Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 813

Paul Czioska

Meeting point locations for shared rides

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5220-8

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover
ISSN 0174-1454, Nr. 339, Hannover 2018

Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 813

Meeting point locations for shared rides

Von der Fakultät für Bauingenieurwesen und Geodäsie

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Vorgelegt von

M.Sc. Paul Czioska

Geboren am 10.12.1986 in Mainz

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5220-8

Diese Arbeit ist gleichzeitig veröffentlicht in:

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover

ISSN 0174-1454, Nr. 339, Hannover 2018

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759

E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender: Prof. Dr.-Ing. Winrich Voß

Referent: Prof. Dr.-Ing. habil. Monika Sester

Korreferenten: Prof. Dr. rer. pol. habil. Dirk C. Mattfeld (TU Braunschweig)
Prof. Dr.-Ing. Steffen Schön

Tag der mündlichen Prüfung: 12.01.2018

ISSN 0065-5325

ISBN 978-3-7696-5220-8

Abstract

Ride-sharing is a common way to utilize available vehicle seat capacity. It offers advantages for users, such as shared travel costs, and also provides benefits for the general traffic, for example less congestion, due to a reduced number of vehicles. Enhancing ride-sharing activity is therefore desirable, from a user and traffic management perspective.

To establish a shared ride, it is necessary for drivers and passengers to negotiate a meeting point for the pick-up (and the drop-off). Often, the doorstep of the passenger is chosen due to its simplicity, or well-known locations such as train stations. However, this may induce unnecessary driver detours through residential or downtown areas. In contrast, meeting points at convenient locations offer the potential for safe and comfortable waiting and boarding, and the detour for the driver can be reduced. Naturally, it requires passengers to walk a certain distance, or to use public transport.

While conventional ride-sharing and demand-responsive transportation systems have been explored thoroughly by the scientific community, less effort has been made to consider shared rides with meeting points. The primary goal of this work is, therefore, to close this gap, by using realistic meeting point locations from real-world map data. The impact, benefits and downsides of using meeting points for ride-sharing and demand-responsive transportation systems are investigated. The city of Braunschweig is used as a spatial template for various simulations. Furthermore, two user surveys were conducted to retrieve personal preferences about meeting points. The survey outcome provides a basis for investigations into how the safety and the convenience of meeting point locations influence intra-urban ride-sharing. In addition, the differences from a door-to-door based service are outlined.

Drivers offering a ride for long-distance (inter-urban) trips undoubtedly prefer meeting points in the vicinity of motorways and arterial roads for a pick-up en route. Such locations reduce driving time and mileage, since drivers do not have to traverse the city. Passengers, on the other hand, can use the public transport system to reach these locations. A location-based approach is presented in this thesis, to enable automatic real-time recommendation of meeting points for this purpose, using a GIS workflow and comprehensive precomputation of travel times.

Finally, a multi-stage workflow is presented, to determine suitable meeting points for demand-responsive transportation (SDRT) systems. First, the customers are grouped, and then appropriate meeting points are assigned to sub-groups. A simulation demonstrates the impact, in comparison with a conventional door-to-door service.

Zusammenfassung

Mitfahrgelegenheiten bieten sowohl für die Teilnehmer, als auch für das Gemeinwohl entscheidende Vorteile: Fahrtkosten können auf alle beteiligten Personen verteilt werden, und durch eine höhere Auslastung der Fahrzeuge wird dessen Anzahl reduziert. Die Förderung von Mitfahrgelegenheiten ist daher ein erstrebenswertes Ziel für Reisende und Verkehrszentralen.

Damit Fahrgemeinschaften zustande kommen, müssen sich Fahrer und Mitfahrer auf einen Treffpunkt (und ggf. Ausstiegspunkt) verständigen. Häufig ist dabei zu beobachten, dass die Mitfahrer der Einfachheit halber zu Hause oder an bekannten Punkten, beispielsweise an Bahnhöfen, abgeholt werden. Das bedeutet allerdings oft Umwege für die Fahrer, die durch Wohngebiete oder Einbahnstraßen fahren müssen. Ein gut gewählter Treffpunkt hingegen kann sowohl die Fahrzeit der Fahrer verringern, als auch einen sicheren und praktischen Ort zum Einsteigen bieten. Allerdings erfordert es von den Mitfahrern, zum Treffpunkt zu laufen oder den öffentlichen Nahverkehr zu nutzen.

