation as traffic congestions highly affect the travel time on a road network. As already mentioned in the introduction to this work, the traffic state information will be used in the proposed ride-sharing approach and, therefore, a short overview on how such information is derived and used in other studies is given.

Methods on traffic state estimation are either used to reconstruct traffic patterns like traffic congestion and mean travel time for a specific road segment or help compute short time predictions of traffic flow. Besides observing traffic state through video cameras or loop detectors, GPS trajectories from probe vehicles can be used to estimate quantities like flow, speed, or density (Sunderrajan et al., 2016). Asakura et al. (2017) analyse probe vehicle's GPS trajectories to identify traffic incidents that lead to traffic congestion by comparing the travel time of two connected road segments. They apply their method on a highway segment in the city of Tokyo, Japan, where they divide this road into segments of equal length. As they work with GPS signals recorded every second, based on vehicle speed and length of the segments, the travel time of each vehicle for each segment can be exactly calculated. Using characteristics like the absolute travel time difference, the ratio of the travel time difference, and the flow rate difference ratio, two consecutive road segments are compared. Wherever the distance difference and ratio are big enough combined with a decrease in the traffic flow, an incident, and a subsequent traffic congestion, is detected.

De Fabritiis et al. (2008) and Kerner et al. (2005) use floating car data to compute mean travel speed, and thus the differences in mean travel time of a road network. This information then serves as an input for short time predictions of traffic speed. The former use floating car data of the Rome Ring Road in Italy containing information about the vehicle speed at each GPS signal. To estimate the current speed of each road segment, the average of the vehicle's speed measures in combination with previously calculated average speed values on that road segment is computed every 3 minutes. Similarly, Kerner et al. (2005) use average traffic speed values to assign them to each road segments as its average speed depending on the time. The travel time is then calculated in combination with the length of the segments.

Nanthawichit et al. (2003) propose a more complex method to estimate traffic state; they also additionally predict traffic flow by applying a macroscopic model and a Kalman filtering technique on a mix of probe vehicle data and data from stationary detectors. The travel time for each road segment based on the GPS signals is again calculated as in studies like Kerner et al. (2005). This is then combined with information about traffic speed and density of stationary detectors. Using this as an input to the algorithm, short time predictions are made. The presented studies are papers that completely focus on traffic state estimation or prediction and their methods, as the study of Nanthawichit et al. (2003) shows, can, therefore, be quite complex. If traffic state information is used in a study where it is not its main focus, it sometimes makes more sense to apply a less complex and, therefore, less time-consuming method.

Two of the presented ride-sharing systems in the section about static ride-sharing work as well with information on the travel time of the road segments, and therefore also, with information on the traffic state. Wang et al. (2018) compute the average taxi travel time for each road segment based on real-world GPS data. As described in their study, they use taxi trajectory data collected over six months to calculate the traffic state. The result is the travel time for each road segment for every hour. It is not mentioned when the data was collected, hence, it is not given that the system truly works with traffic information derived by the same analysed dataset as the data of these six months could have been collected earlier. Moreover, it is not explained how the average travel time is estimated.

Santi et al. (2014a), on the other hand, explain in detail how they estimate the travel times used in their ride-sharing system. Due to the lack of trajectory and speed information in their dataset, they estimate the travel time only based on the pick-up and drop off times of their analysed trips. This means they do not include each GPS signal into their computation as proposed in the presented study of Kerner et al. (2005). First, each pick-up and drop off GPS location is matched to the closest intersection in the road network. Then, knowing the travel time between every pick-up and drop off point lets them estimate the travel time for each road segment of a trip. As a road segment can be part of several trips, the estimation must be done for all trips of a time window at the same time. Like this, they can divide the known travel time of a trip to the individual road segments so that the average relative error (the difference between the actual travel time of a trip and the summed up travel time of the estimated values for each road segment) is minimized. This is done every hour, and therefore 24 different travel time estimations surface as a result. As only 91.7% of the streets of the road network form part of a trip, some road segments without information on the travel time remain. By using a weighted average of the surrounding segments with such information, the missing values are added. This estimated travel time is then used in their fastest path algorithms. Their approach is a good alternative if there is a lack of information in GPS trajectory points, but it only partially represents the real-world circumstances as only two of possible dozens to hundreds of GPS signals per trip are considered.

2.4 Research gap

The presented literature review shows the variety of ride-sharing systems that already have been studied and how they can differ in their objective, their provider, the used algorithm, its performance, and the overall architecture of the model. Nevertheless, there is still room for further research. Besides showing the potential of their system in a sense of e.g. matching rate, total travel distance, or time savings, some studies do as well analyse the influence on the overall results of changes in parameters like the number of passengers per vehicle, the flexibility of the users, the complexity of the algorithms, or user-defined constraints. Considering the scope of research on this problem thus far, what has not yet been analysed is the influence of real-world circumstances like travel speed or traffic congestions on ride-sharing algorithms.

As mentioned in the introduction, the majority of ride-sharing studies assume that the time to reach a destination on a road network only depends on the distance. Just a few studies include information on the travel speed of the vehicles. But except the studies of Wang et al. (2018) and Santi et al. (2014a), all of them assume a constant speed for each road segment, meaning that the traffic state is not taken into consideration. As ride-sharing can not only influence traffic congestions, but is as well affected by it, considering information on traffic state is important to solve the ride-sharing problem more realistically. Two pick-up locations might be close in space, but if the connecting road segment is congested, it could make more sense to share a ride with another user where the traffic state is better. This would not be considered if traffic state information is not used and, therefore, the circumstances of the road network would not be represented realistically enough. Wang et al. (2018) and Santi et al. (2014a) developed a ride-sharing system that uses such information, but as explained in the traffic state estimation section of this chapter, they either compute it based on a different dataset than the analysed one, or they only use a small part of the available GPS signals. Furthermore, they simply include it in their algorithm but do not analyse the influence such information could have on the overall results of a ride-sharing system.

Therefore, this work tries to fill the described research gap by estimating the traffic state based on all the GPS trajectory points that are used in the proposed taxi ride-sharing system, subsequently including this information in the matching process, and finally analysing its influence on the identified ride-sharing paths. As a second contribution, a new similarity measure is introduced to speed up the matching process and hence improve the performance. Last, a taxi ride-sharing system considering real-world circumstances has, to the best of found knowledge, not yet been applied to the used dataset of the city of Chengdu, China. Applying ride-sharing methods to new cities is always useful as each road network has its own characteristics, and therefore can, deliver new findings for the research field of ride-sharing.

3. Research objective

This work aims to develop a framework for the identification of potential ride-sharing paths from GPS taxi trajectory data by considering the traffic state of the underlying road network and implementing a newly developed similarity measurement. By doing so, an attempt is made to solve the ride-sharing problem efficiently and more realistically. The method is only based on GPS taxi trajectory data and road network data. No further inputs are needed. The framework contains all the steps from map-matching the GPS signals, estimating the traffic state to identifying suitable ride-sharing paths. The new approach is based and evaluated on historical data, and thus represents a static ride-sharing system. It is applied to a GPS taxi trajectory dataset of the city centre of Chengdu, China. The overall objective of the system is to minimize the waiting time imposed on the passenger that joins second. Only a maximum of two trip request per taxi is considered for the matching process, as, explained by Cai et al. (2019), the benefit of allowing more than two trip requests to be matched is marginal in comparison to the increase in time consumption of the computation. By comparing the results of the new approach considering the estimated traffic state of the underlying road network and assuming an absence of traffic congestions, the influence of using traffic state information on ride-sharing methods is analysed.

3.1 Research questions

In this study, the following research questions are addressed by working out the framework and used to analyse the influence of traffic state information on ride-sharing systems:

1. How can traffic state information be estimated and included in the process of identifying potential ride-sharing paths?

Traffic state information is estimated based on the GPS taxi trajectory dataset and available in the form of average speed values for a particular road segment at a particular time of the day. This can be used to calculate the resulting travel time of the mentioned road segment. The travel time can be included either while computing a shared path of two trip requests that could be matched together, or as well to select the most suitable computed shared path for an analysed trip request from a set of potential ride-sharing paths. How exactly is this information obtained and where is it included best is addressed by this research question.

2. How can potential ride-sharing paths be efficiently identified from a large GPS taxi trajectory dataset?

The goal of developing a ride-sharing system is to identify the most suitable paths to be shared that fulfil the set constraints and follow the objective of the method. How this result can be efficiently achieved starting with raw GPS signals will be shown by the developed framework. By defining and implementing a new similarity measurement, it is attempted to improve the performance of the identification process of the potential ride-sharing paths, therefore making the system more efficient.

3. What is the influence of considering traffic state information in ride-sharing systems on its results?

This research question analyses the effect of considering information about the estimated traffic state in ride-sharing systems by comparing the results of the developed ride-sharing system between including traffic state information and assuming an absence of traffic congestions. Besides the identified ride-sharing paths, resulting measures like the matching rate, the average waiting time, total saved travel time, total saved driving distance, and the degree of saved CO_2-emissions can be compared. This allows to demonstrate the extent to which other studies are not representing real-world circumstances by assuming an absence of traffic congestions.

3.2 Hypotheses

Concerning the third research question, the following hypotheses were established before the analysis was started. They will be discussed in Chapter 7 based on the results of Chapter 6.

1. Less potential ride-sharing paths are identified when including traffic state information compared to assuming an absence of traffic congestions.

2. The average waiting time for the second passenger is higher when including traffic state information compared to assuming an absence of traffic congestions.

3. Savings in total travel time and total travel distance are smaller when including traffic state information compared to assuming an absence of traffic congestions.

4. Data

The created framework of this study is applied to real-world GPS trajectory data to analyse the influence traffic state information can have on ride-sharing systems. This data consists of two different datasets. One dataset contains the information about the taxi trips, meaning the GPS taxi trajectories, and the other dataset comprises of information about the underlying road network of the study area. In the following section, the study area and the two used datasets are described and its essential characteristics highlighted, to better understand the applied methods of Chapter 5.

4.1 Study area

The study area for this work is situated in the city centre of Chengdu, China. Chengdu is the provincial capital of Sichuan Province, located in south-west China. It is one of its major cities and serves as an economic, cultural, logistical, and technological centre for this region. It has a population of approximately 14 million in a total area of 12'390 km^2 (urban and rural area). The road network consists of a traditional grid-based structure in the centre with four ring roads connecting the different regions. (Qin, 2015)

Figure 7: Location of Chengdu in China, the provincial capital of Sichuan Province (Liu et al., 2014).

The city centre is located inside the first three ring roads. Limited by the extract where the GPS taxi trajectories are provided by Didi-Chuxing, the analysed study area only covers the upper part of the centre as it can be seen in Figure 8. This equals an area of 76.9 km^2.

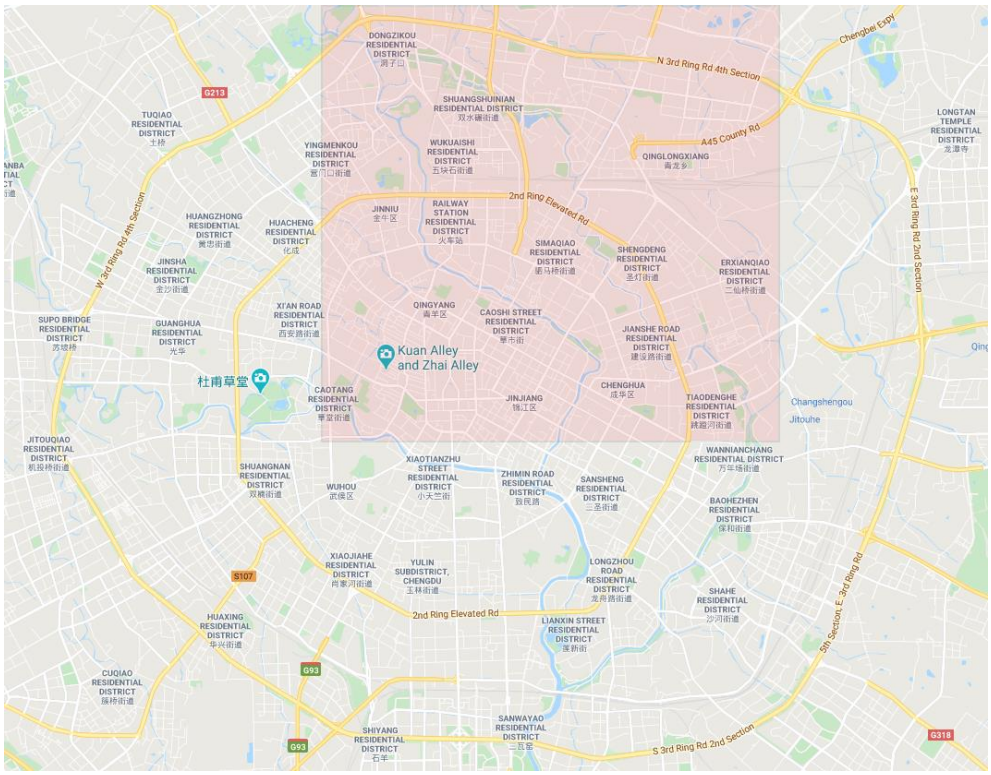


Figure 8: The city centre of Chengdu with its three ring roads and the grid-based structure of smaller streets. The study area of this work is illustrated by the red rectangle located in the upper part of the city centre. The shape of the study area is given by the availability of the data. (Source: Google Maps)

4.2 OpenStreetMap road network

The data of the road network of Chengdu is obtained by the open-source community OpenStreetMap (OSM) and downloaded as a shapefile with the BBBike Extract Service. As mentioned above, the extracted area has an extent of about 77 km^2. The dataset contains 3'136 road segments that are stored as an ordered set of nodes. These nodes can be start and end nodes or additional vertices that represent the shape of the road segment. The selected road network contains 1'038.3 km of road segments. The length of the individual segments differs from 1.4 meters to 6.1 kilometres. On average, a road segment has a length of 331.1 meters. The provided attributes for the road segments are listed in Table 3.

Name	Type	Description
osm_id	Integer	OSM ID of the road segment
name	Text	Street name in Chinese
ref	Text	Reference number or code of the street if available
type	Text	Road type of the segment
oneway	Boolean	Information if the segment is a one-way or a two-way street
bridge	Boolean	Information if the segment is a bridge
maxspeed	Integer	Information on the maximum allowed speed for the segment if available
length	Double	Length of the road segment in meters

Table 3: Attributes of the OSM road network dataset.

The attributes about the name, the reference number, and the maximum allowed speed are not given for all the road segments. Only a small part contains this information. The other attributes are always given. The segments can be divided into 23 different types. They are listed in Table 4; some of them are segments that can only be passed by foot and not by vehicle; some are not located inside the study area. Those are not considered in the further process anymore. As information on the maximum allowed speed is not available for each road type, these values must be added manually. By considering the available values and information provided by Wikitravel (2008), a maximum allowed speed value is set for each road type.

Road type	**Max. allowed speed**	**Considered in study**
bus stop	-	no
construction	-	no
cycleway	-	no
footway	-	no
living street	10 km/h	yes
motorway	100 km/h	yes
motorway link	60 km/h	yes
path	20 km/h	yes
pedestrian	-	no
primary	60 km/h	yes
primary link	60 km/h	yes
residential	20 km/h	yes
road	-	no
secondary	40 km/h	yes
secondary link	40 km/h	yes
service	30 km/h	yes
steps	-	no
tertiary	30 km/h	yes
tertiary link	30 km/h	yes
track	30 km/h	yes
trunk	80 km/h	yes
trunk link	40 km/h	yes
unclassified	30 km/h	yes

Table 4: The 23 different road types available in the OSM road network dataset and information on the maximum allowed speed as well as whether or not will the road type be considered in this study.

As there are many road types with equal or similar maximum allowed speed values, six new categories of road types are built to reduce the complexity of the dataset. Each category has one maximum speed value assigned, which will be later used in the analysis part. The new categories are shown in Table 5. Figure 9 illustrates how many road segments of the study area are assigned to each category. In Figure 10, all the road segments of this part of the road network are coloured concerning their category.

Zhang et al. (2015) analyse in their study the quality of the described OSM road network and show that the dataset for Chengdu has, based on the Shannon-Wiener index, high diversity between 2.13 and 2.46 and high road density between 3.35 and 18.42 km/km^2. The Shannon-Wiener index tells us how well distributed the different road types are. The highest score can be achieved if the number of road segments per road type is equal for all of them. In both categories, Chengdu is part of the group with the highest values for China. Therefore, the quality of this road network is assumed to be adequate enough for it to be used in this study.

Road type category	Max. allowed speed	Grouped road types
Living street	20 km/h	living street, residential, path
Motorway	100 km/h	motorway
Primary street	60 km/h	primary, primary link, motorway link
Secondary street	40 km/h	secondary, secondary link, trunk link
Tertiary street	30 km/h	tertiary, tertiary link, track, service, unclassified
Trunk	80 km/h	trunk

Table 5: The six newly generated road type categories with its maximum allowed speed.

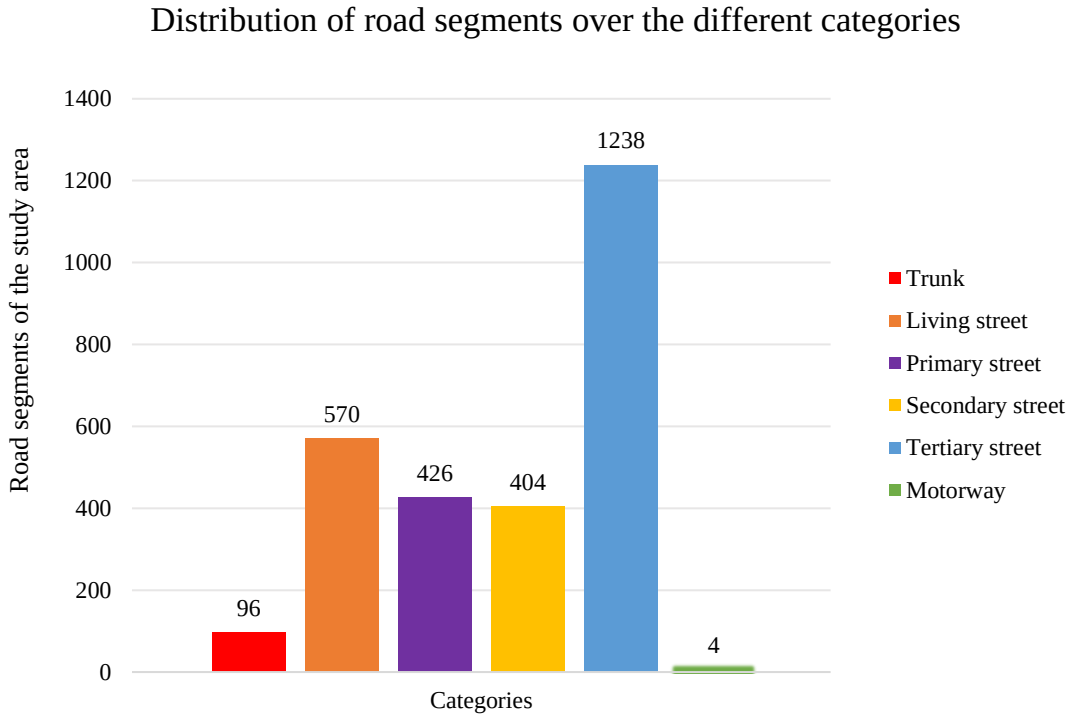


Figure 9: Statistical distribution of the number of road segment per road type category.

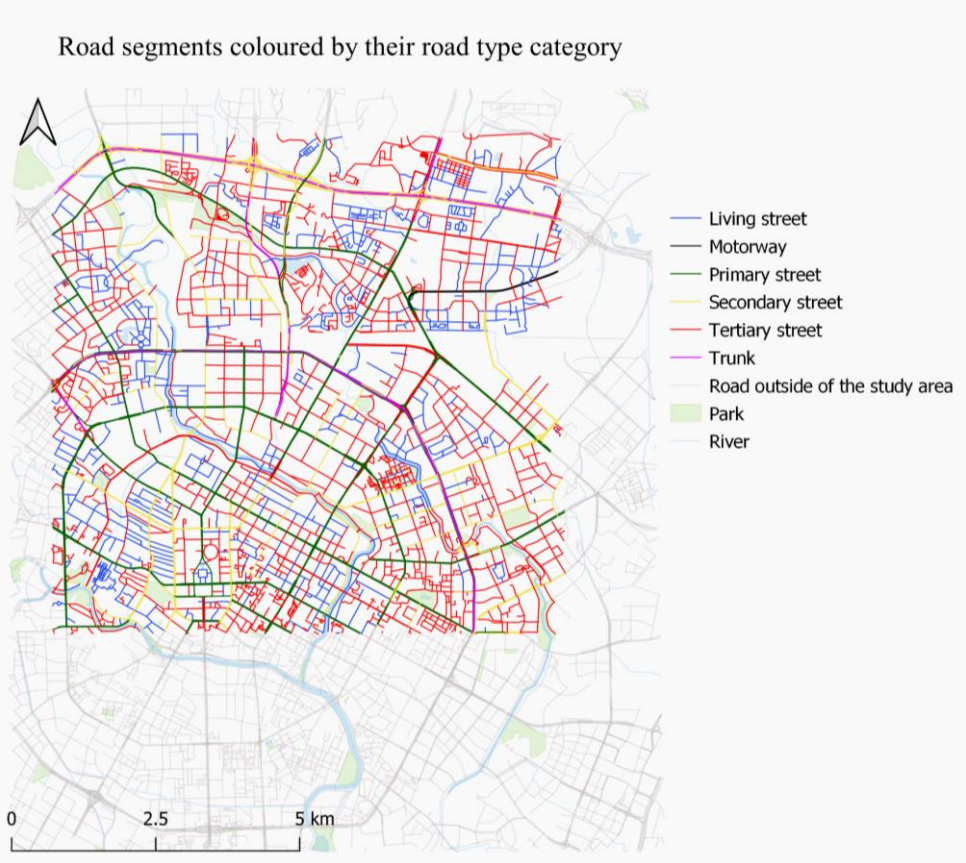


Figure 10: The OSM road network of the city centre of Chengdu. Each road segment inside of the study area is coloured according to its road type category.

4.3 GPS taxi trajectory data

The GPS taxi trajectory dataset used in this study is generated and provided by the Chinese company Didi Chuxing Technology Co. Through their GAIA Open Dataset Initiative, they share part of their collected data for scientific use. Didi Chuxing Technology Co. (in short Didi) is the biggest ride-sharing and -hailing company in China with over 400 million users (Ye, 2018). They even took over Uber in China and see themselves situated in a near-monopoly situation on the Chinese market (Crabtree, 2018). The obtained data is a trajectory dataset of the above-presented study area recorded in the year 2016. It is available in the form of two CSV files. As already mentioned, GPS records outside of the study area are not provided.

The first file contains anonymized information about the routes of their vehicles collected from the 1st until the 30th November 2016 in the form of trajectories. This means that each point of these trajectories has an entry in this CSV file. Table 6 shows the stored attributes for each trajectory point and contains an example entry. The taxi ID is used to identify the vehicle and the order ID to match each trajectory point with a taxi trip. The longitude and latitude represent the coordinates of each GPS record. The time stamp shows the exact time when the GPS signal was recorded. This time is given as a Unix Time Stamp. This temporal reference system counts the number of seconds since its origin on the 1st January 1970 (Cox & Little, 2020). For the given example in Table 6, this means that the Unix time 1477959044 stands for the 1st Nov. 2016 at 00:10 a.m. This is the Coordinated Universal Time (UTC) and must then be translated to the China Standard Time (CST). So, the example GPS signal was recorded at the 1st Nov. 2016 at 08:10 a.m. in the city centre of Chengdu.

Taxi ID	5a25883efb40a7246962ea767ed6f065
Order ID	914cb27d35ba86df0ee95051c0b411f2
Longitude	104.05447960734
Latitude	30.6878976313132
Time stamp	1477959044

Table 6: Example GPS record with its attributes stored in the first CSV file of the trajectory dataset.

The second file contains information about the individual trips and can be linked to the first file by the order ID. Table 7 shows all its attributes. The start and stop time are again given as a Unix Time Stamp and represent the time it took for a taxi to drive a trip. The pick-up and drop off longitudes and latitudes stand for the coordinates of the first and last trajectory point of each trip, in other words their GPS signal.

Order ID	914cb27d35ba86df0ee95051c0b411f2
Start time	1477957963
Stop time	1477959332
Pick-up longitude	104.065129
Pick-up latitude	30.712609
Drop off longitude	104.04777
Drop off latitude	30.68346

Table 7: Example GPS record with its attributes stored in the second CSV file of the trajectory dataset.

While the OSM road network is projected in the WGS-84 coordinate system, the data of the trajectory dataset is projected in the GCJ-02 coordinate system. This is the Chinese coordinate system that is used in its territory. The trajectory points were recorded every 2 to 4 seconds and, therefore, the GPS devices have a high frequency. Around 32 million trajectory points are recorded by these devices per day. These points have been collected by approximately 35’000 taxis (calculated for one day), which equals a total of about 180’000 trips a day. Summing this up to a month, there would have been around 960 million trajectory points and approximately 5.4 million different trips recorded. These numbers show not only the amount of information available for the study, but also indicate the big size of the dataset. The latter is a crucial point for the analysis conducted in this study. As using all the available data would go beyond the scope of this work, only GPS taxi trajectory data of one day will be used from now on. The analysed day is the 1st November 2016, which is a Tuesday. Data from all the 24 hours of that day are analysed.

Figure 11 shows the distribution of the requested trips over the analysed day. It can be seen that during the night, fewer taxi trips were requested on this day. During the morning rush-hour, the number of requested trips rises and remains high until the end of the evening rush-hour. There are two peaks. One in the morning and one in the afternoon. The most trips were requested in the afternoon with approximately 3’250 taxi trips.

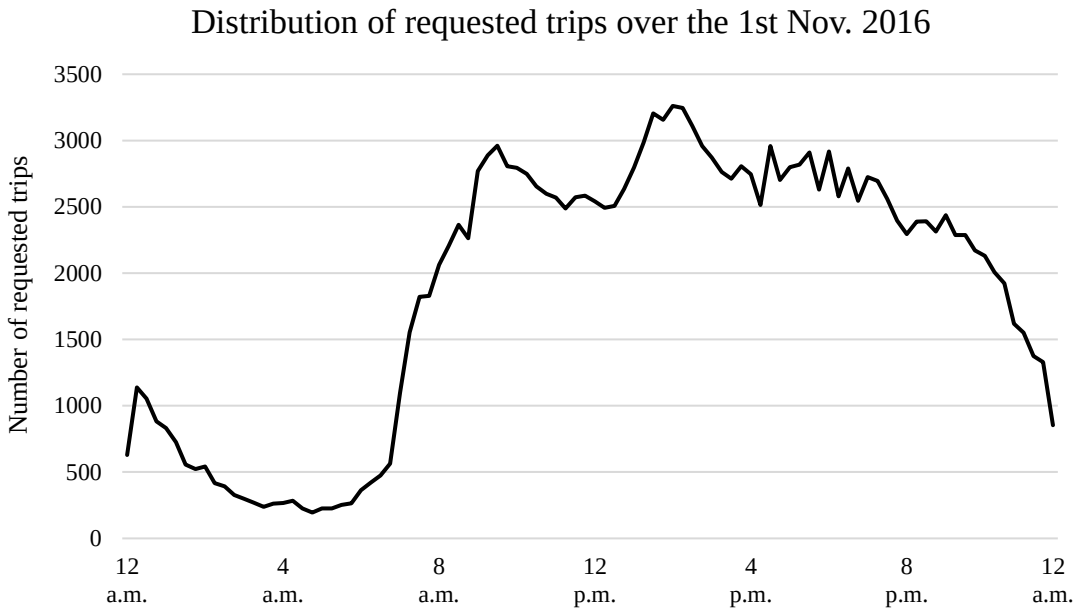


Figure 11: Number of requested trips over the 1st Nov. 2016.

By locating and visualising the trajectory points of these taxi trips in Figure 12, it can be surmised that all the GPS signals were recorded inside the study area. An example trajectory is displayed in Figure 13 to show the density of the available trajectory points for each trip.

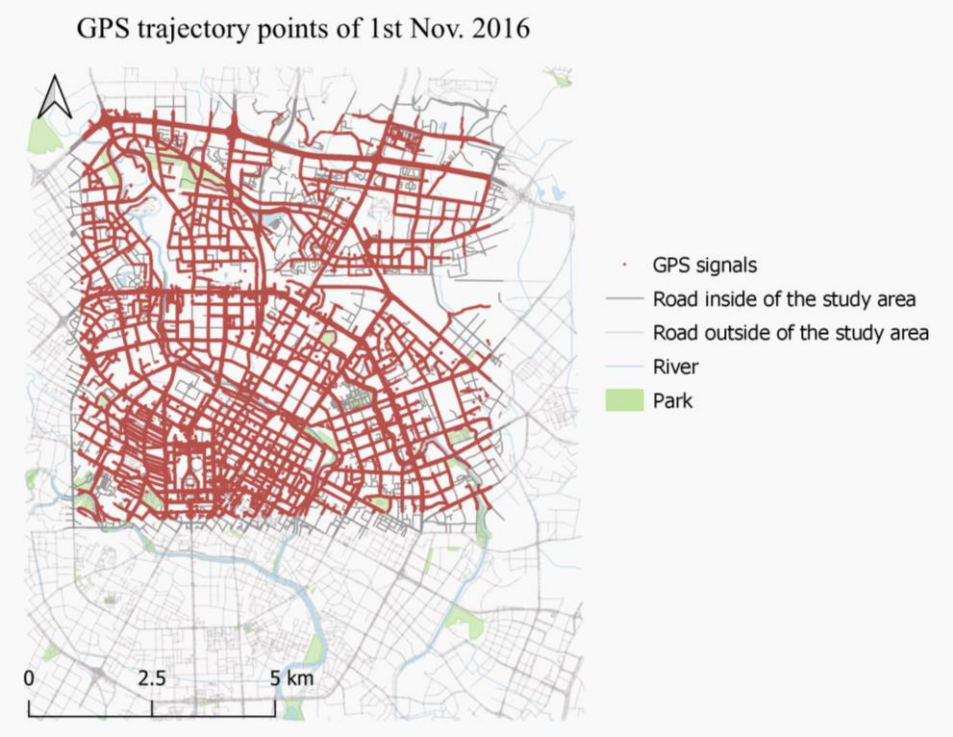


Figure 12: Visualisation of the trajectory points recorded on 1st Nov. 2016 inside of the study area. No GPS records are available outside of the study area.

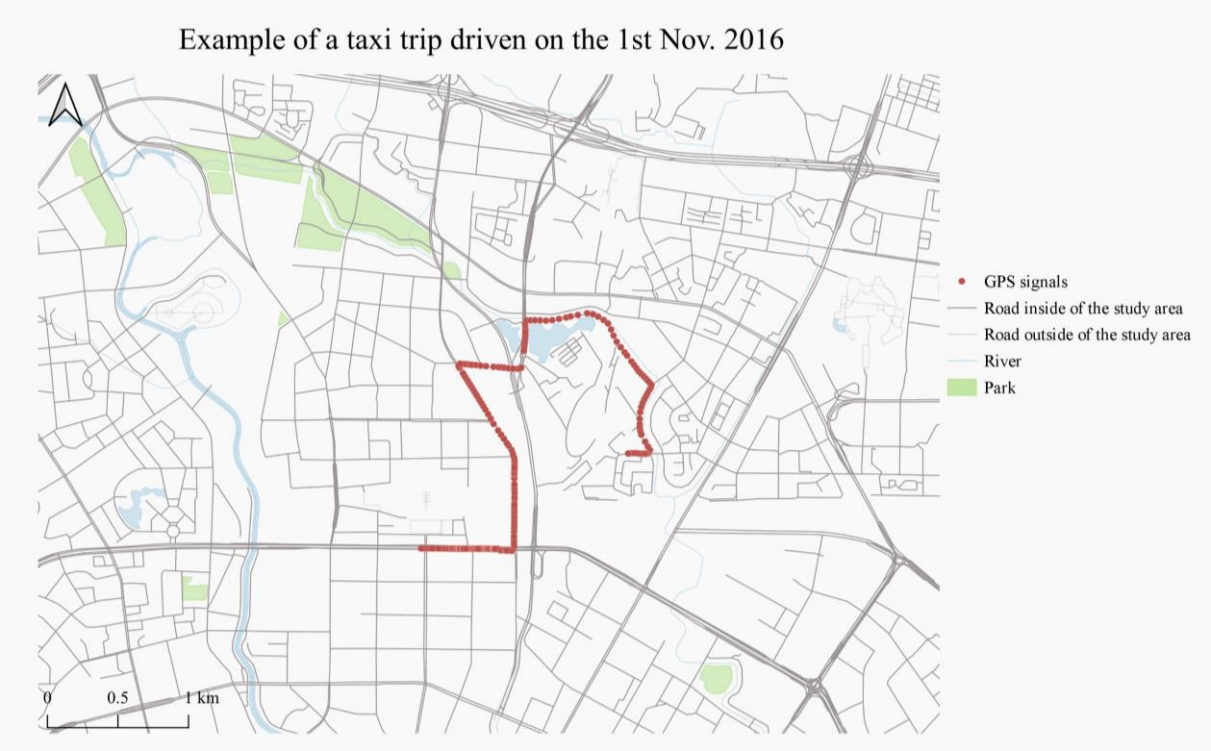


Figure 13: Visualisation of an example taxi trip represented by 385 trajectory points to show the density of the GPS records.

5. Methods

This study aims to identify potential ride-sharing paths from GPS taxi trajectory data and analyse the influence of traffic state information on the ride-sharing results by including the estimated traffic state of the underlying road network into the matching process. To achieve this, a framework is built to explain the steps involved in identifying suitable ride-sharing paths from raw GPS records. This framework is then applied to the previously presented real-world GPS taxi data to conduct the mentioned analysis. By studying the different ride-sharing systems of the related work presented in Chapter 2, suitable insights about the main steps of identifying ride-sharing paths have been gained. Furthermore, opportunities for improvement in the individual steps and the presented research gap were detected. These assist in analysing the best way to include the traffic state information into the system and to build the framework presented in Figure 15.

The four main steps applied in this work are illustrated in Figure 14. First, the real-world GPS and road network dataset must be pre-processed to be used in the next part. Then, these two datasets are map-matched to locate the driven taxi trips on the road network. Using this, the traffic state of the network is computed to later being included in the matching process. This allows identifying the potential ride-sharing paths and analyse the influence that traffic state information can have on ride-sharing systems and their results.

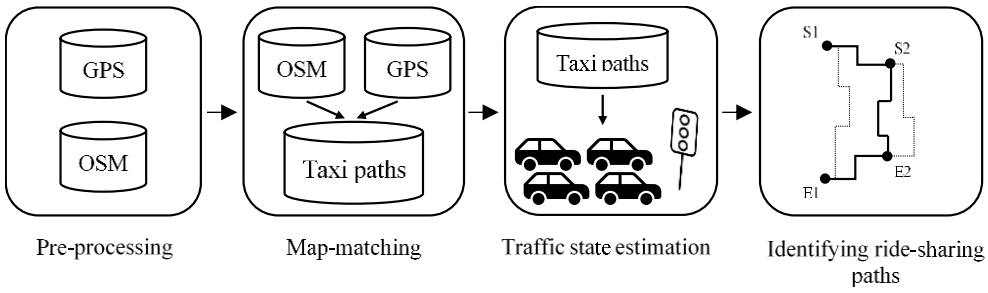


Figure 14: The four main steps of the process of identifying potential ride-sharing paths from GPS taxi trajectory data applied in this work.

Figure 15 illustrates the framework of this study that contains additional sub-processes of the mentioned four main steps. In the pre-processing step, both the GPS and the road network data are slightly changed. Coordinate transformation and resizing form part of these processes. The map-matching step also includes calculating the distance of each trajectory point to the start of its map-matched path, which is later used in the traffic state estimation. There, first, a speed value for each road segment gets calculated and then used for the interpolation. This allows assigning a speed value to each road segment in the network. Finally, the travel time per road segment per time window is computed. In the fourth step, the similarity between a subset of the requested trips is calculated and for similar ones, the fastest shared path is computed. By finding a local optimum, for each trip the ride-sharing path that accomplishes the objective of minimizing the waiting time is identified. By comparing the new method between using traffic state information and assuming an absence of traffic congestion the effect such information can have on ride-sharing systems is analysed. The remainder of this chapter delivers detailed explanations of all the mentioned methods elaborated and applied in this work and an experimental design describes how these methods are used to analyse the influence of traffic state information on ride-sharing systems.

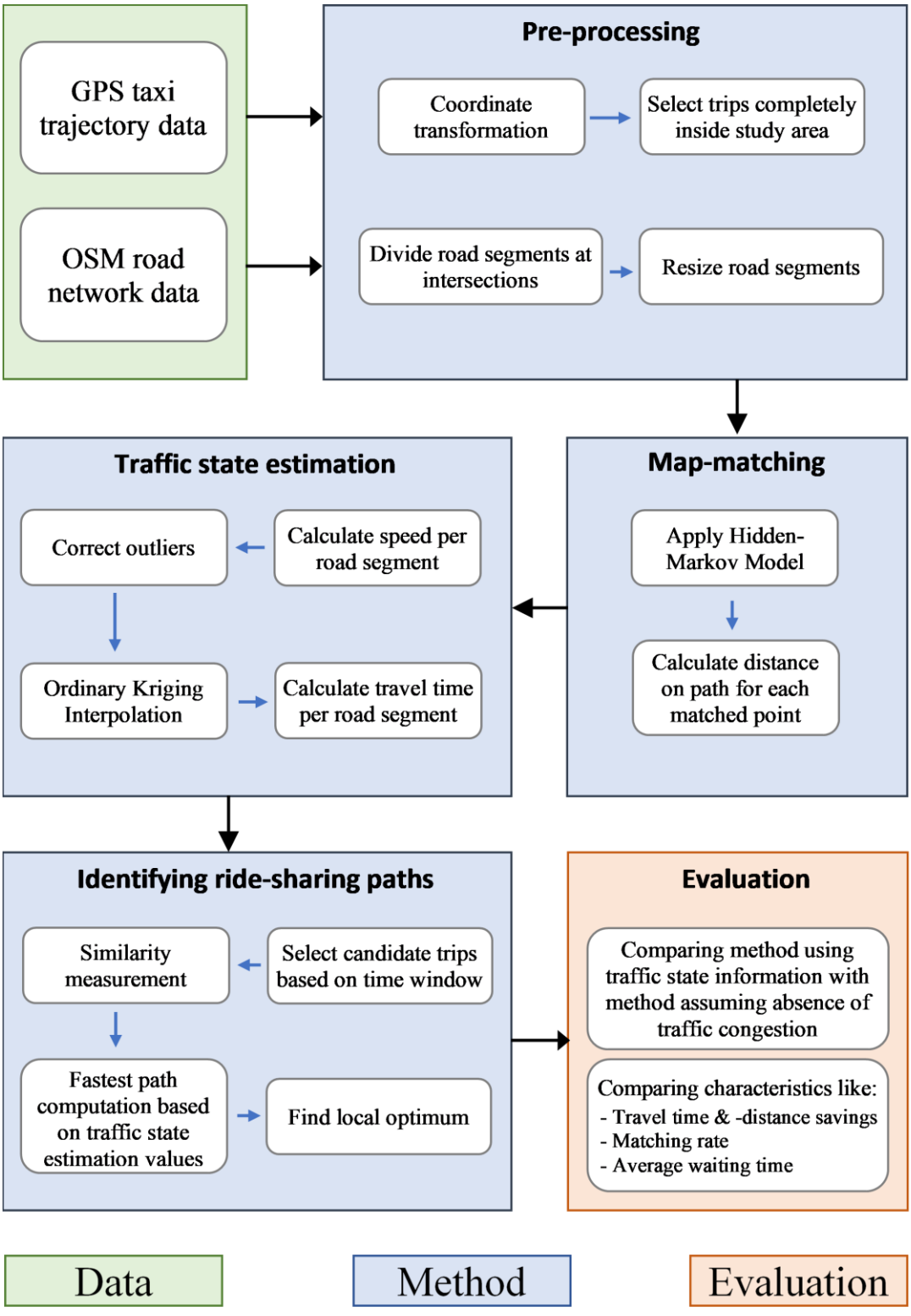


Figure 15: Framework including all the elaborated and applied methods in this work.

5.1 Tools

The methods and processes that are presented in this chapter are computed using different tools. For the coordinate transformation in the pre-processing step, the software environment R 3.5.1 is used (R Core Team, 2018). All the data, original and modified, are stored with the database management system (DBMS) PostgreSQL 12. PostgreSQL is an open-source relational DBMS that originally does not provide a spatial extension, but the open-source solution PostGIS version 3.0 can be installed so that this DBMS can be used with spatial data (Piórkowski, 2011). Most of the analysis part of this study is processed with Python 3.6.10. The code is written using PyCharm 2019.3.1, an integrated development environment (IDE). Table 8 gives an overview of the most relevant Python modules used in this work. Additionally, two Geographic Information Systems (GIS) are used for visualising purposes and analysations. QGIS 3.12.2, an open-source software, is used for all the visualisations and with ArcGIS 10.7 the results of the in Python computed methods are checked. Furthermore, its Python module arcpy is included in several scripts, e.g. the map-matching or interpolation script.

Module
arcpy
geopandas
mapmatcher
math
matplotlib
networkx
pandas
psycopg2
shapely

Table 8: List of most relevant Python modules used in the analysis part of this work.

5.2 Pre-processing

Before an analysis can be conducted, the data typically must be pre-processed. In this study, this needs to be done for both datasets, the GPS trajectories and the OSM road network. First, the pre-processing of the road network and later the pre-processing of the GPS trajectories are explained in detail.

5.2.1 OSM road network

As described in the data section, the original OSM road network consists of road segments which can significantly differ in their length. There are very small road segments of less than 2 meters length and some very large ones of more than 6 kilometres length. As the traffic state is calculated per road segment, it would lead to an inaccurate representation of the traffic conditions if a road segment of 6 km length would only have one traffic state information value for the whole segment assigned. Additionally, the original road segments do not end at intersections, which again would lead to inaccurate results, as an intersection can have a strong influence on the average speed of vehicles on a road segment. Considering this, the original OSM road network is reshaped in two stages.

First, the problem of road segments not divided by intersections is addressed. Each road segment that contains an intersection is divided into two individual segments. This leads to an increase in the number of road segments, but the newly created segments are still referenced to the original road segment by its OSM ID, and, therefore, still contain all the information on e.g. the road type or the maximum allowed speed. As especially long road segments are prone to be divided by this method, the problem with the different lengths is addressed as well. To solve it entirely, in the second step, each road segment that is longer than 500 m is divided into two segments of equal length. This is repeated until no road segments longer than 500 m are left.

The pre-processing of the road network leads to a new total of 8'368 road segments with an average length of 124.1 m. The total length of the road network remains the same and the longest road segment is now 499.8 m long. The statistical distribution of the different road type categories mentioned in the data section has slightly changed, as not all road types are affected equally by these two reshaping steps. How this distribution has changed is visualised in Figure 16.

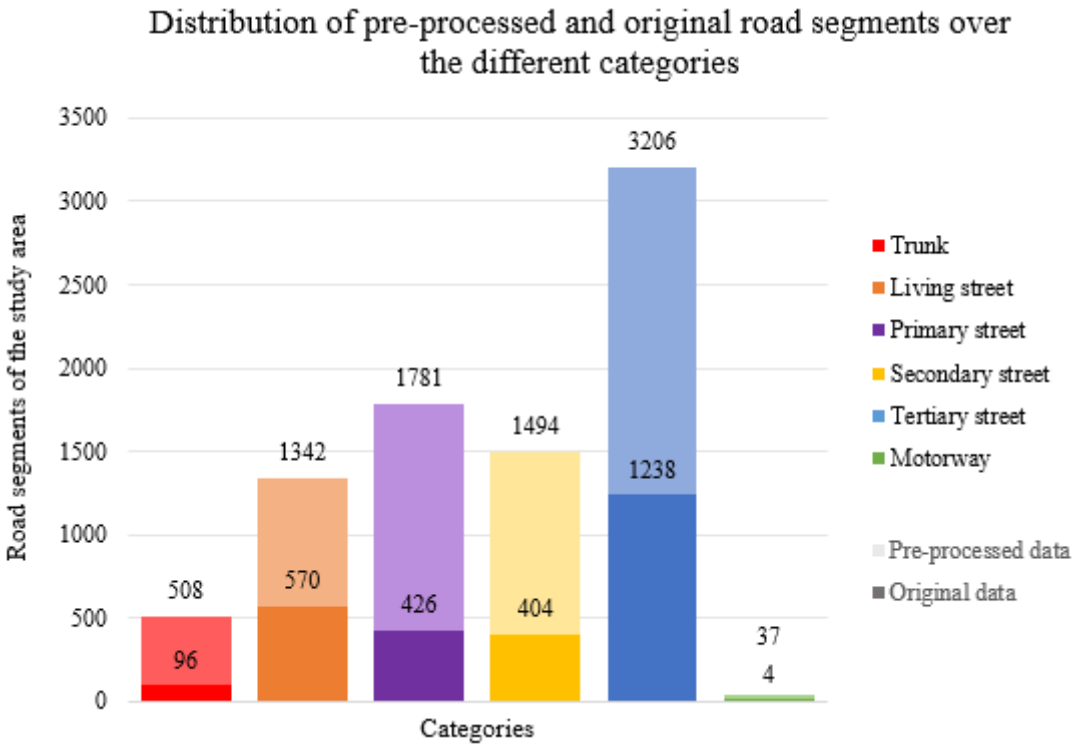


Figure 16: Visualisation of the total number of road segments per road type category. The dark colours represent the numbers for the original data and the light colours the ones for the pre-processed data. The distribution slightly changed as now there exist e.g. more primary streets than living streets.