Die Bestimmung und Auswirkungen von Treffpunkten für Mitfahrgelegenheiten und bedarfsgerechte Verkehrssysteme wurde in der wissenschaftlichen Literatur bisher nur wenig behandelt. Das Hauptaugenmerk dieser Arbeit liegt darauf, diese Lücke zu schließen. Die Stadt Braunschweig fungiert dabei als räumliche Vorlage für verschiedene Simulationen.

Zwei Nutzerumfragen bilden die Basis für Untersuchungen über die Eignung verschiedener Treffpunkte. Dabei fließen die Ergebnisse der Umfrage in eine Simulation von Mitfahrgelegenheiten ein, die neben rein zeitlichen Aspekten auch die persönlichen Präferenzen berücksichtigt. Zudem werden Unterschiede im Vergleich zu einer Abholung an der Haustür aufgezeigt.

Für Langstrecken-Fahrten, bei denen der Fahrer einen Mitfahrer aus einer Stadt entlang der geplanten Route abholen soll, liegt es nahe, einen Treffpunkt in der Nähe von Autobahnausfahrten zu wählen, der ebenfalls gut mit dem öffentlichen Nahverkehr erreichbar ist. Im Vergleich zu bekannten Punkten in der Stadt (z. B. dem Bahnhof) kann so Fahrzeit und -strecke vom Fahrer eingespart werden, da nicht erst die Stadt durchquert werden muss. In dieser Arbeit wird ein Algorithmus vorgestellt, um solche Punkte den Nutzern in Echtzeit vorschlagen zu können.

Zudem wird eine mehrstufige Methode präsentiert, um Treffpunkte für bedarfsgerechte Verkehrssysteme auszuwählen. Dazu werden die Benutzer zunächst gruppiert, und in einem zweiten Schritt werden den Gruppen Treffpunkte zugewiesen. Die Auswirkungen dieser Methode werden mit einer Simulation untersucht, die auch die Fahrzeugrouten berücksichtigt.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Mobility is a basic need for our society, and due to technical progress, people are able to move quickly and safely on a local and global scale. Not only standards of living, but also the global economy builds upon the agility of humans and goods. However, due to rapid urbanization and motorization worldwide, particularly in cities, we increasingly face the negative impacts of this development. Massive capacity overloads lead to congestion and overcrowded public transport systems. The energy needed to keep moving is continually growing, and since most road vehicles are equipped with combustion engines, they progressively pollute our environment. Already, outdoor air pollution kills more than three million people across the world every year, and triggers various health problems, from asthma to heart disease (OECD 2014).

In 2004, transport (passenger and freight) was responsible for 23 % of worldwide energy-related greenhouse gas (GHG) emissions (Ribeiro et al. 2007). Within this amount, standard passenger cars (also called light-duty vehicles) account for 44 % of total transport energy use, making it one of the major contributors to global warming (Ribeiro et al. 2007). Moreover, GHG emissions from the transportation section have increased at a faster rate than any other energy consuming sector. Emissions are even predicted to increase by about 80 % between 2007 and 2030 (Ribeiro et al. 2007). There is strong evidence within the scientific community that a massive reduction in GHG emissions is necessary to prevent serious climate destabilization.

A major reason for constantly-growing traffic is the high demand for individual mobility. Private vehicles still play a major role in satisfying this demand, leading to congestion and pollution. Low occupancy of private cars is one of the reasons for the large number of vehicles being driven on the streets. In Germany, car occupancy (number of people in a vehicle per trip, including the driver) averages out at 1.1 for daily commuting, and at 1.9 for leisure trips (Follmer et al. 2010). Altogether, the report observed an average private car occupancy rate of 1.5 in 2008, and almost two thirds of all private car trips were made alone. Additionally, the modal split of car passengers in Germany is about 15 % (2008) among all trips, and accounts for 24 % of kilometres travelled. In the US, the share of vehicles with more than one passenger is approximately 10.7 % (2012), and only 2 % of transportation trips to work are made with three or more participants (U.S. Department of Transportation 2015). Morency (2007) calculated, in several surveys conducted in the Greater Montreal Area between 1987 and 2003, that approximately 14 % of trips made on a typical weekday are made by car passengers (number of passenger trips over the total number of trips).

The solution: Share rides!

Part of the solution could be to increase the occupancy of vehicles by sharing rides, hence improving the efficiency of road transport by having a higher degree of utilization. Travellers with similar itineraries can travel together and thus reduce the amount of cars driving on the streets. There is much room for improvements: Bicocchi et al. (2015) showed based on a trajectory analysis in Italy that up to 60 % of rides could be saved, assuming a detour acceptance of 1 km. Ride-sharing further offers many advantages both for the individuals as well as for the society. However, the usage of ride-sharing has fluctuated much over time. In the US, the modal share of ride-sharing has, in general, declined over the last decades, with a large drop from 20.4 % in 1970 to only 10.1 % in 2004 (US Census Bureau 2007). According to Ferguson (1997), this is mainly due to decreasing petrol prices and a shift in social trends, such as individual attitudes towards driving alone. Teal (1987) states that a large majority of commuters is simply unmotivated to carpool, due to short commuting distances, a low cost burden and a relatively high vehicle availability. Hence, important motivating factors are vehicle inaccessibility, long trips, and a high commuting cost burden. In recent years, the trend of a declining modal share in the United States seems to be stopped: since 2004 (10.1 %) the value has risen again slightly to 10.7 % in 2008 (US Census Bureau 2007). However, ride-sharing is still a very important factor for sustainable commuting in the US, since there are as many as seven times more passenger miles for commuting trips by carpooling as there are for public transit (Chan & Shaheen 2012). This highlights the availability and quality of public transportation as a key factor influencing usage of ride-sharing (Teal 1987).

Benefits and downsides of ride-sharing

There are many advantages of sharing rides, both individual and societal. The main individual benefit of ride-sharing is undoubtedly the sharing of travel expenses, such as fuel costs or tolls. Also, special incentives such as permission to use HOV (High Occupancy Vehicle) lanes are an essential factor (Caulfield 2009). Carpoolers who commute regularly can often benefit from extended access to preferential parking and additional incentives (Chan & Shaheen 2012). In addition, ride-sharing offers the flexibility and speed of a car to travellers who do not have access to one, or where public transport is inefficient or too expensive. The society, in turn, benefits from reduced vehicle kilometres (which reduces congestion, and hence increases the average speed), savings on fuel, and reduced accidents and emissions (Fellows & Pitfield 2000).

The potential environmental savings are also worth to be mentioned. It is estimated that an increase of 10 % of vehicles sharing at least one seat could result in a saving of 5.4 % in annual fuel consumption (Caulfield 2009). Jacobson & King (2009) calculates that, if no detours are necessary, adding one person for every 100 vehicles could reduce the annual fuel consumption by about 0.8 billion gallons of petrol per year, and up to 7.7 billion gallons with one passenger added in every 10 vehicles. However, the increase in fuel consumption due to detours to pick up passengers also has to be considered, since it may eliminate potential savings. Note that ride-sharing can also result in an increase in vehicle kilometres, when drivers act as a taxi. In particular, in household-based ride-sharing, this happens frequently when parents drive their children to school or to various other activities, often replacing walking, biking and public transit as the preferred mode of transport for such trips. In her surveys, Morency (2007) concludes that around 15 % of shared ride trips are questionable, in the sense that they are exclusively generated for another's individual purpose.

In the last few decades, many online ride-sharing services have become popular for finding travel partners, often with a focus on a particular country or region. Furuhata et al. (2013) lists and classifies the characteristics of 39 matching agencies, and also states that the list is not exhaustive. A list of popular companies includes, for example, BlaBlaCar[1] (Europe), Kangaride[2] (Canada), CoSeats[3] (Australia) or CarPoolWorld[4] (US). Besides the larger companies, there are many regional providers, e.g. Pendlerservice Rhein-Main[5] or MatchRiderGO[6] in the greater Stuttgart region, which are often supported by local municipalities.

[1] http://www.blablacar.com
[2] http://www.kangaride.com
[3] http://www.coseats.com
[4] http://www.carpoolworld.com
[5] http://pendlerservice.ivm-rheinmain.de
[6] http://www.matchridergo.de

Shared demand-responsive transportation systems

A more commercial version of ride-sharing is a *demand-responsive transportation* (DRT) service, where an operator provides a complete door-to-door mobility solution based on own or licensed vehicles driven by employees or licensed freelancers. A DRT is usually operated by a company or statutory authority, then acting sometimes as part of the public transportation. Initially intended as a service with restricted usage (such as for the disabled or the elderly, also known as *paratransit*), it attracted generally more attention in recent years due to emerging mobility solutions and the shortcomings of conventional public transportation systems (Nelson et al. 2010; Navidi et al. 2016). The DRT trend has been further boosted by rapid developments in information and communication technologies in the last decade.