5.2.2 GPS taxi trajectory data

The original GPS taxi trajectory dataset provided by Didi is recorded in the GCJ-02 coordinate system. This is a Chinese coordinate system, which differs from the common WGS-84 system by an applied shifting algorithm. This algorithm can be used to protect the security of China's geographic information and thus using it in combination with data stored in the WGS-84 coordinate system would produce position errors. (Jia et al., 2016)

As the OSM road network dataset is stored in the WGS-84 coordinate system, such position errors are a problem. To solve this, a coordinate transformation must be executed on the trajectory dataset, so that both datasets are stored in the same coordinate system. Lin (2018) published an in R written function on GitHub, a collaborative software development repository, to cope with this transformation. It removes the position shift mathematically and delivers very reliable results. As one day of the dataset already contains around 32 million trajectory points, this transformation can be time-consuming. To speed up the process, these 32 million points are divided into several subsets and then reunited again projected in the WGS-84 coordinate system. An example of these position errors and the trajectory points after the coordinate transformation is shown in Figure 17.

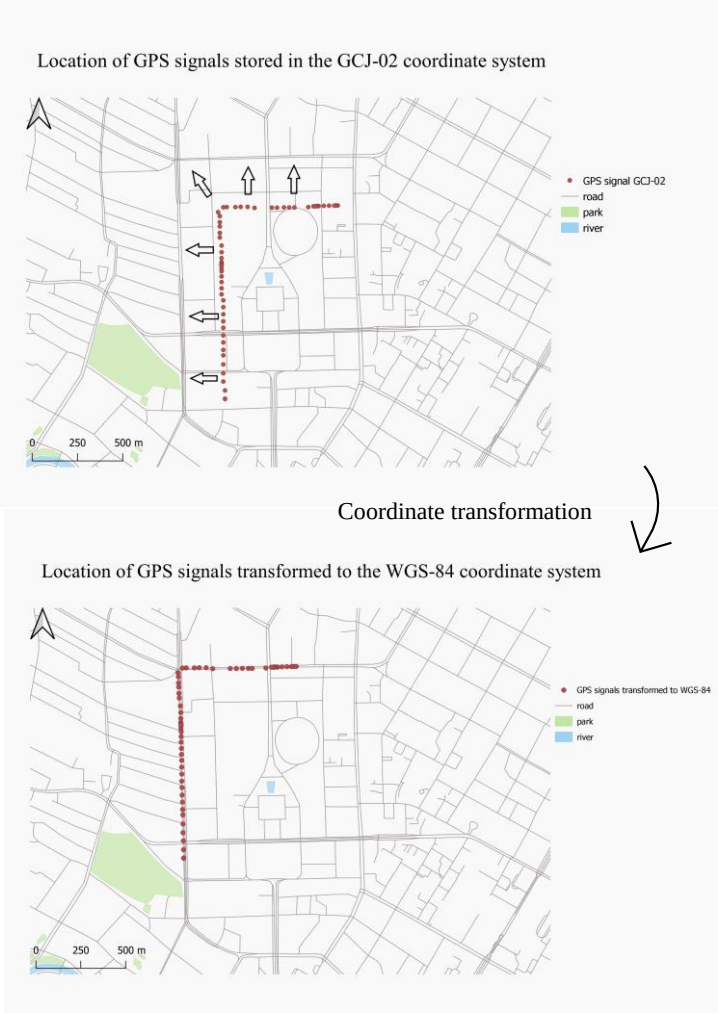


Figure 17: Example taxi trip stored in the GCJ-02 coordinate system (up) and after the transformation to the WGS-84 coordinate system (down), same as the underlying road network. The arrows on the upper hand indicate the effect of the applied shifting algorithm of the Chinese system.

In the second pre-processing step, the transformed GPS trajectories must be filtered by their location. As already mentioned in the data section, Didi only provides data inside the study area, but the taxi trips do not have to stop at the border of this area. This means that a taxi trip can leave the study area and stop outside of it or even return to the study area. The stored GPS signals cover only the part inside the area and the path driven outside of it is missing. Figure 18 shows an example of such a situation, where the taxi trip leaves the study area for a while before returning and finishing the trip inside the study area. Here only the two single parts inside the area are recorded respectively provided and no information is given about the driven path outside the study area.

Figure 18: Example trajectory where part of it is missing because only GPS signals inside the study area are provided by Didi. The trip starts north, leaves the study area, and finishes in the south. Using such trajectories would lead to incorrect paths and potential errors.

Considering these incomplete trajectories in the analysis part would lead to incorrect paths and could produce errors. To solve this, only taxi trips that never left the study area are considered in this work. First, all trajectory points are connected by their order IDs and transformed into line features. Subsequently, a rectangle approximately 150 m smaller on each side as the study area is inserted and only the line features, and thus the trips, that are located completely inside this rectangle are kept. The buffer of 150 m is chosen as assuming a maximum vehicle speed of 110 km/h (100 km/h plus a set buffer of 10 km/h) and the longest time gap between two GPS signals of four seconds, a taxi could reach around 123 m outside the rectangle during this time. Therefore, if a taxi has left the study area but the created line feature still is completely inside this area (possible if the taxi leaves the area and does not return or both, the last GPS record before leaving the area and the first after returning to it, are connected) it will not be selected as it is not located completely inside the rectangle. Additionally, the dataset contains trajectories with a duration of less than one minute. As such taxi trips are not suitable to be considered in a ride-sharing study, they are deleted as well. The GPS signals are recorded every 2-4 seconds, so assuming again the slowest sampling frequency of four seconds would mean that a trip with a duration of one minute is represented by a minimum of 15 trajectory points. This is taken as the threshold to filter out these unsuitable short taxi trips. These pre-processing steps reduces the number of available taxi trips by 76.7%. The corrected dataset contains around 42’000 trips,

driven by 19’000 different taxis, represented by approximately 9 million trajectory points. The pre-processing of the GPS dataset might have an influence as well on the distribution of the taxi trips over the time of the day, but as it is shown in Figure 19, both curves are very similar.

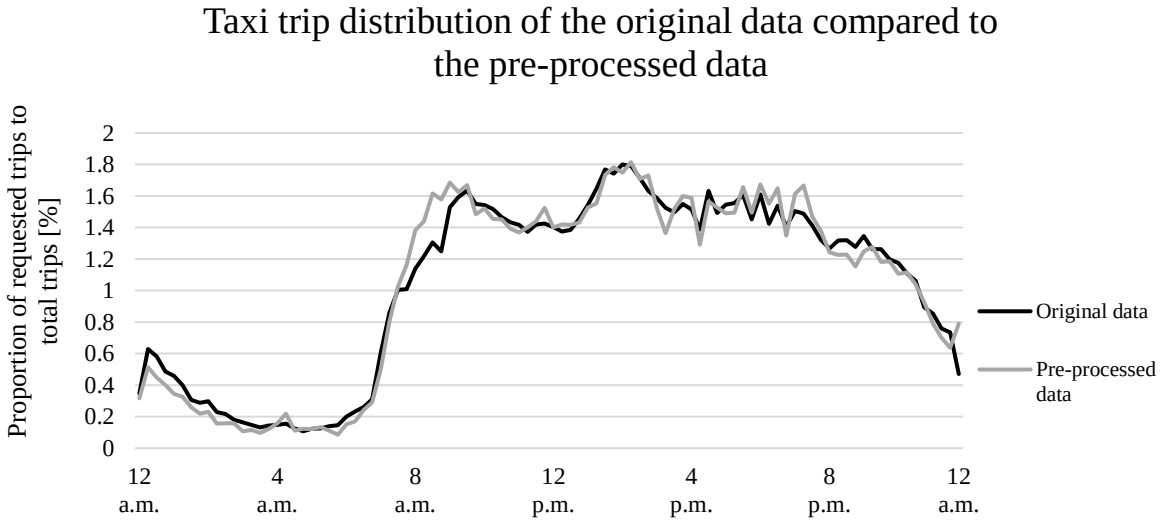


Figure 19: Comparison of the distribution of the number of taxi trips requested per time of the day between the original dataset and the pre-processed dataset. Displayed are the numbers in proportion to the total number of requested trips for both cases.

5.3 Map-matching

To work with the pre-processed data, it is important to know on which road segment each trajectory point is recorded. As already explained in Chapter 2, this process is called map-matching. From all the presented possible methods, in this work, the Hidden Markov Model (HMM) approach provided by the study of Newson & Krumm (2009) is applied. This method is selected because using a simpler approach would lead to inaccurate results, as the road network is dense, and the average GPS error is around 10 meters. Furthermore, their method can provide very reliable matches and is already implemented into a Python script, which is published by Schneider (2017) and freely available for scientific use. In the following section, an explanation of the HMM approach in general and the detailed implementation of Newson & Krumm (2009) is provided. Additionally, a small extension to the existing Python script is presented, to calculate the distance on the route of each map-matched trajectory point.

A Markov model is a statistical model that represents the special case of a Markov process and is named after the Russian mathematician Markov. Given a system with several states, it is used to calculate the probability that a change of state in the system occurs. In a normal Markov model, this probability depends on the transition probability, that shows how likely it is that a state transitions to another, and the initial state probability, that stands for the likeliness that a specific state is detected. In this case, the states refer to observable events. In a Hidden Markov Model, these events are not directly observable, but can be observed through another set of measurements. Differently from a normal Markov model, an HMM, therefore, includes an emission probability, that shows how likely it is, that a measurement can be observed given the state. (Rabiner, 1989)

Newson & Krumm (2009) apply this statistical model to the map-matching problem by using the knowledge about the connectivity of the road network. In their approach, the states of the HMM are represented by the road segments and the state measurements by the GPS signals. The aim is to find for each longitude/latitude measurement pair z_i the road segment on that the vehicle actually was driving. To reduce the computation complexity, a maximum search radius is defined that limits the candidate road segments for each measurement z_i. Each measurement z_1, z_2, … z_N has a specific amount of candidate road segments, on which the vehicle could have been driving. This is illustrated in Figure 20, where r_j stands for the individual road segments. As can be seen, already with only three measurements there are several possible combinations (paths). The method aims to find the most probable one of them. This path must respect both the reasonability of z_i being measured on r_j and that the road segments are connected like this based on the connectivity of the network. This is represented by the introduced emission probability (called measurement probability) and the transition probability. Those must be calculated for each GPS signal to find in the end the most probable path.

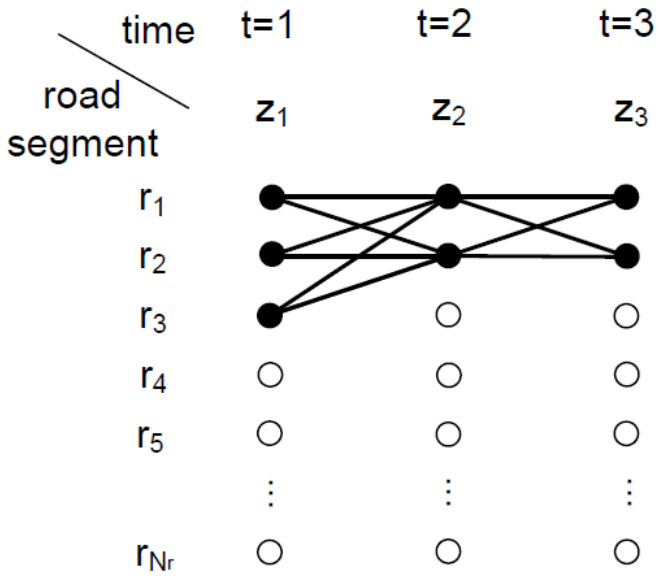


Figure 20: Visualisation of Newson & Krumm (2009). The black dots represent the candidate road segments and their connection stands for the different possible combinations between the three measurements z_i. The white dots are not inside the search radius and therefore ignored. The algorithm must find the path with the biggest probability in the sense of emission and transition probability.

The emission probability stands for the likeliness that a measurement is observed due to a certain state, meaning each road segment inside the search radius of z_i has an emission probability that shows the likeliness that z_i would be observed if the vehicle actually was on road segment r_j. This can be expressed as $p(z_i|r_j)$. In general, the further away a road segment is located from the measurement, the less probable it is to be the correct one. How far away a road segment is located gets measured by the great circle distance of the surface of the earth from the measurement z_i to the closest point on the road segment r_j, denoted as $x_{i,j}$. The great circle distance stands for the distance between two points measured on the surface of the earth while assuming the earth to be nearly spherical. The remaining distance for the correct match is assumed to be the GPS error, that can be modelled as zero-mean Gaussian. This is used to calculate $p(z_i|r_j)$, as this equals the probability density function of a Gaussian distribution in this form:

$$p(z_i|r_j) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-0.5(\frac{great\ circle\ distance\ ||z_i - x_{i,j}||}{\sigma_z})^2} \quad (1)$$

σ_z stands for the standard deviation of the GPS signals. If ground truth data is available, this value can be calculated exactly. In this work, ground truth data is not given and, therefore, this value must be set based on analysing a subset of the GPS signals and testing it in the Python script. In addition to the emission probability, the initial state probability π_i must be computed as well. This normally tells the likelihood that a state can occur. In this approach, for the initial state probability, the probability of the first road segment in each path is taken, based on the first measurement. In other words, π_i is given by $p(z_1|r_j)$.

The transition probability shows the likeliness that a vehicle was moving between two matched road segments given $z_{i,t}$ and $z_{i,t+1}$. This probability is calculated by considering the difference between two distances. First, the distance between the closest point on the first road segment and the closest point on the second is analysed. This distance is represented by the shortest path from $x_{i,t,j}$ to $x_{i,t+1,i}$ and stands for the route distance. The second is the already described great circle distance between $z_{i,t}$ and $z_{i,t+1}$. The smaller the difference between these two distances, the more probable is the analysed path because having to make complicated manoeuvres (what leads to a long route distance and a big overall difference) is unlikely in high-frequency GPS vehicle data. This relation between the difference in these two distances and the probability can be modelled as an exponential probability distribution in the form of:

$$p(d_t) = \frac{1}{\beta} e^{\frac{-d_t}{\beta}} \quad (2)$$

Where

$$d_t = \left| \left\| z_{i,t} - z_{i,t+1} \right\| great\ circle\ distance - \left\| x_{i,t,j} - x_{i,t+1,i} \right\| route\ distance \right| \quad (3)$$

and β again could be calculated if ground truth data would be available, but as before with the σ_z, this is not the case for this study. As β represents the tolerance of non-direct routes, this value must be chosen in a trade-off between accuracy and successfully matched GPS signals. In the end, $p(d_t)$ assigns the transition probability. The relation between the two distances used in this calculation is illustrated in Figure 21.

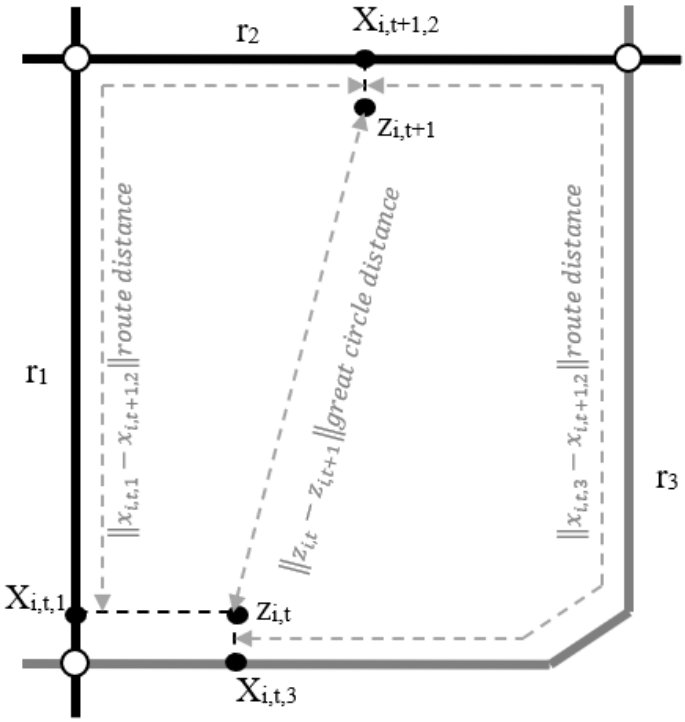


Figure 21: Explanation of the difference between the two distances based on Figure 4 of Newson & Krumm (2009). $z_{i,t}$ has two and $z_{i,t+1}$ has one candidate road segment. For each candidate there exists the closest point $x_{i,t,1}$, $x_{i,t,3}$ or $x_{i,t+1,2}$. The smaller the difference between the route distance of $x_{i,t,1}$ and $x_{i,t+1,2}$ respectively $x_{i,t,3}$ and $x_{i,t+1,2}$ and the great circle distance, the bigger the transition probability.

After calculating these two probabilities (emission and initial state probability taken as one) for each trajectory point of a taxi trip the aim is to find the optimal path with the highest probability. This path maximizes the product of emission and transition probability for each trajectory point of the trip. Here, they apply the Viterbi algorithm to find this optimal path for each taxi trip. The Viterbi algorithm is a dynamic programming algorithm that is useful together with an HMM as it does not have to compute the probability for each possible collocation in a network (in this case a road network) and is, therefore, less time-consuming (Theodoridis & Koutroumbas, 2009).

As the data used in this work is different from the one used in the study of Newson & Krumm (2009), the previously mentioned parameters must be newly set for the map-matching script. The search radius is set to 50 m due to the dense road network in the city centre. σ_z is set to 50 m as well. The average GPS error, analysed by a subset of the data, is about 10 m, but using 50 m as the parameter gives more successfully map-matched taxi trips. β, the last parameter, is set to 3000 m as sometimes there are big gaps between two trajectory points in the used dataset and these trips would not be map-matched with a lower β. Nevertheless, there can still be some taxi trips that produce errors and cannot be map-matched successfully.

The explained HMM map-matching approach provided by Newson & Krumm (2009) is, as already mentioned, implemented into a Python script, mainly based on arcpy, and provided on GitHub (Schneider, 2017). Only the road network, the trajectory points of a trip ordered by time and the mentioned parameters are needed as the input. The script calculates the presented emission-, initial state- and transition probabilities based on these parameters. The output of the original script is a complete path containing the matched road segments and its ID. For each trajectory point, the nearest road segment of this matched path is selected, which in this case must be the map-matched road segment, and its ID is added to the point. Like this, each trajectory point of a trip that is map-matched successfully contains the ID of the matched road segment. To improve the performance of this method unnecessary trajectory points of the input trip are removed so that only the important ones that keep the shape of the trip are left. By trying and resetting the degree of simplification it is assured that this process does not negatively influence the result but leads to a less time-consuming computation. Furthermore, an addition to the script is made, that calculates the distance each trajectory point has to the start of its assigned path. As the output of the map-matching algorithm is a complete matched path, each trajectory point can be located on that path and the distance from the start of the path to the trajectory point represents its distance value. It is important to mention that with this step, for each point the network distance instead of the Euclidean distance is calculated as every turn is considered in the computation. Especially in dense urban road networks, the Euclidean distance between two points can be smaller than the network distance, as the mentioned turns and junctions lead to a larger distance than just taking the direct way. This is illustrated in Figure 22 for better understanding. The calculated distances respectively the distance between two trajectory points are used in the traffic state estimation and, therefore, this step forms an important part.

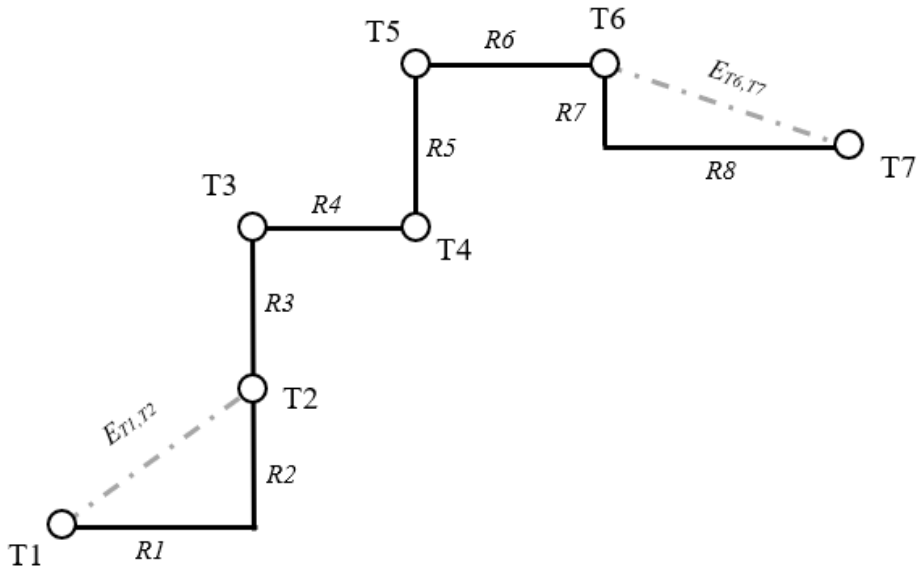


Figure 22: The black line shows the matched path containing eight road segments. The seven trajectory points are located on the route for the distance computation. Using the network distance gives ΔT2-T1 = *R1+R2* and using the Euclidean distance gives ΔT2-T1 = $E_{T1,T2}$. The former is larger as it considers the involved turn. The same situation appears at T6/T7. This shows that working with the Euclidean distance would lead to inaccurate measurements what later would affect the quality of the traffic state estimation.

5.4 Traffic state estimation

The main goal of this work is to identify potential ride-sharing paths including information on the traffic state of the underlying road network and analyse the influence such information can have on the results. The information on traffic state is not given or derived from any source and must, therefore, be estimated first. As one of the contributions of this work, in the following section, it is presented how traffic state can be estimated from raw GPS taxi trajectory data. Information used from the dataset is the vehicle speed, the maximum allowed speed per road segment, the type of the road segment, and the length of it. Normally GPS data used in traffic state estimation or prediction studies contain information on the speed of the vehicle at each GPS record. Unfortunately, such information is not given in the used dataset of this work. Therefore, the first step is to estimate the vehicle speed at each GPS signal. Subsequently, an average speed value per time window per road segment is computed. This gives information on how fast a vehicle is driving on average at a particular time on a particular road segment. The final estimated traffic state is then represented by the travel time for each road segment. In this section, first, the vehicle speed calculation method is explained in detail, then an interpolation approach is presented to estimate the traffic state on every road segment of the network and finally the method on computing the travel time is described.

5.4.1 Vehicle speed

To calculate how fast a taxi was driving at each GPS record, the pre-processed and map-matched data is used. As each trajectory point comes with information about the exact time it was recorded and thanks to the map-matching process as well with information about the distance to the start of the trip, the differences in this information between several points can be used. In other words, the difference in the time stamps and the distance of two trajectory points is used to calculate the vehicle speed. Let us consider the situation illustrated in Figure 23, where a trip only contains trajectory point A, B, and C. Each trajectory point contains information about its exact recording time, given in seconds, and the distance to the start of the trip. Furthermore, the ID of the map-matched road segment is stored for each point as well. How the vehicle speed is calculated depends on the position in the order of the trajectory points.

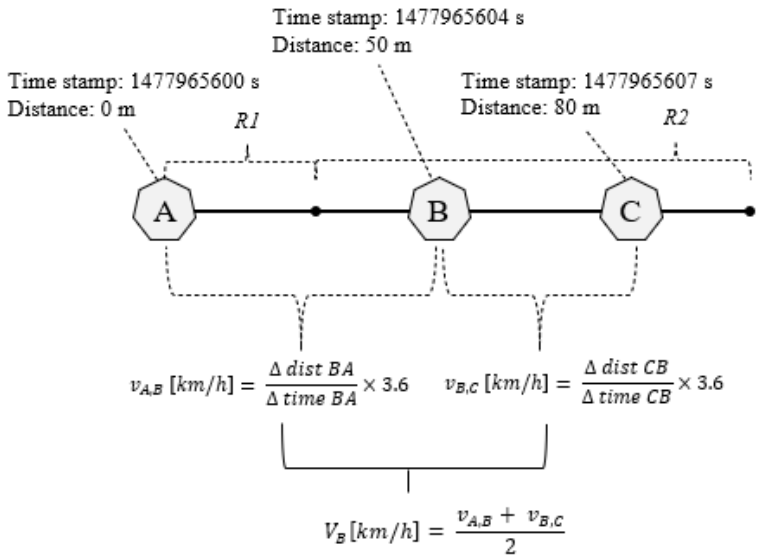


Figure 23: Explanation of the vehicle speed calculation for trajectory point B based on the driven speed between A-B and B-C. The speed is derived by using the differences in time and distance.

To calculate e.g. the speed of trajectory point B, the average of the speed between point A and B and point B and C is taken. The speed between A and B is simply calculated by:

$$vehicle\ speed\ [km/h] = \frac{(distance\ of\ point\ B - distance\ of\ point\ A)}{(time\ stamp\ of\ point\ B - time\ stamp\ of\ point\ A)} \times 3.6 \quad (4)$$

The distance stands for the in the map-matching part calculated network distance to the start of the trip, meaning the real distance the taxi was driving, which is important to get a more accurate speed value. As the distance is given in meters and the time stamp in seconds, the result must be converted into km/h to be in the same format as the given maximum allowed speed per road segment. Equation 4 is used the same way for calculating the speed between point B and C. Considering the values given in Figure 23, the vehicle speed between point A and B is 45 km/h and between B and C 36 km/h. The average of these two speed values, 40.5 km/h, represents how fast the vehicle was driving at the GPS record B. While calculating the vehicle speed for point A or C, there is only one other trajectory point available. Because of this, the speed for point A or C is not the average of its surrounding speeds, but just the vehicle speed calculated between point A and B (45 km/h) respectively between B and C (36 km/h).

The described method is used to calculate an average speed value for each road segment based on the trajectory points' vehicle speed. As the trajectory points are recorded over a whole day and the vehicle speed at a particular road segment is not equal all the time, the 24 hours of the 1st Nov. 2016 must be divided into short time windows. For each time window, only information of trajectory points that are recorded inside this window is used to estimate the traffic state. Santi et al. (2014a) work with a time window of one hour and in Kong et al. (2013) they calculate the traffic state every four minutes. Regarding the time-consuming computations, in this work, a time window size between the two mentioned ones of 15 minutes is set. This means the 24 hours are divided into 96 equal intervals. Getting back to Figure 23, to calculate the average speed value at the given time window for the road segments R1 and R2, the following is done. First, all the trajectory points are filtered by their time stamp, so that only trajectory points inside the specific time window are considered. In this example, this means all the points that are recorded between 10 a.m. and 10:15 a.m. Here, only the three points are given, so no other points must be filtered out. To calculate the average speed for road segment R1, from the remaining points, only the ones which are map-matched to R1 are selected. Now, the average of all the speed values of these points represents the vehicle speed for road segment R1 between 10 a.m. and 10:15 a.m. As in Figure 23 only trajectory point A is map-matched to R1, the average vehicle speed of R1 is 45 km/h for that time window. R2 has two map-matched trajectory points and therefore the average of 40.5 km/h and 36 km/h represents the speed value for this road segment, which equals 38.25 km/h. Like this, the road segments contain information about how fast on average a vehicle drives at a particular time of the day, which represents the first way to estimate the traffic state. The second one will be represented by the estimated travel time, which is explained later.

While a taxi is picking-up or dropping off a customer, its speed value drops to 0 km/h. Because this is a special behaviour of the taxis' movement, it does not represent the speed a taxi normally could drive on the specific road segment. Therefore, these start- and stop movements of a taxi trip must be filtered out before computing the average speed value per road segment. If during the trip the taxi must stop, because of traffic or traffic lights, it can be seen as a representation

of the state at this location. To not mix these two patterns, only trajectory points at the start and the end of the trip are analysed. In more detail, starting with the first trajectory point, if the calculated speed value is lower than 20 km/h, all the following trajectory points are filtered out until its speed exceeds the 20 km/h threshold. The same is done for the last trajectory point of each trip. This threshold is set because the minimum of the allowed speed values of the network equals 20 km/h for streets of type residential (excluding living streets with a maximum speed of 10 km/h as they are very rare). So, speed values at the start an end of a trip below 20 km/h are classified as not normal, and thus, as start- and stop movements. For better understanding, such a situation is illustrated in Table 9.

a)

Trajectory point	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}
Speed [km/h]	0	5	21	4	0	0	15	28	37	45	23	10	0	0

b)

Trajectory point	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}
Speed [km/h]	-	-	21	4	0	0	15	28	37	45	23	-	-	-

Table 9: a) shows the calculated speed values per trajectory point for an example trip. b) shows the remaining speed values after filtering out the start- and stop movements. The speed values below 20 km/h during the trip are not classified as such a movement and, therefore, included in the average speed calculation for the road segments.

As either the GPS records can be erroneous or the distance calculation in the map-matching part can produce wrong values, unrealistically high speed values are possible after the vehicle speed computation. For each road type a maximum allowed speed is given and, therefore, too high speed values can be detected easily. Including them in further processes would lead to a bad representation of the estimated traffic state. Thus, these detected values must be corrected. To set the threshold to the maximum allowed speed per road type would be unrealistic, as vehicles may drive faster than allowed. In China usually after driving more than 10 km/h faster than allowed a fine is issued (Angloinfo China, 2020). Considering this, the threshold for too high speed values is set to 10 km/h above the given maximum allowed speed. If a calculated speed value of a trajectory point is more than 10 km/h higher than the maximum allowed speed, the calculated value is replaced by the maximum allowed one. So, if for some reason a trajectory point, that is matched to a road segment of type “primary”, has a speed value of 90 km/h assigned, this value is replaced by 60 km/h, the maximum allowed speed for roads of type “primary”. With the mentioned steps of filtering and correcting, each road segment where a taxi was driving has reasonable vehicle speed values for the 96 time intervals assigned.

5.4.1.1 Interpolation

The pre-processed road network contains 8’368 road segments, but not all of them are visited by a taxi during the 1st Nov. 2016. This means, that there exist road segments with no GPS records assigned to and, therefore, no speed value estimated. If focusing on the 96 time intervals individually, even less segments are visited. As a potential ride-sharing path can lead through each of the 8’368 segments, the traffic state must be estimated for all of them. This assures that there is no information missing while computing e.g. the resulting travel or waiting time of a shared path. The unvisited road segments get a speed value assigned by interpolating the already estimated values. This is done by applying a Kriging interpolation method.

Kriging is a geostatistical approach to estimate unknown values based on the autocorrelation in the distance to the measured values. The basic principle is that the further away the unknown point lies from the measured one, the less autocorrelated they are. The missing value can be estimated by the sum of a trend of observable factors and a random error component. The spatial autocorrelation is found in this error component. By plotting the variance over the lag distance of the known values, an experimental variogram is created that models this component. After fitting a curve to the variogram, important parameters can be detected, that serve as an input to the Kriging model. They are known as the nugget, the partial sill and the range. The nugget represents the variation that remains unresolved, the partial sill stands for the spatially correlated variance and the range gives the threshold distance at which the variance stabilizes. The curve that is fitted to the variogram and detects the presented parameters is represented either by an exponential, spherical or Gaussian model. Besides the model of the curve and the three parameters, the type of Kriging interpolation must be chosen as well based on the given data as an input to the function. If the mentioned trend of observable factors is known, then the Simple Kriging method can be used. If the trend depends on explanatory variables, the Universal Kriging approach should be applied. If explanatory information is lacking, the Ordinary Kriging method fits best. (Oliver & Webster, 2014 and Wang & Kockelman, 2009)

The work of Wang & Kockelman (2009) shows that the described Kriging interpolation method is a useful approach for transportation studies. They analyse the utility of Kriging to interpolate traffic count values on a road network in Texas, USA. As a pre-processing step, they divide the road segments based on their type into several groups so that only similar road segments are used to estimate the missing value of a road of the same type. Then, the described parameters and the model of the curve are analysed for each group individually and later used as an input to the global Ordinary Kriging model (assuming a lack of explanatory information). Additionally, they show that using the Euclidean distance instead of the network distance does not severely worsen the quality of the interpolation but significantly reduce the complexity. The result is a continuous distribution of traffic count values over the whole study area for each group of road segments. Based on the location of the road segments, the interpolated value of the underlying Kriging surface can be extracted and assigned to the segments' attributes.

Based on their study, the procedure of the interpolation of the speed values is implemented similarly in this work. The Ordinary Kriging model is applied to the estimated speed values per road segment. As it does not make sense to include speed values of road segments of the type "living street" to interpolate a missing value of a road segment type of "motorway" (on average totally different speed values), the road network first must be divided into sub-networks based on the road type categories presented in Chapter 4. Like this, the 8'368 road segments are divided into six sub-networks, shown in Table 10. The Ordinary Kriging is applied to each sub-network individually, to assure that only speed values of the same range are used to estimate the missing values. As the used speed values of a sub-network are similar over the whole study area, it is waived to apply a local interpolation and, thus, global Ordinary Kriging interpolation is implemented. Furthermore, the interpolation must be done 96 times, meaning for each time-window separately, as the estimated values per road segment differ in time. The global Ordinary Kriging is, therefore, run 576 times. To reduce the time consumption of this process, the parameters and curve model are analysed only once per sub-network and not 96 times. The time window used for analysing the variograms of the six sub-networks is between 12:00 p.m. and 12:15 p.m. The six variograms including the chosen parameters are illustrated in Figure 24.

Sub-network	Number of road segments
Living street	570
Motorway	4
Primary street	426
Secondary street	404
Tertiary street	1238
Trunk	96

Table 10: The six sub-networks used for the interpolation and the number of road segments per sub-network.

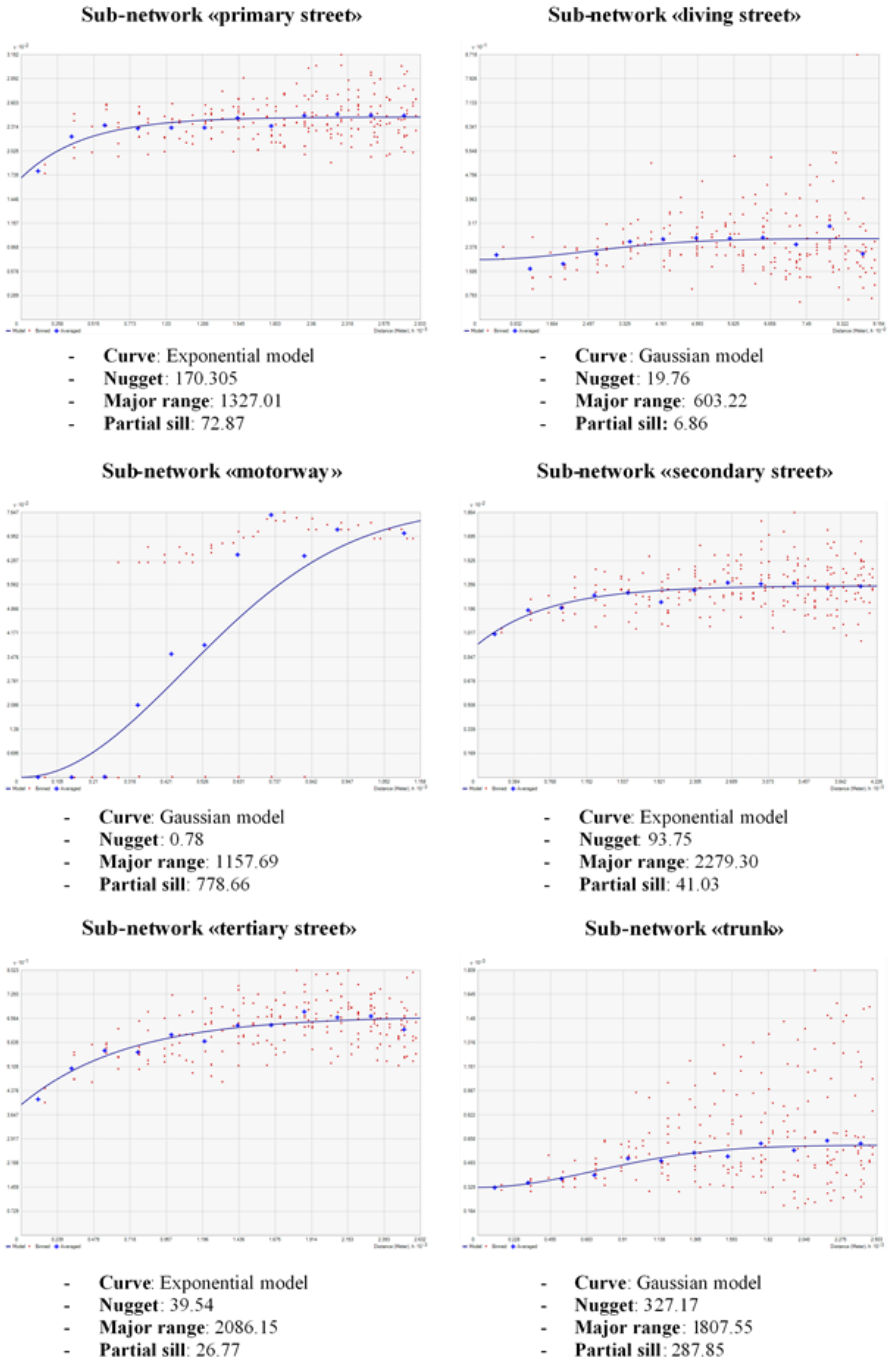


Figure 24: Variograms and used parameters of the six sub-networks. The parameters are analysed for the time window between 12:00 p.m. and 12:15 p.m. and taken as the input for the Ordinary Kriging interpolation method. A bigger size of the figure is given in the appendix of this work.

The already estimated speed values are assigned to a road segment, meaning they represent an attribute for a feature of the geometry type line. As these input measurements, on which the interpolation is based, must be given as point geometries for the Kriging function, this problem first must be solved. Therefore, the vertices of each road segment of each sub-network are extracted. Complex line features can have more than two vertices to store the shape of the line and thus, these vertices must not only be the start and end points. Each extracted point contains an attribute that keeps the connection to its road segment ID and an attribute that stands for the speed value estimated for this segment. The Ordinary Kriging interpolation is then run 96 times for each sub-network based on these point features with its speed value and the analysed parameters of Figure 24. The result is a continuous distribution of the speed values over the whole study area. The road segments with the missing speed values are represented as well by their vertices as explained before. The interpolated speed value of the continuous distribution is then extracted at the exact location of each vertex and assigned to its attributes. Through this step, each vertex of the sub-network has an estimated speed value assigned. In the end, the geometry type representing the road segments must again be a line feature and, therefore, the original road segments must be reconstructed. If a road segment where no taxi was recorded, is split into e.g. three vertices, then all of them can potentially become slightly different speed values assigned, as their distance to the given points is not equal. The reconstructed road segment of these three vertices can only have one speed value assigned, thus, the three newly estimated values must be transformed into one. This is done by computing the average speed value of the three individual ones. For better understanding, an example for the time window between 12:00 p.m. and 12:15 p.m. of the sub-network “trunk” is illustrated in Figure 25.

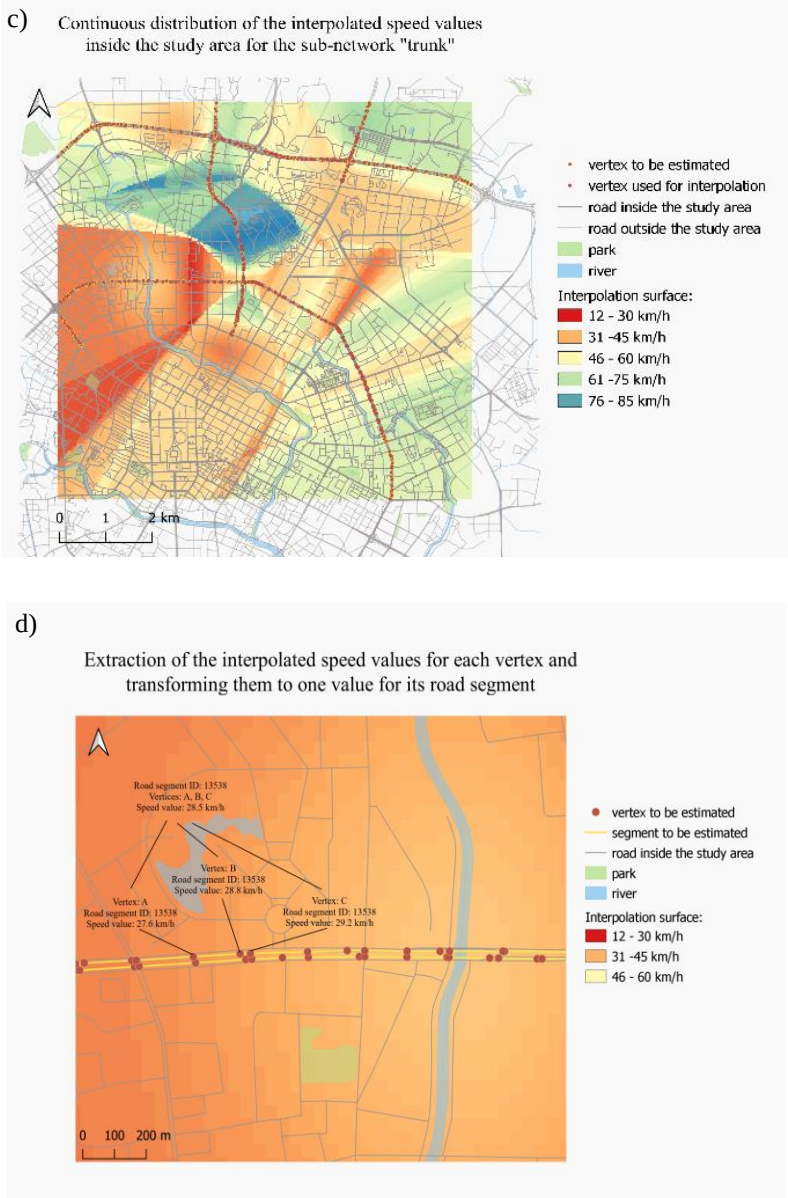


Figure 25: a) shows the sub-network and the road segments with the missing values. In b) the line features are split into its vertices. The resulting interpolated speed values are illustrated in c). As shown in d), the average of the speed values of the three vertices, that are extracted from the Kriging surface, represents the final interpolated speed value for the specific road segment. A bigger size of the figure is given in the appendix of this work.

A special case of the explained procedure is when for a given time window no road segment of the whole sub-network is visited by a taxi. This means, that not even one speed value is available and could be used for the interpolation. To deal with such situations, for each road segment in the sub-network, the average of the estimated speed values of one time window before and one after the missing time window is calculated and assigned to the analysed road segment. If only one value is given, either before or after the analysed time window, then this value represents the speed for the specific road segment. Through this, an average value of the time windows before and after the analysed one is calculated, and this is likely to be accurate because the speed value usually does not change significantly in 15 minutes at the same road segment.

Additionally, a post-processing method is applied, as there is no upper boundary of the interpolated speed values, meaning that if a road segment is located far away from the other segments of its type, the assigned speed value might be unreasonably high (as it occurred before the interpolation). To cope with this problem, the same correction as before the interpolation step is done again. Thus, the speed values more than 10 km/h higher as the maximum allowed speed value of this road type are readjusted and set to the maximum allowed speed.

5.4.2 Travel time

Besides vehicle speed values for each road segment depending on the time, the travel time of a road segment can represent the traffic state as well. Moreover, it is a very useful way of informing on the traffic state, as the travel time can be included in the trip duration of e.g. a taxi ride. Through the presented interpolation and re-correcting method, each road segment in the study area has 96 speed values assigned. As mentioned in the data section, for each road segment the length in meters is given as well. By transforming this length to kilometres and using the calculated speed value, the travel time for each road segment and for all the 96 time windows is computed. The resulting time is given in hours. As most of the road segments' travel times are much shorter than one hour, the travel time will be stored and later used in minutes.

5.5 Identifying potential ride-sharing paths

Without the presented steps of pre-processing, map-matching and traffic state estimation, the final identification of potential ride-sharing paths would not be possible or at least the influence of using traffic state information could not be analysed. The following methods aim to use the gained information most effectively. The general procedure to identify the potential ride-sharing paths, as presented in studies of e.g. Santi et al. (2014a), Barran et al. (2017) or Wang et al. (2018), would be to select a subset of all the requested or driven taxi trips and compute between all of them individually a shared path. These paths would then be ranked by some characteristics and either based on a local or a global optimum, the ride-sharing paths that maximize the objective of the method are chosen.

5.5.1 Time window size

The mentioned subset is based on a time window where the selected taxi trips act as candidate trips for ride-sharing. This is necessary as it does not make sense to share a ride with a user that requests a trip several hours later than the first user. Therefore, only trips that are requested at a similar time are useful to potentially be shared. How big this time window is, depends on how long a user is willing to wait until a ride-sharing partner is found. In general, the bigger the time window is, the more candidate trips are available and, therefore, the percentage of matched trips, or simply the "matching rate" increases. The implemented time windows in the literature differ from one minute up to ten minutes. Santi et al. (2014a) analyse in their work how significant the impact of enlarging the time window is on the matching rate and based on this select the most effective time window. They show that using a time window of one minute is the most effective choice for their data as enlarging it would lead to a relatively bigger increase in the computation time than in the matching rate. As this result might be different depending on the given data, a similar evaluation is done for the dataset used in this study.

Figure 26 shows the relation between a bigger time window and a bigger candidate trip repertory. Enlarging the time window by one minute, from one minute to two minutes, doubles the number of available candidate trips. Considering another minute more, the increase in candidate trips is slightly reduced to 150%. The curve visualises the decline in the increase of the candidate trips. After four minutes there is a too small increase considering the rising computation time and the decrease in the user-friendliness of the system, as a user must potentially wait longer to be matched to a ride-sharing partner with a bigger time window.

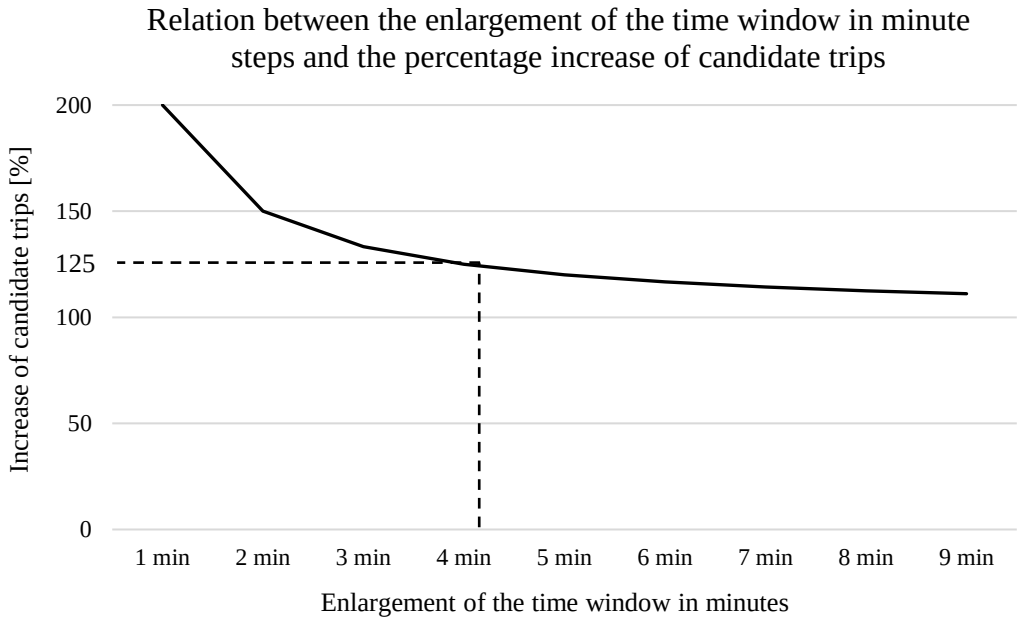


Figure 26: The decline in the percentage increase of candidate trips by an enlargement of the time window of one minute. Starting with 46 candidate trips, by enlarging the time window one minute the number of candidate trips gets doubled. Adding another minute enlarges the 92 candidate trips by again 46 candidates, what equals an increase of 150%. After the enlargement of four minutes, the percentage increase is too small to justify the arising increase in the computation time and the decrease in the user-friendliness of the system.

Considering this short analysis, a time window of five minutes (enlargement of the original time window of one minute by four minutes) is implemented in the ride-sharing approach presented in this study. This means for an analysed taxi trip T_t all the trips started five minutes before and five minutes after this trip are considered as candidate trips for ride-sharing. Depending on the number of taxi trips in the dataset and their temporal distribution during the day, there can be dozens, hundreds or thousands of candidate trips selected. The system of Santi et al. (2014a) would then compute a fastest shared path between the analysed trip and each of the candidate trips individually. Filtering and ranking them based on specified constraints and characteristics lets the system identify the ride-sharing paths that globally optimise their objective of minimizing the total travel time. Such an approach is very time intensive and a lot of fastest path computations are conducted between sets of two paths that are not useful to be shared at all. To reduce the time consumption and prevent unnecessary computations, in this work a new simple yet reliable similarity measurement is presented and implemented to filter out unsuitable candidate trips before computing the optimal ride-sharing paths that locally optimise the objective of minimizing the waiting time for the second passenger to be picked up. Figure 27 provides an overview of the so far and in the following explained methods.

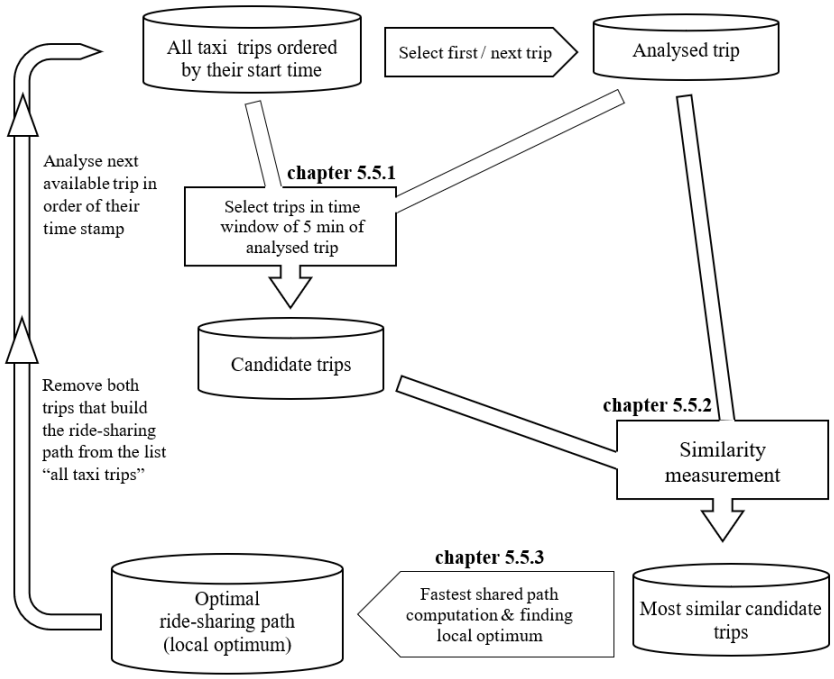


Figure 27: Illustration of the steps implemented in the identification process of potential ride-sharing paths in this work. Different from previous studies, a similarity measurement is developed to reduce the time consumption and prevent unnecessary fastest shared path computations.

5.5.2 Similarity measurement

There is a higher probability that the optimal ride-sharing path for a trip is combined with a very similar candidate trip than with a less similar one and therefore two trips must be similar to a certain degree in order to count as suitable to be shared, so that minimal detour emerges to pick up the second passenger. Thus, measuring the similarity between the analysed and the candidate trip can be used to filter out unnecessary candidates and reduce the time consumption of the computation. In ride-sharing studies where the service provider is a matching agency, a common way to measure the similarity between two vehicle trajectories is to compare the two trips in their entirety through methods like Dynamic-Time-Warping (DTW) or Longest-Common-Subsequence (LCSS) (Besse et al., 2016). This means two trajectories are only similar if a significant part of the two trips are similar in route. Thus, the similarity depends a lot on the vehicles' chosen path. To show this, one can assume that two start and end points are very close in space, but the rest of the trips take completely different paths and are far away from each other. This would lead to a small similarity measure despite the fact the start and end points are close. Such route choices are realistic, as it can be that a driver needs to pass a certain place on her or his path before the destination is reached. An example could be a kindergarten, where a father needs to drop off his child. Generally, it can be defined that considering the two trips in their entirety makes sense for vehicle trajectory data of ride-sharing systems provided by matching agencies, as there can exist hidden patterns in their paths that influence the shareability of the trips. In a taxi ride-sharing system the similarity between the two paths in their entirety is not relevant because the final optimal shared path will visit both start and end points in the fastest possible way and the route of the two individual trips does not influence that. Therefore, a different similarity measurement must be applied. To the best of found knowledge, there does not exist a taxi ride-sharing study that implements a similarity measurement and thus the presented method in this section represents a novelty in the research field of ride-sharing.

The developed and implemented similarity measurement only considers the distance between each start and end point of two taxi trips and, if necessary, the closest trajectory point of the opposite trip to either a start or end point. It thus does not consider the distances between all trajectory points of both paths. This assumes that trips whose start and end points, and sometimes the closest point of the other trip, are to some extent close in space and time (already taken into account by the applied time window to select the candidate trips) are similar and, therefore, suitable to be shared. A naive approach would be to simply measure the Euclidean distance between both start and end points. This is functionally insufficient as it would not consider when a start point is located rather far away from the first passenger's start point but very close on the way of this passenger, and therefore could be picked-up while en route (at the closest point of the first trip to the secondary passengers' start point, illustrated in Figure 29). Such special cases must be included in the measurement as well. Thus, the similarity measurement assumes three different collocations of two taxi trips which cover all the possible positions two trips can have to each other. In the following, these collocations are explained based on the start points. The same situations apply as well for the end points.

1. Considering Figure 28, in this collocation the closest point of the second trajectory T_2 to the start point of the first trajectory S_{T1} equals the start point of the second trajectory S_{T2}. The same is valid for the other way around, meaning that the closest point of the first trajectory T_1 to the start point of the second trajectory S_{T2} is equal to the start point of the first trajectory S_{T1}. This represents the simplest collocation.

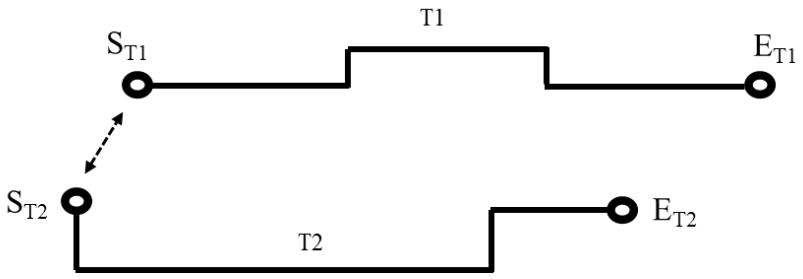


Figure 28: 1st possible collocation of two taxi trips, where the closest point of the other trajectory is for both its start point. This represents the simplest case.

2. As shown in Figure 29, in this collocation the closest point of the second trajectory T_2 to the start point of the first trajectory S_{T1} is one of its trajectory points T_{T2} but not its start point S_{T2}. On the other hand, the closest point of the first trajectory T_1 to the start point of the second trajectory S_{T2} equals the start point of the first trajectory S_{T1}. The same counts as well if the closest point of the first trajectory T_1 to the start point of the second trajectory S_{T2} is one of its trajectory points T_{T1} but not is start point S_{T1}.

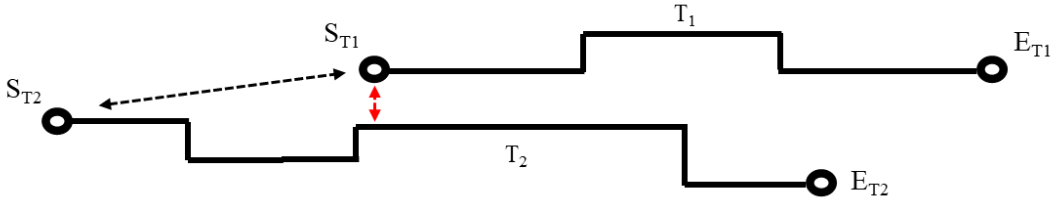


Figure 29: 2nd possible collocation of two taxi trips, where the closest point of the other trajectory is not for both its start point. This represents a more complex case, where one passenger could be picked up on the way of the other passenger without generating a big detour.

3. Figure 30 represents the collocation where the closest point of the second trajectory T_2 to the start point of the first trajectory S_{T1} is one of its trajectory points T_{T2} but not its start point S_{T2}. The closest point of the first trajectory T_1 to the start point of the second trajectory S_{T2} is as well one of its trajectory points T_{T1} but not its start point S_{T1}.

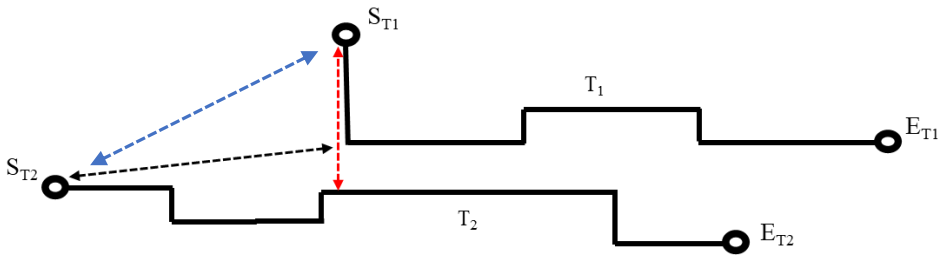


Figure 30: 3rd possible collocation of two taxi trips, where the closest point of the other trajectory for both is not its start point. This represents the most complex case, as picking up a passenger on the way would only be possible if the other passenger would be willing to walk to a meeting location.

Depending on the given collocation of the two trips, the distance between the two start and end points can be calculated slightly different. In the end one distance value, given in meters, for the two start points and one for the two end points is computed. The overall Similarity Measurement Index (SMI), that shows how similar two trips are, is the average of the distance between the two start and the two end points. A small SMI signifies a short average distance between the start and end points of both trips and represents, therefore, high similarity. How the distance is calculated for the three collocations is explained in the following:

1. The value that represents the distance between the two start points (the same is valid for end points) in the 1st collocation, visualised in Figure 28, is retrieved by calculating the Euclidean distance between both start points. As for each start point the closest point on the other trajectory is its start point, no other distance than the one between these two points is possible.

2. Given the situation in Figure 29, only for one start point (respectively end point) the closest point on the other trajectory is its start point. For the second there is a closer point on the first passenger's trajectory. This can be used to pick up this passenger in a more efficient way as driving from S_{T1} to S_{T2}. The closest point X on T_2, notated as X_{T2}, represents the location from where the detour to S_{T1} begins and therefore, the value showing the distance of the start points between these two trips is calculated by measuring the Euclidean distance between X_{T2} and S_{T1} (red arrow in Figure 29). As this distance is always shorter than the distance between S_{T1} and S_{T2}, the latter is no option for this collocation.

3. In the 3rd collocation, for each start point the closest point on the other trajectory is not its start point but a trajectory point X_{T2}, respectively X_{T1}. Driving from S_{T2} to X_{T1} or from S_{T1} to X_{T2} to pick up the other passenger would assume that the users are willing to walk to a meeting point, because driving e.g. from S_{T2} to X_{T1} to S_{T1} is illegitimate and therefore, no better option. As forcing passengers to walk is not user-friendly and thus, excluded in this system, the best option is to directly drive from S_{T2} to S_{T1}. Consequently, the value representing the distance between the two start points in this collocation is again calculated by measuring the Euclidean distance between them (blue arrow in Figure 30).

The distance used to calculate the SMI in all the three explained cases is the Euclidean distance and not the network distance. This is chosen as implementing the network distance would mean that the shortest path must be calculated for each distance (either between X_{T2} and S_{T1}, respectively X_{T1} and S_{T2} or between S_{T1} and S_{T2}). This increases the time consumption of the method, what would be off-target as with this similarity measurement the computation time should be minimized. Thus, using the Euclidean distance assures the sought time reduction. The technical details of the developed similarity measurement are notated in the algorithm illustrated in Figure 31.

Algorithm Similarity measurement

Input: (1) An analysed taxi trip $T_1 = \{T_{0T1}, T_{1T1}, \ldots, T_{nT1}\}$ with $T_{0T1} = S_{T1}$ and $T_{nT1} = E_{T1}$
(2) A candidate taxi trip $T_2 = \{T_{0T2}, T_{1T2}, \ldots, T_{nT2}\}$ with $T_{0T2} = S_{T2}$ and $T_{nT2} = E_{T2}$

Output: Similarity Measurement Index SMI in meters

for S_{T1} **do**
  find the closest trajectory point X_{T2} of T_2;
  if $X_{T2} = S_{T2}$ **then**
    a = 1 and d_a = *EuclideanDistance* (S_{T1}, S_{T2});
  else if $X_{T2} = T_{iT2}$ *or* $X_{T2} = E_{T2}$ *and* $X_{T2} \neq S_{T2}$ **then**
    a = 2 and d_a = *EuclideanDistance* (S_{T1}, X_{T2});
  end if
end for
for S_{T2} **do**
  find the closest trajectory point X_{T1} on T_1;
  if $X_{T1} = S_{T1}$ **then**
    b =1 and d_b = *EuclideanDistance* (S_{T2}, S_{T1});
  else if $X_{T1} = T_{iT1}$ *or* $X_{T1} = E_{T1}$ *and* $X_{T1} \neq S_{T1}$ **then**
    b = 2 and d_b = *EuclideanDistance* (S_{T2}, X_{T1});
  end if
end for
if *a = 1 and b = 1* **then**
  distance between start points $\Delta S = d_a = d_b$;
else if *a = 1 and b = 2* **then**
  $\Delta S = d_b$;
else if *a = 2 and b = 1* **then**
  $\Delta S = d_a$;
else if *a = 2 and b = 2* **then**
  ΔS = *EuclideanDistance* (S_{T1}, S_{T2});
end if
for E_{T1} **do**
  find the closest trajectory point X_{T2} of T_2;
  if $X_{T2} = E_{T2}$ **then**
    a = 1 and d_a = *EuclideanDistance* (E_{T1}, E_{T2});
  else if $X_{T2} = T_{iT2}$ *or* $X_{T2} = S_{T2}$ *and* $X_{T2} \neq E_{T2}$ **then**
    a = 2 and d_a = *EuclideanDistance* (E_{T1}, X_{T2});
  end if
end for
for E_{T2} **do**
  find the closest trajectory point X_{T1} on T_1;
  if $X_{T1} = E_{T1}$ **then**
    b =1 and d_b = *EuclideanDistance* (E_{T2}, E_{T1}) ;
  else if $X_{T1} = T_{iT1}$ *or* $X_{T1} = S_{T1}$ *and* $X_{T1} \neq E_{T1}$ **then**
    b = 2 and d_b = *EuclideanDistance* (E_{T2}, X_{T1});
  end if
end for
if *a = 1 and b = 1* **then**
  distance between end points $\Delta E = d_a = d_b$;
else if *a = 1 and b = 2* **then**
  $\Delta E = d_b$;
else if *a = 2 and b = 1* **then**
  $\Delta E = d_a$;
else if *a = 2 and b = 2* **then**
  ΔE = *EuclideanDistance* (E_{T1}, E_{T2});
end if
$SMI_{T1, T2} = (\Delta S + \Delta E) / 2$;

Figure 31: Algorithm of the new developed and implemented similarity measurement.

Besides the presented three possible collocations, there exist two special cases; for one start point the closest point on the other trajectory is its end point. This represents a situation where the first trip has already finished before the second trip is started and is therefore not considered as a ride-sharing situation. Nevertheless, this end point is treated as a normal trajectory point. Such a situation is illustrated in Figure 32. The distance gets calculated like in Figure 29 and this could lead to a very small SMI and would therefore incorrectly signal a high similarity. If such a trip collocation occurs and would be selected as one of the most similar candidate trips for ride-sharing, it would get eliminated in the next step, as only ride-sharing paths that reduce the total travel time of the shared path compared to the sum of the two individual paths are considered to be suitable for ride-sharing systems. Therefore, this special case is not added as a 4th possible collocation, but rather represented by the 2nd collocation. The mentioned condition is explained in more detail in the next section. The same elimination counts for the situation illustrated in Figure 33 (same distance calculation method for the start points as in the 3rd collocation, and same method for the end points as in the 1st one), where two trips are allocated in the opposite direction as sharing these two trips would again not reduced the total travel time if the two start points must be visited first. Once again there is no 4th collocation needed.

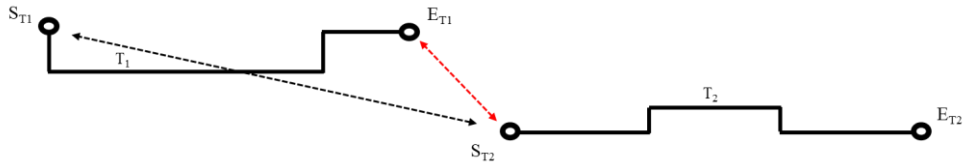


Figure 32: Collocation where the first trip has already finished before the second one even started and is therefore not considered as a ride-sharing situation. Nevertheless, E_{T1} is treated like a normal trajectory point T_{T1} and the red arrow shows the distance used for the SMI. In the next process such wrongly similar candidate trips would get eliminated.

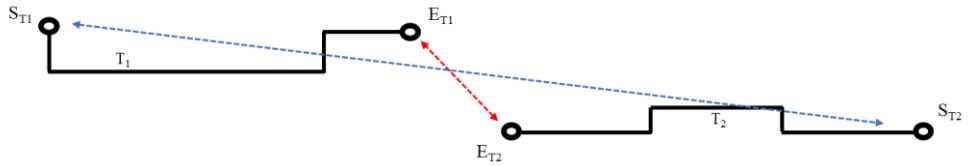


Figure 33: The situation where two taxi trips are collocated in the opposite direction. The blue arrow shows the distanced used for the similarity between the two start points (as in the 3rd collocation) and the red arrow represents the similarity between the two end points (as in the 1st collocation). If such a candidate trip is selected as one of the most similar ones it would again get eliminated in the upcoming process.

The presented similarity measurement is applied to all the candidate trips of an analysed trip and each set of two paths gets an SMI appended. By ordering them from small to big, the three smallest SMI values, if available, and in relation to this the three most similar candidate trips are selected as the remaining candidates for the identification of the optimal ride-sharing path. All the other candidate trips are considered as unsuitable for sharing their ride with the analysed trip and are discarded. This reduces the computation time and portrays the utility of the developed method. The approach is quite simple yet reliable as it considers the different possible collocations of two taxi trajectories. Using this method in real-time applications, instead of the recorded trajectory (which in this case would not be available) the shortest path connecting the known start and end point of a trip could be used.

5.5.3 Optimal path computation

Due to the similarity measurement, instead of considering all candidate trips, an optimal ride-sharing path must only be computed between each analysed trip and a maximum of three other taxi trips. This optimal ride-sharing path is identified by applying a set of fastest path algorithms and constraints to the three (or less) possible combinations of the analysed and the candidate trip. The taxi trips get analysed ordered by their start time. For the primary taxi trip of the 1st Nov. 2016 initially all the candidate trips are selected based on the time window of five minutes (in this case only trips that started up to five minutes after the first one as there are no trips available before this time stamp) and then the similarity measurement is applied. From the remaining candidate trips the optimal ride-sharing path is selected which represents the fastest path for the given combination of analysed and candidate trip, fulfils the set constraints, and maximizes the objective. The identified ride-sharing path is then stored, and the two input taxi trips are removed from the candidate list. This means that they are no longer an option for sharing a ride with a trip started later. If for some reason for an analysed trip no optimal path that fulfils the constraints is identified or no candidate trips are found and therefore the similarity measurement cannot be applied, then this path is stored to be driven individually and as well removed from the candidate list to not be considered anymore. Subsequently, the next taxi trip in order of their start time that is not removed so far from the candidate list is analysed. As for each trip the optimization problem is solved independently from the following trips and their possible optimal paths, the matching process of this system represents a local optimum case and not a global optimum as implemented in the study of Cai et al. (2019). This means that for the analysed trip at this moment the identified ride-sharing path is the optimal solution, but on a global perspective potentially its ride-sharing partner could have been matched to an even more optimal trip to share the ride. On the point of a strong increase in the computation time and the complexity of the method, it is decided to work with a local optimum as just described. Moreover, Wang et al. (2018) show, that finding a local optimum instead of a global one, can deliver very reliable results as well.

5.5.3.1 Fastest path

The optimal ride-sharing path must visit each start and end point of the two combined taxi trips in the fastest but not necessarily the shortest way. This way it is ensured that the travel time, depending on the driven speed, is considered to find the optimal path and not its distance. This travel time can have two different sources, meaning the considered speed values are derived differently. First, as the speed values the given maximum allowed speed per road segment can be used, which represents the travel time needed to pass a road segment assuming an absence of traffic congestion. Second, and this represents a new approach, the in the traffic state estimation calculated speed values, or directly the estimated travel time, can be implemented into the fastest path algorithm. By this function, the necessary time to travel on a road segment based on the raw GPS taxi trajectory points is used and thus the traffic state information is included in the identification process of potential ride-sharing paths. To find the fastest path, a weighted Dijkstra's shortest path algorithm is implemented. Given a start node, an end node and a network graph, the algorithm computes the shortest path between the two nodes on the underlying graph (Goldberg & Tarjan, 1996). A normal Dijkstra's shortest path algorithm would identify the shortest path based on the total distance, given by the length of the individual edges of the graph (road segments). By using a weighted Dijkstra's shortest path algorithm, a weight can be added that replaces the length of the edges. This weight is either the travel time based on the maximum allowed speed values or the travel time calculated in the traffic state

estimation. The sum of the travel times of each selected edge must be minimal for the fastest path. The results of this fastest path algorithm are the edges of the path and the total travel time. As a shared path must visit four nodes but the Dijkstra's shortest path algorithm only works with two nodes, more than one fastest path is computed for an optimal shared path. Given the first start point S_1, the first end point E_1, the second start point S_2 and the second end point E_2, for the complete shared path, three fastest paths must be computed. Additionally, there is more than one possible order of how the points are visited. The only rule is that both start points must be visited before serving an end point and dropping off a passenger, as the opposite would not represent a ride-sharing situation. Thus, the following four collocations are possible:

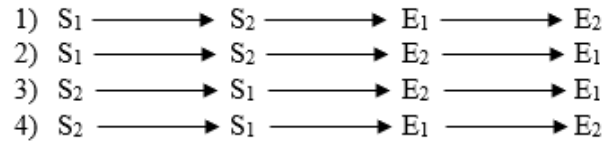


Figure 34: The four different collocations of the start and end points of the two trips to be shared. For each collocation three fastest paths must be computed and summed up to the final shared path of this collocation. The collocation that leads to the shortest total travel time represents the fastest path for the combination of the two matched trips.

For the first collocation e.g., the fastest path between S_1 and S_2, between S_2 and E_1 and between E_1 and E_2 must be computed. Considering this, for an optimal path, 12 fastest paths are computed. Summing up the total travel time and edges of the three sub-paths give the end measures for each collocation. The one leading to the shortest total travel time represents the final fastest shared path for the combination of the analysed trip with this candidate trip. This is done a maximum of three times if there are three very similar candidates left, and each trip combination is represented by one fastest shared path. Later, these paths are tested on the constraints and the one of the remaining paths that maximizes the objective is identified as the final ride-sharing path.

As aforementioned, the weighted Dijkstra's shortest path algorithm needs as the input the two nodes, the network graph and the weight of the edges. If for the weight the travel time based on the maximum allowed speed is used, the weight of the edges remains the same over the whole day. On the contrary, when using the travel time calculated in the traffic state estimation depending on the start time of the two trips that are shared a different weight is used, as the travel time is always given for a time window of 15 minutes. Before running the fastest path algorithm, the estimated travel time for the time window in which the second trip started must first be selected and added as the weight to the algorithm. The time window of the start time of the second trip is chosen as the shared path will not start earlier because the trip request of the second user is not available until this time. Of course, a shared trip can have a duration longer than 15 minutes and then the traffic state and with this the travel time might change. However, it is assumed that taxi drivers like to stick to the at the beginning computed route and thus the travel time of only one time window is considered. The network graph is represented by nodes and edges and is based on the start and end vertices of the road segments. Furthermore, the information on the one-way restrictions is included, meaning a two-way road segment is included twice in the network graph (once with inverted order of the nodes). This ensures that the optimal path only considers road segments where it is allowed to drive.

As the origin and destination of a taxi trip are only given as a trajectory point that is map-matched to the road network, meaning the ID of the map-matched road segment is stored as its attribute but the point itself is not part of the segment, these origin and destinations are not available in form of nodes. Thus, besides including the one-way restriction and selecting the right travel time for the weights of the edges, the origin and destination of each input trip must be transformed to start and end nodes of the network graph to be used in the algorithm. There exist three different approaches on how to transform the origin and destination of a trip to start and end nodes of the network graph. These approaches are visualised in Figure 35. One possible way is to select the closest start or end vertex of the map-matched road segment to the first and last trajectory point of a trip, meaning the closest node representing this segment in the network graph (Figure 35b). To make this method more accurate, as seen in Figure 35c, the road segments could be split into smaller segments so that the distance to the closest node of the map-matched road segment gets smaller. Unfortunately, this leads to a strong increase in the complexity of the network graph and therefore to a loss of performance, as there would be created a lot of new nodes. The third approach, visualised in Figure 35d, is to snap each start and end trajectory point of a trip to the map-matched road segment and divide it into two new segments. This would again lead to an increase in the complexity and the time consumption as each new road segment means two new nodes are inserted in the network graph. Due to the mentioned downsides of the other methods, the approach visualised in Figure 35b is implemented in this work.

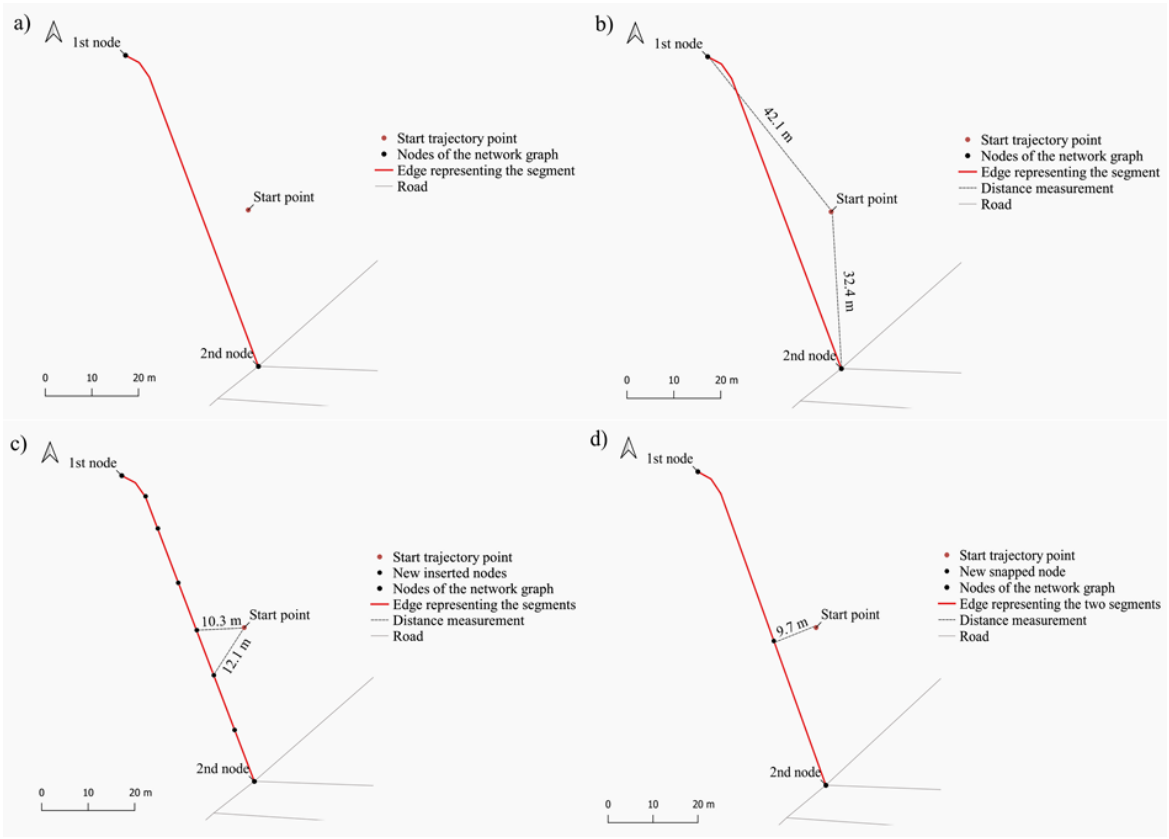


Figure 35: a) shows the problem that the start trajectory point of a trip is not part of the map-matched segment and, therefore, not stored as a node in the network graph. In b), the closer node of the map-matched segment is chosen as the start node that represents the start trajectory point. The method in c) divides the road into smaller segments so that the nearest node of it is even closer to the start trajectory point. In d), the start trajectory point gets snapped to the segment and divides it at this location so that the start trajectory point is directly represented in the graph.

The selected approach measures the Euclidean distance from the start trajectory point to both start and end vertex of the map-matched road segment. The node that represents the vertex with the closer distance to the start trajectory point is chosen as the start node of this taxi trip. In the case of figure 35b, the 2nd node is chosen as the start node of that trip. The same is done for the end trajectory points and the vertices of its map-matched road segment. With this method, each start and end point of a taxi trip is represented by a start and end node in the network graph and can be used as an input for the weighted Dijkstra's shortest path algorithm. As seen in Figure 35b, with this approach the location of the start point gets moved slightly (in this case 32.4 m). The average length of a road segment is 331.1 m and thus the average maximum shift is around 165 m. This is not a significant shift and both the start and end point get shifted randomly, meaning the trip could be enlarged or shortened with the same likelihood. With this considered as well as the utility of the method, the choice of the approach is regarded as legitimate.

5.5.3.2 Objective and constraints

From the computed fastest paths, the one that fulfils all the constraints and maximizes the objective of the ride-sharing system is identified as the optimal ride-sharing path for the given combination of the analysed and the candidate trip. Different from the presented studies in Chapter 2, the objective of the system of this work is to minimize the emerging waiting time for the second user to be picked up. Another possible objective would have been to minimize the total travel time and with this the total driven kilometres and therefore the total CO_2-emissions. Following this objective makes sense as in a global perspective the aim of a ride-sharing system is to some extent to minimize the CO_2-emissions, but the more important aspect will be the user-friendliness of the system which is highly related to the waiting time. A ride-sharing system only works if the users are willing to join, and this depends, in some part, on additional costs like the mentioned waiting time. Therefore, in a holistic view, a ride-sharing system aiming to minimize the CO_2-emissions might not save more CO_2-emissions than a system that focuses on the user-friendliness, as fewer users will join the system and thus less shared rides are identified.

Before the fastest path that minimizes the waiting time can be identified, the remaining paths must be tested on three constraints. Only the fastest shared paths that lead to a total travel time smaller than the sum of the travel times of the two individual trips are suitable for ride-sharing. This means that only ride-sharing paths that save time remain. The others are removed from further analysis as they will not save CO_2-emissions and are not user-friendly. The second set constraint is that the shared path must also save distance compared to the situation without any ride-sharing involved. In other words, the total driven kilometres must be less for the shared path than the sum of the driven kilometres of the two individual trips. Here again CO_2-emissions and user-friendliness are the reason to remove the unsuitable paths. The last constraint is related to the objective of the system. By setting a maximum acceptable waiting time, it is prevented that a user would have to wait for too long until the taxi arrives and decides to not use the ride-sharing system. In the study of Cao et al. (2015) they apply a maximum acceptable waiting time of 15 minutes. As Rayle et al. (2014) analyse in their survey, 90% of the users of taxi-like transportation systems as e.g. Uber or Lyft say that they wait on average less than ten minutes for the requested vehicle to arrive. Only a few users specified a waiting time bigger than ten minutes and barely any users mentioned waiting times over 20 minutes. Therefore, the applied maximum waiting time of 15 minutes by Cao et al. (2015) is regarded as reasonable and will be used in this study as well. So, if the computed fastest shared path leads to a waiting time

bigger than this threshold, it is not considered as suitable anymore and will be removed. After applying the fastest path algorithm, all three or less computed shared paths are checked on the mentioned constraints. Only shared trips that fulfil all the set constraints are considered to be suitable for ride-sharing. As soon as one constraint is not met, the candidate trip is removed from the list. The remaining shared paths are ordered by their waiting time and the one that minimizes this waiting time is identified as the optimal shared path for the analysed trip. As already mentioned, this shared path with its indicators is stored and both the analysed and the candidate trip are removed from the list for further combinations of ride-sharing paths. If no shared trip fulfils all constraints, the analysed trip is considered as unsuitable for ride-sharing and will be served individually. This information is stored as well, and the trip is removed from further computations. Subsequently, the next taxi trip in order of its start time is analysed. The total distance of the shared and the two individual trips is derived by summing up the lengths of each road segment that is part of the path. The waiting time equals the time needed to get from the first start point to the second one. This sub-path is represented by the computed fastest path from S_1 to S_2 or vice versa. The resulting travel time of this fastest sub-path stands, therefore, for the emerging waiting time of the final shared trip.

5.6 Experimental design

To analyse the influence of considering traffic state information in ride-sharing systems, the described optimal path identification process is applied twice. First, the travel time calculated in the traffic state estimation is used as the weight of the edges in the Dijkstra's algorithm. Second, the travel time based on the maximum allowed speed value of each road type is taken as the weight. This leads to potentially different results in measures like the average waiting time, the saved total travel time, or the reduction in the size of the needed taxi fleet. Analysing these differences is used to answer and comment on some of the presented research questions and hypotheses of Chapter 3. Furthermore, the computation is done first without considering the constraint on the distance reduction and then with including this addition. This means, once only the travel and waiting time constraints are applied and therefore the identified shared path must not necessarily reduce the total travel distance. This is used to show the influence such a constraint can have on the overall results of a ride-sharing system and its importance regarding the impact of ride-sharing on the natural environment. So, the whole process is run four times:

1) Using traffic state information but not considering the constraint on saving distance
2) Using traffic state information and considering the constraint on saving distance
3) Not including information on traffic and not considering the distance constraint
4) Not including information on traffic but considering the distance constraint

Figure 36: The four different variations of implementing the identification process of the optimal ride-sharing path.

All results of these four variations are presented and compared in the next chapter. Based on them, in Chapter 7, the influence of using traffic state information is discussed and the results are put into perspective to the research questions and hypotheses of Chapter 3. Figure 37 shows the algorithm that delivers again a detailed view of the described process of identifying the optimal ride-sharing path and is connected to the developed similarity measurement. Presented is the variation where the traffic state information is included, and the distance constraint considered. The algorithms for the other three variations are very similar.

Algorithm Identification of optimal ride-sharing paths

Input: (1) All map-matched taxi trips $T_n = \{T_{0Tn}, T_{1Tn}, \ldots, T_{nTn}\}$ with $T_{0Tn} = S_{Tn}$ and $T_{nTn} = E_{Tn}$
(2) The road network R including the one-way restrictions and the max allowed speeds
(3) Traffic state information t_{Rntn} given for each road segment at each time interval

Output: Optimal ride-sharing paths with its indicators and individually driven trips

```
create graph G based on R
for each trip Tn do
    calculate travel time time_Tn
    calculate distance distance_Tn
    select node Gn where minDist (Gn, STn) as start node
    select node Gm where minDist (Gm, ETn) as end node
    select trips started +/- 5 min of start time of STn as candidate trips CTn
    if CTn is not empty then
        for each candidate trip CnTn in CTn do
            calculate SMI = Similarity measurement (Tn, CnTn)
        end for
        select the CnTn with the 3 (or less) smallest SMI values as CnTn'
        for each CnTn' do
            select node Gi where minDist (Gi, SCnTn') as start node
            select node Gj where minDist (Gj, SCnTn') as end node
            set weight of edge we = tRntn where maxTime {STn, SCnTn'} in time interval tn
            create weighted graph weighted_G = weightedGraph (G, we)
            for each collocation [A-B-C-D] = {(S1-S2-E1-E2),(S1-S2-E2-E1),(S2-S1-E1-E2),(S2-S1-E2-E1)}
            do
                travel_time_1, path_1 = weighted_Dijkstra's_shortest_path (weighted_G, A, B)
                travel_time_2, path_2 = weighted_Dijkstra's_shortest_path (weighted_G, B, C)
                travel_time_3, path_3 = weighted_Dijkstra's_shortest_path (weighted_G, C, D)
            end for
            total_travel_time = travel_time_1 + travel_time_2 + travel_time_3
            shared_path = path_1 + path_2 + path_3
            fastest_shared_path sharedCnTn' = shared_path where min (total_travel_time[A,B,C,D])
            time_sharedCnTn' = total_travel_time
            waiting_timeCnTn' = travel_time_1
            distance_sharedCnTn' = sum of lengths of Rn in sharedCnTn'
            timeCnTn' = travel time of CnTn'
            distanceCnTn' = sum of lengths of Rn in CnTn'
        end for
        for each sharedCnTn' do
            if time_shared_CnTn' > (time_CnTn' + time_Tn)
            or
            if distance_shared_CnTn' > (distance_CnTn' + distance_Tn)
            or
            if waiting_time_CnTn' > 15 min then
                drop sharedCnTn'
            end if
        end for
        if remaining trips C_nTn'' > 0 then
            optimal shared path sharedTn,CnTn'' = sharedCnTn'' where min (waiting_timeCnTn'')
            save sharedTn,CnTn'', time_sharedTn,CnTn'', waiting_timeTn,CnTn'', distance_sharedTn,CnTn''
            remove Tn and CnTn'' from the candidate list of the up-following trips
        else
            save: Tn is unsuitable for ride-sharing and served individually
            remove Tn from the candidate list of the up-following trips
        end if
    else
        save: Tn is unsuitable for ride-sharing and served individually
        remove Tn from the candidate list of the up-following trips
    end if
end for
```

Figure 37: Algorithm of the identification process of the optimal ride-sharing paths.

6. Results

This chapter presents the results of applying the developed framework and its methods to the real-world GPS taxi trajectory and the OSM road network data. As each sub-process produces its results, this chapter is divided into these respective processes. First, the results of the applied map-matching method are presented. Based on example taxi trips, it is shown how the resulting map-matched trips look like and statistics about the successfully map-matched trajectories are provided. In Chapter 6.2, the estimated traffic state information is described. Results about the vehicle speed calculation, the interpolation of these values, and the estimated travel times are delivered. By providing examples of traffic maps, the final estimated traffic state is illustrated. Before presenting the main results of the optimal path identification, the product of the implemented similarity measurement is shown. Subsequently, examples of the identified ride-sharing paths are visualised, and different measures provided. These ride-sharing results are described for each of the explained variations in Figure 36 of Chapter 5.6. The presented results are then discussed in Chapter 7 and related to the research questions and hypotheses of this work.

6.1 Map-matching

By applying the map-matching algorithm, for each trajectory point of a taxi trip the ID of the map-matched road segment results. Furthermore, a distance value measured from the start of the trip is given for each point. By connecting all the map-matched road segment IDs, the path where the taxi was most likely to be driving is reconstructed. Table 11 shows the resulting map-matched road segment IDs and calculated distance values for an example taxi trip. This taxi trajectory consists of 320 GPS records and has a length of approximately 5.2 km.

Order ID of taxi trip: 5d546bc7354521f7f004bc13f0c9b84b

Trajectory point	Map-matched road segment ID	Calculated distance from start
1	8936	0 m
2	8936	7.756 m
3	8936	20.727 m
4	8938	31.931 m
5	8938	57.496 m
6	8938	89.711 m
7	8938	124.148 m
8	8938	147.476 m
.	.	.
.	.	.
.	.	.
311	13018	5044.883 m
312	13018	5044.883 m
313	13018	5048.236 m
314	13032	5062.754 m
315	13044	5070.023 m
316	13044	5078.969 m
317	13078	5135.283 m
318	13078	5165.206 m
319	13110	5190.905 m
320	13504	5202.889 m

Table 11: Example of the map-matching algorithm results in written form. Shown are the map-matched road segment IDs and the calculated distance from the start for a subset of the taxi trip.

As illustrated in the previous table, a road segment can have more than one trajectory point matched to. This is presented in more detail in Table 12. The 320 trajectory points are map-matched to 42 different road segments. The number of matched points to a road segment differs between 1 and 44. This difference is due to the varying length of the segments, the different vehicle speed at each road section and the unequal distribution of the trajectory points. Road segment 13018 for example has 44 trajectory points matched to and a length of 182.9 m. Road segment 8942, on the other hand, has only one trajectory point matched to and a length of just 16.9 m. In Figure 39 a), the distance between the trajectory points is not always the same and this influences the number of matched points to each road segment as well. Either a higher vehicle speed or a decrease in the density of the recorded trajectory points are the reason for this occurrence.