In contrast to conventional taxicabs, which accommodate usually only one customer (or a customer group) at a time, DRT systems generally focus on larger vehicles, so that multiple passengers can share the ride. In order to distinguish from single-customer DRT operators, such as Uber[7] or Lyft[8], the shared mode is often explicitly called a *shared* demand-responsive transportation (SDRT) system. The difference is that idle resources are utilized by combining several requests with similar itineraries and time schedules. As a result, a trip is often partly or completely shared with other travellers, and they have only limited control over the journey path. The advantages and disadvantages are quite similar to private ride-sharing: there are possible detours, but the service costs can be shared among all participants. Recently, many service providers have launched new SDRT services, including popular companies such as UberPOOL[9] and Lyft Line[10], and smaller local start-up companies, such as Bridj[11] (Boston, Kansas), Via[12] (New York, Chicago, Washington D.C), CleverShuttle[13] (Berlin, Leipzig, München), and Allygator[14] (Berlin).

1.2 Meeting points for shared rides

To establish a shared ride, travellers (hereinafter divided into drivers and passengers) need to negotiate a *meeting point*. While public transportation uses pre-defined and designated stops for boarding and deboarding procedures, there is nothing comparable for shared rides. A meeting point has to be selected for every ride, which is particularly difficult if the customers are not familiar with the environment. For long-distance ride-sharing in particular, this frequently results in meeting locations that are well-known, simple to describe, and easily reachable by public transport, e.g. the central train or bus station, or prominent landmarks in a city. While this is generally a reasonable choice, such locations are usually located in inner city districts, producing unnecessary detours and time loss for drivers. Here, meeting points close to motorways or arterial roads, and furthermore, easily reachable by public transport could reduce the driving time, driving distance and congestion in urban areas.

Also, most DRT services offer door-to-door transportation, where passengers are picked up and dropped off at their home, or their current location. However, this can lead to considerable detours, since the driver has to stop at many different places and take side roads. In addition, these are sometimes one-way streets, which further extend the trip. A meeting point in the vicinity of a customer's location could reduce the

[7] `http://www.uber.com`
[8] `http://www.lyft.com`
[9] `http://www.uber.com/de/ride/uberpool`
[10] `http://www.lyft.com/line`
[11] `http://www.bridj.com`
[12] `http://ridewithvia.com/`
[13] `http://clevershuttle.org/`
[14] `http://www.allygatorshuttle.com/`

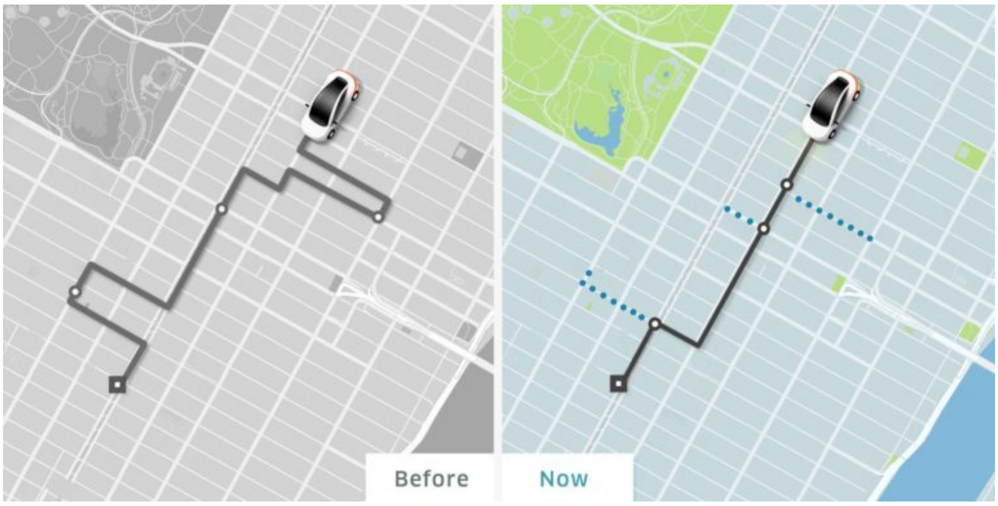


Figure 1.1: Meeting points at UBER Pool (Trivedi 2017). Left: Standard mode, right: new meeting point mode.