Order ID of taxi trip: 5d546bc7354521f7f004bc13f0c9b84b

Road segment ID	Number of matched points	Road segment ID	Number of matched points
8936	3	10910	4
8938	6	10981	1
8939	41	11165	11
8942	1	11299	19
8944	4	11691	5
8948	6	11692	12
9082	9	11721	1
9171	1	12032	6
9173	2	12033	17
9176	5	12363	4
9177	25	12364	4
9217	11	12654	6
9374	3	13017	4
9555	6	13018	44
9580	9	13032	1
9798	21	13044	2
10045	5	13047	1
10227	3	13055	1
10455	2	13078	2
10528	2	13110	1
10591	8	13504	1

Table 12: Distribution of the matched trajectory points to the 42 different road segments of the example taxi trip. 320 GPS signals are map-matched in total. Depending on the length of the segment, the speed of the vehicle at each road, and the density of the trajectory points, between one and 44 points are matched to each road segment.

In Figure 38, the described results are visualised on the road network. The red points in a) represent the 320 GPS signals of the example taxi trajectory. The trip starts south-west and ends north-east. The resulting map-matched taxi path is displayed in b). This path is derived by connecting each of the mentioned road segments of Table 12. The start and end of the trip are now represented by a start and an end road segment, not by a point anymore. While this slightly enlarges the path, it is not problematical as the trajectory points are used for both the traffic state estimation and the similarity measurement, instead of the road segments. Only the optimal path identification algorithm gets affected by this as it works with the closest node of each start and end segment to each start and end trajectory point. As explained in the previous chapter, however, this change of the original path is miniscule, thus regarded as legitimate.

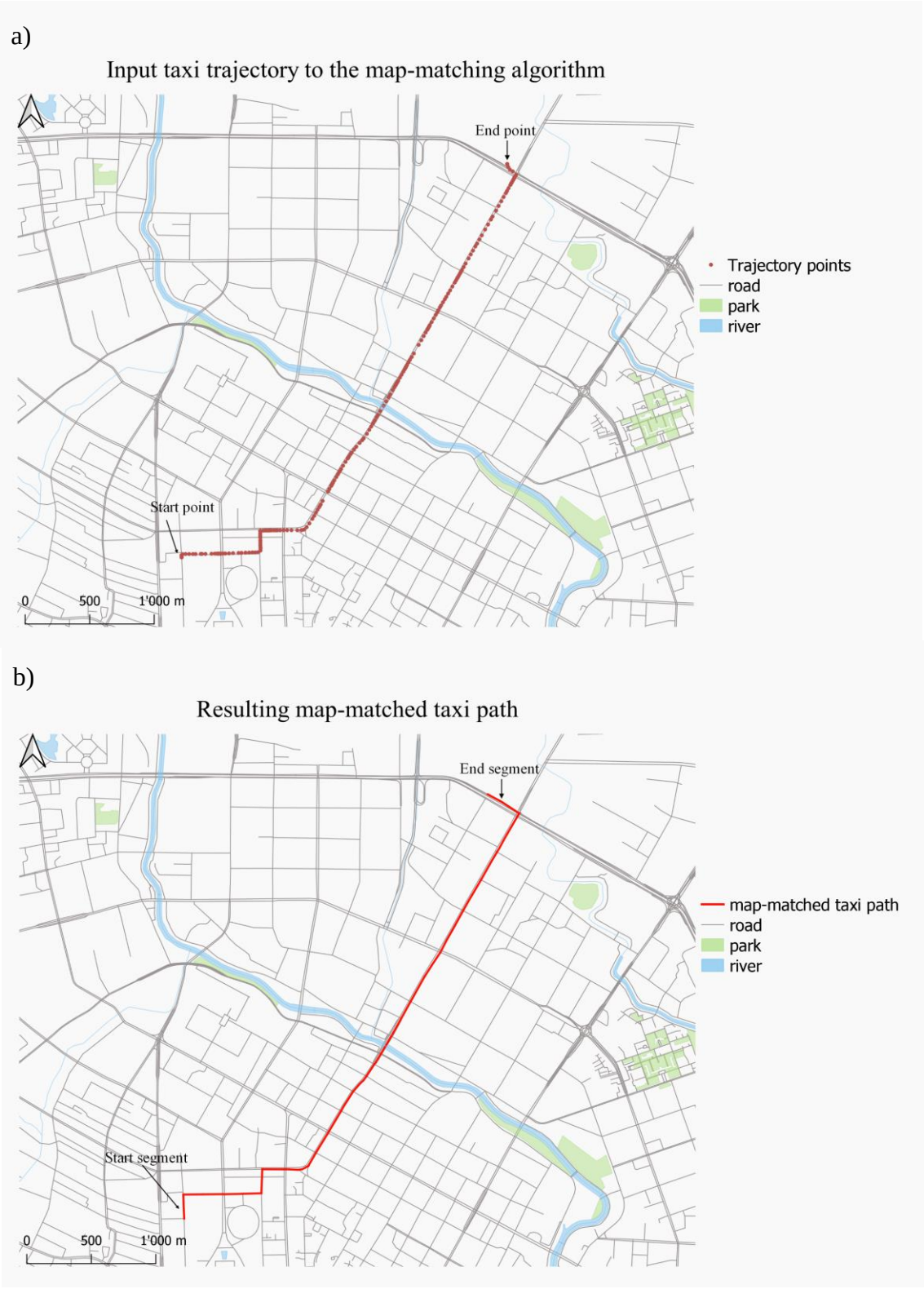


Figure 38: Visualisation of the map-matching result of an example taxi trip. In a) the 320 GPS signals, that are used as the input for the algorithm, are located on the road network of Chengdu. The resulting map-matched taxi path is displayed in b). The start segment of the trip is located south-west and the end segment north-east. The trip consists in total of 42 road segments and is approx. 5.2 km long.

A proof that the implemented map-matching algorithm works as explained in Chapter 5 is given in Figure 39. Displayed is a section of the discussed example taxi path, where the results appear correctly only due to the selected map-matching method. Considering the trajectory points in the centre of the visualisation in a), the main part of them tend to be located on the left road segment instead of the map-matched segments on the right side. But if the whole section is inspected, it becomes clear that the trajectory points must be matched to the road segments on the right, as both in the beginning as well as in the end of the section, the trajectory points are located exactly on this road. Thus, the right path is identified by the map-matching algorithm despite appearing to be different if analysing each point separately.

The situation in b) explains this in more detail. Focusing on point 215, two candidate road segments are given. In this case, the road segment 11692 is obviously the correct choice for point 215 as it is located closer to it and, as shown in a), all the previous points are matched to the same segment. Point 220 is again correctly map-matched even though road segment 12040 would have been closer. This is again because the forthcoming points are matched to the same segment. The choice for point 217 is more complex. This point could potentially be matched to road segment 11710, 11712, or 11692. Considering only the distance to each candidate segment, road segment 11712 would be chosen instead of segment 11692 as it is the furthest candidate segment. Nevertheless, the algorithm correctly chooses this segment, and therefore makes it clear that by considering the previous and forthcoming trajectory points, the taxi trips can successfully be map-matched.

This example taxi trip shows how exact the map-matching results can be. Unfortunately, no ground truth data is available to make a conclusion on the concrete accuracy of the adopted HMM map-matching method. The only option to roughly control the quality of the results is by visual analysis of random samples; this means to randomly select order IDs from the database and comparing the map-matched paths with the related trajectory points, just like it was done for the example trip. This control has shown that most of the trajectory points were map-matched correctly, hence the overall accuracy is considered to be good.

Nevertheless, as already mentioned in Chapter 5.3, different errors can occur during the map-matching process. If the distance between two consecutive trajectory points is too big, or perhaps an outlier GPS signal is recorded, the algorithm cannot continue computing the needed probabilities and fails. Therefore, the input parameters of the algorithm must be chosen wisely to minimize such errors. Furthermore, errors influencing the traffic state estimation can appear while calculating the network distance of each trajectory point. This is either due to erroneous data or problems in the algorithm, but as already mentioned in Chapter 5.4, these errors are later corrected by the applied post-processing approach. Additionally, the problem of incorrect map-matched trajectory points must be considered as well as this is always possible in such methods. Otherwise, the accuracy of these approaches would be 100%. The incorrect map-matched points, however, do not always have to lead to a completely incorrect reconstructed taxi path as only some of its trajectory points are erroneous, thus only a few road segments are wrongly added to the reconstructed path. An example of such incorrect map-matched trajectory points is given in Figure 40. The path correctly starts at road segment 8531 and would then go through segments 8462 and 8449 and follow the rest of the correct matched route. However, segments 8463 and 8460 are wrongly identified as part of the matched path. The corresponding trajectory points, therefore, are map-matched incorrectly.

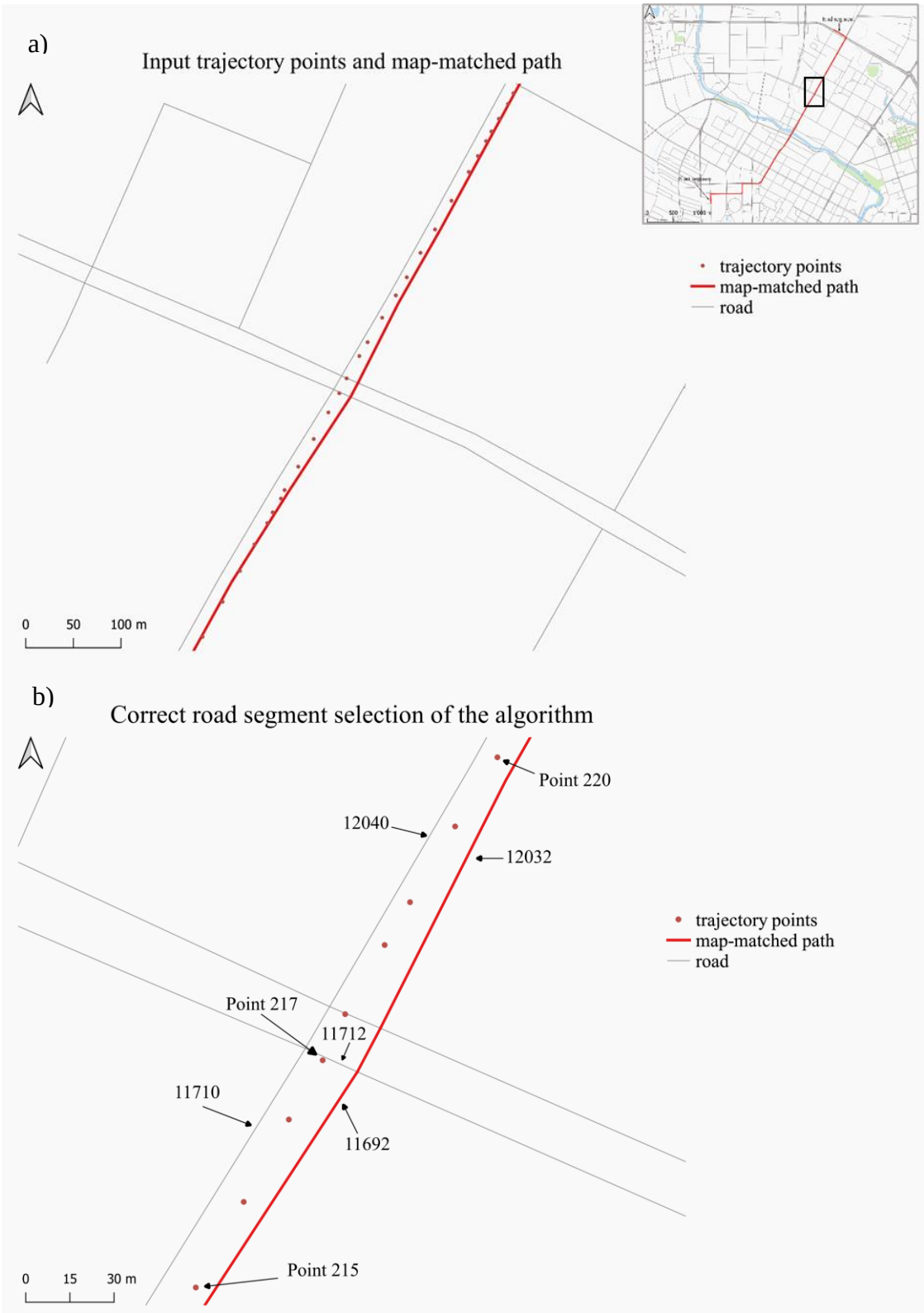


Figure 39: Visual proof that the implemented map-matching method can work as explained in Chapter 5. By analysing each trajectory point separately, the situation in a) would lead to incorrectly matched road segments as the points tend to be closer to the segments on the left side than to the ones one the right side. This is explained in more detail in b). Point 217 could potentially be matched to three road segments and selecting the closest one would be incorrect. As the algorithm considers the previous and forthcoming points of each GPS signal, the correct map-matched taxi path is identified.

As the information about the map-matched road segments is only used for the traffic state estimation, more specifically to calculate the traffic speed and to define the start and end nodes of the fastest path, the consequences of these incorrectly selected segments are not too serious. For the given example, only wrong speed values would be used to calculate the average speed of the road segments 8463 and 8460. As the start segment is matched correctly regardless, the right start node of the fastest path is going to be selected.

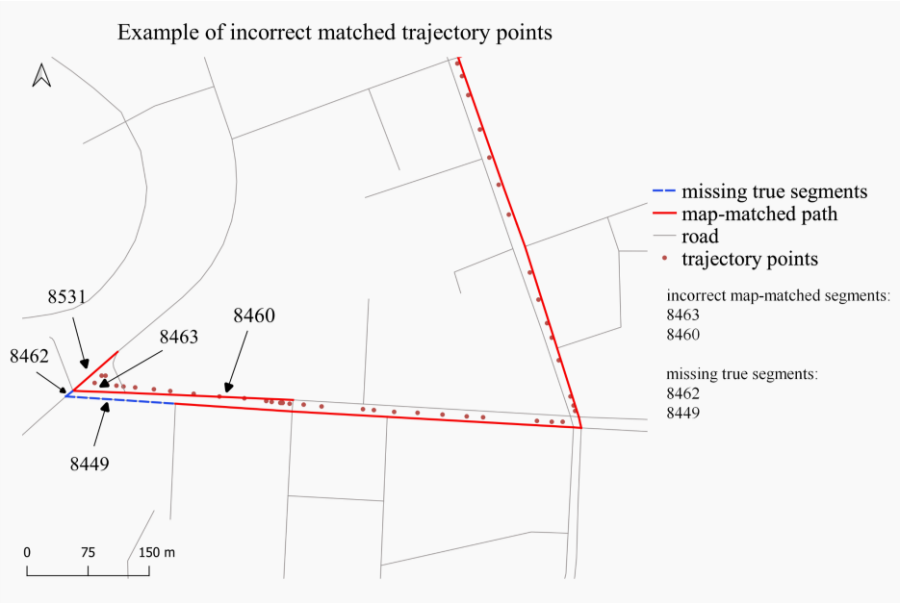


Figure 40: Example of an incorrect map-matched taxi path. The road segments 8463 and 8460 are wrongly identified as part of the path. The true path starts at segment 8531 and goes through segments 8462 and 8449 and follows the rest of the route. The two blue dashed segments are therefore missing.

By far the biggest source of errors for the map-matching method is related to the Python module of ArcGIS named arcpy. For unknown reasons, arcpy produces errors if the map-matching algorithm is running for an extended period of time. The result is that after a specific time all the trips that are being processed throw an error. As arcpy plays an important part in the applied map-matching algorithm, these errors cannot be prevented. Thus, this limitation on the successfully map-matched taxi trips is acknowledged for the remaining part of the work.

In detail, this means that only 15'347 trips are successfully map-matched from the original 41'828 taxi trips. This equals to 3'181'904 instead of 9'053'673 trajectory points. Overall, 36.7% of the available taxi trips and 35% of the available trajectory points are map-matched successfully. Hence, 65% of the data is lost while map-matching the GPS signals. This seems to be grave in the first instance, but fortunately the successfully map-matched trips are equally distributed over the whole day and the total number of available trips is still big enough to apply the remaining part of the framework and follow the research objective; Figure 41a shows the distribution of the remaining taxi trips over the 1st Nov. 2016; in Figure 41b, these numbers are visualised in proportion to the total number of successfully map-matched taxi trips, which in this case is 15’347. Additionally, this curve is compared to the proportional distribution before the map-matching process, where the total number of trips is 41’828. This illustrates that only the total amount of available trips is reduced while at the same time the distribution remains the same. Therefore, the magnitude of the occurring errors is kept within a reasonable limit.

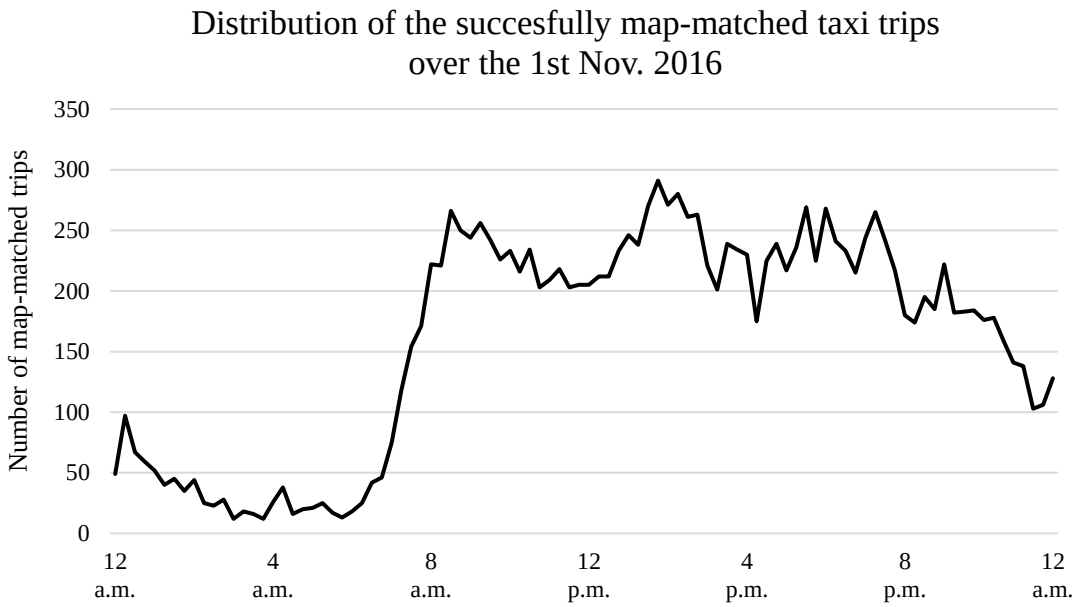


Figure 41a: Visualisation of the distribution of the total number of successfully map-matched taxi trips over the 1st Nov. 2016. Due to the errors in the map-matching process, the total number of available taxi trips is strongly reduced.

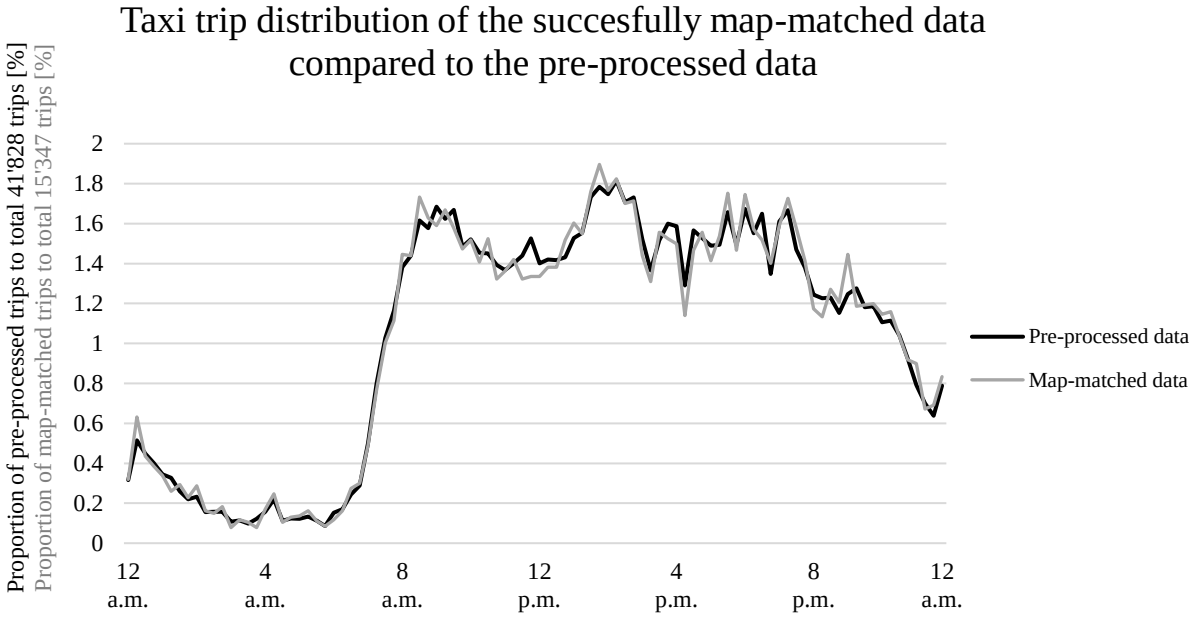


Figure 41b: Visualisation of the comparison between the proportional distribution of successfully map-matched taxi trips, given a total of 15’347 trips, and the proportional distribution of available trips after the pre-processing step, given a total of 41’828 trips. The sum of the proportions over the time equals to 100%. Even if the total number of available taxi trips strongly decreases, the distribution over the whole day of the 1st Nov. 2016 remains almost the same. This shows that the magnitude of the occurring errors is kept within a limit.

6.2 Traffic state estimation

By using the information on the differences in the location and the time stamp of the 3'181'904 map-matched trajectory points, the traffic state of the road network in the city centre of Chengdu is estimated. As explained in the methodology, the vehicle speed is calculated in the first step. Table 13 shows the calculated speed values for a subset of the trajectory points of an example taxi trip. This trajectory contains 55 points and is 1.2 km long. The trajectory points in the table are ordered by their time stamp and numbered on the left side (the subset contains points 34 to 53). In the second column, the network distance of each point to the start of the path is given. Additionally, the exact time stamp in seconds is provided. Calculating the ratio of the difference in the distance values and the difference in the time stamp values, a vehicle speed between e.g. point 34 and 35 can be computed. Doing the same for point 35 and 36 results in two different speed values. Taking the average of them gives the vehicle speed value for trajectory point 35, which is illustrated on the right side of the table. These values are uncorrected, and therefore represent the original vehicle speed. In Figure 42, the nine road segments that correspond with the 55 trajectory points are visualised and coloured based on these uncorrected vehicle speed values. The average of the speed values of all the trajectory points that are map-matched to the same road segment is taken as the speed value for this road segment. The same procedure is applied to compute the traffic speed of the whole network, the only difference being using all the available trips per time window instead of just one example trip as it is the case in this figure. The vehicle's speed at the start and the end of the trip is very small. This is due to start and stop movements. The maximum speed is achieved in the curved blue road segment and amounts to approximately 42 km/h. Considering all the speed values of the nine road segments, an average vehicle speed of 28.2 km/h is computed for the given example taxi trip.

Order ID of taxi trip: 0e8120b4b81c75780493cc43fbb9940f

Trajectory point	Distance from start	Time stamp	Orig. vehicle speed
34	615.443 m	1477967631 s	14.22 km/h
35	628.400 m	1477967634 s	18.61 km/h
36	646.462 m	1477967637 s	26.31 km/h
37	672.245 m	1477967640 s	34.31 km/h
38	703.640 m	1477967643 s	40.30 km/h
39	739.416 m	1477967646 s	39.36 km/h
40	769.234 m	1477967649 s	40.83 km/h
41	807.468 m	1477967652 s	47.51 km/h
42	848.422 m	1477967655 s	53.52 km/h
43	896.665 m	1477967658 s	41.78 km/h
44	918.052 m	1477967661 s	32.71 km/h
45	951.177 m	1477967664 s	40.02 km/h
46	1018.340 m	1477967670 s	42.42 km/h
47	1055.452 m	1477967673 s	44.55 km/h
48	1092.584 m	1477967676 s	42.28 km/h
49	1125.925 m	1477967679 s	28.17 km/h
50	1157.668 m	1477967686 s	50.76 km/h
51	1204.995 m	1477967688 s	50.19 km/h
52	1213.435 m	1477967690 s	7.60 km/h
53	1213.435 m	1477967691 s	1.52 km/h

Table 13: Subset of an example taxi trip containing the calculated vehicle speed values for points 34 to 53. These values are derived by computing the average of two ratios between differences in the distance and time stamp of two points and are uncorrected.

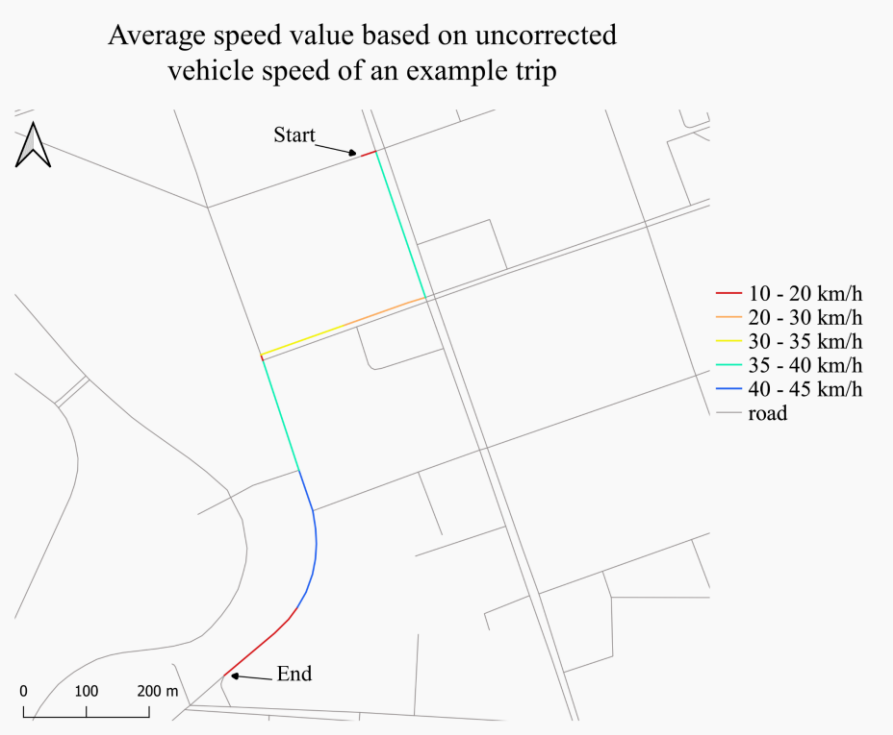


Figure 42: Computed average speed values of the nine map-matched road segments based on the uncorrected values of Table 13. The segments are coloured based on these values. The vehicle speed of the start and end segment is very small due to start and stop movements. The top speed is calculated for the blue curved segment and amounts to approx. 42 km/h.

As explained in the methodology of the traffic state estimation, the calculated vehicle speed is corrected in two stages. The first correction is related to the individual trajectories. Each taxi trip is analysed separately to filter out the mentioned start and stop movements as they should not be included in the average speed of the map-matched road segment. The threshold to filter them out is, as already explained in the methodology section, 20 km/h. If applied to the example taxi trip, the first ten and last four speed values are filtered and kept out of the remaining process. As not all trajectory points are listed in Table 13, only the last two points are affected by the correction. While computing the average speed of the nine road segments again, different values can occur. This is visualised in Figure 43. Both the start and end segments are changed. As the start segment only contains small speed values below 20 km/h, this trip is not suitable to be used to compute the traffic speed of the start segment, and therefore the whole segment is left out in this figure. The second change is visible in the end segment. By filtering out the small speed values of the last four trajectory points, the average speed value of this segment strongly increases and represents now the top speed of the trip with approximately 43 km/h. As so far only the start and stop movements are corrected, the part between the start and end segment of the path is not changed.

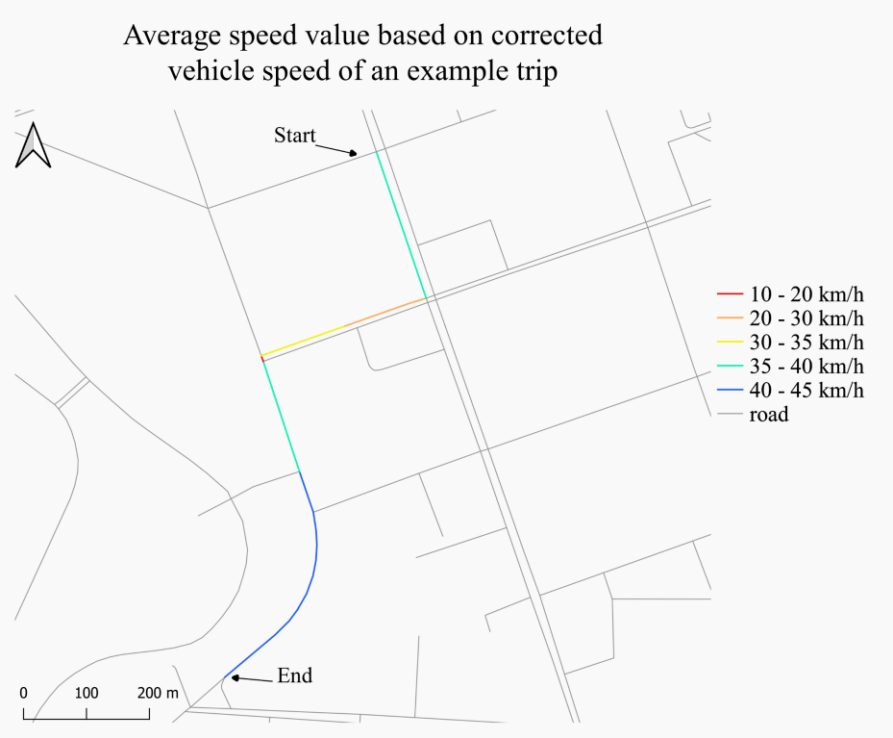


Figure 43: Computed average speed values of the nine matched road segments based on the corrected values of the example trip. The segments are again coloured based on these values. The small vehicle speed of the start and end segment is filtered out, and the average speed of these segments has changed. The top speed is now calculated for the end segment and amounts to approx. 43 km/h.

The second correction is related to the already computed average speed value of a road segment, not to the individual values anymore. If unrealistically high average speed values are calculated for a road segment, they must be re-corrected so that the traffic state is represented as close to reality as possible. The threshold used for this correction is 10 km/h above the maximum allowed speed value for the road type of the segment. Road segments containing average speed values higher than this threshold get their average speed reduced to the maximum allowed speed value. This re-correction is applied to the example trip as well and displayed in Figure 44. The first segment is an example of the type primary street. The maximum allowed speed value for roads of this type is 60 km/h. The next two road segments represent the type secondary street and their maximum allowed speed amounts to 40 km/h. The last five road segments, including the end segment, are tertiary streets and its maximum allowed speed value is 30 km/h. The calculated values for the segments of the types primary street and secondary street are below the threshold, and therefore remain unchanged. From the road segments that are of the type tertiary street, only the first two segments contain speed values below the threshold. The last three road segments all contain a speed value over 40 km/h, which exceeds the maximum allowed speed value by more than 10 km/h. Thus, their speed value is reset to 30 km/h, which is the maximum allowed speed for roads that are of the type tertiary street. Consequently, as Figure 44 portrays, these three road segments are now coloured in orange and not blue anymore. After this two-stage-correction, the average speed value of the example taxi trip amounts to 29.4 km/h.

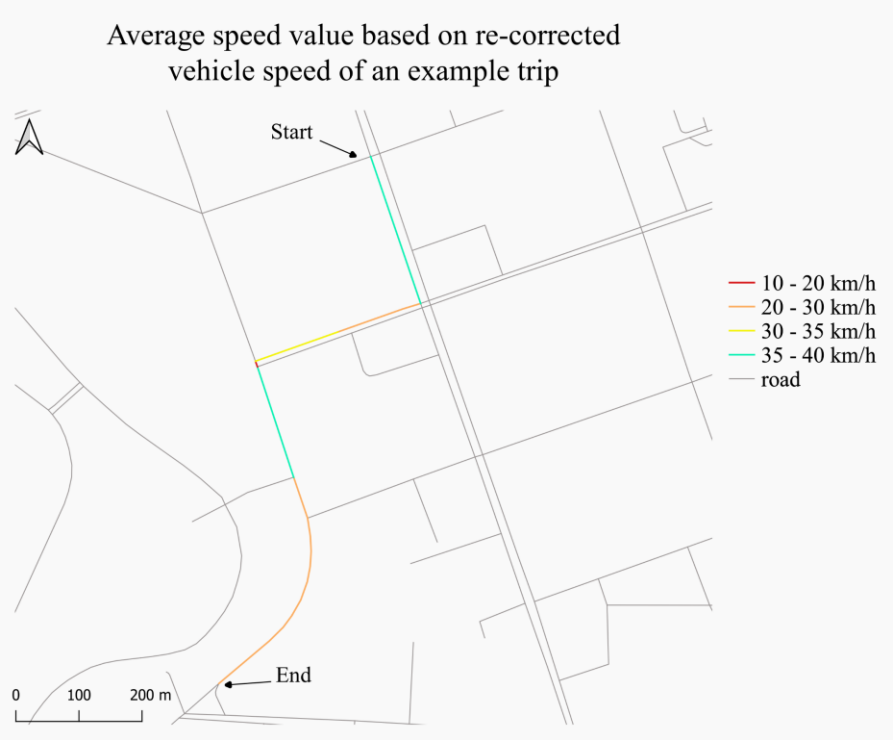


Figure 44: Computed average speed values of the nine map-matched road segments based on the re-corrected values of the example trip. The segments are still coloured based on these values. The average speed values of the last three road segments are above the defined threshold and are, therefore, reset to the maximum allowed speed value for a road of this type. The top speed is now calculated in the first segment and amounts to approx. 36 km/h.

As previously described, the traffic speed for a specific time window gets computed in a similar fashion; the only difference is that this time around, all the available map-matched trajectory points recorded in this time window are used. First, each taxi trip is filtered based on the start and stop movements and then for each road segment the average of the vehicle speeds of the map-matched trajectory points is computed. Subsequently, the calculated traffic speed is re-corrected by resetting unrealistic high speed values to the maximum allowed speed of its road type. Figure 45 presents the computed traffic speed maps for the road network of Chengdu during the time window in which the example trip started. This is between 10:30 a.m. and 10:45 a.m. on the 1st Nov. 2016. The speed values used for the map in a) are the uncorrected values and in b) the corrected values. At first sight, the traffic speed map of the uncorrected and the corrected values appear identical. But focusing on the high speed values in a), coloured in dark blue, a difference can be detected. In the map with the uncorrected values, there are small road segments in the city centre where the average speed amounts to 70 km/h and higher. This is very unrealistic as it is not possible or responsible to drive that fast in a dense city centre. Comparing these segments with the ones in b), the effect of the correction becomes visible. As they are obviously too high and, therefore, reset to the maximum allowed speed, they are coloured differently in the second map and the traffic speed is overall slower.

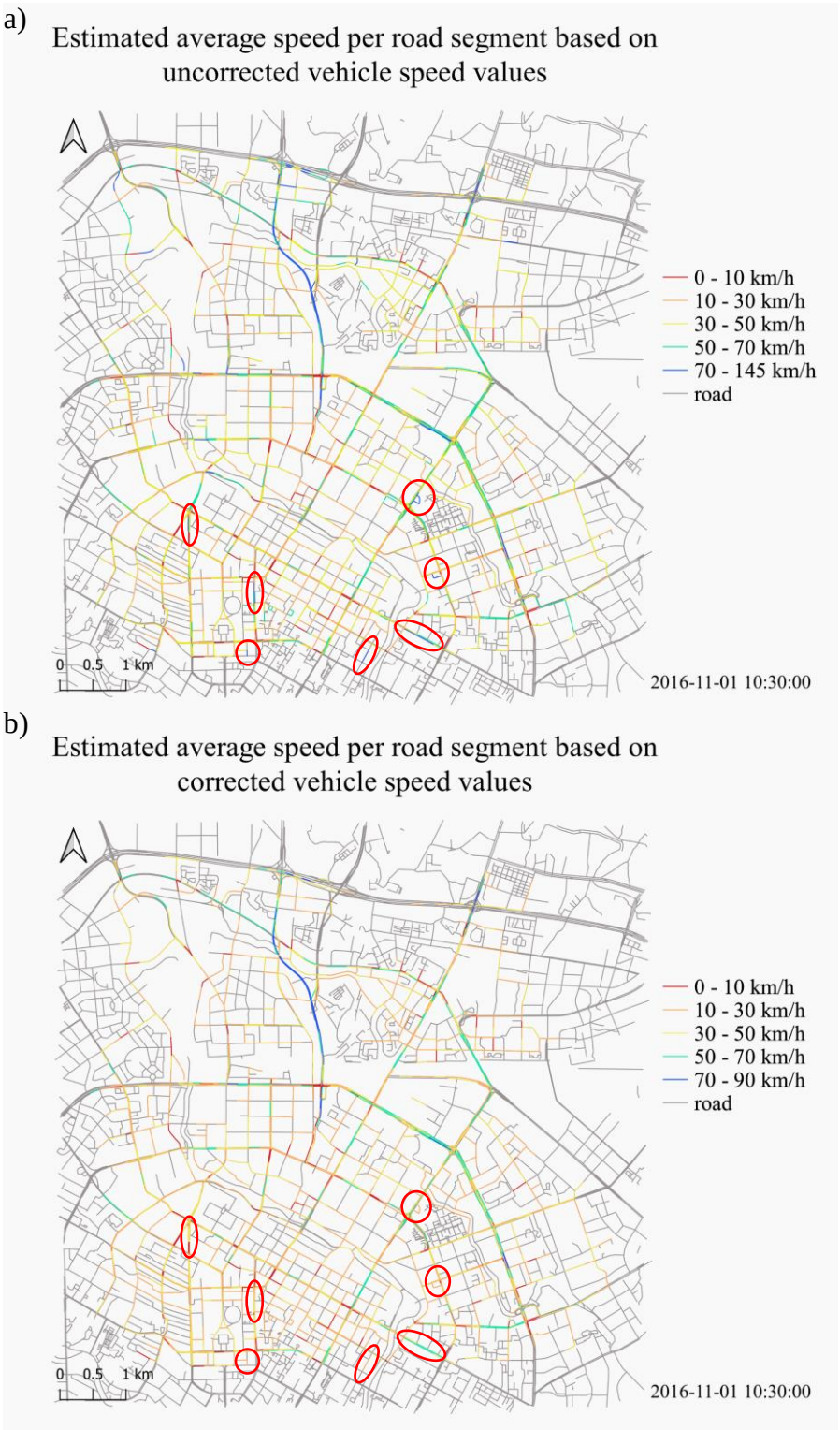


Figure 45: The computed traffic speed maps for the road network of Chengdu between 10:30 a.m. and 10:45 a.m. on the 1st Nov. 2016. In a) the uncorrected speed values are used for the traffic map. By resetting too high average speed values to the maximum allowed speed per road type, the values used for the traffic map in b) are corrected. Examples for such corrections are visualised by the red circles. In both maps, the grey roads represent the roads where no taxi was recorded during this time window and, therefore, no information on the speed is given.

Another difference is displayed in the legend of the maps. In a), there exist speed values up to 145 km/h and in b) the limit is at 90 km/h. So, the top speed is smaller with the applied correction than before. If a road segment of type motorway would be included in the calculated segments as well, a limit up to 110 km/h could be possible. In this time window, no taxi was recorded on such a motorway and, therefore, the top speed is limited to 90 km/h. The only road segments where the traffic speed is very high for both maps is the connection in the north between the 3rd and the 2nd ring road, coloured in dark blue. These road segments are of the trunk type where a maximum speed of 90 km/h (80 km/h plus the 10 km/h buffer) is allowed. So, the applied correction plays an important role in the traffic state estimation even though the visible differences are not that obvious. This gets even clearer when considering the interpolation process. All the grey coloured road segments in Figure 45 represent roads where no taxi was recorded during the analysed time window. As it is important to estimate the traffic on those roads as well, the calculated values are used to interpolate these missing values. The detailed methodology is explained in Chapter 5.4.1.1. If the unrealistic high speed values are now used for the interpolation, even higher and more unrealistic values can be expected. Thus, it is very important to first correct the calculated values before applying the interpolation algorithm to estimate the traffic state as realistically as possible. The resulting corrected and interpolated traffic speed map for the same time window is visualised in Figure 46.

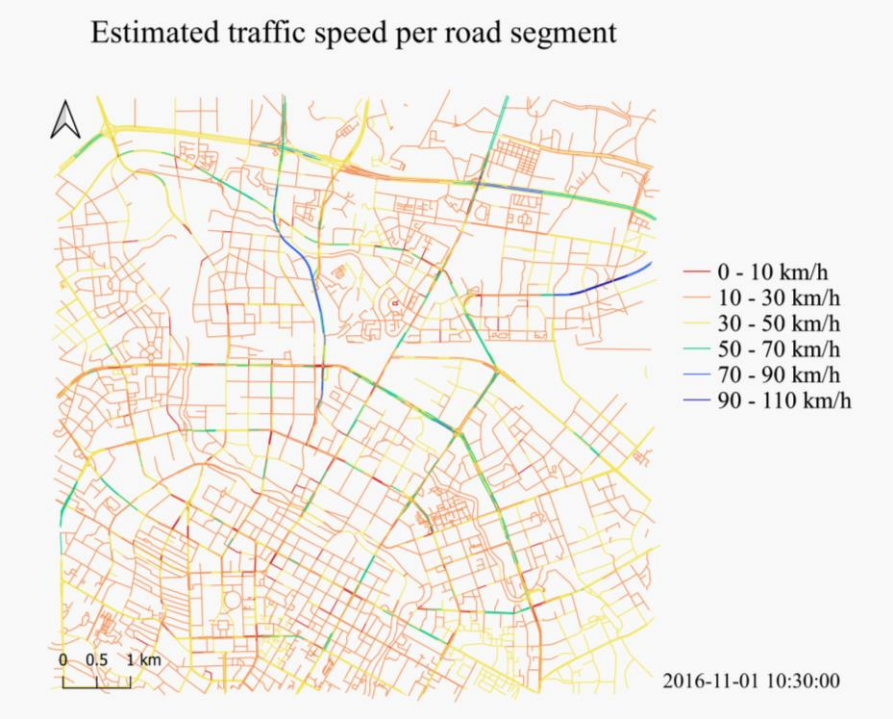


Figure 46: The interpolated traffic speed map for the time window between 10:30 and 10:45 a.m. Based on the corrected vehicle speed values, each road segment has an estimated value assigned.

This figure shows the final estimated traffic state for this specific time window. Most of the road segments are coloured orange or yellow, meaning speeds between 10 km/h and 50 km/h are the most common ones. This makes sense as the data is collected in a dense urban road network. High speed values are only estimated at the already mentioned connection of the 3rd and 2nd ring road in the north and the motorway coloured in blue and dark blue in the north-east. Speed values between 50 km/h and 70 km/h are mostly located on the ring roads or on some primary streets connected to them. In the middle of the city centre, speed values below

10 km/h are sometimes estimated. Such small speed values are due to traffic congestions triggered by traffic lights, jam-packed roads, or car accidents. Overall, the quality of the estimated traffic speed is good, especially as there do not exist unrealistic high speed values in the middle of the city centre or big changes in the speed values between two connected road segments.

To further evaluate the quality of the traffic state estimation, the map in Figure 46 is compared to the traffic speed map based on the maximum allowed speed values per road type, illustrated in Figure 47. The biggest difference is visible on the ring roads. In Figure 47, speed values bigger than 50 km/h are possible on each road segments of a ring road or primary street connecting these ring roads. Sometimes, even speed values over 70 km/h are given. Values below 10 km/h are barely seen. Only a small section of a living street in the north of the city centre contains such speed values. Not a big difference is given for the tertiary streets coloured in orange. In both maps, these small roads contain values of 10 km/h to 30 km/h and are represented in great quantity. Additionally, the trunk road connecting the 3rd and 2nd ring road and the motorway are almost equal for both cases. Overall, in Figure 46, the speed values in the city centre are smaller compared to the maximum speeds allowed of Figure 47. This represents another quality sign as such differences are expected to occur when considering the traffic state. Furthermore, it shows that not working with information on the traffic state assumes too high speed values compared to the reality. Thus, by assuming an absence of traffic congestions, the real-world circumstances get distorted.

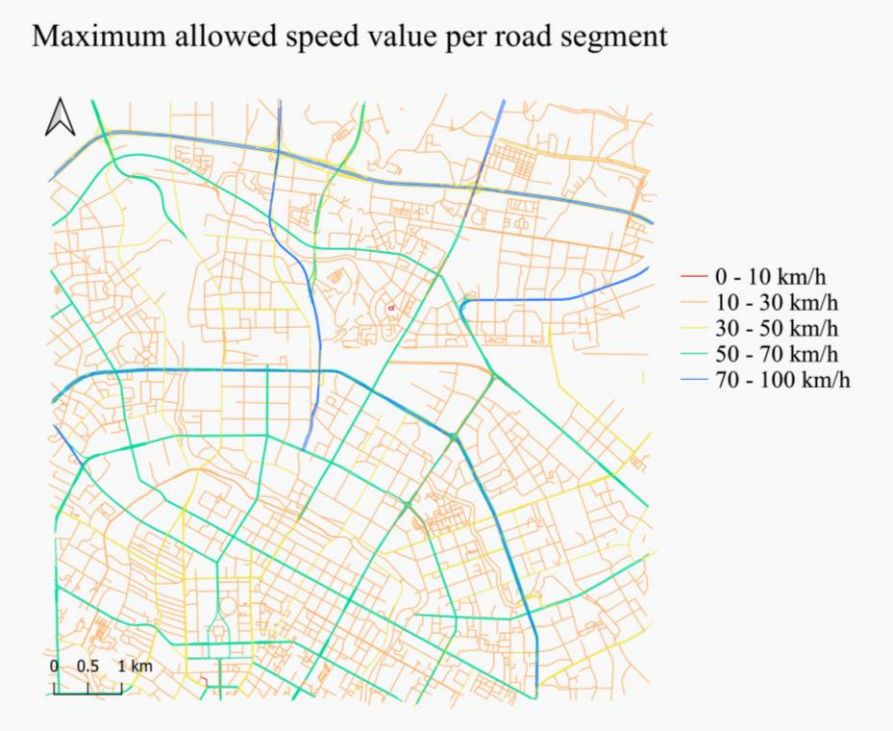


Figure 47: Traffic speed map based on the maximum allowed speed value for each road type.

In theory, traffic state is highly dependent on the time of the day. Thus, the average speed value of a road segment normally differs a lot during the day. As in the traffic state estimation part the traffic speed is computed for every 15 minutes, the created maps can be analysed on this phenomenon. This is done through the visualisation in Figure 48. Shown are four traffic speed maps of four different time windows during the 1st Nov. 2016. Focusing on a), the high speed values of the ring roads and some primary roads stand out. In general, the road network seems to be less congested and the vehicles can drive faster on average. This map represents a situation during the night in the city centre of Chengdu. The selected time window is between 3:15 and 3:30 a.m. In b), the previously discussed time window is given. It is clearly visible that the traffic increased and that the network has become more congested. The average traffic speed in c) is even slightly smaller than in b). This situation represents the rush hour between 6:30 and 6:45 p.m. The ring roads are more congested than in b) and overall, for both cases, the traffic is slow-moving. The situation relaxes in d), where like in a), the ring roads are less congested, and the average speed is higher. The selected time window is between 11:30 and 11:45 p.m. This short analysis shows the transformation of the road network conditions of the study area during that day and confirms the mentioned theory about the time dependency of traffic state.

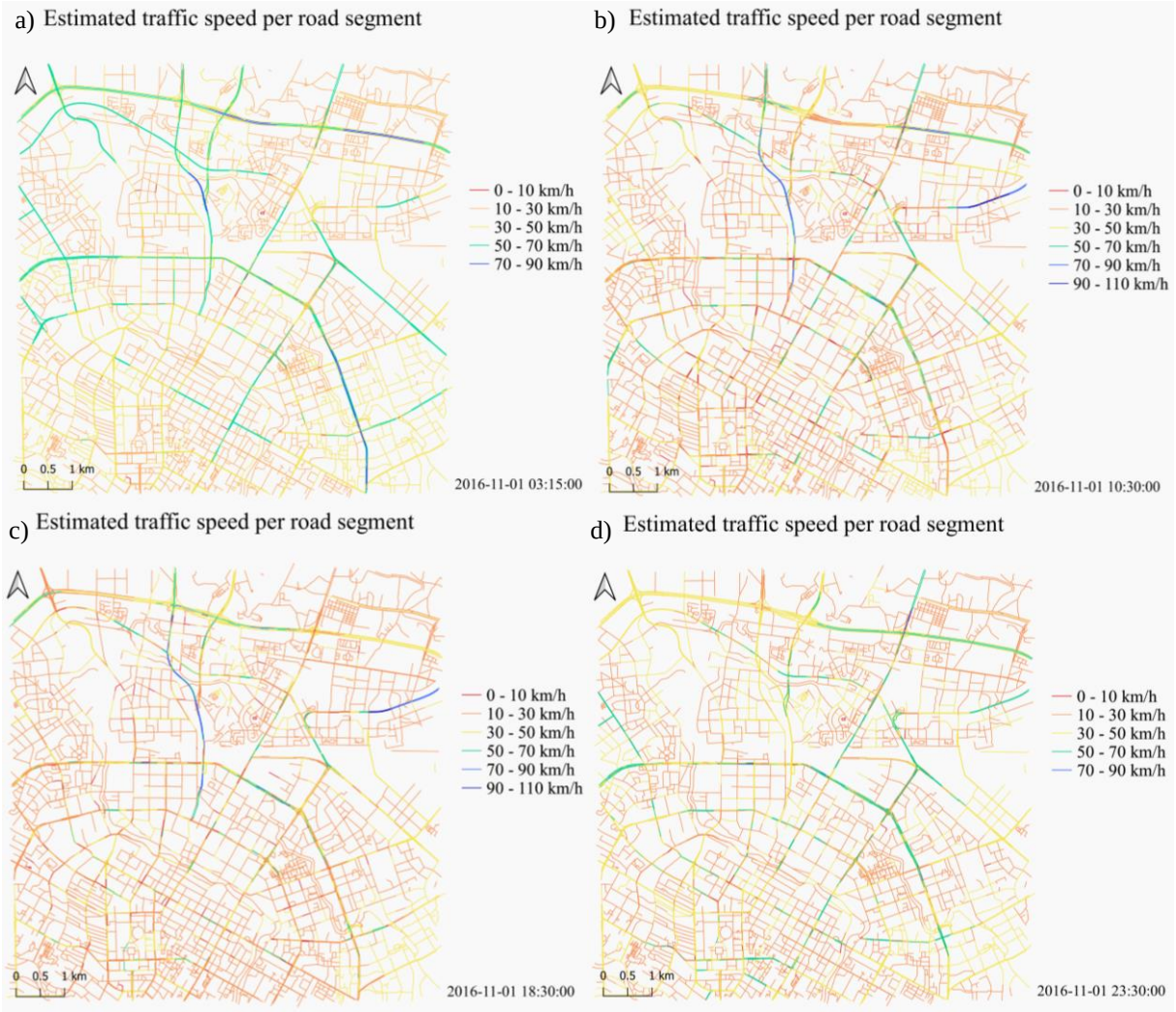


Figure 48: Analysis of the estimated traffic state by comparing the traffic speed maps of four different time windows. In a) the traffic speed map during the night is shown. The map in b) is the same as in the previously discussed time window. In c) the time window is between 6:30 and 6:45 p.m. and d) represents the situation a few minutes before midnight (between 11:30 and 11:45 p.m.).

The final estimated traffic state can either be represented by the traffic speed as illustrated and discussed in Figure 48, or it can be given by the travel time needed to pass a road segment. The latter is only available if the traffic speed is estimated. Using these values and the information about the length of each road segment, a value given in minutes and seconds is calculated telling how much time on average is needed at a specific time of the day to drive through a specific road segment of the network. As described in the methodology section, this travel time is then used as the weight of the edges in the weighted Dijkstra's shortest path algorithm. Because of the direct connection between travel time and traffic speed, these values vary also during the day. To present the scale of the size of these travel times, a map of the estimated travel time per road segment for the known time window between 10:30 and 10:45 a.m. is displayed in Figure 49. Analysing the change in the values during the day would result in the same findings as previously detected for the traffic speed, just in a different unit, and therefore is not repeated here. The map shows that the travel time discrepancy ranges from a few seconds up to 5 minutes. The main influence on these values, besides the traffic speed, is given by the length of the segment. A long road segment has automatically a longer travel time assigned independent of the traffic speed. Thus, this map must be interpreted with caution. The results of including the presented estimated traffic state into the identification process of potential ride-sharing paths are provided in the subsequent sections.

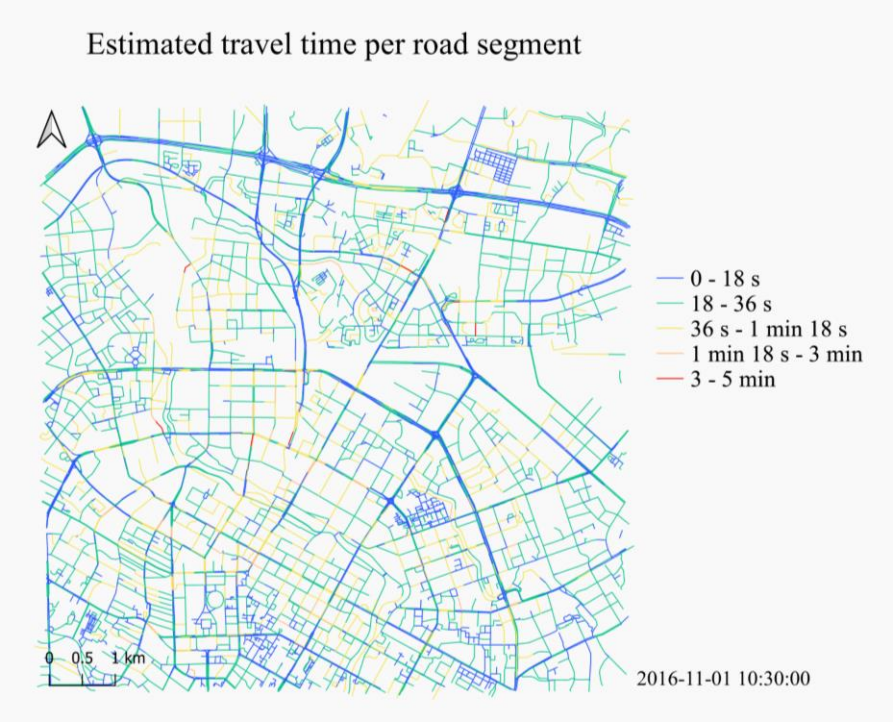


Figure 49: Visualisation of the estimated travel time per road segment for the time window between 10:30 and 10:45 a.m. in the city centre of Chengdu. The values vary from a few seconds up to five minutes. As the travel time, besides the traffic speed, is strongly dependent on the length of each road segment, this map must be interpreted with caution.

6.3 Similarity of trajectories

Before the presented traffic state information is included in the optimal path identification process, the similarity between the candidate and the analysed taxi trips is measured not only to exclude unsimilar trips, but also to avoid needless fastest shared path computations. The similarity is measured between an analysed trip and each candidate trip that started inside the defined time window of five minutes. Depending on the number of requested taxi trips and, therefore, on the time of the day, the total amount of available candidate trips for the identification process can vary. This is visualised in Figure 50. Illustrated are the changes in the number of candidate trips over the time of the day for the original data and four variations, which is explained in more detail in the following paragraph. The black line represents how many candidate trips are generally available over the day. It is visible that during the night, on average, fewer candidate trips are available mainly due to the decrease in the requested taxi trips. During the day, the availability varies between approximately 130 and 170 candidate trips and drops slightly at the end of the day. On average, 140 candidate trips are available for each analysed trip.

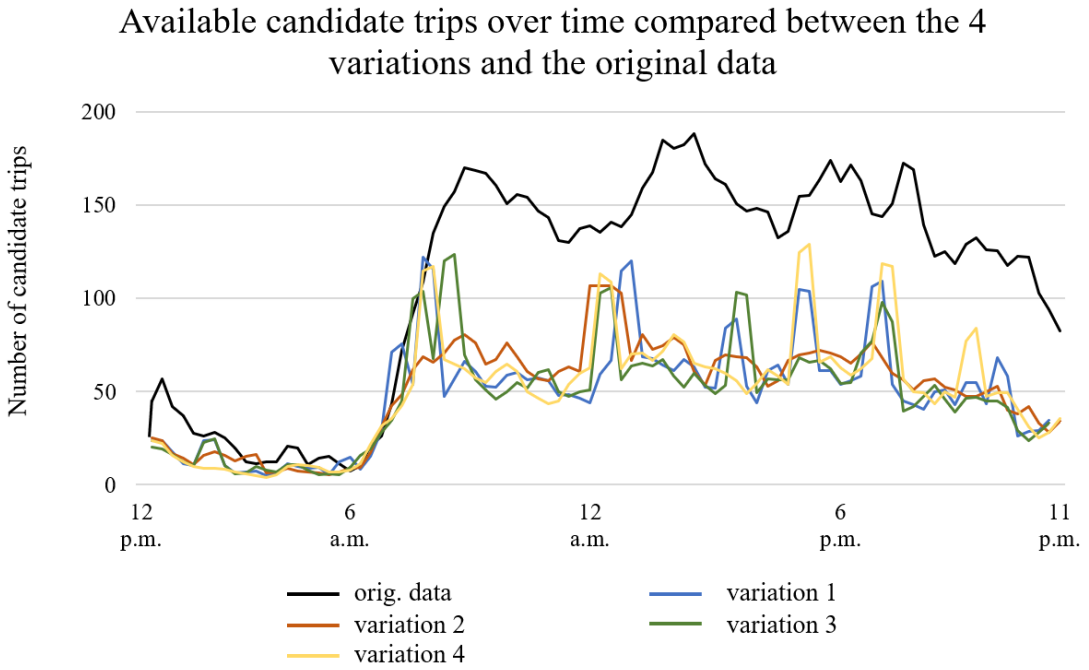


Figure 50: Comparing the number of available candidate trips over the day between the four different applied variations and the original data. The total amount of available candidate trips is smaller for all four variations compared to the original data as already identified ride-sharing partners are not considered anymore as candidate trips in the identification process.

As explained in the methodology, when a ride-sharing duo is identified or a trip is marked as unsuitable for ride-sharing, the trips get eliminated from the candidate list so that they will not be considered for subsequent matching anymore. This means that the total amount of available candidate trips can decrease in time if a lot of ride-sharing matches are found. Thus, the black line in Figure 50 changes over the day. For better understanding, the available candidate trips over the day for each of the in Chapter 5.6 presented four variations are displayed as well. The first two variations stand for including the traffic state information in the identification process: once excluding the distance savings constraint and once including it. The other two variations stand for assuming an absence of traffic congestions: in variation three while excluding the

distance constraint and in variation four while including it. This distance constraint means that the identified shared paths must save travel distance compared to the sum of the distances of the two individual trips. It is clearly visible that especially during the day, the number of available candidate trips for all the four variations is smaller compared to the black line. Between the four variations, no big difference is visible. This can be seen as well by comparing the average of available candidate trips for each variation. In the first case, on average 58 candidate trips are available. Considering the distance savings constraint in variation two as well, the average rises to 66 candidate trips. In situation three, on average only 57 candidate trips are available. This number then slightly increases to 62 trips for variation four. The only difference is that when the distance savings constraint is included, the average of available candidate trips is slightly higher than without this constraint (for considering the traffic state the average contains 8 more candidates and for assuming an absence of traffic congestions the average amounts to 5 more candidate trips). This allows for an assumption that less ride-sharing paths are identified while the additional constraint is included as like this more often only one (the analysed trip that is marked as unsuitable for ride-sharing), and not two trips, are eliminated from the candidate list and, thus, more candidates are available in the end.

In the first variation, the most similar candidate trips have on average a measured Similarity Measurement Index (SMI) of 567.12. This means that both the start and the end points are on average 567.12 m away from each other (either the two start respectively end points or one of them and the closest trajectory point of the other trip). For the second variation, an average SMI of 521.12 is measured for the most similar candidate trips. With 513.36, an even smaller average SMI value is given for the variation where neither the traffic state nor the distance constraint is considered. The highest average SMI for the most similar candidate trips is measured for the fourth variation and amounts to 628.06.

How the measured SMIs differ between the three most similar candidate trips and an unsimilar trip is illustrated in Figure 51a and 51b. In a), an example trip and its most similar candidate trip are displayed. Each start and end point is marked, and the distance that represents the value for the similarity between both start points and end points is visualised. As for each start point the closest point on the other trajectory is not its start point, the distance of 497 m between the two start points is measured. For the end point of trajectory two, the closest point on the other trip is its end point, but this does not count for the end point of trajectory one. Thus, the displayed distance of 273 m is measured. Taking the average of these two values, a final SMI of 385 results. This is the smallest SMI for the analysed example trip and, therefore, this candidate trip is identified as the most similar one. The candidate trip in b) is the second most similar trip. Its SMI of 1’008 is significantly higher and is only considered to be similar because the value that represents the distance between both end points is rather miniscule. The two start points are located very far from each other. As it can take the taxi a long time to drive from the first to the second start point, the emerging waiting time for the second user is potentially bigger than the threshold of 15 minutes. Thus, this candidate trip will eventually be excluded in the ongoing process. In c) of Figure 51b, the analysed and the third most similar trip are shown. Their collocation represents a special situation, where one trip has finished before the other even started. As explained in the methodology, the SMI still gets measured the same way. This results in a distance for the start points of 2’703 m and a distance for the end points of 337 m. For both start points, the closest point on the other trajectory is not its start point and, therefore, the direct distance between both points represents the distance value for the start points.

Whereas the closest point on the first trajectory for the second end point is its end point, the closest point on the second trajectory for the first end point is not its end point. Thus, the short distance of 337 m represents the distance value between the end points. Taking the average of both values, an SMI of 1’520 results. As this is the third most similar candidate trip, it will be used for the matching process, even though it is obviously not a representation of a ride-sharing situation as combining these two trips will not result in a shared part. To prevent wrongly identifying this collocation as an optimal ride-sharing path, the travel time and distance savings constraints are applied. Those dictate that a shared path must reduce the total travel time and, in variation two and four, also the total distance compared to the sum of both individual trips. As a situation where one trip has finished before the other started cannot surpass these constraints, the presented collocation of c) in Figure 51b is excluded in the explained step.

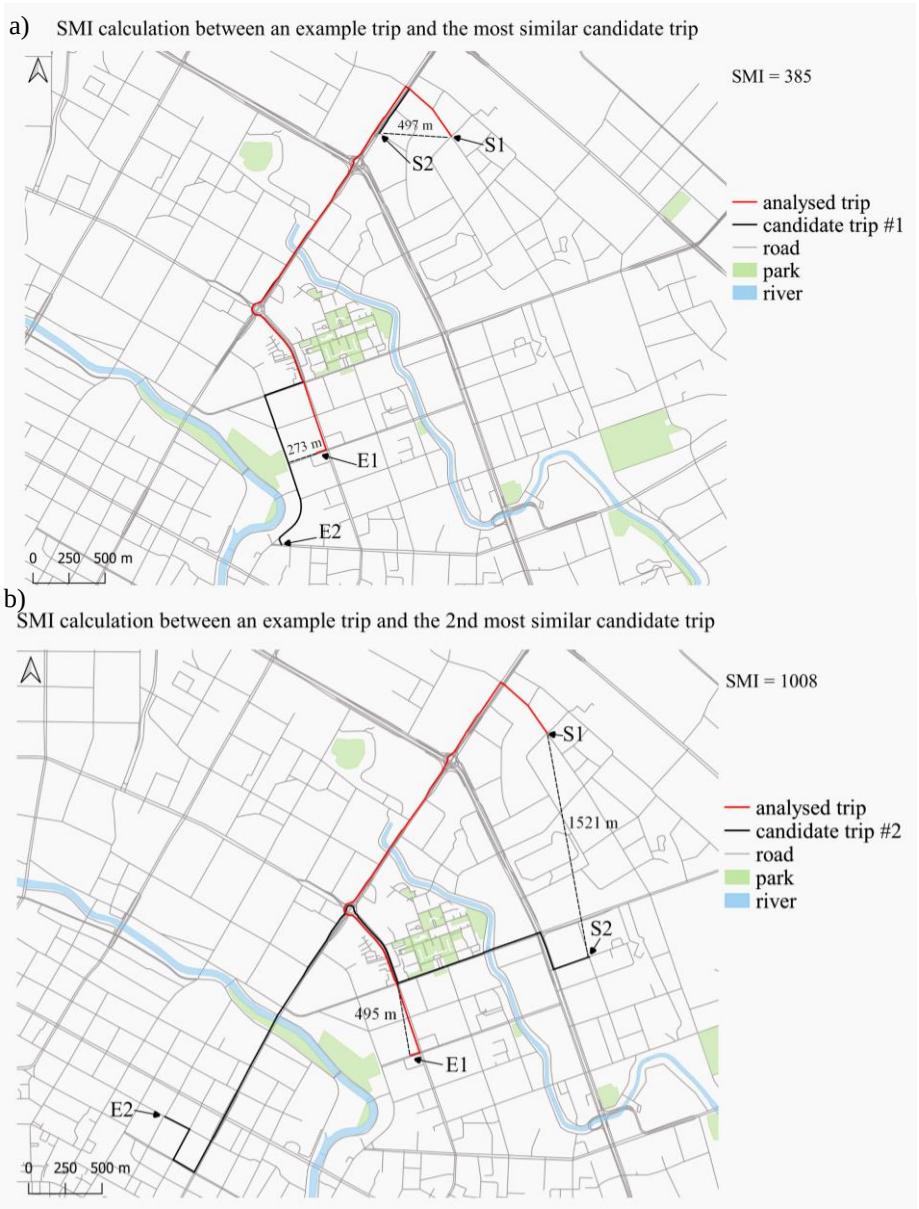


Figure 51a: Visualisation of the analysed and the most similar candidate trip in a) and the analysed and the second most similar candidate trip in b). Illustrated with the dashed black line are the distances that represent the final distance value between both start and end points and are used for the SMI calculation. The collocation in b) is excluded from further processes as it possibly does not fulfil the set constraint about the emerging waiting time below 15 minutes.

As both the second and third most similar candidate trips do not fulfil the set constraints and the remaining candidate trip is the most similar one anyways, the fastest path serving both start and end points of the analysed and the candidate trip in situation a) will represent the optimal identified ride-sharing path for the candidate and the example trip. The situation in d) is an example of an unsimilar candidate trip. Besides the fact that both the start and end points are located very far away from each other, the two trips are aligned in the opposite direction. So, it is obvious that these two trips should not be shared. The big SMI of 5'676.5 confirms this. In other ride-sharing systems where each trip is a possible candidate, a lot of needless fastest path computations would have been done for such cases. This is prevented with the implemented similarity measurement and, thus, the time consumption of the computation is reduced.

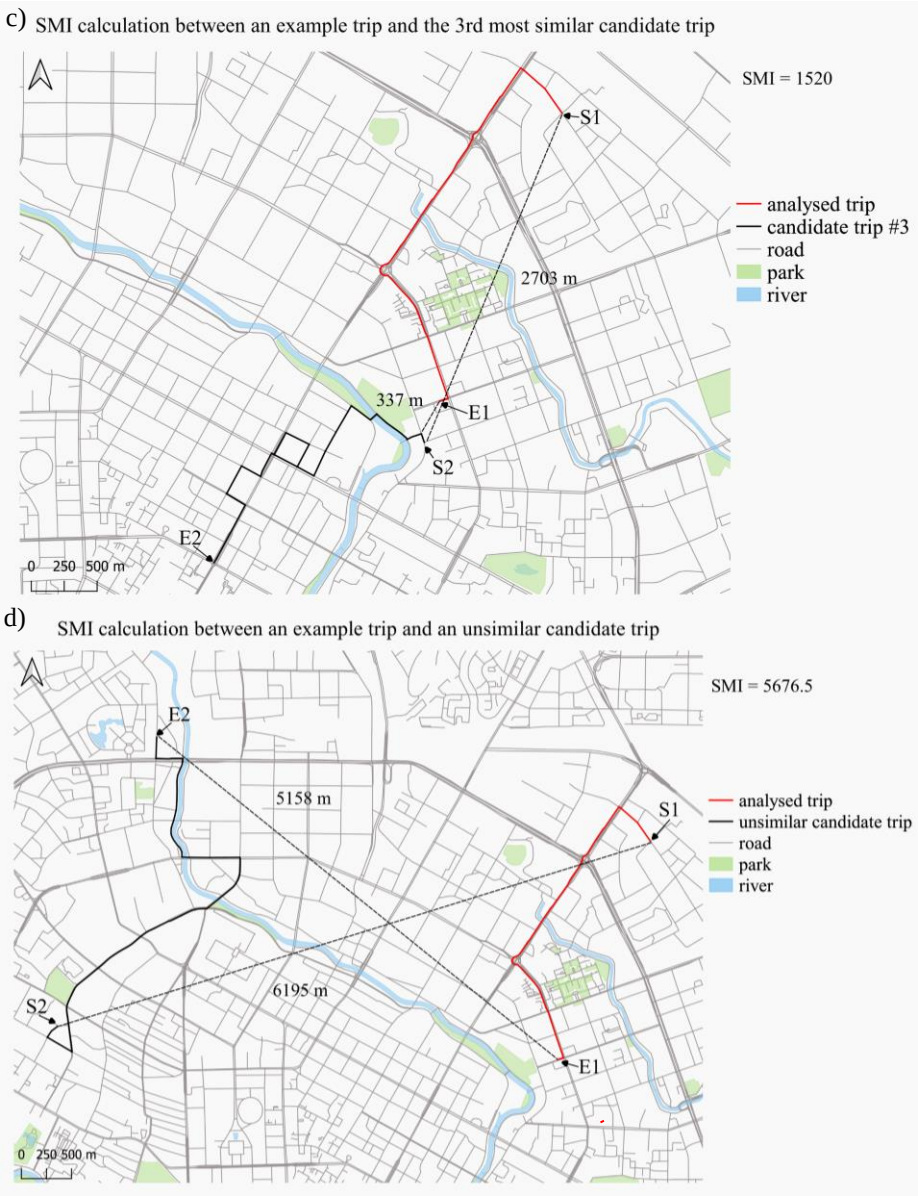


Figure 51b: Visualisation of the analysed and the third most similar candidate trip in c) and the analysed and an unsimilar candidate trip in d). Illustrated with the dashed black line are the distances that represent the final distance value between both start and end points and are used for the SMI calculation. Both collocations are excluded from further processes as in c) the set constraints are not fulfilled and the candidate trip in d) is not one of the three most similar candidate trips.

6.4 Identified potential ride-sharing paths

The result of the similarity measurement for the analysed example trip is the presented three most similar candidate trips. As explained in the previous section, only the most similar candidate trip is suitable for ride-sharing and fulfils the set constraints, and therefore the result of the fastest path computation is only shown for this one. Figure 52 contains three different maps. In a), the map-matched trajectory of the analysed example trip is visualised and in b), the map-matched trajectory of the most similar candidate trip is given. During the identification process of the optimal ride-sharing path, one of the four possible collocations described in Figure 34 that leads to the fastest shared path for the two illustrated trips must be selected. As shown in c), by the identified optimal ride-sharing path first S2 gets served and then the passenger of the analysed trip is picked up. After following the red visualised ride-sharing path, the passenger of the analysed trip gets dropped off at E1 and then the shared ride is finished at E2.

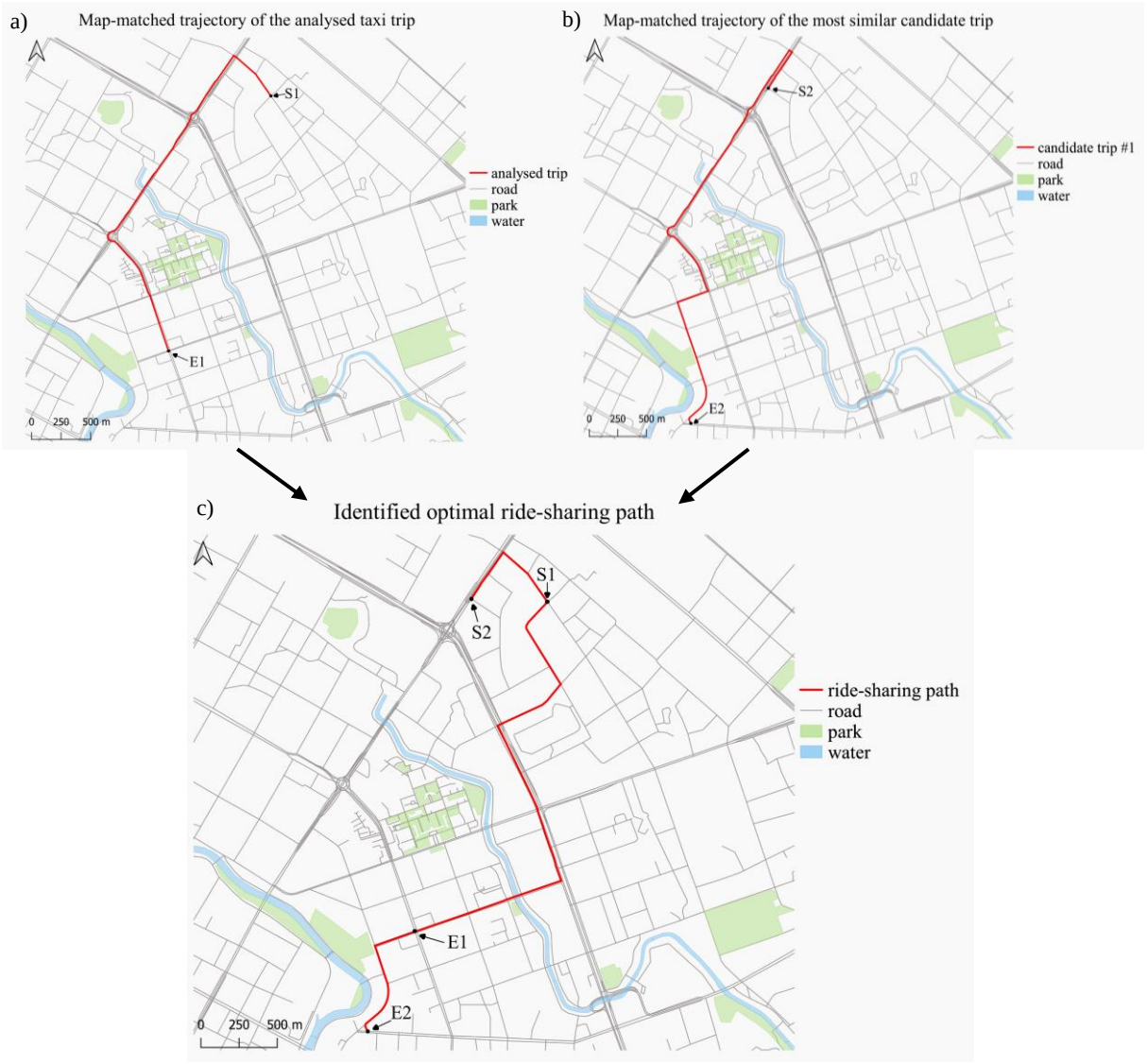


Figure 52: Visualisation of the identified optimal ride-sharing path for the analysed example trip of the previous section. In a), the analysed example trip is illustrated and in b) the most similar candidate trip is given. The resulting ride-sharing path that optimises the objective of the system and fulfils all the set constraints is displayed by the red line in c).

As S2 is located on a road going north, it would be very unsuitable if the shared trip would start at S1, because then the taxi would have to first drive more south, change its direction to the north, and then pick up the second passenger. This would lead to an unnecessary detour and, thus, the algorithm correctly identifies the best collocation of the two start and end points. The discussed path represents the identified optimal ride-sharing path for the situation where the estimated traffic state information is not included in the identification process, but the distance savings constraint must be met. Considering this, the algorithm must find a path that leads through road types of high maximum allowed speed limits while keeping the total driving distance small. This is maintained as the shared path follows a direct way after picking up the second passenger at the start point of the analysed trip through roads of type tertiary street (max. allowed speed of 30 km/h) to the trunk running north to south-east, instead of generating a big detour to stay on the road of type primary street running north to south-west (60 km/h). On this part of the identified path, the maximum allowed speed amounts to 80 km/h. Subsequently, the taxi drives through a road of type secondary street (max. allowed speed of 40 km/h) until it reaches the road that leads directly to E2 and is again of the tertiary street type.

Summing up the two travel times that would occur if the analysed and the most similar candidate trip had been served individually results in a total travel time of 19 min 31 s. The total travel time of the identified ride-sharing path amounts to 7 min 53 s. Thus, 11 min 38 s of travel time can be saved by implementing the proposed ride-sharing system for this specific example. In other words, the shared path leads to a total travel time saving of 59.6%. The second passenger, in this case the passenger that requested the analysed trip, must wait only 1 min 53 s after the trip started until he gets picked up. The travel distance of the analysed taxi trip is 4.08 km and the candidate trip is 4.41 km long. Summing this up, a total driving distance of 8.49 km occurs. On the contrary, the shared path leads to a total driving distance of 6.43 km. So, besides saving travel time, the shared path can save 2.06 km of travel distance. This equals a total driving distance saving of 24.2%.

In addition to reducing the overall travel costs for passengers and the number of vehicles on the road networks, ride-sharing can have an influence on emissions as well, most importantly the emission of carbon dioxide (CO_2). Thus, it is interesting to analyse how much CO_2 emissions can be saved by implementing a proposed ride-sharing system. Several other studies do the same and assume a linear correlation between the driven kilometres and the CO_2 emissions. Santi et al. (2014b), for instance, show that even though vehicle emissions can be highly non-linear because of factors like the speed, the traffic signals, or the driver mentality, assuming all things being equal is legitimate for the purpose of showing the potential of ride-sharing systems in relation to the natural environment. Therefore, in this work, a linear correlation between the driven kilometres and the CO_2 emissions is assumed as well. As calculated and published by Mobitool (2016), a factor of 197.23 g CO_2 / km can be used for mobility in Switzerland. Even though this study is based on data collected in China, the Swiss factor is applied, as the performance of taxis are expected to be similar all over the world. The saved travel distance of 2.06 km for the discussed example trip leads, therefore, to an emission reduction of 406.29 g CO_2. The presented fastest shared path is an example of an effectively identified match with a short waiting time of less than 2 minutes, travel time savings of more than 50%, and distance savings of nearly 25%. All the mentioned results are summarised in Table 14.

	Travel time	Travel distance	CO_2 emission	Waiting time
Analysed trip	9 min 27 s	4.08 km	804.69 g CO_2	
Candidate trip #1	10 min 4 s	4.41 km	869.78 g CO_2	
Sum of the individual trips	19 min 31 s	8.49 km	1.675 kg CO_2	
Shared trip	7 min 53 s	6.43 km	1.268 kg CO_2	1 min 53 s
Savings [min / km / g CO_2]	11 min 38 s	2.06 km	406.29 g CO_2	
Savings [%]	59.6%	24.2%	24.2%	

Table 14: Summary of the resulting measures for the identified optimal ride-sharing path of the analysed example trip of this and the previous section.

The above-presented results refer only to one identified optimal ride-sharing path. Even more interesting are the results of these individual measures for all the analysed data. Additionally, the differences between the mentioned four variations and with this the influence of the traffic state information and the distance saving constraint are of big interest. Therefore, in the following two sections, first, the overall results of the ride-sharing system including the estimated traffic state information are provided. In the second part, the results for the same measures assuming an absence of traffic congestions are shown. To explain how the traffic state information and the distance saving constraint influence the matching process, the identification of the fastest shared path and the resulting measures of another example trip with its three most similar candidate trips are presented and compared between the four possible variations. In the end, an overview of the overall results for all four variations of the implemented ride-sharing system is provided.

6.4.1 Including traffic state information

The results provided in this section are obtained by implementing the proposed ride-sharing system while considering the estimated traffic state of the underlying road network. To show how the distance saving constraint influences the identified ride-sharing paths, this section compares two situations. First, the traffic state information is included, but the distance savings constraint ignored. Second, the traffic state information is again included and additionally, the distance savings constraint must be met. This comparison is based on another analysed example taxi trip and its three most similar candidate trips. The analysed trip (illustrated in red) and each candidate trip (the black lines) are visualised in Figure 53 a) to c). In a), the most similar candidate trip with an SMI of 499.34 is shown. For the second most similar candidate in b) an SMI of 841.43 is measured. With an SMI of 869.90 in c), the third most similar candidate trip is displayed. The identified optimal ride-sharing path for the analysed trip given the first variation of not including the distance saving constraint is illustrated in d). This shared path is a combination of the analysed and the second most similar candidate trip. First, the taxi visits S1 before driving through the roundabout to pick up the second passenger. After following the red line, the second passenger gets dropped off at E2, and then the ride concludes at E1. The main difference to the previous example trip is that this time around, not the most similar but the second most similar candidate trip is selected to build the optimal ride-sharing path. As in the first variation, only the total travel time must be smaller than the sum of the travel times of the two individual trips, which means that all three candidate trips could be potentially selected for building the ride-sharing path so far. Whether or not is the final driving distance smaller or even bigger than before applying ride-sharing does not matter in this variation. As it can be seen in c), the two start points are located quite far from each other and, thus, the emerging waiting time is slightly bigger than the threshold of 15 minutes. From the two remaining

candidate trips, the one with the analysed trip shared path that minimizes the waiting time and, therefore, maximizes the followed objective of the system, is selected as the optimal ride-sharing path. As sharing the analysed trip with the second most similar candidate trip leads to a shorter waiting time of 4 min 4 s than with the most similar candidate trip (4 min 32 s), this combination is identified as the optimal ride-sharing path for the analysed trip. The sum of the travel times, if the analysed and the second most similar candidate trip had been served individually, is 20 min 27 s. The total travel time of the identified shared path is 17 min 3 s and, thus, 3 min 24 s of travel time can be saved with the proposed ride-sharing system for the analysed taxi trip in variation one. As already mentioned, when the distance saving constraint must not be met, the total driving distance of the shared path might even be bigger than the sum of the travel distances of the two individual paths. This can occur when the shared path selects a road where the driven speed is much higher and, therefore, the travel time in total shorter but reaching this road leads to a detour. This is the case in variation one as the driving distance of the shared path is 9.82 km long and, thus, 1.13 km longer than the sum of the distances of the two individual trips of 8.69 km. By still assuming a linear correlation between driven kilometres and CO_2 emissions, this identified shared path would lead to an increase in emissions of 13% what equals 222.87 g CO_2.

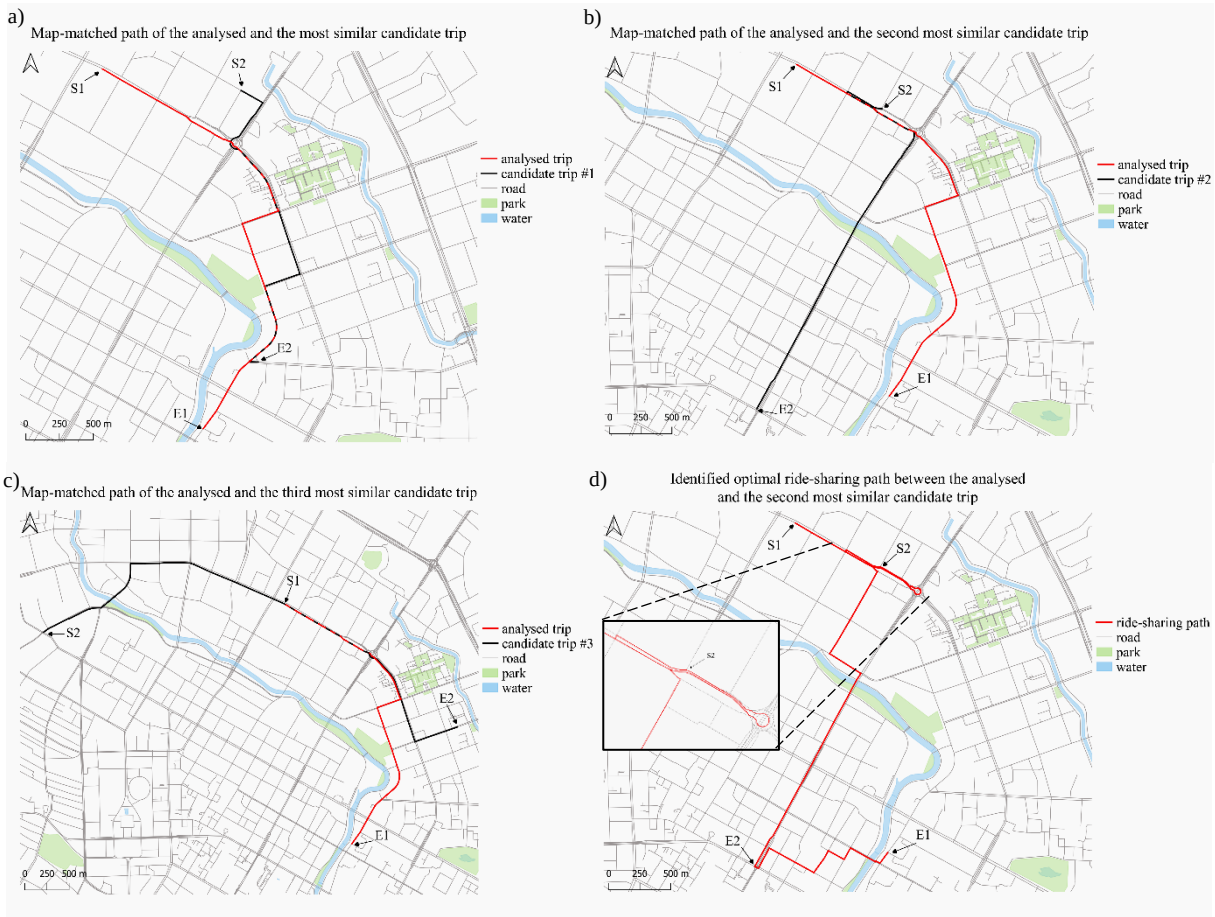


Figure 53: Visualisation of the three most similar candidate trips and the optimal ride-sharing path for the analysed trip illustrated by the red line in a) to c). The final shared path is displayed in d). This path is a combination of the analysed with the second most similar candidate trip. As in variation one the distance savings constraint must not be met, the total driving distance of the shared path can be bigger than the sum of the distances of the two individual taxi trips, as it is the case in d). A bigger size of the figure is given in the appendix of this work.

In the second variation, the optimal ride-sharing path must reduce the total driven kilometres in comparison to the sum of the two individual trips while the estimated traffic state information still is considered. To show what differences in the selection of the optimal candidate trip occur using this additional constraint, the same example trip is analysed again. As shown in Figure 54 a) to c), the three most similar candidate trips remain the same as in Figure 53. As it is prohibited that the shared path lets the total driving distance increase, the second most similar candidate trip in b) is not an option for sharing the ride with the analysed trip anymore. Only the most similar in a) and the third most similar candidate trip in c) remain. As already explained in the first variation, in c) the waiting time emerging for the passenger that joins the ride second at the start point of the analysed trip is bigger than the set threshold of 15 minutes (16 min 19 s). Considering the traffic speed map as displayed in Figure 55, besides the big distance between the two start points, a second reason for this too big waiting time stands out. When focusing on the road segments between S2 and S1 of figure 53 c), in Figure 55 (dashed buffer) mostly orange and red coloured lines can be identified. This means that at this time the taxi on average drives only between 0 km/h and 10 km/h or 10 km/h and 30 km/h. So, due to the big distance and the bad traffic state given, the shared path with this candidate trip is not suitable for ride-sharing. The remaining shared path between the analysed and the most similar candidate trip fulfils all the set constraints and is, therefore, identified as the optimal ride-sharing path for the analysed example trip given the second variation. This ride-sharing path is visualised in Figure 54 d). So, in the second variation a different optimal shared path is identified than in the first one and, thus, the only difference between Figure 53 and 54 is given in the maps in d).