detours required. Also, passengers are, in principle, mostly willing to walk a certain distance in order to meet at a place where safe and convenient boarding can be established. In fact, some well-known SDRT service providers recently switched to a meeting point-based mode. Passengers are expected to walk to a meeting point recommended by the system, and after the ride, they are dropped off at another location close to their desired destination. UberPool, one of the big players in DRT services, stated in a blog entry in May 2017 that they realized that a short walk to meeting points could save people both time and money, so they have decided to adapt their algorithms to offer a more flexible and affordable carpooling product for Manhattan, including meeting points (Trivedi 2017). Figure 1.1 illustrates the route change; it can clearly be seen that necessary detours can be significantly decreased by using meeting points. Also, other service operators, such as Bridj or Via, offer pickup and drop-off at meeting-points. Sometimes, multiple passengers are grouped together and picked up (and dropped off) at the same location, hence reducing the number of stops and the necessary service time.

In summary, there are several reasons why meeting points can be advantageous for shared rides:

- **Safety and Convenience**: Meeting point locations can be chosen so that safe boarding is possible, without other traffic around and with sufficient parking space, so that luggage can be placed beneath the vehicle. In addition, meeting points can, as an option, be filtered according to certain facilities, such as seating possibilities, shelter or heating.
- **Identification**: By using well-defined meeting points, the driver and the passenger(s) know exactly where to go, and where to find each other. Meeting locations at the doorstep can be ambiguous, for example if there are several entrances to a building. Also, meeting points that are not well-defined can be problematic, e.g. if a multi-lane junction is negotiated, but the exact location is not clearly determined.
- **Service time**: The total service time (including boarding and de-boarding procedure) can be reduced for DRT services when more than one passenger can meet simultaneously, due to a reduction of the total amount of necessary stops.
- **Privacy**: The actual origin and destination of customers are not necessarily disclosed, or can be obfuscated through some techniques (Aïvodji et al. 2016; Goel et al. 2016).

– **Health**: The incorporation of walking into the daily transportation route can be seen as a contribution to a healthier and more sustainable urban transportation. Estimations show that 75% of US adults do not get enough physical activity in their daily lifes, which may lead to severe medical consequences (Ewing et al. 2003, 2014). Hence, sustainable urban planning that naturally includes or encourages walking into a daily transport plan can assist in creating healthier cities, reducing environmental and social risk factors (Giles-Corti et al. 2016).

– **Designation**: Feasible meeting point locations can optionally be designated by the service operator or the municipal traffic management, to make it an official boarding place with reserved boarding areas. In particular, in view of the emergence of autonomous vehicles, it can be beneficial to have locations at hand where a parking and boarding procedure is known to be achievable.

1.3 Research questions

The choice of appropriate meeting points is not a trivial task. Of course, there are simple solutions available in order to operate on a meeting-point base. UberPool, for example, states that simply the best nearby corner for a passenger is chosen as meeting point (Trivedi 2017). This algorithm is straightforward and quick, but it lacks some important properties, such as safety or easy identification. Furthermore, in the scientific community, the use of meeting points has not gained much attention, compared with conventional DRT or ride-sharing systems. Although there are some simulation studies that investigate the impact of meeting points, the actual determination of eligible meeting points in a real city environment is mostly neglected. Often, the Euclidean plane is used for simulation, or all vertices of the street network are considered as potential meeting point locations. In reality, however, safe and convenient meeting and divergence locations are not ubiquitous, since it may not be possible to stop at a junction or in the middle of a street. Moreover, feasible meeting point candidates, such as public parking areas, are usually unequally distributed within a city area, and not equally reachable by vehicles and pedestrians. Additionally, the road network may contain obstacles and one-way streets that require large detours to reach some meeting points. Hence, the impacts of these limitations need to be investigated.

The goal of this work is to close this gap in the current research, by transferring the meeting point problem more effectively into the real world, using meeting point locations and street networks from map data. Since the ride-sharing problem is mostly modelled as an optimization problem, with the goal of finding a good (or even optimal) matching of drivers and passengers, this work aims at extending the model in order to obtain driver-passenger matches at appropriate meeting points. In particular, the following research questions are tackled:

– Which properties and facilities are important for drivers and passengers concerning a meeting point, and how can personal preferences about meeting points be incorporated into the selection?

– What is the impact (benefits and downsides) of using real-world meeting points for intra-urban ride-sharing?

– How can appropriate meeting points be automatically recommended to ride-sharing customers, particularly for long-distance trips?

– What is the impact (benefits and downsides) of using real-world meeting points for intra-urban demand-responsive transportation (DRT) systems?