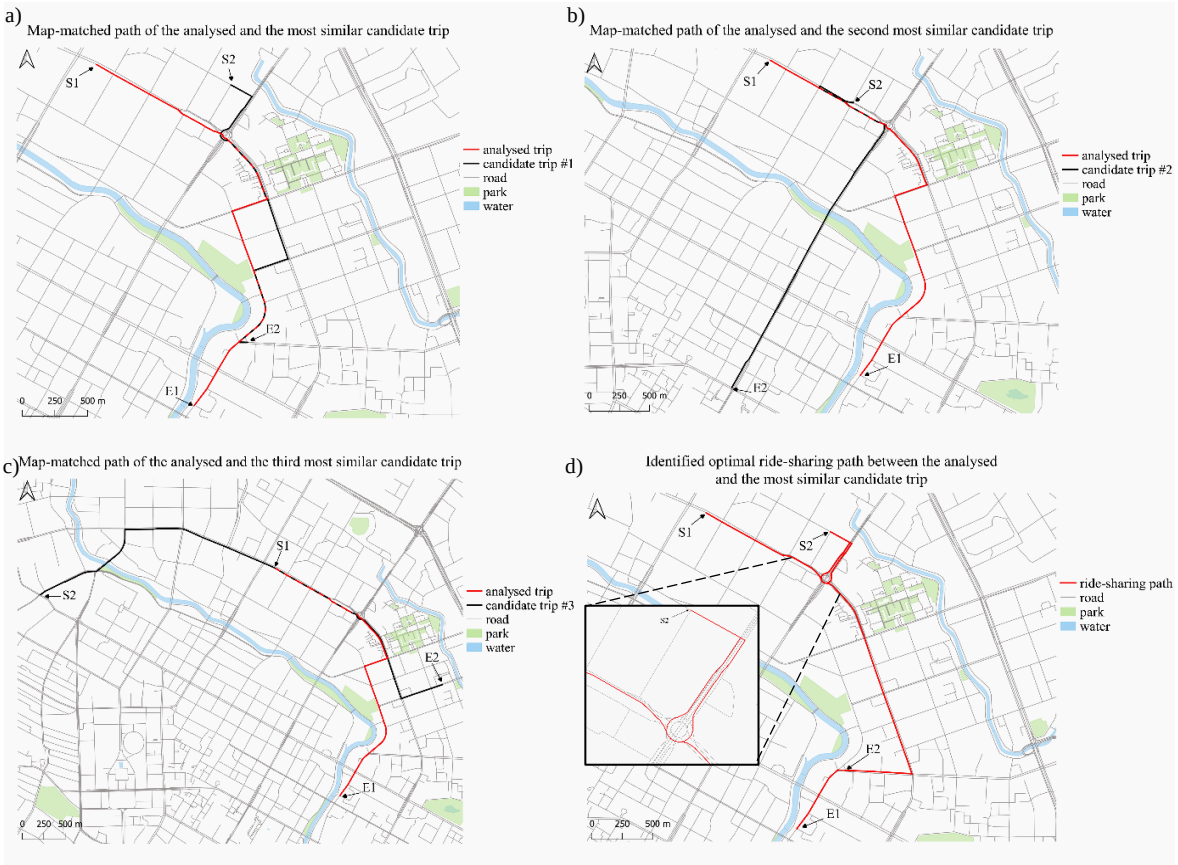


Figure 54: Visualisation of the three most similar candidate trips in a) to c) and the optimal ride-sharing path in d) for the analysed trip. The final ride-sharing path of the second variation is a combination of the analysed with the most similar candidate trip. A bigger size of it is in the appendix.

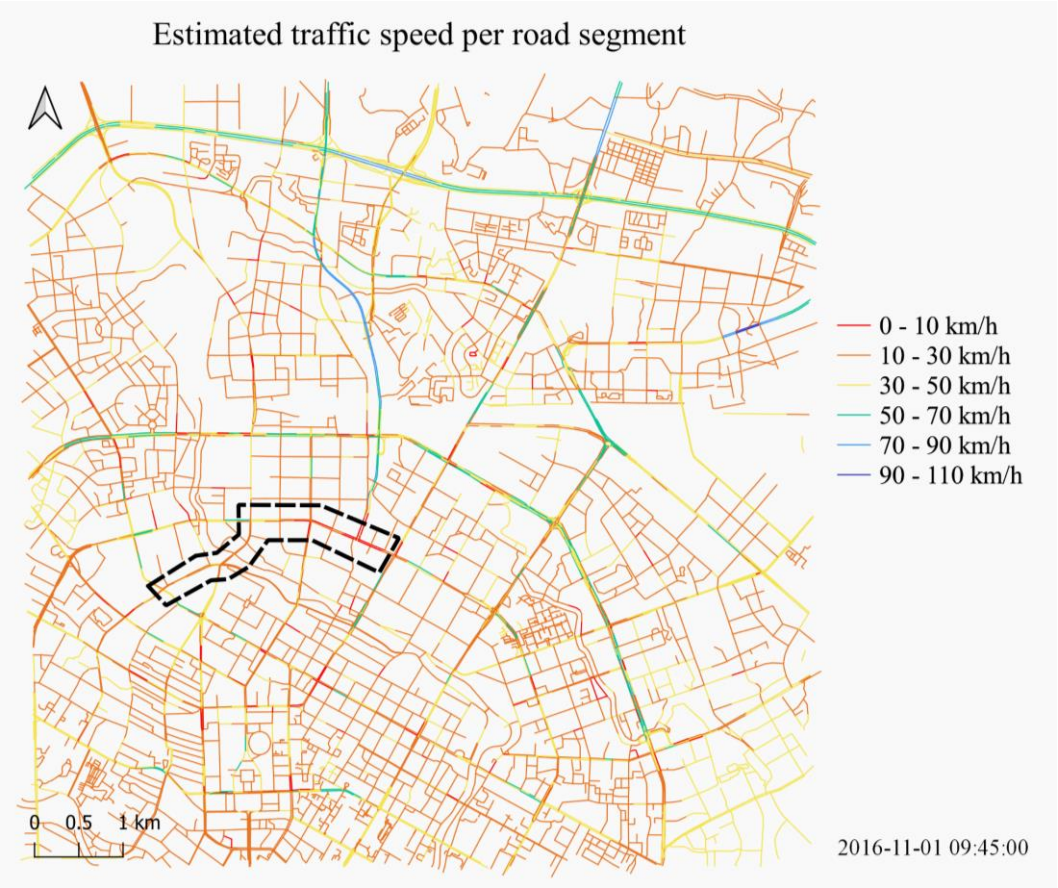


Figure 55: Traffic speed map for the time window when the third most similar candidate trip would have started when served individually. The black dashed line shows the part of the path between both start points. As the distance is large and mostly orange and red coloured segments are visible, the emerging waiting time for the second user exceeds the set threshold of 15 minutes.

The identified optimal shared path begins at S1, goes through the roundabout and then north to pick up the second passenger. After following the same way back to the roundabout and heading on to E2, the end point of the most similar candidate trip, the passenger that joined second gets dropped off. The ride is finished at E1. The emerging waiting time for the second passenger amounts to 4 min 32 s. The sum of the travel times of the two individual trips is 17 min 42 s and the total travel time of the shared path is 15 min 15 s. With 2 min 27 s, a total travel time saving of 13.9% is achieved. Considering the distance, a total value of 7.69 km for the shared path emerges. As the sum of the travel distances of the two individual trips amounts to 7.83 km, 140 m of distance is saved. Including the additional constraint leads therefore, notwithstanding the reduction of the total travel time savings, to a decrease in the driving distance, though a very small one. In comparison to the first variation, the second one saves 27.61 g CO_2 for the analysed example. Table 15 provides an overview on all the presented measures of the two variations. Variation one stands for including the traffic state information and ignoring the distance savings constraint, and in variation two, the traffic state is still considered, but, additionally, the distance savings constraint must be met. The differences in the results of the two discussed variations show how important it is to include the distance savings constraint for it has a substantially positive impact on our environment. Otherwise, the problem of too high emissions only moves from having too many vehicles on the network to enlarging each driving distance.

	Variation 1	Variation 2	Difference
Travel time savings [min]	3 min 24 s	2 min 27 s	↓ 57 s
Travel time savings [%]	16.6%	13.9%	↓ 2.7%
Travel distance savings [km]	- 1.13 km	0.14 km	↑ 1.27 km
Travel distance savings [%]	- 13%	1.7%	↑ 14.7%
CO_2 emission savings [g CO_2]	- 222.87 g CO_2	27.61 g CO_2	↑ 250.48 g CO_2
CO_2 emission savings [%]	- 13%	1.7%	↑ 14.7%
Waiting time [min]	4 min 4 s	4 min 32 s	↑ 28 s

Table 15: Summary of the measures of both variations including the estimated traffic state. Displayed are the results of the identification process of ride-sharing paths for the analysed example trip. In variation one the travel distance slightly increases by the proposed ride-sharing method. A - stands for a negative influence on the travel distance and CO_2 emissions, meaning an increase in these measures.

The measures presented in Table 15 are calculated based on only one identified ride-sharing path. To assess how effective a developed ride-sharing system is, the overall results must be analysed as well. This means that for the waiting time, an average value of all the resulting waiting times must be calculated. The travel time and distance savings are also computed for all the processed taxi trips (shared and unshared). Just like that, an incomparably more significant statement can be made about the savings of CO_2 emissions. In addition to this, and with the overall results in mind, a value for the taxi fleet reduction can be calculated as well. This shows to what extent the total number of taxis could be reduced thanks to ride-sharing and, thus, how such a system could contribute to an overall reduction of the number of vehicles on the road network. To calculate this reduction, the number of unique taxi IDs of all the involved taxi trips must be counted first; this represents how many taxis are in service without ride-sharing involved. When an analysed trip is shared with a candidate trip, it is assumed that the taxi of the analysed trip will serve the shared ride. Therefore, for all shared paths, the number of unique taxi IDs of the analysed trips is elaborated. The taxi trips that are unsuitable for ride-sharing are served individually by its assigned taxi. So, the final number of taxis in service, when ride-sharing is implemented, is the sum of the unique taxi IDs of the analysed trips of the shared paths and the unique taxi IDs of the individually served trips. The difference between the two counted numbers gives the potential taxi fleet reduction. Another measure that only makes sense when considering all the identified ride-sharing paths is the matching rate of the systems. This value shows how many taxi trips can be shared by the proposed method. The higher this number, the more attractive a system gets for the users; and with more users involved, normally a higher reduction in the CO_2 emissions and the number of vehicles on the road network can one expect to achieve. The matching rate of the proposed system in this work is calculated by counting how many of the 15’347 available map-matched taxi trips are involved in the identified ride-sharing paths. The rest of the taxi trips are served individually.

The mentioned measures are again computed for the case where the distance savings constraint is not considered and for the case where this constraint must be met. For both variations, the estimated traffic state of the underlying road network is included. The resulting values are given in Table 16. When traffic state information is used during the identification process, but the distance savings constraint must not be met, 7’412 ride-sharing paths result. This means that only 523 of the 15’347 available taxi trips are not involved in a ride-sharing path. Thus, the matching rate for this variation amounts to 96.59%. On average, the passenger that joins the

ride second must wait 3 min 24 s until the taxi arrives at its pickup location. If all 15'347 trips of the analysed day would be served individually, a total travel time of 2'930 h 36 min and a total driving distance of 67'554.6 km would emerge. Applying the proposed ride-sharing system leads to a total travel time of 1'904 h 16 min and, as the distance savings constraint must not be met, potentially leading to an increase in the distance, to a total driving distance of 75'374.7 km. The travel time savings are, therefore, 1'026 h 20 min or 35.02%, while no travel distance is saved as 7'820.1 km more are driven, which equals an increase of 11.57%. As already seen by the analysed example trip, the proposed ride-sharing system in variation one does not reduce the total CO_2 emissions, but in fact lets them increase by 1'542.3 kg CO_2. This equals a rise of 11.57%. The 15'347 available trips would be served by 10'760 different taxis if no ride-sharing is applied. This number decreases to 6'535 taxis if the proposed system under variation one is implemented. So, 4'225 taxis can be saved and removed from the road network. This equals a taxi fleet reduction of 39.27%.

When traffic state information is used and the distance savings constraint is required to be met, the resulting measures differ. From the 15'347 available taxi trips, 8'420 trips are involved in the identified ride-sharing paths. This means that 4'210 shared paths are found, and 6'927 trips are served individually. The resulting matching rate, therefore, drops to 54.86%. The average waiting time for the second passenger to be picked up decreases slightly and amounts to 3 min 14 s. The total travel time and the total driving distance that emerges when all the trips would be served individually are still 2'930 h 36 min and 67'554.6 km. By applying the proposed system considering the distance savings constraint, a total travel time of 2'147 h 38 min and a total driving distance of 61'576.4 km results. So, 782 h 58 min or 26.72% travel time and, different from the previous variation, 5'978.2 km or 8.85% driving distance are saved. This leads to savings of 1'179.1 kg in CO_2 emissions, which equals a reduction by 8.85%. The size of the taxi fleet, granted the trips are served individually, remains the same with 10'760 taxis. By applying the ride-sharing system including the distance savings constraint, the number drops to 8'484 different taxis. So, here only 2'276 taxis can be removed from the road network, which equals a taxi fleet reduction of 21.15%.

	Variation 1	**Variation 2**	**Difference**
Matching rate [%]	96.59%	54.86%	↓ 41.73%
Waiting time [min]	3 min 24 s	3 min 14 s	↓ 10 s
Travel time savings [h, min]	1'026 h 20 min	782 h 58 min	↓ 243 h 22 min
Travel time savings [%]	35.02%	26.72%	↓ 8.3%
Travel distance savings [km]	- 7'820.1 km	5'978.2 km	↑ 13'798.3 km
Travel distance savings [%]	- 11.57%	8.85%	↑ 20.42%
CO_2 emission savings [g CO_2]	- 1'542.3 kg CO_2	1'179.1 kg CO_2	↑ 2'721.4 kg CO_2
CO_2 emission savings [%]	- 11.57%	8.85%	↑ 20.42%
Taxi fleet reduction [taxis]	4'225 taxis	2'276 taxis	↓ 1'949 taxis
Taxi fleet reduction [%]	39.27%	21.15%	↓ 18.12%

Table 16: Resulting measures for the developed ride-sharing system based on real-world GPS taxi data of Chengdu, China. In the first variation the estimated traffic state is included but the distance savings constraint is not considered. In the second variation both the traffic state and the distance savings constraint are included. The main differences are visible in the matching rate, the travel distance savings, and the CO_2 emission reduction. A - stands for a negative influence on the travel distance and CO_2 emissions, meaning an increase in these measures.

6.4.2 Assuming absence of traffic congestions

The results provided in this section are obtained by implementing the proposed ride-sharing system while assuming an absence of traffic congestions on the underlying road network. This means that the vehicle speed is a static value that only depends on the type of the road segment and its maximum allowed speed. Therefore, travel time in this vacuum does not depend on the time of the day. To show how the distance savings constraint and the traffic state information influence the identified ride-sharing paths, this section again compares the results between not considering the distance savings constraint and including this additional constraint. Furthermore, the findings are compared with the previous section. The comparison is based on the same analysed example trip as before, but the three most similar candidate trips have slightly changed. Figure 56 provides again the overview of the analysed and its three most similar candidate trips in a) to c). The most similar candidate trip in a) is the same as while including the traffic state information and still contains an SMI of 499.34. The second most similar candidate trip in the previous section does no longer count as a candidate for the analysed trip, as it is already used in a ride-sharing path with a trip that started earlier. This difference is due to the change in the travel time values. While considering the estimated traffic state, this old candidate trip is not suitable to be shared with the taxi trip that started earlier than the analysed one as either the travel time is not reduced or the waiting time is too long because of the bad traffic state. Without traffic state information this shared path can be driven in less time and the waiting time gets reduced as well. Thus, this combination is considered as the most suitable one for the earlier trip and is not available anymore for the analysed example trip. The second most similar candidate trip for the analysed example trip is, therefore, the previously as third most similar candidate selected trip. So, in Figure 56 b), the same candidate trip as in Figure 53 c) and 54 c) is illustrated. The SMI of this trip is still 869.90. The third most similar candidate trip in Figure 56 c) given the assumption of an absence of traffic congestions is a new candidate. In the first and the second variation, this candidate trip is ranked as the fourth most similar candidate trip. As in the third variation the ranking changes, the mentioned candidate is no longer the fourth but third most similar candidate trip with an SMI of 1’013.26.

Each candidate trip would lead to a shared path that fulfils the two set constraints in this variation. The most similar candidate trip is already considered to be suitable when the estimated traffic state is included and hence, here again. The candidate given in Figure 56 b) is not considered to be suitable for ride-sharing when traffic state information is used. It is argued that there is a too long waiting time sourcing from the big distance between both start points and the bad traffic state on the driven road segments, as displayed in Figure 55. Figure 57 shows the same dashed line around the subject road segments but with the maximum allowed speed value of each road type. During most of the analysed part of the path speed values of 40 km/h and 60 km/h are allowed. So, in the third variation the vehicles on average can drive much faster due to the absence of traffic congestions and therefore the emerging waiting time for the shared path of Figure 56 b) is clearly smaller than the threshold of 15 minutes. Thus, while not using traffic state information, this trip is considered to be suitable as well. The third most similar candidate trip is obviously less suitable to be shared with the analysed trip than e.g. the most similar candidate trip, but it still fulfils the set constraints because of the allowed high speed values and is therefore considered to be suitable for ride-sharing too. If more than one candidate trip fulfils the constraints the trip with the shortest waiting time is identified as the optimal ride-sharing path. Following this rationale, the optimal shared path for the analysed trip given the third variation is built with the most similar candidate trip and visualised in Figure 56 d).

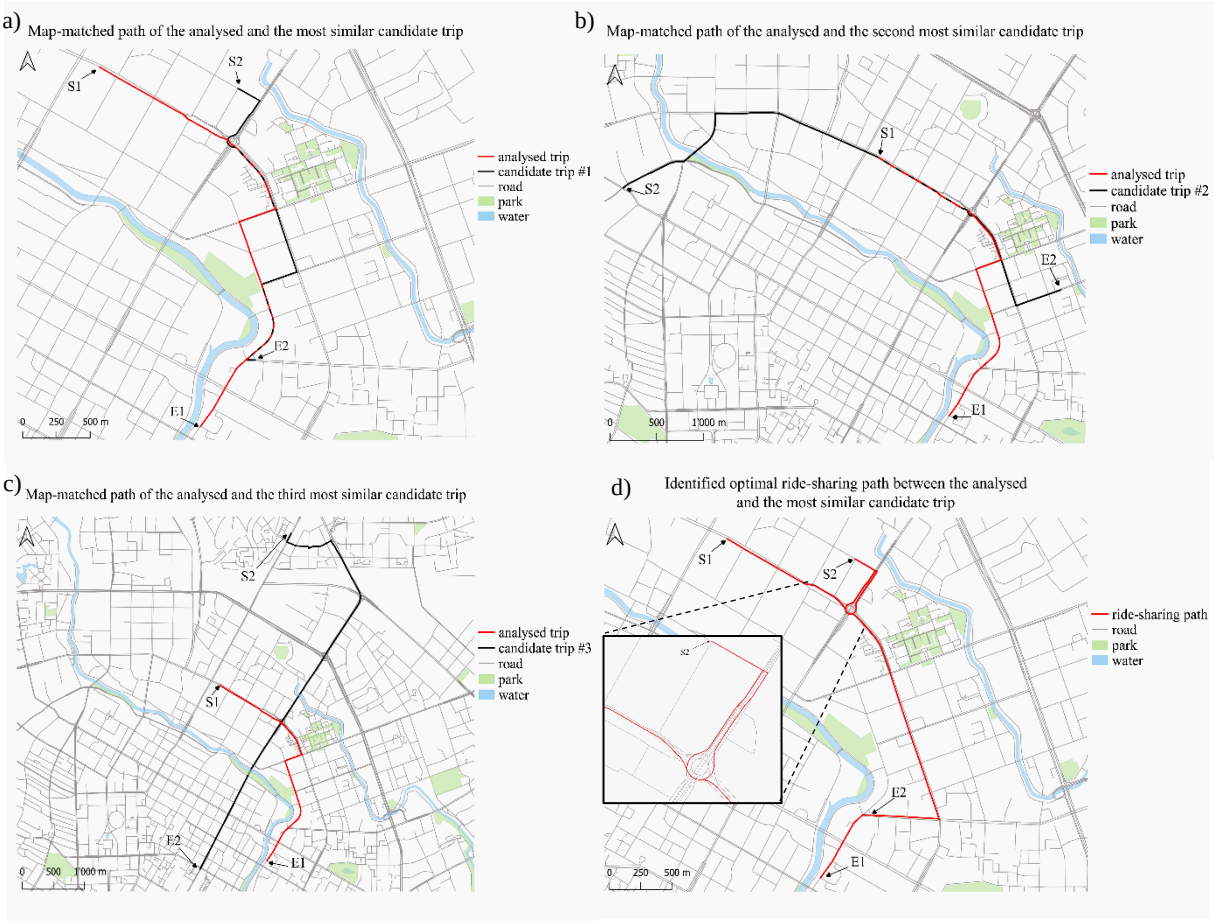


Figure 56: Visualisation of the three most similar candidate trips in a) to c) and the optimal ride-sharing path in d) for the analysed example trip. The final ride-sharing path of the third variation is a combination of the analysed with the most similar candidate trip. A bigger size of it is in the appendix.

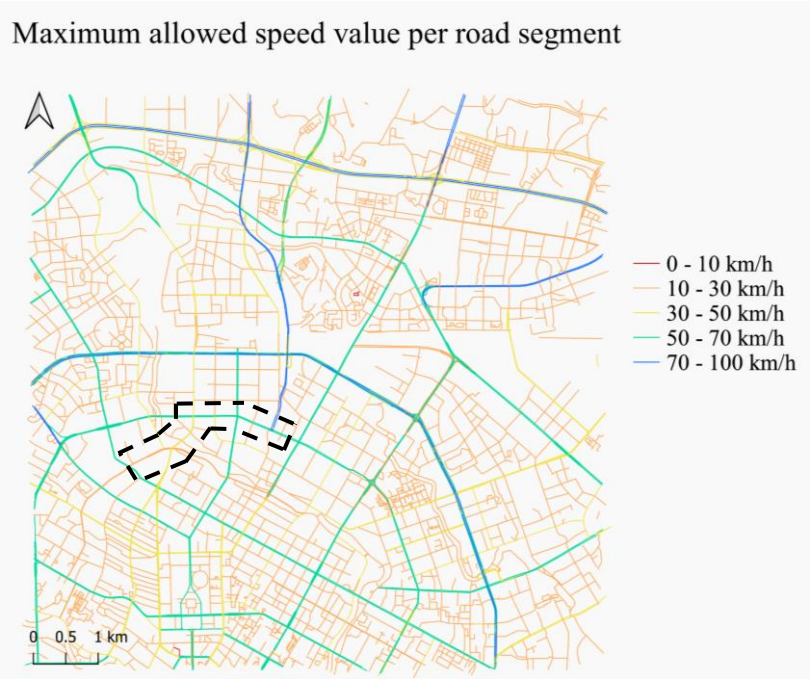


Figure 57: Visualisation of the maximum allowed speed values per road type. The black dashed line shows the same part of the path between both start points like in Figure 55. As the sub paths' average speed value is quite high, the emerging waiting time is smaller than the threshold of 15 minutes.

The identified shared path is the same as in variation two, given in Figure 54 d) and goes through the same road segments in the same order. Due to the absence of traffic congestions, the emerging waiting time for the second passengers is now only 2 min 16 s. The sum of the travel times of the two individual trips is still 17 min 42s but the total travel time of the shared path amounts now to 7 min 46 s. The travel time of the shared path is now clearly shorter because of the higher speed values. Like this, 9 min 56 s of travel time is saved what equals 56.13%. As the path is exactly the same, the sum of the travel distances of the two individual trips still amounts to 7.83 km and the distance of the shared path 7.69 km. The travel distance savings of 140 m remain the same for this variation and so do the CO_2 emission savings of 27.61 g CO_2. The assumption of an absence of traffic congestions thus only affects the travel time savings and the emerging waiting time for this example trip but not the route and the distance of the shared path compared to the second variation.

In the fourth variation again an absence of traffic congestions is assumed but different from the third variation, the distance savings constraint must be met. This means that as in the second variation, the sum of the driving distances of the two individual trips must be bigger than the driving distance of the shared path. Analysing the example trip, the three most similar candidate trips are again slightly different than before. All the trips are shown in Figure 58 a) to c) and in d) the identified ride-sharing path is visible. Completely different from all the other three variations, the most similar candidate trip is a new and even more similar trip than the most similar one has been so far. This trip given in a) is combined with an earlier started trip for the first three variations. Due to the higher allowed speed values and the set distance savings constraint, it represents in this variation the most similar candidate trip for the analysed example trip with an SMI of 445.77. The previously most similar candidate trip is now ranked as the second most similar candidate trip with the same SMI of 499.34 and is displayed in b). The third most similar candidate trip given in c) is the same as the second most similar candidate in the first two variations with an SMI of 841.43. The in the third variation as the second most similar candidate ranked trip is combined with an earlier started taxi trip for this case and is thus not represented anymore in the three most similar candidate trips.

As for the combination in a) the first part of the path until the roundabout is equal for both trips and the end points are located close to each other, their shared path fulfils all the set constraints including the distance savings constraint. The second most similar candidate trip already fulfils the set constraints in the second variation where the additional constraint is included as well. As in the fourth variation the average speed value of the road segments is much higher than in the second, the shared path will be even faster and the waiting time shorter and thus this trip is again considered to be suitable for sharing the ride with the analysed example trip. The candidate trip in c) is very similar to the candidate in a). The only difference is that the third most similar candidate trip will lead to a bigger waiting time as the start points are located a little bit further away from each other. Nevertheless, this candidate fulfils all the set constraints as well. So again the situation is given where all three candidate trips are suitable for sharing the ride with the example trip. As the most similar candidate trip starts at the same place as the analysed trip, no waiting time emerges and therefore this trip is identified as the final ride-sharing candidate. The optimal shared path of this combination is displayed in d). It starts for both passengers in the north, goes east until the roundabout and then south to E2. After dropping off the first passenger, the trip is finished at E1, the end point of the analysed trip.

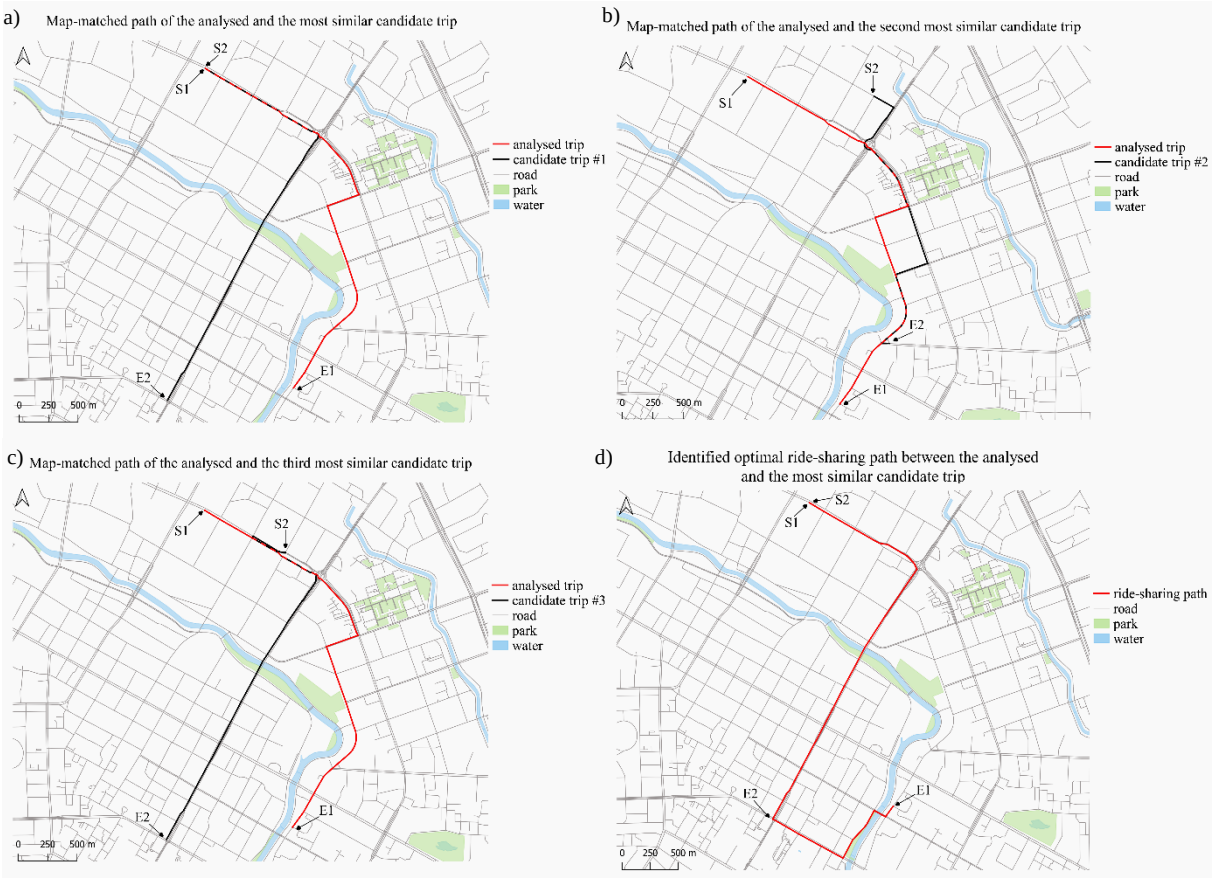


Figure 58: Visualisation of the three most similar candidate trips in a) to c) and the optimal ride-sharing path in d) for the analysed trip. The final ride-sharing path of the fourth variation is a combination of the analysed with the most similar candidate trip. A bigger size of it is in the appendix.

The sum of the travel times of the two individual trips amounts to 19 min 10 s. The travel time of the shared path is only 6 min 41 s. Thus, 12 min 29 s of travel time is saved what equals a decrease of 64.88%. The decrease can be explained as most of the shared path goes through road segments of type primary street, where the speed limit amounts to 60 km/h. So, compared to the situation where the traffic state is considered, the average speed is almost three times higher and thus a much shorter travel time occurs. Summing up the length of both paths, a total driving distance of 8.78 km emerges. The shared path on the other hand, is 5.98 km long. The 2.8 km saved travel distance represent a reduction of 31.89%. The identified optimal ride-sharing path saves 552.24 g CO_2 and reduces these emissions by 31.89%. All the presented measures of both variations are as in the previous section summarized in Table 17.

By comparing the resulting measures between the variations where for both cases the distance savings constraint is included but once the traffic state is excluded, especially the smaller travel time savings stand out for the situation of considering the estimated traffic state. Comparing the situation where for both cases the traffic state is not included but only once the distance savings constraint must be met, a big difference in the travel distance savings is given. As soon as this constraint must be met, the savings in the distance strongly increase. So, besides the effect on the selection of the three most similar candidate trips and moreover, on the identification of the optimal combination for the final ride-sharing path, after analysing only one example trip, for both the traffic state information and the distance savings constraints, an influence on the resulting measures can be detected.

	Variation 3	Variation 4	Difference
Travel time savings [min]	9 min 56 s	12 min 29 s	↑ 2min 33 s
Travel time savings [%]	56.13%	64.88%	↑ 8.75%
Travel distance savings [km]	0.14 km	2.8 km	↑ 2.66 km
Travel distance savings [%]	1.7%	31.89%	↑ 30.19%
CO_2 emission savings [g CO_2]	27.61 g CO_2	552.24 g CO_2	↑ 524.63 g CO_2
CO_2 emission savings [%]	1.7%	31.89%	↑ 30.19%
Waiting time [min]	2 min 16 s	0 s	↓ 2 min 16 s

Table 17: Summary and comparison of the measures of the third and fourth variation. Displayed are the resulting values of the identification process of the optimal ride-sharing path for the analysed example trip. The values of the fourth variation for the example trip are better in each category.

To analyse the influence traffic state information and the additional constraint can have on the whole system, the overall results must be checked as well for the situation of assuming an absence of traffic congestions. Thus, the in the previous section mentioned measures are again computed for this situation. Once the distance savings constraint is considered and then ignored. The resulting measures for the whole system are listed in Table 18. If instead of the estimated traffic state information the travel times resulting from the maximum allowed speed values are used during the identification process and the system does not have to save travel distance, 7'626 ride-sharing paths result. From the 15'347 available taxi trips, 15'252 trips are involved in ride-sharing and only 95 trips are not suitable to be used in the proposed system. This equals a matching rate of 99.38%. The average waiting time that emerges for the passenger that is picked up second amounts to 2 min 8 s.

As the data used to calculate the total travel time if all trips would have been served individually is the same as for the case where the traffic state is considered, the total travel time remains 2'930 h 36 min for the 15'347 taxi trips. The same counts for the total driving distance of 67'554.6 km. A difference occurs by applying the ride-sharing system without the traffic state involved, as this leads to a total travel time of 1'185 h 24 min and if the distance constraint is ignored, to a total driving distance of 68'925.85 km. In other words, the travel time savings are 1'745 h 12 min, which equals 59.56% and the travel distance is lengthened by 1'371.25 km or 2.03%. These values are better for the case of the travel time and more optimal for the travel distance compared to considering the estimated traffic state in the first variation. The same counts for the CO_2 emissions. With 270.4 kg CO_2 or 2.03%, even more emissions are generated than without ride-sharing, though the magnitude shrinks. Focusing on the taxi fleet reduction, the in the third variation proposed system lets the number of taxis decrease from 10'760 to 6'386. So, 4'374 taxis can be saved and removed from the road network. This represents a taxi fleet reduction of 40.65%.

If an absence of traffic congestions is assumed but additionally the distance savings constraint must be met, the resulting measures differ again, as it was the case for the first two variations. 10'830 of 15'347 taxi trips form part of a ride-sharing path. So, 5'415 different shared paths are identified, and 4'517 trips are served individually. This results in a decrease in the matching rate to 70.57%. The average waiting time that occurs for the second passenger gets slightly shorter with 2 min 6 s. Again, no changes are made to the values of the total travel time and travel distance given each trip is served individually. Thus, a travel time of 2'930 h 36 min and

a travel distance of 67'554.6 km are given. Different from the third variation, while considering the additional constraint, a total travel time after applying the ride-sharing approach of 1'467 h 52 min results. The new total driving distance is 58'908.5 km. This means, that 1'462 h 44 min or 49.91% of travel time and 8'646.1 km, which equals 12.8%, of travel distance, are saved. This leads to a reduction of 1'705.3 kg CO_2. So, in other words 12.8% of the CO_2 emissions can be avoided by the proposed system and the given circumstances. The number of taxis that are needed to serve all the requests if no ride-sharing is involved remains 10'760. Due to ride-sharing, in the fourth variation, 2'969 taxis can be removed from the road network, which leads to a taxi fleet size of 7'791 vehicles. So, compared to variation three, the taxi fleet reduction amounts only to 27.59%.

	Variation 3	**Variation 4**	**Difference**
Matching rate [%]	99.38%	70.57%	↓ 28.81%
Waiting time [min]	2 min 8 s	2 min 6 s	↓ 2 s
Travel time savings [h, min]	1'745 h 12 min	1'462 h 44 min	↓ 282 h 28 min
Travel time savings [%]	59.56%	49.91%	↓ 9.65%
Travel distance savings [km]	- 1'371.25 km	8'646.1 km	↑ 10'017.35 km
Travel distance savings [%]	- 2.03%	12.8%	↑ 14.83%
CO_2 emission savings [g CO_2]	- 270.4 kg CO_2	1'705.3 kg CO_2	↑ 1'975.7 kg CO_2
CO_2 emission savings [%]	- 2.03%	12.8%	↑ 14.83%
Taxi fleet reduction [taxis]	4’374 taxis	2'969 taxis	↓ 1'405 taxis
Taxi fleet reduction [%]	40.65%	27.59%	↓ 13.06%

Table 18: Resulting measures for the developed ride-sharing system based on real-world GPS taxi trajectory data of Chengdu, China. In the third variation an absence of traffic congestions is assumed, and the distance savings constraint is not considered. In the fourth variation the traffic state is still not included but the distance savings constraint considered. The main differences are visible in the matching rate, the travel distance savings, and the CO_2 emission reduction. A - stands for a negative influence on the travel distance and CO_2 emissions, meaning an increase in these measures.

In the last column of Table 16 and Table 18, the differences in the resulting measures between the different variations are listed. These differences occur due to the distance savings constraint and thus by looking at these absolute numbers and percentages, the influence this constraint has on the results can be assessed. If the traffic state information is included in the identification process, the additional constraint leads to a decrease in the matching rate of 41.73%. Assuming an absence of traffic congestions, this decrease amounts only to 28.81%. So, the distance savings constraint has a stronger influence on the results when the traffic state is considered. This can be explained as with the additional constraint more direct paths must be created and these can lead through areas of bad traffic state. Hence, the travel time increases and can exceed the threshold of the sum of the travel times of the two individual trips. Consequently, more shared paths are filtered out and thus the matching rate decreases stronger. The average waiting does for both situations, with traffic state or without, slightly decrease due to the additional constraint, but there is no specific influence detectable on this measure, as still the objective of the system is to minimize this waiting time and, thus, it remains short for both cases. Considering the distance savings constraints worsens the travel time savings as well. For including the traffic state, the savings decrease by 8.3% and for assuming an absence of traffic, they decrease by 9.65%. As these two values are quite similar, the traffic state information does not affect the influence of the additional constraint on the travel time savings. The decrease

occurs because most of the shared paths that save a lot of time are not allowed anymore as they normally tend to increase the travel distance (driving through ring roads of high speed values which are not located very close to the original path). A significant improvement by the additional constraint is visible in the travel distance savings. For using traffic state information, the savings increase by 20.42% and for not using this information, 14.83% more distance is saved as when this constraint is not an obligation. These values are rather high as the distance savings values change from negative savings, meaning an increase in the distance, to positive savings of 8.85% respectively 12.8%. The bigger change is visible in the situation, where the traffic state information is used, but this is only due to the stronger increase in the travel distance caused in the variation where the distance savings constraint is not considered. Focusing only on the savings in variation two respectively four, the value is higher while assuming an absence of traffic congestions. Thus, traffic state information negatively influences the effect the additional constraint has on the distance savings, as again more direct ways are possible if no bad traffic state is given. In other words, the road network is not congested.

In addition, these numbers show how important the distance savings constraint for the positive impact of ride-sharing systems on the natural environment is. This impact is first discussed by the analysed example trip in Chapter 6.4.1 but expounded upon with these presented values. Not forcing the ride-sharing system to identify only shared paths that save travel distance results in more driven kilometres and hence more CO_2 emissions. With this additional constraint, the conserved travel distance of 8.85% and 12.8% leads to an emission reduction of 1’179.1 kg CO_2 and 1’705.3 kg CO_2 for just the 15’347 available taxi trips. This reduction could be even bigger if extended to the whole taxi fleet of Didi in Chengdu. In general, the distance savings constraint has a negative influence on all the measures except the travel distance savings, the CO_2 emissions, and the average waiting time. However, the additional constraint should be included in the algorithm of a ride-sharing system to achieve useful results, as the three positively influenced measures are amongst the most important ones of such a system.

To analyse the influence traffic state information can have on the different measures in more detail, what is used in the next section to discuss the related research question, Tables 19 and 20 compare the provided values between variation one and three respectively variation two and four. The last column delivers once again the differences between the retrieved values. The difference in the matching rate when the distance constraint must not be met is rather small. Both rates are very high, and the traffic state does not have a special influence on that. If the additional constraint is included, the traffic state information leads to a worse matching rate than simply assuming an absence of traffic congestions. The reason for this, as already explained in the previous paragraph, are the areas of bad traffic where the more direct shared paths pass through. The resulting too big travel times consequently reduce the matching rate. The strongest impact traffic state information has, is given in the average waiting time and the travel time savings. The former decreases in both situations by more than a third when the traffic state information is not included in the identification process. The main reason for this might be the higher average speed of the vehicles when no traffic congestions are given and the resulting shorter travel time from one start point to the other. The same counts for the increase in the travel time savings when no traffic state is considered. As shown in Chapter 5.4, the travel speeds are for the biggest part of the study area below the maximum allowed speed value and thus the shared paths take more time to be completed when traffic state information is included than without. Consequently, considering the traffic state in the system negatively influences the

travel time savings. As it is shown in Tables 19 and 20, the distance savings, and linearly correlated to this the reduction of CO_2 emissions, slightly increase (Table 20) and are more optimal without traffic state information (Table 19). Additionally, it is apparent that excluding traffic state information has a stronger effect on these two measures when the distance savings constraint must not be met. Focusing on the differences in the taxi fleet reduction, it can be surmised that including traffic state information negatively influences this decrease, even though only slightly. This means that for both situations, considering the traffic state of the underlying road network removes fewer taxis from this network as when an absence of traffic congestions is assumed. Overall, the results show that including the estimated traffic state the way it is proposed in this work negatively affects all the presented and discussed measures, independent of adding a third constraint or not. To summarise the discussed results of this section, Table 21 delivers an overview of all the computed measures for all four variations.

	Variation 1	**Variation 3**	**Difference**
Matching rate [%]	96.59%	99.38%	↑ 2.79%
Waiting time [min]	3 min 24 s	2 min 8 s	↓ 1 min 16 s
Travel time savings [h, min]	1’026 h 20 min	1'745 h 12 min	↑ 718 h 52 min
Travel time savings [%]	35.02%	59.56%	↑ 24.54%
Travel distance savings [km]	- 7’820.1 km	- 1'371.25 km	↑ 6448.85 km
Travel distance savings [%]	- 11.57%	- 2.03%	↑ 9.54%
CO_2 emission savings [g CO_2]	- 1’542.3 kg CO_2	- 270.4 kg CO_2	↑ 1'271.9 kg CO_2
CO_2 emission savings [%]	- 11.57%	- 2.03%	↑ 9.54%
Taxi fleet reduction [taxis]	4’225 taxis	4’374 taxis	↑ 149 taxis
Taxi fleet reduction [%]	39.27%	40.65%	↑ 1.38%

Table 19: Comparison of the resulting measures for the developed ride-sharing system between the variation where the distance savings constraint is not considered, once with the traffic state information included (variation 1) and once excluded (variation 3). In the last column, the difference that shows the influence traffic state information can have on the results is given. A - stands for a negative influence on the travel distance and CO_2 emissions, meaning an increase in these measures.

	Variation 2	**Variation 4**	**Difference**
Matching rate [%]	54.86%	70.57%	↑ 15.71%
Waiting time [min]	3 min 14 s	2 min 6 s	↓ 1min 8 s
Travel time savings [h, min]	782 h 58 min	1'462 h 44 min	↑ 679 h 46 min
Travel time savings [%]	26.72%	49.91%	↑ 23.19%
Travel distance savings [km]	5’978.2 km	8'646.1 km	↑ 2’667.9 km
Travel distance savings [%]	8.85%	12.8%	↑ 3.95%
CO_2 emission savings [g CO_2]	1’179.1 kg CO_2	1'705.3 kg CO_2	↑ 526.2 kg CO_2
CO_2 emission savings [%]	8.85%	12.8%	↑ 3.95%
Taxi fleet reduction [taxis]	2’276 taxis	2'969 taxis	↑ 693 taxis
Taxi fleet reduction [%]	21.15%	27.59%	↑ 6.44%

Table 20: Comparison of the resulting measures for the developed ride-sharing system between the variation where the distance savings constraint must be met, once with the traffic state information included (variation 2) and once excluded (variation 4). In the last column, the difference that shows the influence traffic state information can have on the results is given.

	Traffic state information included		Assuming an absence of traffic congestions	
	Distance savings constraint ignored	Distance savings constraint considered	Distance savings constraint ignored	Distance savings constraint considered
Ride-sharing paths	7'412	4'210	7'626	5'415
Individually served trips	523	6'927	95	4'517
Matching rate [%]	96.59 %	54.86 %	99.38 %	70.57 %
Waiting time [min]	3 min 24 s	3 min 14 s	2 min 8 s	2 min 6 s
Travel time (single) [h, min]	2'930 h 36 min	2'930 h 36 min	2'930 h 36 min	2'930 h 36 min
Travel time (shared) [h, min]	1'904 h 16 min	2'147 h 38 min	1'185 h 24 min	1'467 h 52 min
Travel time savings [h, min]	1'026 h 20 min	782 h 58 min	1'745 h 12 min	1'462 h 44 min
Travel time savings [%]	35.02 %	26.72 %	59.56 %	49.91 %
Travel distance (single) [km]	67'554.6 km	67'554.6 km	67'554.6 km	67'554.6 km
Travel distance (shared) [km]	75'374.7 km	61'576.4 km	68'925.85 km	58'908.5 km
Travel distance savings [km]	- 7'820.1 km	5'978.2 km	- 1'371.25 km	8'646.1 km
Travel distance savings [%]	- 11.57 %	8.85 %	- 2.03 %	12.8 %
CO_2 emission savings [g CO_2]	- 1'542.3 kg CO_2	1'179.1 kg CO_2	- 270.4 kg CO_2	1'705.3 kg CO_2
CO_2 emission savings [%]	- 11.57 %	8.85 %	- 2.03 %	12.8 %
Taxi fleet reduction [taxis]	4'225 taxis	2'276 taxis	4'374 taxis	2'969 taxis
Taxi fleet reduction [%]	39.27 %	21.15 %	40.65 %	27.59 %

Table 21: Summary of all the resulting measures after applying the developed ride-sharing system to real-world GPS taxi trajectory data. The results are computed for four different variations. In the first two variations traffic state information is included in the identification process and in the last two variations an absence of traffic congestions is assumed. Variation one and two respectively three and four differ as once the distance savings constraint is ignored and once it must be met. A - stands for a negative influence on the travel distance and CO_2 emissions, meaning an increase in these measures.

7. Discussion

By developing and implementing a framework on how to identify potential ride-sharing paths from raw GPS taxi trajectory data, the conducted analysis addressed the influence considering traffic state information has on the results of a ride-sharing system. Additionally, the effect of including a third constraint focusing on the total distance reduction of ride-sharing paths was outlined. By introducing a new similarity measurement for taxi trips and describing how to estimate traffic state information based on raw GPS data, additionally two new contributions were made to the research field of ride-sharing. In this section, the newly developed methods and the resulting measures are put into perspective to the three research questions and the hypotheses of Chapter 3.2 are discussed. Furthermore, the approaches and findings are related to the presented literature in Chapter 2. At the end of the chapter, the limitations of this work are described.