– How can meeting points be used by municipal traffic management?

In addition, the scalability of all proposed methods should be considered.

This work is organized as follows. First, the relevant fundamentals are explained in detail (chapter 2), before the current state of the art concerning meeting points is discussed (chapter 3). In chapter 4, two user surveys are presented, to clarify human needs and preferences regarding meeting points. Chapter 6 shows an experiment highlighting inner-city ride-sharing, followed by an algorithm and experiment focusing on inter-urban, long-distance ride-sharing (chapter 7). In chapter 8, meeting points for demand-responsive transportation systems are investigated. All experiments are based on a common data base, which is described in chapter 5. Finally, the results are summarized in the conclusion (chapter 9).

2 Fundamentals

This chapter provides an overview of relevant topics of importance for understanding this work, including basic fundamentals, problem settings and methods for mathematical optimization (Section 2.1), vehicle routing (Section 2.2) and ride-sharing (Section 2.3).

2.1 Mathematical optimization

Mathematical optimization (or simply *optimization*), aims at determining the variable constellation that yields the *best* solution for a given problem. Optimization is an important and basic mathematical framework, with many real-world applications in various fields, such as engineering, manufacturing, economics, transportation, scheduling and many more.

A standard optimization problem consists of an *objective function* $f(X)$, yielding an output value that should be minimized or maximized, a set of *decision variables* $X = \{x_1, x_2, \cdots\}$ as input of the objective function, and, optionally, a set of constraints, limiting the valid solution space. There are many different types of optimization problems, depending on the constraints and characteristics of the objective function:

- Unbounded (no constraints) or bounded
- One or multiple decision variables
- Discrete or continuous decision variables
- Linear or non-linear objective function
- Static (no changes over time) or dynamic
- Deterministic or stochastic (randomness involved)

In this chapter, various problems and concepts of optimization relevant to this work are introduced. Firstly, Section 2.1.1 provides a short abstract of *linear programming (LP)*, covering the special case having continuous decision variables, a linear objective function and linear constraints. If the decision variables are forced to be integer, and hence discrete, the problem is called *integer programming (IP)*, as explained in Section 2.1.2. Furthermore, if the decision variables are discrete and there are constraints that limit the solution space, then the problem has a finite number of solutions. This special case is known as *combinatorial optimization*, which is described in Section 2.1.3 including typical problems and solutions.

2.1.1 Linear Programming

A linear program is, in general, an optimization problem consisting of a linear objective function that should be either minimized or maximized, and a set of linear constraints. The solution space forms a *polyhedron*, which has a convex shape (Korte & Vygen 2012). The *optimum* is the solution for which the value of the objective function is the best, and all constraints are satisfied.

A simple example of a linear optimization problem, taken from Google OR Tools[1], is given as follows:

$$\max (3x + 4y) \tag{2.1}$$

subject to

$$x + 2y \leq 14 \tag{2.2}$$

$$3x - y \geq 0 \tag{2.3}$$

$$x - y \leq 2 \tag{2.4}$$

The constraints define the feasible solution space (polyhedron) for the problem, which is visualized in Figure 2.1. The optimal solution for a linear optimization problem is proven to be at one of the corner points of the polyhedron (Korte & Vygen 2012). A well-known algorithm to find the optimal values efficiently is the *simplex* algorithm, originally developed by Georg Dantzig in 1947, where feasible solutions on the boundary of the polyhedron are explored subsequently until the optimal corner point is found (Bradley et al. 1977). However, the optimal solution to a linear program is not always unique, because the objective function may be parallel to one of the constraints, so that a range of values becomes optimal. In the example above, the optimal point is at (6, 4), a corner point of the polyhedron (Figure 2.1).

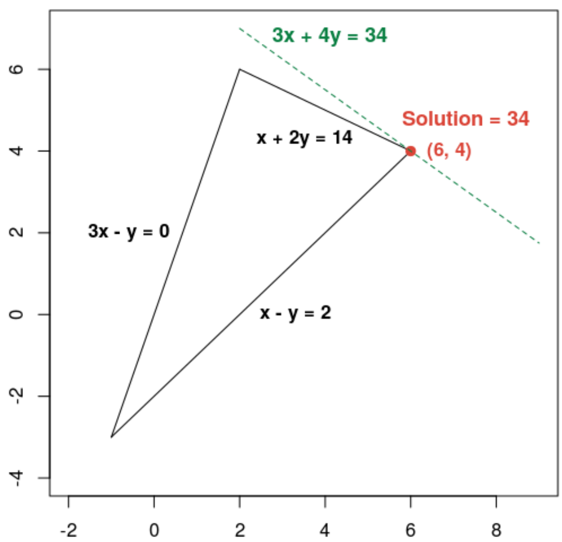


Figure 2.1: Optimization problem with the feasible region, valid integer candidates and the optimum for the Linear Programming and Integer Programming case. Source: Google OR Tools[1]

The constraints can be further divided into *hard* and *soft* constraints. Hard constraints are compulsory to be satisfied, hence forming the solution space. Soft constraints are, in contrast, not modelled as real

[1] `https://developers.google.com/optimization/introduction/using`, accessed 11.10.2017

constraints, but are part of the objective function as a penalty term. If very high penalty terms are used as soft constraints, the effect is similar to a hard constraint, but with the theoretical possibility of violating the constraint if there is no other choice.

If the objective function contains nonlinear parts, the problem of finding an optimal solution becomes much more difficult, since the optimal solution is no longer proven to be at a corner point of the feasible region (Bradley et al. 1977). The problem is then called *Nonlinear programming*. A well-known early heuristic to find a (local) minimum or maximum of a nonlinear objective function is the *Nelder-Mead method*, which can be regarded as a Hill Climbing (or downhill) method based on the simplex principle (Nelder & Mead 1965).

2.1.2 Integer Programming

While the solution space in linear programming is continuous (i.e. fractional values are allowed), *Integer Programming (IP)* has to be applied when the solutions to the decision variables are forced to be integer values. If only some variables are restricted to being integer, the problem is called *Mixed-Integer Programming (MIP)*. In contrast to linear programming, IP is much harder to solve and known to be NP-hard (see Section 2.1.3). Hence, the size of problems that can be solved successfully is usually limited and much smaller than for LP (Bradley et al. 1977). A special case of IP uses only binary variables (usually 0 and 1), which can be useful for yes-no decisions. A common application area of binary Integer Programming is to solve combinatorial optimization problems (Section 2.1.3), e.g. by introducing a binary decision variable for each possible solution, and then adding several constraints such that the correct combination of true values is required.

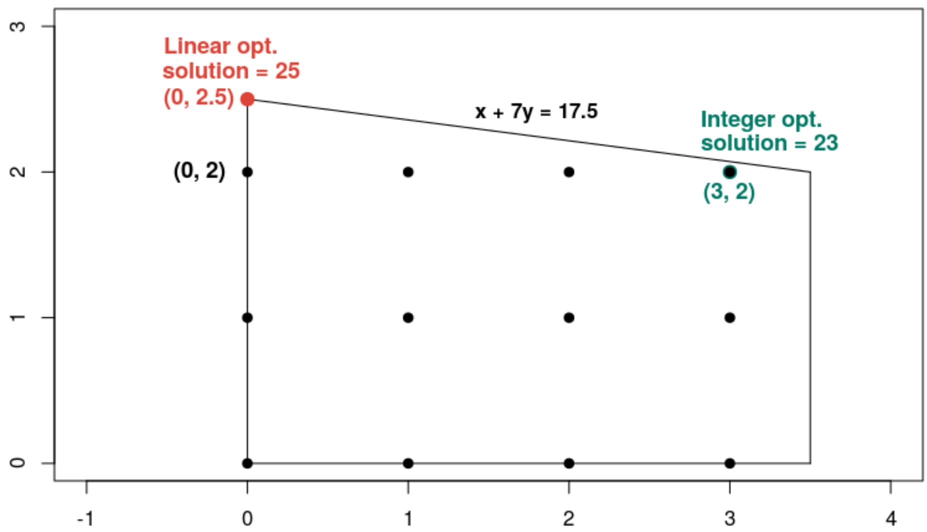


Figure 2.2: Optimization problem with the feasible region, valid integer candidates and the optimum for the Linear Programming and Integer Programming case. Source: Google OR Tools[2]

If an IP instance is bounded (i.e. constraints in every direction of the search space), the set of possible solutions becomes countable, hence it is also a *combinatorial optimization problem*, which is explained in more detail in Section 2.1.3. Figure 2.2 shows exemplarily the feasible region of a linear optimization problem containing all valid integer candidates. In this special case, the optimal linear programming solution differs clearly from the optimal integer programming solution.