7.1 Estimating traffic state information

The first research question addresses how traffic state information can be estimated given the available data sources and how this information can be included in the process of identifying potential ride-sharing paths. As explained in detail in the method section, only the road network and the raw GPS records are needed for the presented approach. By using the differences in the distance and the time stamps of the GPS signals, speed values for the individual trajectory points can be calculated. By simply taking the average of the speed values of all records that are map-matched to the same segment, the overall estimated speed value for a road segment is derived. This procedure is repeated 96 times as the traffic state is estimated for time windows of 15 minutes during the whole day. Based on these traffic speed values and the information about the length of the individual road segments, derived from the OSM dataset, the travel time of each road segment for the different time windows is calculated. This information is then integrated into the algorithm that identifies the optimal ride-sharing paths by adding it as the weight to the Dijkstra's shortest path algorithm. Besides using the traffic state information for computing the potential ride-sharing paths, this information is as well used while selecting the for ride-sharing suitable paths and identifying the optimal solution by calculating the travel time and the waiting time based on that information.

Compared to existing approaches in the literature, the developed process is very simple and does not depend on additional data sources or complex probability functions as e.g. in Nathawhichti et al. (2003), where they use a complex macroscopic model and additional data from stationary detectors. In the study of Santi et al. (2014a), due to incomplete data they are forced to apply a quite complex estimation method as well. The results of this are expected to represent the reality less accurate than with the proposed approach of this work as only a small part of the real-world data is considered in the estimation. Wang et al. (2018) presumably estimate the average traffic state based on historic data and not only on the same dataset as analysed by the ride-sharing method. This means, that again additional data is needed in their approach.

The strength of the proposed traffic state estimation method is that the values are only based on the input data to the ride-sharing system and still produce reliable results. This is shown in Chapter 6.2, where the traffic speed maps of different time windows are analysed and compared. No unrealistic values or abrupt changes in the road network are detectable. The disadvantages of this method are the strong dependency on the quality of the GPS data and the map-matching approach, as well as the decreasing accuracy when not many location fixes are available. If the GPS data is erroneous, the vehicle speed cannot be calculated accurately enough. Incorrect map-matched trajectory points make this even worse. Nevertheless, the traffic state estimation method presented in this work is very simple to be applied while delivering reliable results and includes the estimated values in the fastest path computation and selection of the optimal shared ride to produce real-world circumstance based findings.

7.2 Identifying ride-sharing paths from raw GPS data

The second research question focuses on how potential ride-sharing paths can be efficiently identified from a large GPS taxi trajectory dataset. The complete framework of all the involved steps that are developed and explained in detail in this work is given in Figure 15 of Chapter 5. The first three main steps can be seen as preliminary processes that must be done to successfully identify ride-sharing paths from raw GPS data. These steps include the pre-processing of both input datasets, the map-matching of the trajectory points and the traffic state estimation. Without these steps, the whole process is dysfunctional as the subsequent approaches need correct information about the exact location of the trips on the road network and the conditions of this network in sense of traffic state or maximum allowed speed values. The final identification process consists again of two sub-steps. These are the newly developed similarity measurement and the matching process itself. By computing the fastest paths between the analysed trips and a set of candidate trips, which are selected based on the similarity measurement, a local optimum solution for each analysed trip is found. These optimal shared paths must reduce the total travel time compared to the sum of the travel times of the two individual trips, lead to a waiting time below the set threshold of 15 minutes and in two of the four different analysed situations to a reduction in the total distance of the shared path compared to the sum of the two individual trips. Moreover, they must follow the objective of the system by minimizing the occurring waiting time for the second passenger of each ride. Thanks to the implemented similarity measurement, only a few fastest paths must be computed for each analysed trip, and therefore the ride-sharing paths can be identified more efficiently.

As discussed in the related work section of Chapter 2, many different ride-sharing systems have been developed and presented in the last several years. Because the basic settings of these systems are very similar, it is interesting to show how the proposed ride-sharing system of this work can be distinguished from already existing ones and how the architecture of the identification process is built to achieve higher efficiency. Pre-processing and map-matching the GPS data form part of almost every study if the system is applied to real-world data. Here, different approaches are chosen to map-match the trajectories but mostly no detailed explanation about the method is given, as it is simply seen as a basic step of each system. The main differences occur in the matching process. These have been identified as on what type of algorithm the shared paths are built, what conditions must be met, what the objective of the whole system is and if the results represent a local or a global maximum. The most similar architectures of ride-sharing systems compared to the presented one are provided by the works of Cai et al. (2019), Wang et al. (2018) and Santi et al. (2014a).

All of them develop a static taxi-sharing system based on real-world GPS data, as it is the case in this study. Cai et al. (2019) and Santi et al. (2014a) both are identifying a global optimum solution with their matching process. Different from them and similar to Wang et al. (2018), the proposed system focuses on a local optimum. The constraints included in these three studies differ from saving driving distance, not creating a too big delay, or waiting time, to reducing the taxi fare. Given the three constraints applied in this work, in this aspect, the system does not strongly differ from the others. A novelty compared to the three similar systems is the followed objective of this model. Minimizing the waiting time has so far not often been addressed as the main goal of a system. They rather follow objectives like minimizing the total driven distance, minimizing the total travel time, or maximizing the matching rate. The reason for following this objective and not a more common one is already explained in the method chapter. Like in Santi et al. (2014a) and Wang et al. (2018) this system works with the fastest path algorithm to build the potential ride-sharing paths. Cai et al. (2019) on the other hand, use the shortest path algorithm as no information on travel time is included.

The main difference from the discussed papers and in general from the existing literature is the number of considered candidate trips for building a shared path with an analysed trip. In the three mentioned studies, the shortest or the fastest path is computed between an analysed trip and a big number of candidate trips, which is only limited by a defined time window. In some studies, even shared paths between all available trips are built and then tested on the constraints. Like that, a lot of fastest paths are computed between candidate and analysed trips which are not suitable to be shared at all. They are then excluded in a subsequent step as they do not fulfil the set constraints and are therefore needlessly computed.

Due to the developed similarity measurement, this work suggests a different approach as only the three most similar candidate trips are analysed in detail and only between them shared paths are built. Assuming that the fastest path computation is equally time-consuming for both systems, the proposed ride-sharing system could therefore identify the potential ride-sharing paths more efficiently. This is intensified because only a local optimum and not a global optimum is searched, as the complexity of a system strongly increases with global optimization problems. Thus, by the developed framework and especially by the implemented similarity measurement, an alternative simple and efficient ride-sharing system is proposed in this work.

7.3 The influence of using traffic state information in ride-sharing systems

With the third and last research question, the effect of considering traffic state information in a ride-sharing system on the resulting measures is analysed. To be able to analyse this influence, the in the experimental design presented four variations of the proposed ride-sharing system are applied to real-world GPS taxi trajectory data of the city centre of Chengdu, China. By comparing the differences in the resulting measures of Chapter 6.4, this research question is addressed. Subsequently, the three established hypotheses regarding the influence of traffic state information are discussed.

The effect the additional constraint about the savings of driving distance has on the resulting measures is already presented in the last part of the result chapter. The conclusion is that regardless of including traffic state information or not; the matching rate, the travel time savings and the taxi fleet reduction decrease while the waiting time gets shorter and the total driving distance savings as well as the reduction of CO_2 emissions increase. This additional constraint

does not affect the resulting measures 100% positively. Nevertheless, as a ride-sharing system should have a positive impact on the natural environment, what is not given without this constraint, the positive impact on the waiting time, the distance savings and the emission reduction outmatch the negative effects. Hence, it is considered necessary to include this additional constraint in ride-sharing systems. Therefore, to analyse the influence traffic state information can have, only the differences in the measures of the second and fourth variation are discussed. The significance of this is that the results between the situation where the traffic state information is included and the distance savings constraint must be met and the situation where an absence of traffic congestions is assumed but the distance constraint still must be fulfilled, are compared.

Considering real-world circumstances by including the estimated traffic state in the proposed ride-sharing system lets the matching rate decrease by 15.71% from 70.57% to 54.86%. Thus, only slightly more than half of the available taxi trips form part of a shared ride. This is explained by the fact that in general the traffic state is congested for the study area and thus the travel times of the shared trips and additionally the travel distances increase. This then leads in many cases to not fulfilling the set constraints anymore and therefore less shared rides can be identified. This effect is further confirmed by the change in the total travel time savings and the reduction of the total driving distance. Including the traffic state lets the travel time savings decrease by 46.47%. When no traffic congestions are given, 1'462 h 44 min or 49.91% travel time can be saved by applying ride-sharing. These numbers shrink to 782 h 58 min or 26.72% due to the mentioned reasons after considering the traffic state information. For the travel distance, 8'646.1 km or 12.8% are saved while an absence of traffic congestions is assumed and only 5'978.2 km respectively 8.85% of travel distance can be reduced with real-world circumstances considered. This equals a decrease of 2'667.9 km or 30.86% in the distance savings. As a linear correlation between the driven distance and the CO_2 emissions is assumed, as well 30.86% less CO_2 can be reduced due to the traffic state. Instead of 1'705.3 kg, only 1'179.1 kg CO_2 is saved by the applied ride-sharing system under the given circumstances. The second positive effect of ride-sharing systems, besides reducing emissions, is a potential reduction in the taxi fleet size and thus a decrease in the number of vehicles on the road network. While assuming an absence of traffic congestions, 2'969 taxis are removed from the network, which equals a taxi fleet reduction of 27.59%. These values are also influenced by the traffic state and diminish to 2'276 removed taxis respectively a taxi fleet reduction of 21.15%. The traffic state information lets this positive effect decrease by 23.34%. The last measure that gets influenced by the real-world circumstances is the waiting time that emerges for the second passenger. While this measure amounts to 2 min 6 s when no traffic congestions are given, it increases by 53.81% to 3 min 14 s as soon as traffic state information is included.

Focusing on the research question, generally, the findings are that considering traffic state information in ride-sharing systems has a negative influence on all the presented measures. This results from the smaller average vehicle speed triggered by congestions, traffic lights or accidents. As the speed is directly connected to the travel time, this measure and additionally the waiting get influenced the most by traffic state information. Other measures like the distance savings or taxi fleet reduction are negatively influenced as well, but less severely. This negative effect is coupled with the influence on the travel time as the given circumstances a shared ride tries to lead through streets where a higher speed value is possible and this lets the distance increase and, hence, the savings decrease.

Before conducting the presented analysis of this work, three hypotheses connected to the third research question were constructed. In the following, these three hypotheses are discussed.

1. Less potential ride-sharing paths are identified when including traffic state information compared to assuming an absence of traffic congestions.

This hypothesis was constructed due to the thought that considering traffic state normally leads to speed values smaller than the maximum allowed ones as traffic congestions that trigger this decrease exist in almost every city centre. Smaller speed values and resulting bigger travel times can produce problems with the set constraints and, thus, less ride-sharing paths were expected to be marked as suitable. As shown in Chapter 6.2, the traffic state of the city centre is bad, and the network congested. This means, that on average the possible vehicle speed is, as expected, smaller than the maximum allowed one. This indeed leads to the described effect and reduces the number of identified potential ride-sharing paths. Expressed in numbers, this means that assuming an absence of traffic congestions 5'415 potential ride-sharing paths are identified with the presented system for the 15'347 available taxi trips. Including the traffic state, this number decreases by 22.25% to 4'210 identified potential ride-sharing paths. Thus, this hypothesis is corroborated.

2. The average waiting time for the second passenger is higher when including traffic state information compared to assuming an absence of traffic congestions.

The second hypothesis was again established due to the expected increasing travel times caused by considering the traffic congestions. As on average for each road segment, more time is needed to pass it, the time emerging driving from one start point to the other is increasing as well. Thus, the average waiting time is expected to be longer with traffic state information included than while assuming an absence of traffic congestions. This hypothesis is corroborated by comparing the two average waiting times of both scenarios. Not considering the traffic state leads to an average waiting time of 2 min 6 s. Including this information, the average waiting time rises by 1 min 8 s or 53.81% to 3 min 14 s and is clearly higher.

3. Savings in total travel time and total travel distance are smaller when including traffic state information compared to assuming an absence of traffic congestions.

The last hypothesis was established due to the same reasons as already explained in the two previously discussed hypotheses. It was expected that the due to traffic congestions on average bigger travel time per road segment leads to an increase in the total travel times of most of the identified shared paths. As the sum of the travel times of the two individual trips that are combined in a shared path remains the same, the difference between their travel time and the travel time of the shared path was predicted to decrease or in other words, the travel time savings to shrink. As to diminish the loss of travel time savings the shared path tries to lead through road segments where a higher vehicle speed is possible, normally an increase in the travel distance can be forecasted, as it does not represent a very direct way anymore. Hence, the average travel distance of a shared path was expected to increase as well. This would lead to a shrinkage in the total travel distance savings. These expectations are confirmed by the conducted analysis. If no traffic state information is given, 49.91% of the total travel time can be saved by the ride-sharing system. Considering real-world circumstances shrinks the savings

by 46.47% and results in 26.72% saved total travel time. The same is valid for the travel distance savings as without traffic state information included 12.8% of the total driving distance is saved with the proposed method and while considering the traffic state only 8.85% of the total driving distance can be reduced. Hence, the total travel distance savings decrease by 30.86% due to including traffic state information.

To assess the efficiency of the proposed approach and highlight again the influence of traffic state information, some of the resulting measures can be compared to the results of related studies. Starting with the matching rate, studies of Aydin et al. (2020), Barann et al. (2017), Cai et al. (2019) or Stiglic et al. (2015) produce matching rates of 65.75%, 48.34%, 77% or 74.83%. As all these studies do not include traffic state information, their values are compared to the value of the proposed system for the situation where an absence of traffic congestions is assumed. With 70.57%, the resulting matching rate of this work is part of the upper range of the given measures of other studies and therefore represents a solid value. The distance savings differ between the four studies from 33.48%, 18.98%, 33% to 29.63%. As the computed total distance savings amount only to 12.8% in the proposed ride-sharing system, this measure is rather small compared to other studies.

A reason for this might be the different objective of the system. Most of the other studies follow the objective of maximizing the distance savings or the matching rate while the system of this work follows the objective of minimizing the waiting time. As the waiting time is coupled with the travel time, this measure is on average small as well. Hence, the travel time savings of 49.91% of this work are very high compared to e.g. the travel time savings in Barran et al. (2017) of 22.42%. The same study computes a taxi fleet reduction of 24.17% which is in the same range as the taxi fleet reduction of the system of this work with 27.59%. As the presented measures of the other studies are computed based on different data, the results must be analysed with caution as comparing them only allows to study if they are more or less in the same range but not to calculate numerical differences in the absolute values. In general, this comparison shows that the proposed ride-sharing system produces results that are in the same range of results of other studies while having a very simple architecture thanks to the developed similarity measurement. The strength can be seen in the travel time savings whereas the weakness is given by the distance savings.

As already described, the compared values are computed for the situation where traffic state information is not considered. If this information is included, the values of the matching rate, the travel time savings, the distance savings, and the taxi fleet reduction shrink to 54.86%, 26.72%, 8.85% and 21.15%. These values are clearly smaller and for most of the measures not in the range of the presented results of the other studies anymore. This highlights again the negative influence traffic state information can have on ride-sharing systems. Considering this, the conclusion can be made that ride-sharing studies that do not include traffic state information, as in some of the mentioned papers, distort their results as they are embellished. Working with real circumstances of the underlying road network would lead to worse measures. The users of such systems might have to wait longer for a taxi to arrive as calculated or the trip duration unexpectedly increases. As this is not user-friendly, the traffic state information should be considered in ride-sharing systems to be useful for real-world applications.

7.4 Limitations of this work

In this subchapter, some limitations of this work are presented. The first two are related to the used subset of the data. Originally one month of GPS taxi trajectory data is available. Due to the time consumption of the methods based on the rather slow server infrastructure and the limited time window in which this work is done, only one day of the dataset is analysed. Analysing all the available data would be interesting especially for the traffic state estimation. With data of only one day, the changes in the traffic state between working days and weekends cannot be analysed. Considering this, further conclusions could be made on the differences in the influence of traffic state information based on different weekdays. Additionally, using more data would improve the accuracy of the estimation method. The second limitation, which represents the biggest limitation, is given by the problems while map-matching the GPS taxi trajectory data. The errors occurring with the arcpy Python module significantly reduce the size of the data. Though still enough trips are left to conduct the analysis, having more data available would lead to better and more accurate results.

During the interpolation process of the calculated vehicle speed values, a global Kriging approach is implemented. It is argued that as the speed values of the generated sub-networks are in the same range, independent from the location on the network, using a local interpolation method is not required. Nevertheless, it would be an improvement to show the differences in the two methods and then decide based on this which one truly is more suitable. Another improvement could be done by considering more than one traffic state time window for the weights of the Dijkstra's shortest path algorithm for trips with a duration bigger than 15 minutes. As aforementioned in Chapter 5.5.3.1, the rather simple approach of only considering one time window is legitimated with the fact that taxi drivers like to stick to the at the beginning computed route. Nonetheless, by changing this more detailed results could be obtained.

The last limitation is related to the similarity measurement. As the goal of this method is to reduce the complexity of the matching process and make the system more efficient, it would be nice to monitor the computation time of this step to analyse how much more efficient the system gets compared to already existing ones. Unfortunately, this additional analysis is not conducted in the study, as it would go beyond the scope of this work.

8. Conclusion

This work shows how potential ride-sharing paths can be efficiently identified starting with raw GPS taxi trajectory data while considering the estimated traffic state of the underlying road network. By applying the developed framework to real-world GPS data, it is analysed what influences such information about the traffic state can have on ride-sharing systems. The major findings are that traffic state information leads to more conservative (and thus likely more realistic) matching of trips, which shows itself in lower respectively worse values for the calculated measures. Most affected by the traffic state are the total travel time savings, which get reduced by 46.47%, and the average waiting time with an increase of 53.81%. The total distance savings (reduction of 30.86%), the CO_2 emission savings (decrease of 30.86%), the taxi fleet reduction (23.34% fewer savings) and the matching rate (reduction of 15.71%) are affected less severely.

Comparing the resulting measures with existing ride-sharing studies for the situation where the traffic state information is not considered shows that besides being built less complex and more efficient, the results of the proposed system can keep up with the other studies. Analysing the differences between results of existing literature and the resulting measures when the traffic state information is included highlights the negative effect traffic state information has on ride-sharing systems. This allows claiming that ride-sharing system not considering traffic state information distort their results as they are embellished. This can lead to a decrease in the user-friendliness of a system as unexpected different waiting times or delays can emerge. Thus, this study shows that including traffic state information can be a very important point to make a ride-sharing system more useful to real-world applications.

Another finding is that not forcing an identified ride-sharing path to result in a shorter travel distance than the sum of the travel distances of the two individual trips that build this shared path, has a rather strong influence on the measures of a ride-sharing system. The matching rate, the travel time savings and the taxi fleet reduction are negatively affected; meaning they decrease, while the waiting time, the total distance savings and the CO_2 emission reduction are influenced positively. This means that the waiting time gets reduced and the total distance savings as well as the savings of CO_2 emissions increase. Despite not affecting all the resulting measures positively, it is shown that this additional constraint should be included in the matching process of a ride-sharing system to have a positive impact on the natural environment.

In addition to the presented major findings, this work provides three further contributions. First, a framework containing all the necessary steps beginning with raw GPS taxi trajectory data and a road network to efficiently identify potential ride-sharing paths is developed. Different from previous ride-sharing studies, by considering the information on the traffic state of the underlying road network, real-world circumstances are included in the ride-sharing system. Second, with the described traffic state estimation method an alternative approach to existing methods in the literature is presented that is based only on the already in the ride-sharing system used data. No further inputs are needed. Moreover, the result of this estimation is included twice in the identification process of potential ride-sharing paths to solve the problem more realistically.

The third contribution is represented by a newly developed similarity measurement that filters out unsimilar and thus for ride-sharing unsuitable taxi trips that initially serve as candidates to build a ride-sharing path with an analysed trip. Different to existing similarity measurements in the literature, this method is built for ride-sharing systems of taxis and only measures the distance between the start and end points of two taxi trips respectively between start or end points and the closest point of the opposite trip. The significance of this is that the route of a trip is less important while the start and end points are weighted stronger. Being able to filter out highly unsimilar candidate trips and having to compute only the fastest paths for a maximum of three times between an analysed trip and its three most similar candidates, allows the complexity of the system to decrease and leads to higher efficiency. Besides considering real-world circumstances, due to the similarity measurement, the framework can represent a compared to existing ride-sharing systems less complicated and more efficient solution.

For future research, the presented ride-sharing system of this work must be compared in more detail, in sense of time consumption and quality, to systems of other studies. Analysing the same dataset with the proposed system and an existing one allows to monitor how much time is needed for the computations of both systems and what the resulting measures are. Through this method it will be shown if because of the implemented similarity measurement, the system truly is more efficient and if yes, how much computation time can be saved. Furthermore, by comparing the absolute values of the measures, what is not possible given results based on different datasets, it can be validated whether besides being more efficient in computation time, the quantity and quality of the results can keep up with existing systems.

In addition, future research studies about ride-sharing systems including traffic state information must analyse the influence such information has on the resulting taxi fare. The aspect of the price of a shared ride is not considered in this work, but eventually, it is influenced by traffic state information as well. Normally, the taxi fare of a trip or at least the range of it, if requested through an application, is known before starting the ride. If no traffic state information is included, the algorithm potentially computes a too short travel time or distance, which could lead to an incorrect taxi fare. How including the traffic state influences this price and to what extent the taxi company and the users earn respectively spend more, must be addressed in future work. Last, the proposed static ride-sharing system must be transformed into a dynamic system keeping the structure of the approach (using the traffic state information and simplify the matching process by the similarity measurement). This way it can be implemented into a mobile application to be used in real-world scenarios.

Literature

Agatz, N. A. H., Erera, A. L., Savelsbergh, M. W. P. & Wang, X. (2011). Dynamic ride-sharing: A simulation study in metro Atlanta. *Transportation Research Part B: Methodological*, 45 (9), 1450–1464.

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.

Aly, H., Basalamah, A. & Youssef, M. (2016). Robust and ubiquitous smartphone-based lane detection. *Pervasive and Mobile Computing*, 26, 35–56.

Angloinfo China (2020). Speed Limits and Types of Roads. *https://www.angloinfo.com/how-to/china/transport/driving/on-the-road* [Accessed: 11.03.2020].

Armant, V. & Brown, K. N. (2014). Minimizing the Driving Distance in Ride Sharing Systems. *IEEE 26th International Conference on Tools with Artificial Intelligence*, 568–575.

Asakura, Y., Kusakabe, T., Nguyen, L. X. & Ushiki, T. (2017). Incident detection methods using probe vehicles with on-board GPS equipment. *Transportation Research Part C: Emerging Technologies*, 81, 330–341.

Aydin, O. F., Gokasar, I. & Kalan, O. (2020). Matching algorithm for improving ridesharing by incorporating route splits and social factors. *PLoS ONE*, 15(3), 1–23.

Barann, B., Beverungen, D. & Müller, O. (2017). An open-data approach for quantifying the potential of taxi ridesharing. *Decision Support Systems*, 99, 86–95.

Bathla, K., Raychoudhury, V., Saxena, D. & Kshemkalyani, A. D. (2018). Real-Time Distributed Taxi Ride Sharing. *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2044–2051.

Bernstein, D. & Kornhauser, A. (1996). An Introduction to Map Matching for Personal Navigation Assistants. *New Jersey TIDE Center*, 1–16.

Besse, P. C., Guillouet, B., Loubes, J. M. & Royer, F. (2016). Review and Perspective for Distance-Based Clustering of Vehicle Trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17(11), 3306–3317.

Bierlaire, M., Chen, J. & Newman, J. (2013). A probabilistic map matching method for smartphone GPS data. *Transportation Research Part C*, 26, 78–98.

Cai, H., Wang, X., Adriaens, P. & Xu, M. (2019). Environmental benefits of taxi ride sharing in Beijing. *Energy*, 174, 503–508.

Cao, B., Alarabi, L., Mokbel, M. F. & Basalamah, A. (2015). SHAREK: A Scalable Dynamic Ride Sharing System. *Proceedings - IEEE International Conference on Mobile Data Management*, 1, 4–13.

Chan, N. D. & Shaheen, S. A. (2012). Ridesharing in North America: Past, Present, and Future. *Transport Reviews*, 32(1), 93–112.

Cox, S. & Little, C. (2020). Time Ontology in OWL. *https://www.w3.org/TR/2020/CR-owl-time-20200326/* [Accessed: 25.06.2020].

Crabtree, J. (2018). Didi Chuxing took on Uber and won. Now it's taking on the world. *https://www.wired.co.uk/article/didi-chuxing-china-startups-uber* [Accessed: 05.11.2019].

De Fabritiis, C., Ragona, R. & Valenti, G. (2008). Traffic Estimation And Prediction Based On Real Time Floating Car Data. *IEEE Conference on Intelligent Transportation Systems*, 197–203.

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.

Gökay, S., Heuvels, A. & Krempels, K. H. (2019). On-demand Ride-sharing Services with Meeting Points. *VEHITS 2019 - Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, 117–125.

Goldberg, A. V. & Tarjan, R. E. (1996). Expected Performance of Dijkstra ' s Shortest Path Algorithm. Princeton University, Computer Science Department, 1-6.

Greenfeld, J. S. (2002). Matching GPS Observations to Locations on a Digital Map. *Transportation Research Board*, 3, 13.

Haddad, Y., Cohen, Y. & Goldsmith, R. (2013). A Dynamic Real Time Car Sharing System. *International Conference on Soft Computing and Software Engineering*, 1-7.

Hashemi, M. & Karimi, H. A. (2014). A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*, 48, 153–165.

He, M., Zheng, L., Cao, W., Huang, J., Liu, X. & Liu, W. (2019). An enhanced weight-based real-time map matching algorithm for complex urban networks. *Physica A: Statistical Mechanics and Its Applications*, 534, 122318, 1-13.

He, W., Hwang, K. & Li, D. (2014). Intelligent Carpool Routing for Urban Ridesharing by Mining GPS Trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 15(5), 2286–2296.

Hosni, H., Naoum-Sawaya, J. & Artail, H. (2014). The shared-taxi problem: Formulation and solution methods. *Transportation Research Part B*, 70, 303–318.

Jia, T., Luo, W., Jia, H., Zhu, H. & Li, X. (2016). Research on Remote Diagnosis System Based on Baidumap API and OBD II diagnosis technology. *2016 International Conference on Communication Problem-Solving*, 1–3.

Jung, J., Jayakrishnan, R. & Park, J. Y. (2013). Design and Modeling of Real-time Shared-Taxi Dispatch Algorithms. *Transportation Research Board*, 1–20.

Kerner, B. S., Demir, C., Herrtwich, R. G., Klenov, S. L., Rehborn, H., Aleksić, M. & Haug, A. (2005). Traffic State Detection with Floating Car Data in Road Networks. *IEEE Conference on Intelligent Transportation Systems*, 700–705.

Kong, Q. J., Zhao, Q., Wei, C. & Liu, Y. (2013). Efficient Traffic State Estimation for Large-Scale Urban Road Networks. *IEEE Transactions on Intelligent Transportation Systems*, 14(1), 398–407.

Lin, S. (2018). ChinaCoordinate. *https://github.com/versey-sherry/ChinaCoordinate* [Accessed: 23.11.2019].

Liu, Y., Xu, J. & Luo, H. (2014). An integrated approach to modelling the economy-society-ecology system in urbanization process. *Sustainability*, 6, 1946–1972.

Mobitool (2016). mobitool-Faktoren v2.0. *https://www.mobitool.ch/admin/data/files/tool/tool_file_de/5/mobitool-faktoren-v2.0.2.xlsm?lm=1491413883* [Accessed: 20.08.2020].

Nanthawichit, C., Nakatsuji, T. & Suzuki, H. (2003). Application of Probe Vehicle Data for Real-Time Traffic State Estimation and Short-Term Travel Time Prediction on a Freeway. *TRB 2003 Annual Meeting*, 1–16.

Newson, P. & Krumm, J. (2009). Hidden Markov Map Matching Through Noise and Sparseness. *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 336–343.

Obradovic, D., Lenz, H. & Schupfner, M. (2006). Fusion of Map and Sensor Data in a Modern Car Navigation System. *Journal of VLSI Signal Processing*, 45, 111–122.

Ochieng, W. Y., Quddus, M. A. & Noland, R. B. (2004). Positioning algorithms for transport telematics applications. *Journal of Geospatial Engineering*, 6(2), 10–30.

Oliver, M. A. & Webster, R. (2014). A tutorial guide to geostatistics: Computing and modelling variograms and kriging. *Catena*, 113, 56–69.

Ota, M., Vo, H., Silva, C. & Freire, J. (2015). A Scalable Approach for Data-Driven Taxi Ride-Sharing Simulation. *2015 IEEE International Conference on Big Data*, 888–897.

Petit, S. (2017). World Vehicle Population Rose 4.6% in 2016. *https://wardsintelligence.informa.com/WI058630/World-Vehicle-Population-Rose-46-in-2016* [Accessed: 12.03.2020].

Piórkowski, A. (2011). MySQL Spatial and PostGIS - Implementations of spatial data standards. *Electonic Journal of Polish Agricultural Universities*, *14*(1), 1–8.

Qin, B. (2015). City profile: Chengdu. *Cities*, 43, 18–27.

Quddus, M. A., Ochieng, W. Y. & Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5), 312–328.

R Core Team. (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), 257–286.

Rayle, L., Shaheen, S., Chan, N., Dai, D. & Cervero, R. (2014). App-Based, On-Demand Ride Services: Comparing Taxi and Ridesourcing Trips and User Characteristics in San Francisco. *University of California Transportation Center*, 1-19.

Reed, T. (2020). Global Traffic Scorecard. *INRIX Research*, 1, March, 1-20.

Ren, M. & Karimi, H. A. (2009). A hidden Markov Model-Based Map-Matching Algorithm for Wheelchair Navigation. *Journal of Navigation*, 62, 383–395.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H. & Ratti, C. (2014a). Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences of the United States of America*, 111(37), 13290–13294.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. & Ratti, C. (2014b). Supporting Information for quantifying the benefits of vehicle pooling with shareability networks. 1–24.

Schneider, S. (2017). Map matching in ArcGIS *https://github.com/simonscheider/mapmatching/wiki/Map-matching-in-ArcGIS* [Accessed: 12.01.2020].

Schwieterman, J. & Smith, C. S. (2018). Sharing the ride: A paired-trip analysis of UberPool and Chicago Transit Authority services in Chicago, Illinois. *Research in Transportation Economics*, 71, 9–16.

Shen, B., Huang, Y. & Zhao, Y. (2015). Dynamic Ridesharing. *SIGSPATIAL Special*, 7(3), 3–10.

Shete, A., Bhandare, V., Londhe, L. & P.B.Mali, P. B. M. (2015). Intelligent Carpooling System. *International Journal of Computer Applications*, 118(4), 26–31.

Stemler, A., Evans, J. & Himebaugh, B. (2019). The Chinese Experiment: Lessons from the Regulation of Ridesharing in China. *Indiana University Kelley School of Business Research Paper*, 19–48, 1–54.

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.

Sun, Y., Chen, Z. L. & Zhang, L. (2020). Nonprofit peer-to-peer ridesharing optimization. *Transportation Research Part E*, 142, 1–26.

Sunderrajan, A., Viswanathan, V., Cai, W. & Knoll, A. (2016). Traffic State Estimation Using Floating Car Data. *Procedia Computer Science*, 80, 2008–2018.

Syed, S. & Cannon, M. E. (2004). Fuzzy Logic Based-Map Matching Algorithm for Vehicle Navigation System in Urban Canyons. *Proceedings of the National Technical Meeting*, 982–993.

Theodoridis, S. & Koutroumbas, K. (2009). Chapter 9 - Context-Dependent Classification. In Theodoridis, S. & Koutroumbas, K. (2009). Patter Recognition, 4, 521-565.

Tian, C., Huang, Y., Liu, Z., Bastani, F. & Jin, R. (2013). Noah: A dynamic ridesharing system. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 985–988.

van Kreveld, M. & Luo, J. (2007). The Definition and Computation of Trajectory and Subtrajectory Similarity. *Proceedings of the 15th International Symposium on Advances in Geographic Information Systems*, 1–4.

Velaga, N. R., Quddus, M. A. & Bristow, A. L. (2009). Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies*, 17(6), 672–683.

Wang, X. & Kockelman, K. M. (2009). Forecasting network data spatial interpolation of traffic counts from texas data. *Transportation Research Record*, 2105, 100–108.

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.

Wikitravel (2008). Driving. *http://www.china.org.cn/travel/beijingguide/2008-06/04/content_15616950_3.htm* [Accessed: 19.04.2020].

Yang, D., Cai, B. & Yuan, Y. (2003). Improved Map-Matching Algorithm Used in Vehicle Navigation System. *Proceedings of Intelligent Transportation Systems Conference*, 2, 1246–1250.

Yang, H., Cheng, S., Jiang, H. & An, S. (2013). An enhanced weight-based topological map matching algorithm for intricate urban road network. *Procedia - Social and Behavioral Sciences*, 96, 1670–1678.

Ye, J. (2018). Big Data at Didi Chuxing. *SIRIP: Industry Days*, 18, 1341–1341.

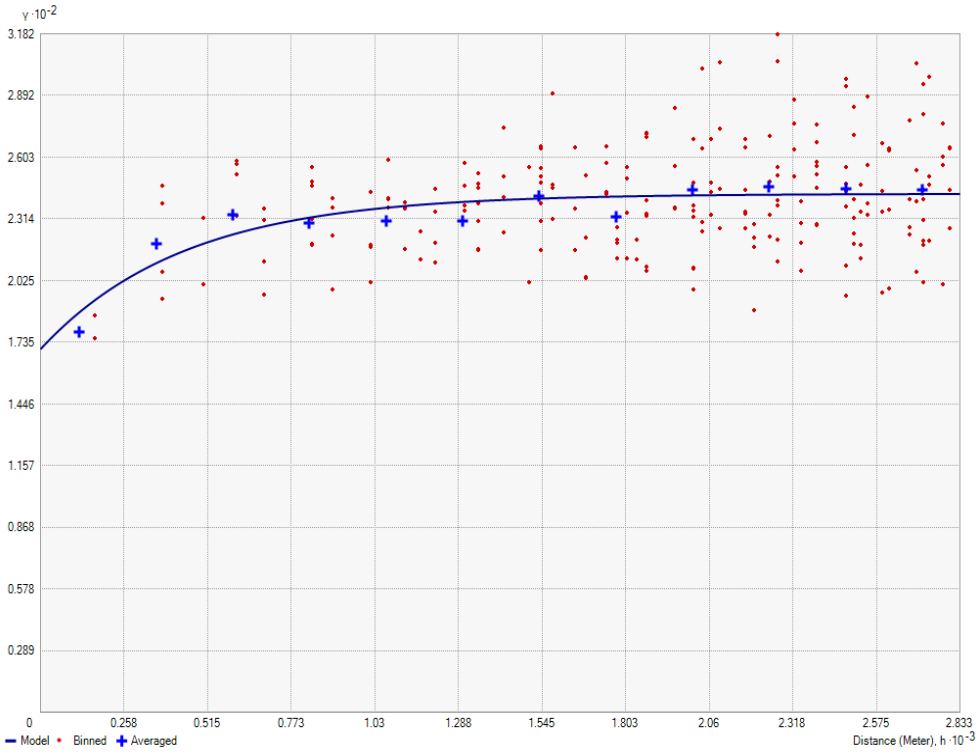
Yu, H., Raychoudhury, V. & Silwal, S. (2020). Dynamic Taxi Ride Sharing using Localized Communication. *ICDCN 2020: Proceedings of the 21st International Conference on Distributed Computing and Networking*, 1–10.

Zhang, Y., Li, X., Wang, A., Bao, T. & Tian, S. (2015). Density and diversity of OpenStreetMap road networks in China. *Journal of Urban Management*, 4, 135–146.

Appendix

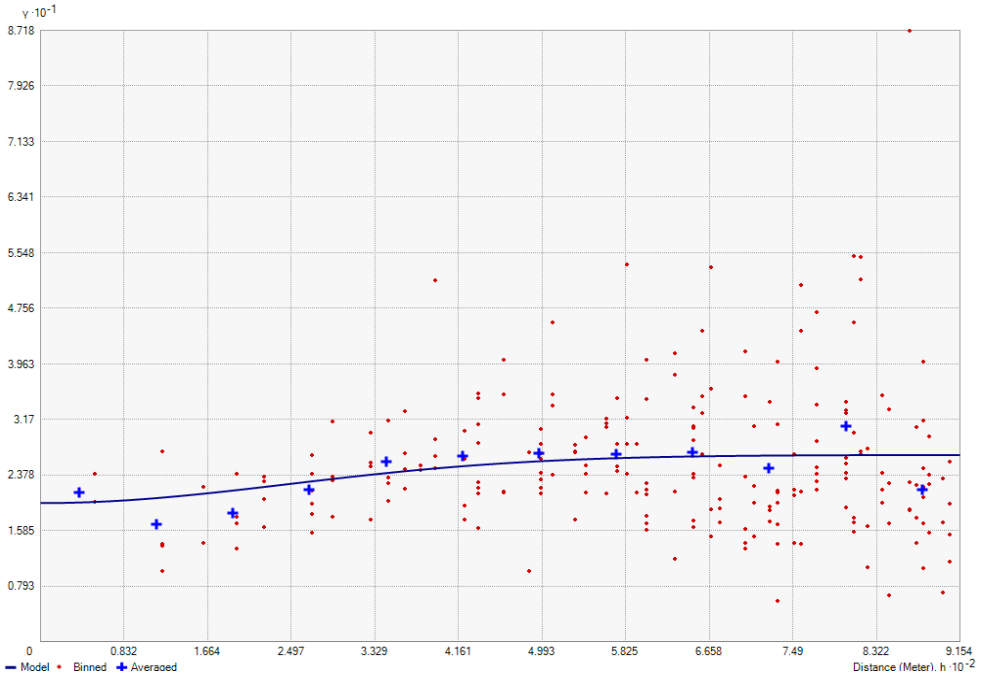
In this appendix, figures which appear slightly too small in the main text are again illustrated in bigger size. The numbering of them remains the same. This means that each figure in this appendix is labelled with the same number as in the main text. Hence, in the figure catalogue the figures of the appendix are not repeated.

Sub-network «primary street»



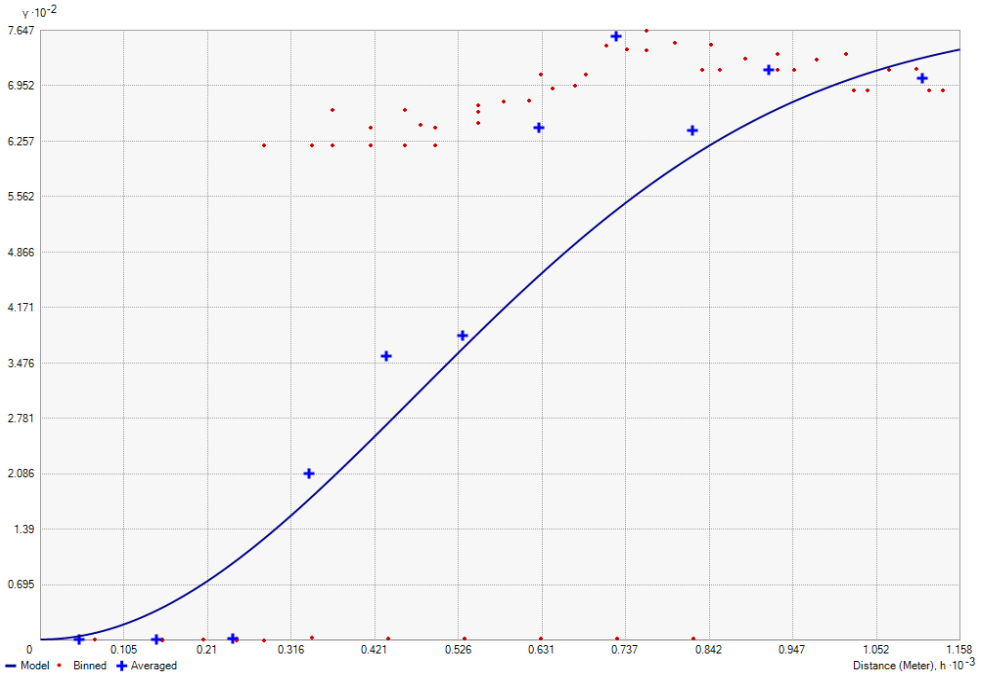
- **Curve**: Exponential model
- **Nugget**: 170.305
- **Major range**: 1327.01
- **Partial sill**: 72.87

Sub-network «living street»



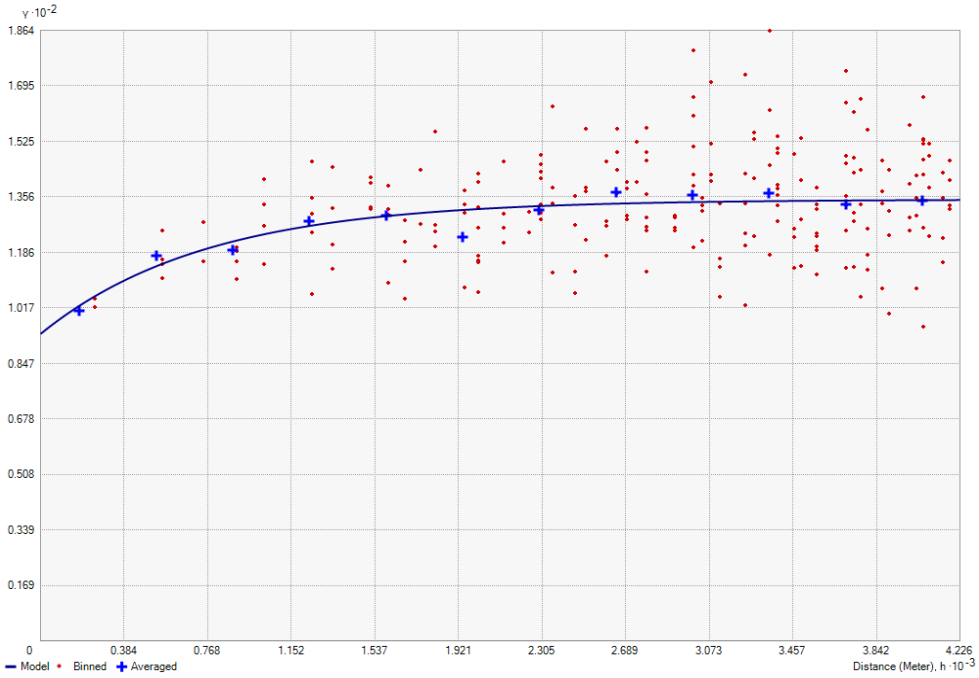
- **Curve**: Gaussian model
- **Nugget**: 19.76
- **Major range**: 603.22
- **Partial sill:** 6.86

Sub-network «motorway»



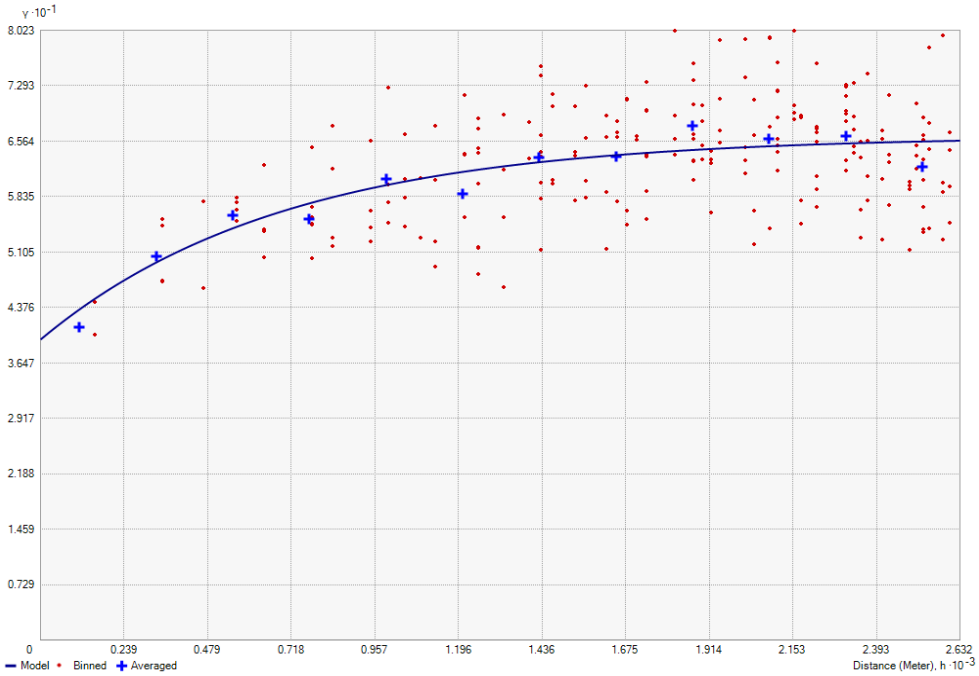
- **Curve**: Gaussian model
- **Nugget**: 0.78
- **Major range**: 1157.69
- **Partial sill**: 778.66

Sub-network «secondary street»



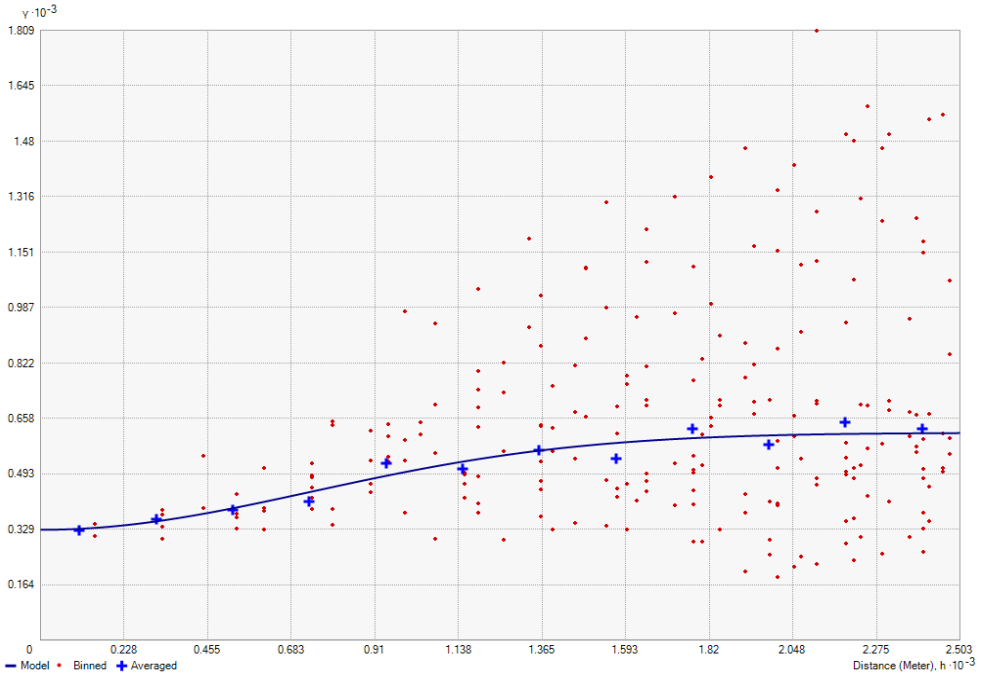
- **Curve**: Exponential model
- **Nugget**: 93.75
- **Major range**: 2279.30
- **Partial sill**: 41.03

Sub-network «tertiary street»



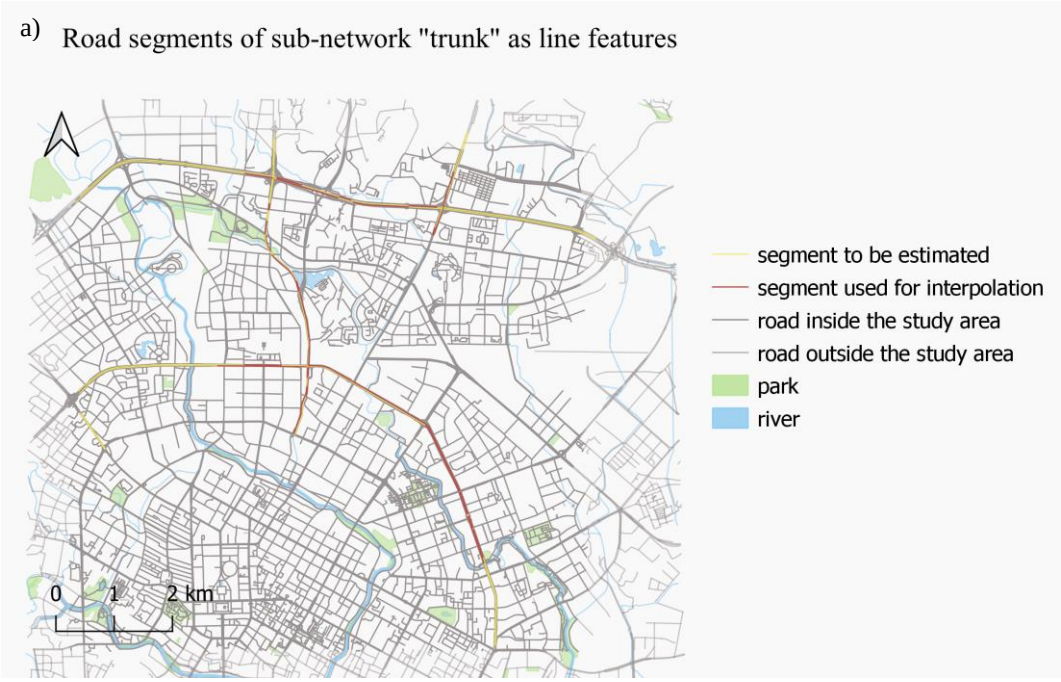
- **Curve**: Exponential model
- **Nugget**: 39.54
- **Major range**: 2086.15
- **Partial sill**: 26.77

Sub-network «trunk»



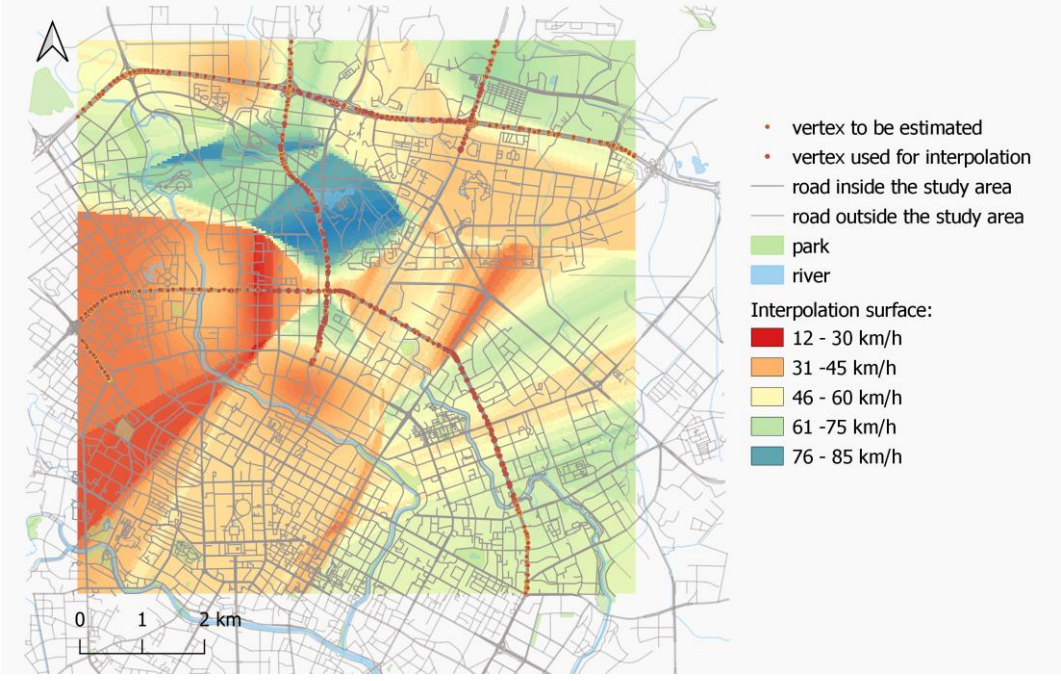
- **Curve**: Gaussian model
- **Nugget**: 327.17
- **Major range**: 1807.55
- **Partial sill**: 287.85

Figure 24: Variograms and used parameters of the six sub-networks. The parameters are analysed for the time window between 12:00 p.m. and 12:15 p.m. and taken as the input for the Ordinary Kriging interpolation method.



b) Road segments of sub-network "trunk" as point features

c) Continuous distribution of the interpolated speed values inside the study area for the sub-network "trunk"



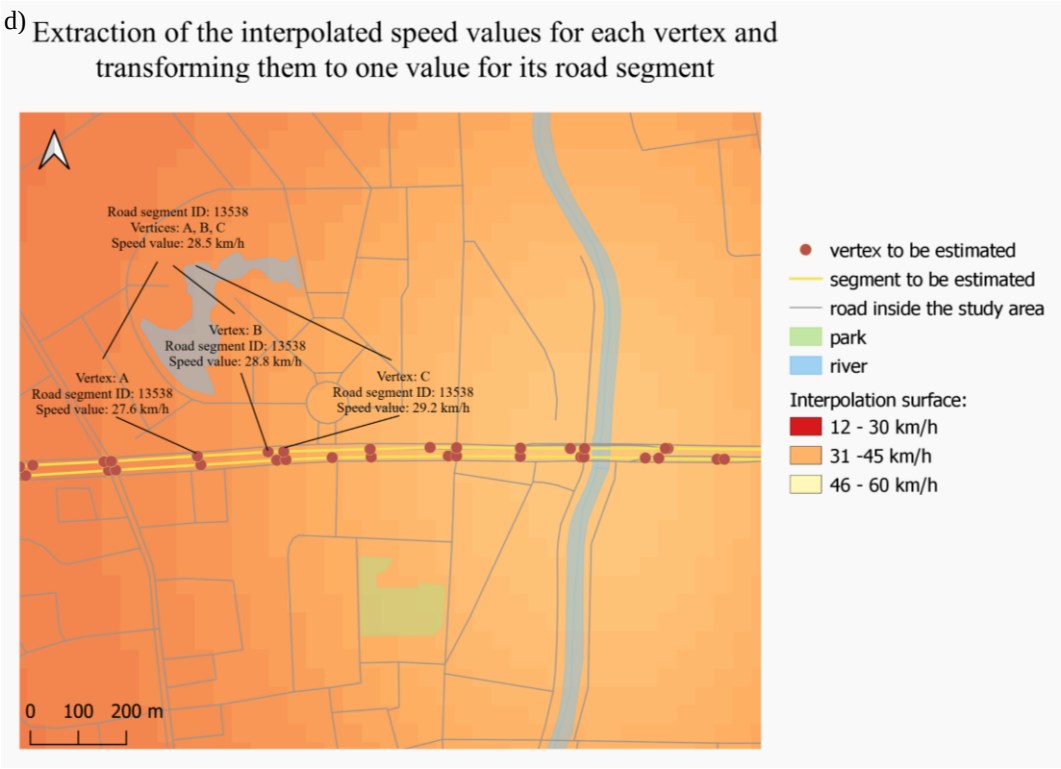
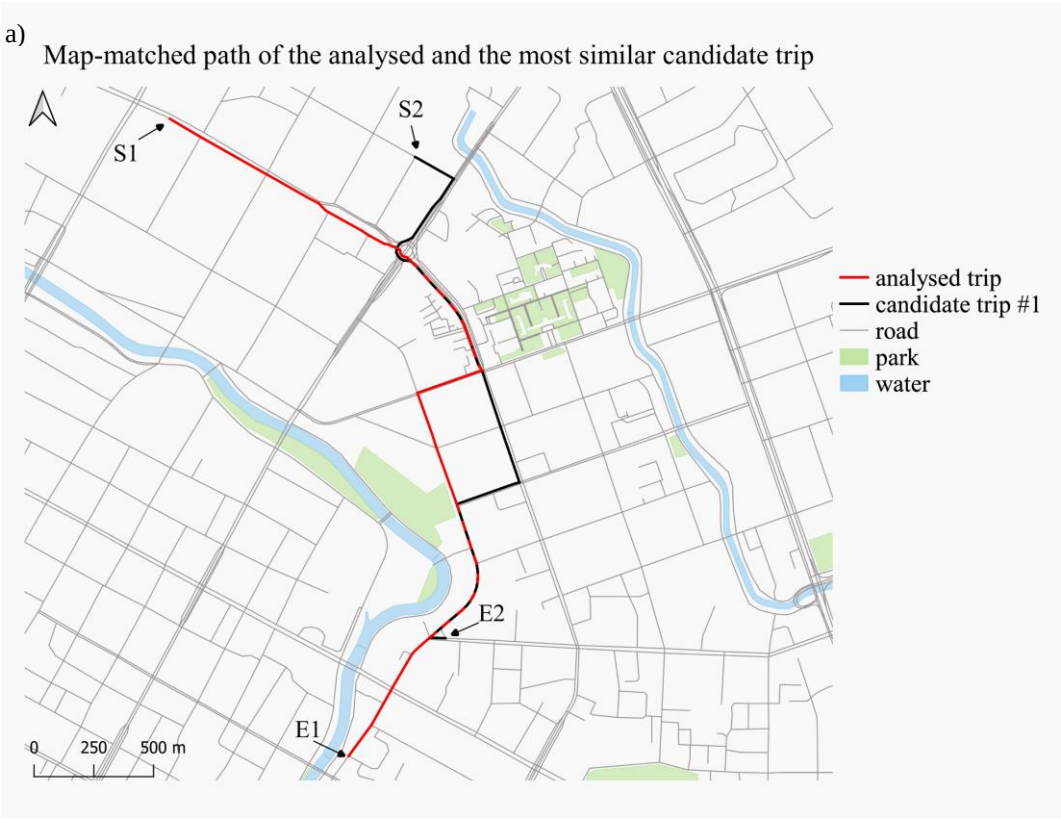


Figure 25: a) shows the sub-network and the road segments with the missing values. In b) the line features are split into its vertices. The resulting interpolated speed values are illustrated in c). As shown in d), the average of the speed values of the three vertices that are extracted from the Kriging surface represents the final interpolated speed value for the specific road segment.

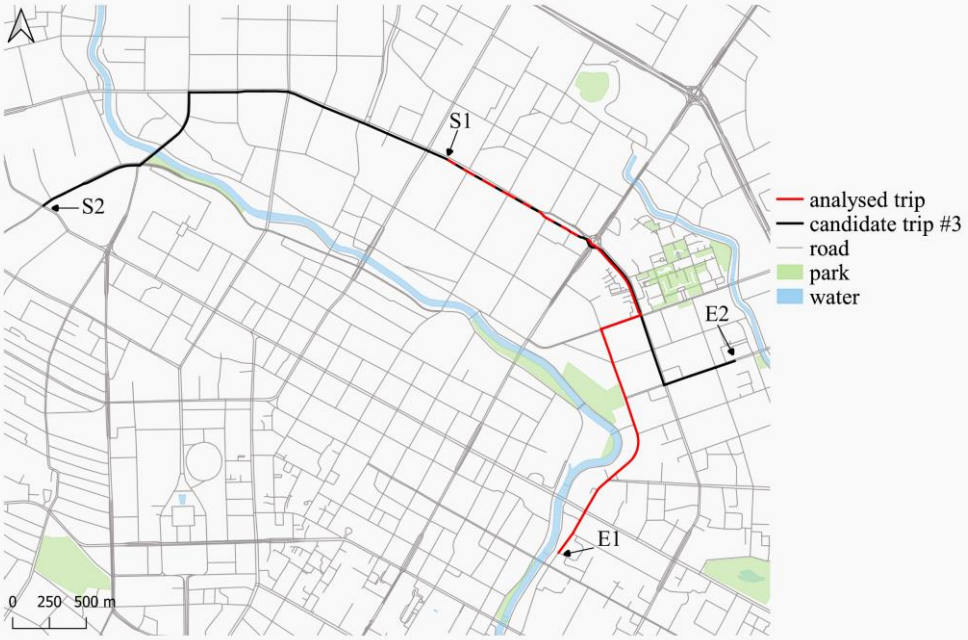


b)

Map-matched path of the analysed and the second most similar candidate trip

c)

Map-matched path of the analysed and the third most similar candidate trip



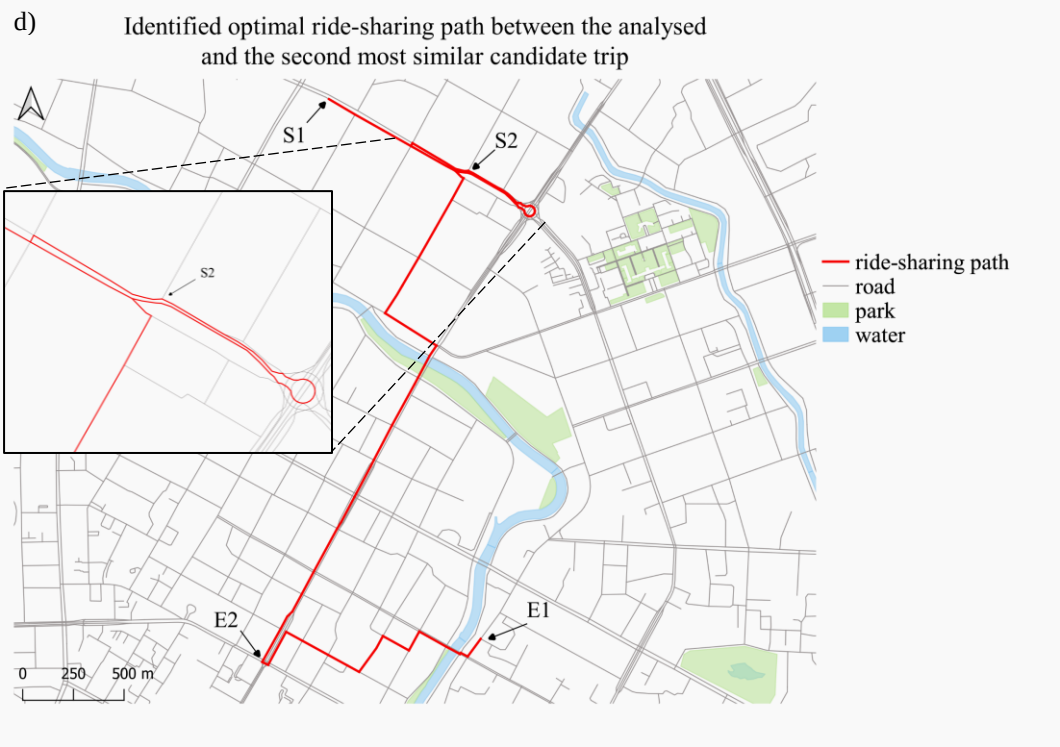
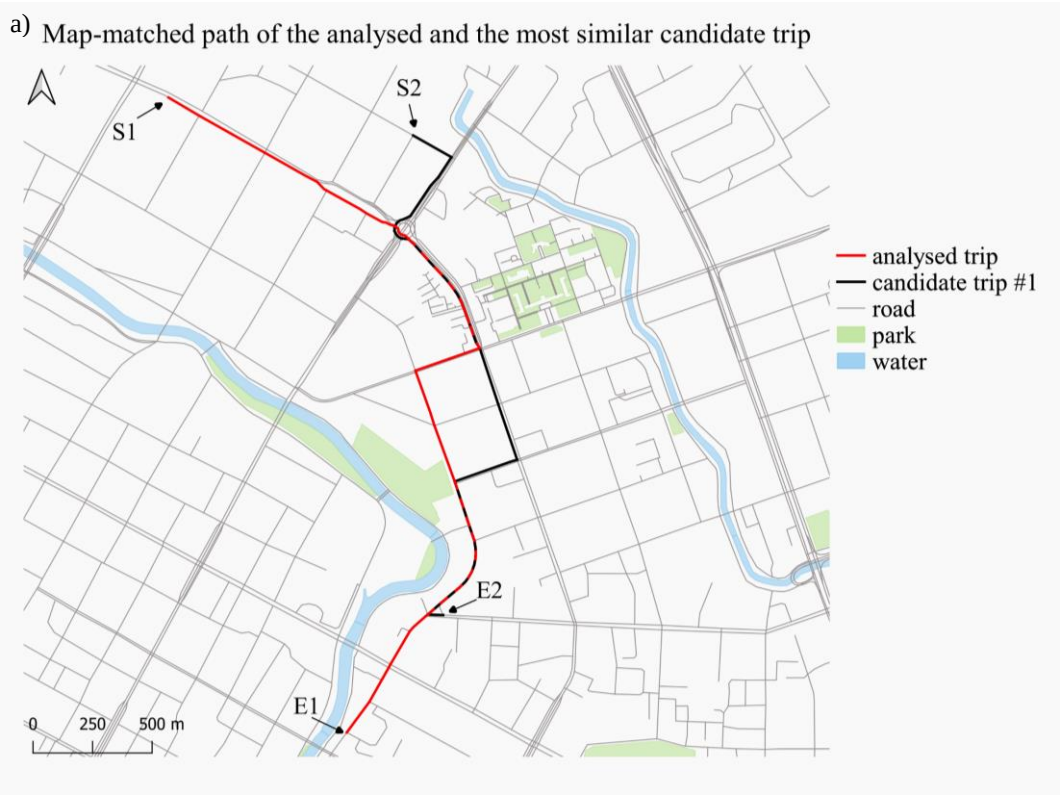


Figure 53: Visualisation of the three most similar candidate trips and the identified optimal ride-sharing path for the analysed example trip illustrated by the red line in a) to c). The final ride-sharing path is displayed in d). This path is a combination of the analysed with the second most similar candidate trip. As in variation one the distance savings constraint must not be met the total driving distance of the shared path can be slightly bigger than the sum of the driving distances of the two individual taxi trips, as it is the case in d).

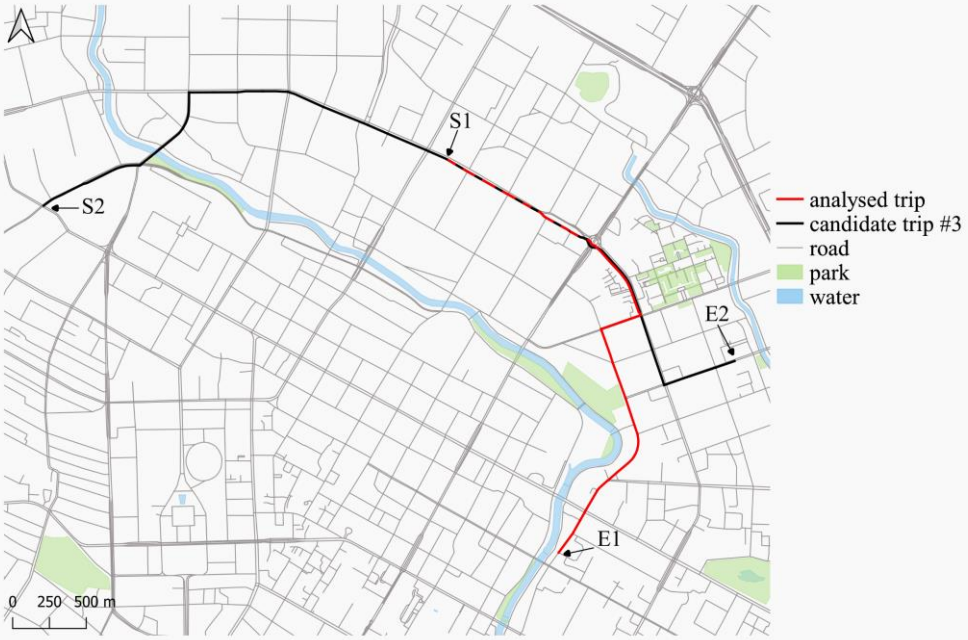


b)

Map-matched path of the analysed and the second most similar candidate trip

c)

Map-matched path of the analysed and the third most similar candidate trip



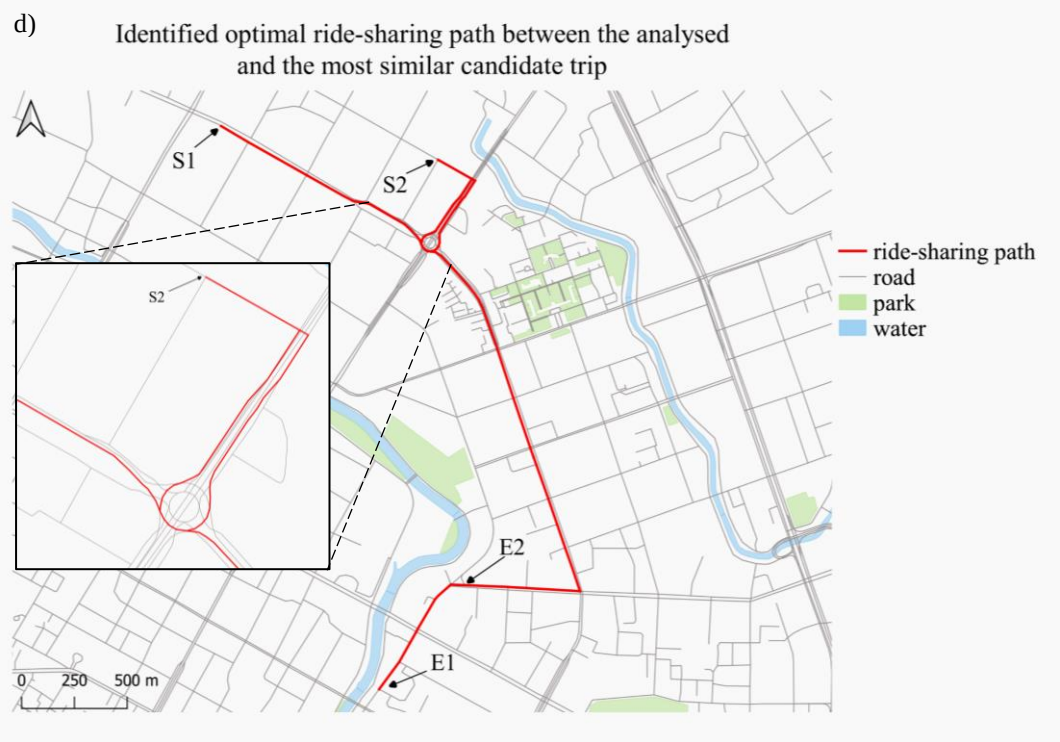
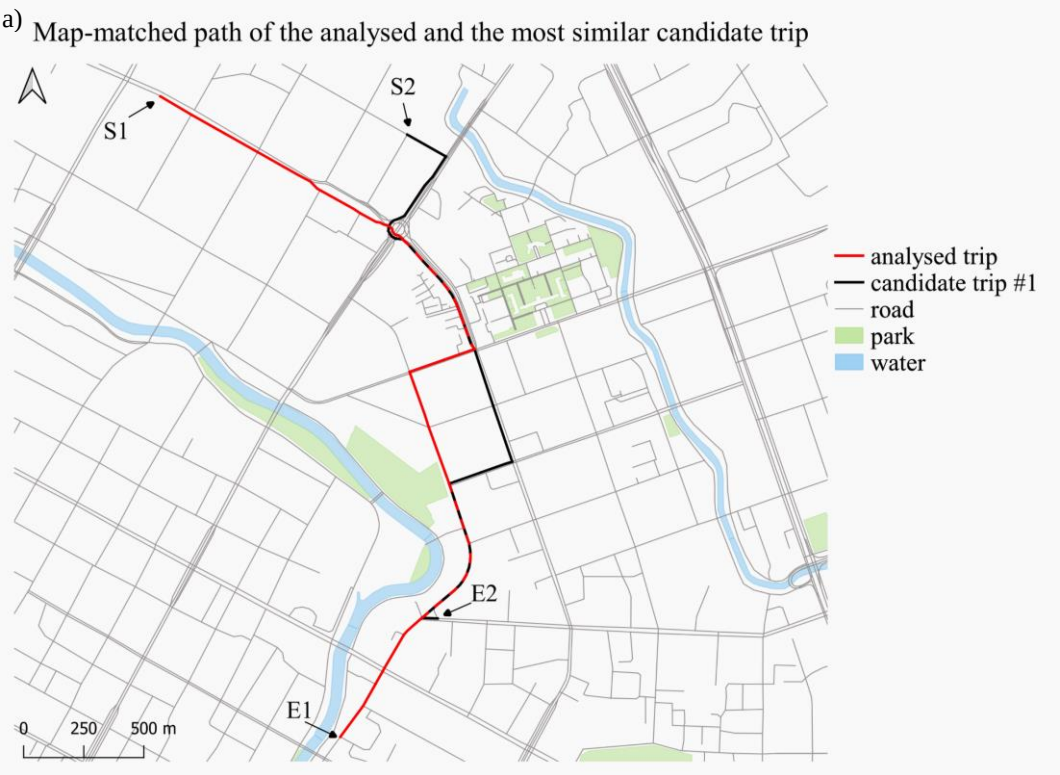
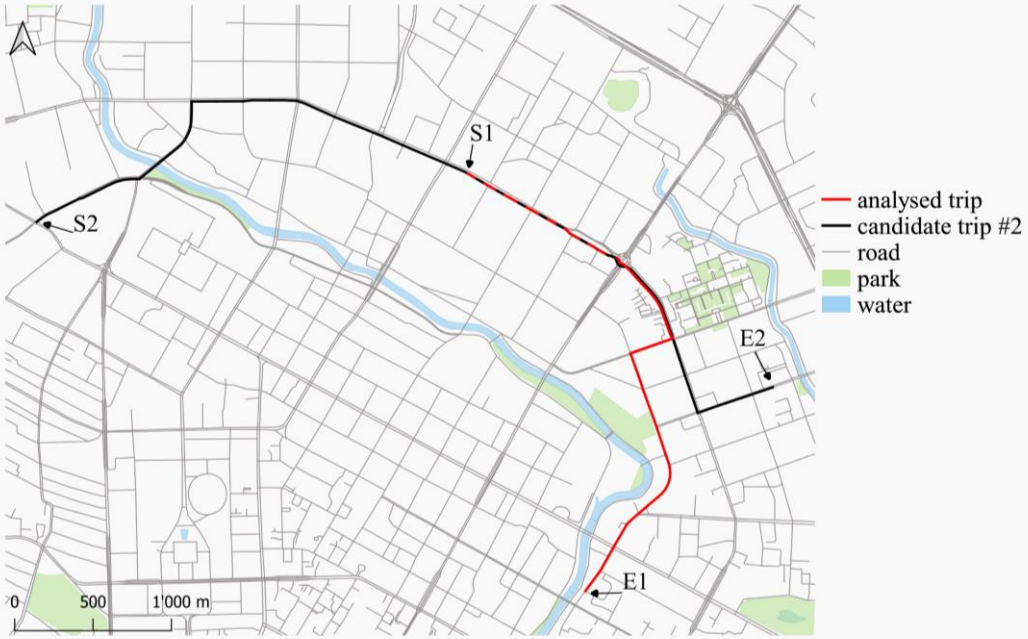


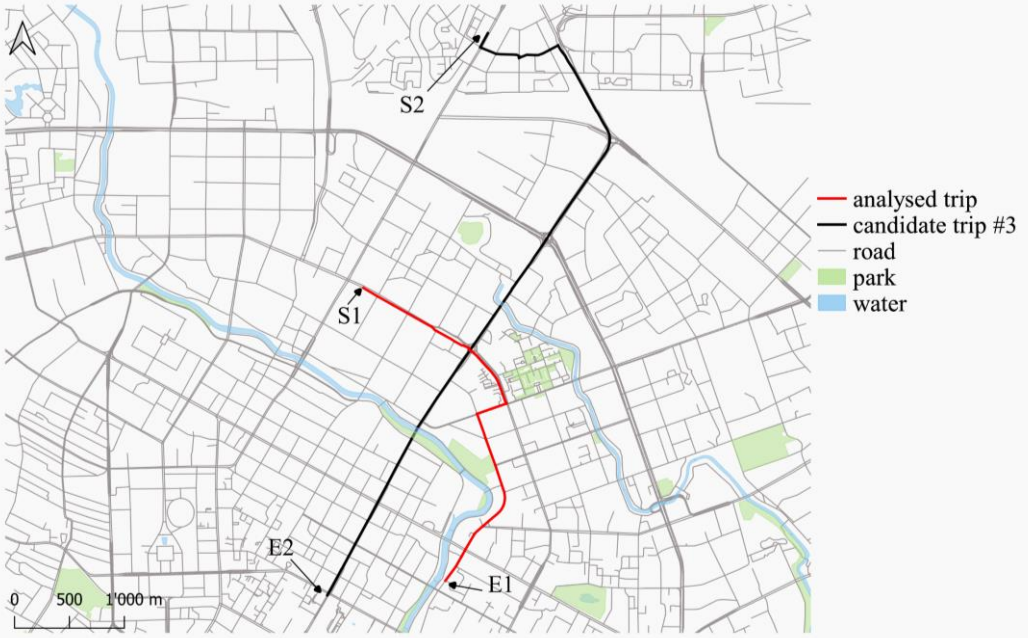
Figure 54: Visualisation of the three most similar candidate trips in a) to c) and the identified optimal ride-sharing path in d) for the analysed example trip. The final ride-sharing path of the second variation is a combination of the analysed with the most similar candidate trip.



b)
Map-matched path of the analysed and the second most similar candidate trip



c)
Map-matched path of the analysed and the third most similar candidate trip



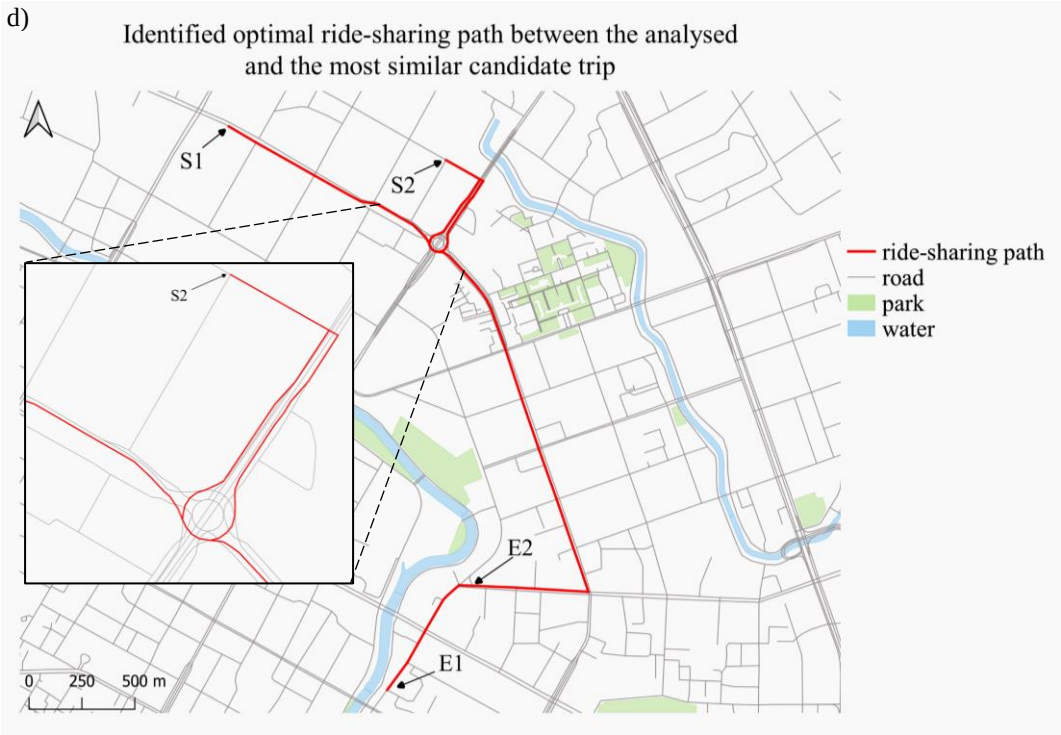
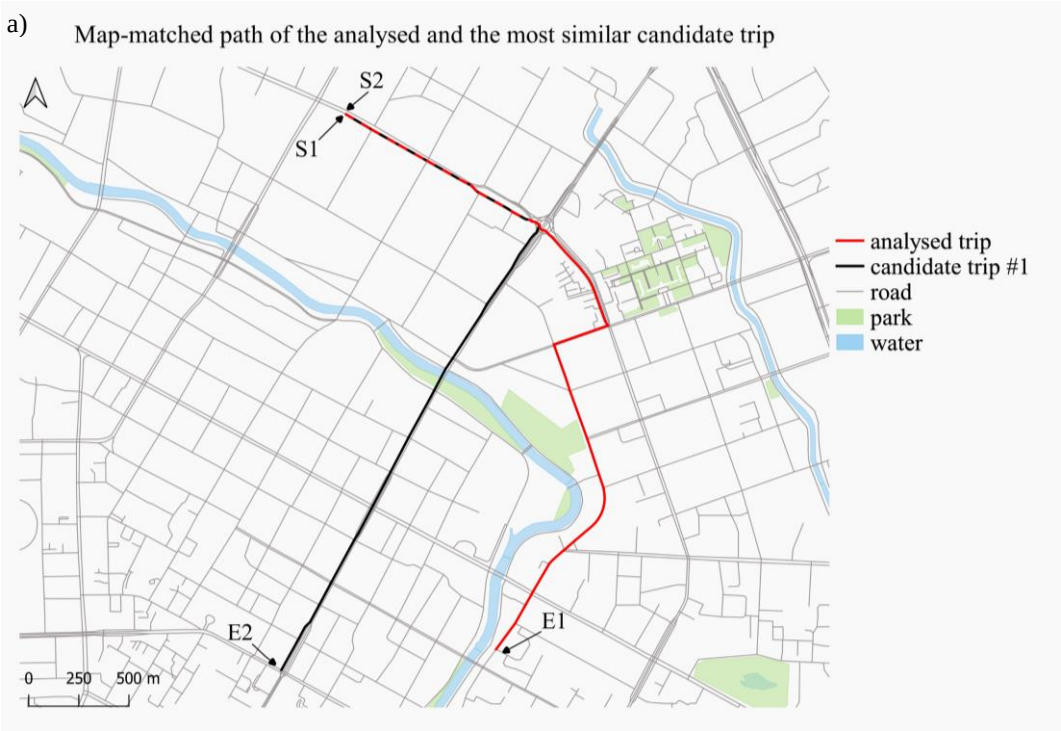
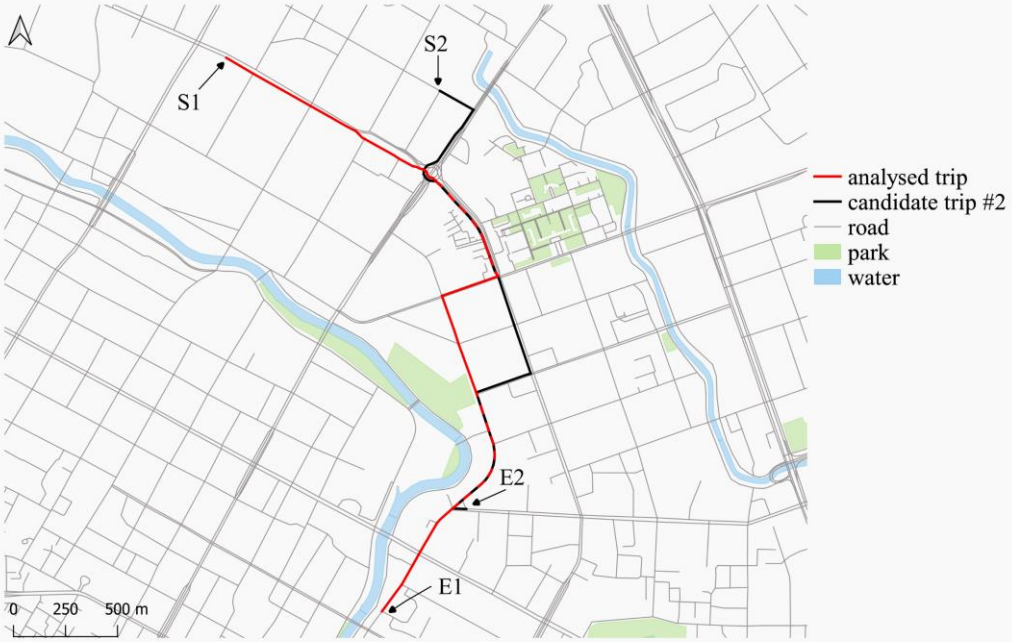


Figure 56: Visualisation of the three most similar candidate trips in a) to c) and the identified optimal ride-sharing path in d) for the analysed example trip. The final ride-sharing path of the third variation is a combination of the analysed with the most similar candidate trip.

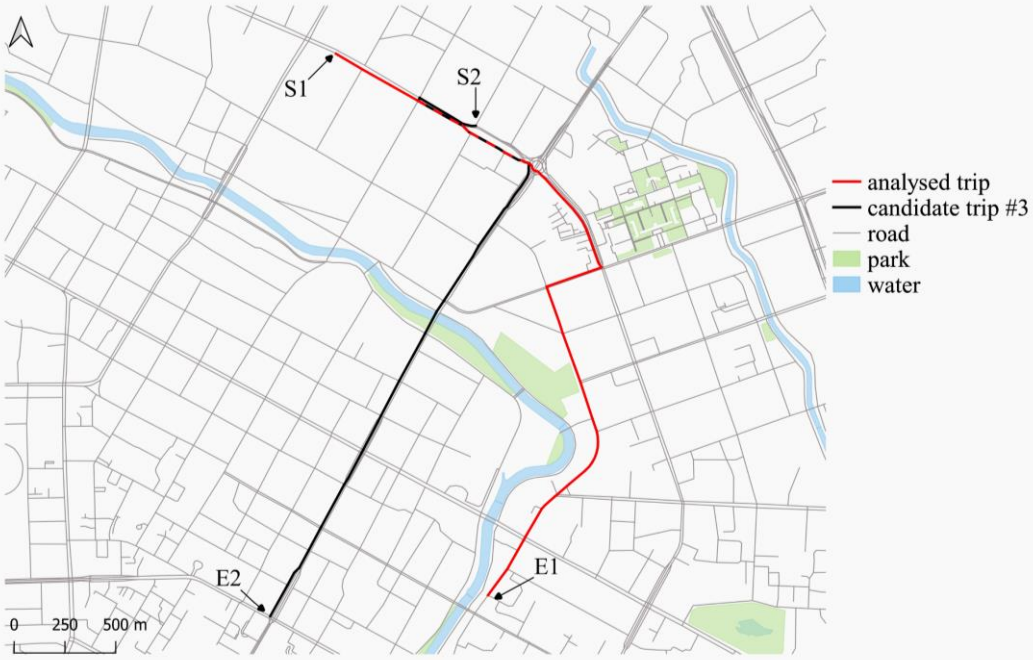


b)

Map-matched path of the analysed and the second most similar candidate trip



c) Map-matched path of the analysed and the third most similar candidate trip



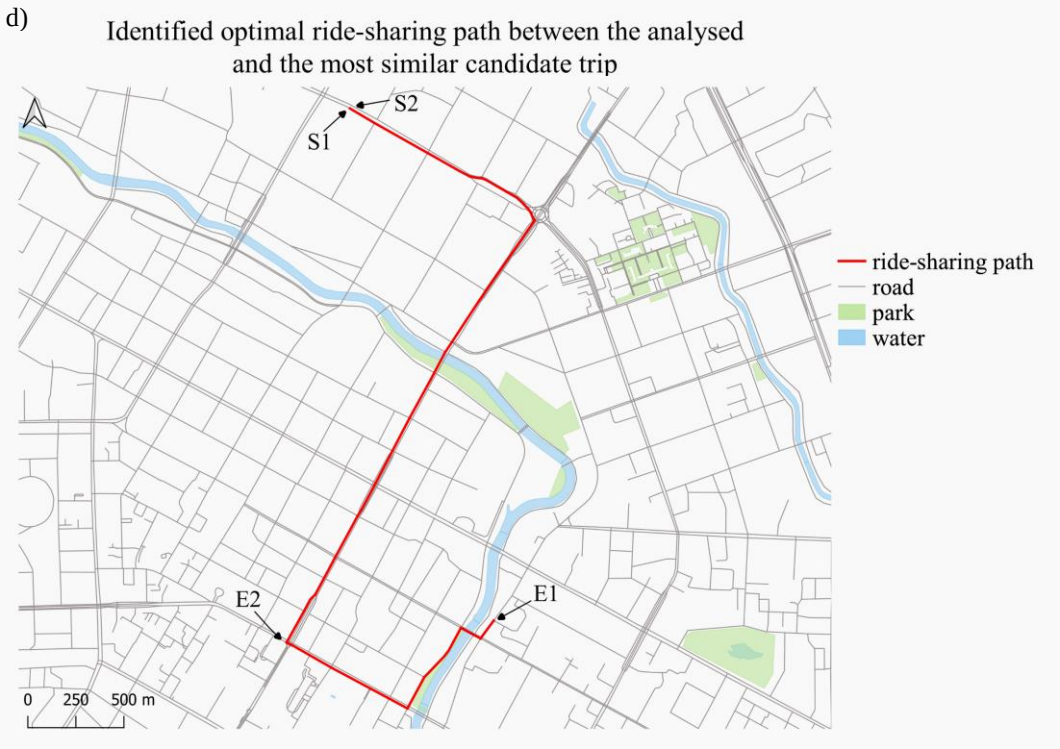


Figure 58: Visualisation of the three most similar candidate trips in a) to c) and the identified optimal ride-sharing path in d) for the analysed example trip. The final ride-sharing path of the fourth variation is a combination of the analysed with the most similar candidate trip.

Personal declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Zurich, 30th of September 2020 Christian Grass