A usual way of solving IP problems is by applying a *linear relaxation*, meaning that the problem is first transformed into a non-integer version that can then be solved to optimality. In the best case, the result is already an Integer solution. If not, the solution space can at least be shrinked by introducing a *cut* (linear inequality). The search is then continued in the limited search space until a valid integer solution has been found (Kelley 1960). Such algorithms based on a cutting of the search space are generally known as *Cutting Plane Methods*.

2.1.3 Combinatorial Optimization

Combinatorial optimization is the umbrella term for finding the optimal or a near-optimal solution from a finite set of solutions for a given problem. It has its roots in combinatorics, operations research, applied mathematics and theoretical computer science. Many problems that arise in real-world situations can be formulated as combinatorial optimization problems and solved with methods that have been developed in this scope, such as the shortest path problem or the well-known travelling salesman problem (TSP), but also as assignment or bin packing problems.

In theory, a combinatorial optimization problem can be solved very straightforwardly by enumeration (also known as *brute force*): given a set of possible solutions, try out all of them, compute the cost for each, and select the one with the best value. In reality, however, the problem is the complexity of the algorithm, describing how much the running time of the algorithm grows depending on the size of data input. While for smaller instances enumeration is a reasonable choice, the algorithm running time is often the limiting factor for larger instances. As an example, in the well-known travelling salesman problem (see Section 2.2.1) the number of possible paths grows with $n!$ depending on the number of input points n. Already for $n = 20$ there are $20! = 2\,432\,902\,008\,176\,640\,000 \approx 2.4 \cdot 10^{18}$ different paths to investigate, so that even the fastest computers would need several years to complete (Korte & Vygen 2012). The class of problems which are currently not able to be solved in polynomial time is called NP (*nondeterministic polynomial time*). All presented problems can be divided into the classes of P (polynomial-time) and NP.

The algorithms used to solve combinatorial optimization problems can roughly be divided into three categories: *exact* algorithms, *approximation* algorithms and *heuristics*. While exact methods are able to figure out the optimal solution for a given problem (sometimes also called *global minimum*), approximation algorithms are used to find near-optimal solutions within polynomial time. Often, a certain percentage can be provided that guarantees how close the solution will be to the optimum (Korte & Vygen 2012). A heuristic also returns non-optimal solutions, but the focus is more on fast running times than on proving a certain solution quality. Since the size of NP problems that can be solved to optimality is naturally limited, approximation algorithms or heuristics must be used for large instances, when the algorithm should finish in a reasonable time.

There are several approaches to solving combinatorial optimization problems exactly. Often, combinatorial problems can be formulated as an Integer Programming problem (Section 2.1.2) or modelled as a graph. One common way is to apply cutting plane methods, which are frequently used for Integer Programming (see Section 2.1.2). Another popular class of methods involves *Branch-and-Bound* approaches. The idea behind

[2] `https://developers.google.com/optimization/mip/integer_opt`, accessed 11.10.2017

this principle is to enumerate all solutions and build a tree, with all solutions at the root. The algorithm then explores the branches of the tree, but before the solutions of a branch are investigated, the solution candidates are checked against lower and upper bounds. If the branch is not able to deliver a solution better than the current optimum value, it can be discarded (Lawler & Wood 1966). As a consequence, the efficiency of Branch-and-Bound algorithms depends heavily on fast estimation of lower and upper bounds for a solution subset. The branch-and-bound method can also be combined with the cutting plane principle, which is consequently named *Branch-and-Cut*.

However, as stated above, if the instances are becoming very large, approximation algorithms or heuristics are necessary to solve the problems in reasonable time. Besides problem-specific heuristics, a number of metaheuristics can be applied, which are feasible for solving a large variety of problems. Popular metaheuristics are, for example:

- *Hill Climbing* or the *Greedy Algorithm* (Start at a random place in the solution space and move towards the direction of best improvements; fast but prone to finding local optima)
- *Simulated Annealing* (Neighbourhood search with slowly decreasing probability of moving to more remote solutions)
- *Tabu Search* (Hill Climbing with restricted movements and the possibility of escaping local optima by performing worsening movements)
- *Ant Colony Optimization* (inspired by the foraging behaviour of real ants, this algorithm is based on swarm intelligence, where many individuals with limited knowledge are able to find global good solutions)

Of course, there are many more algorithms available which cannot be enumerated in this work. The reader is here referred to the work by Gendreau & Potvin (2010).

2.1.4 Dynamic Programming

Another approach to solving combinatorial optimization problems to optimality is *Dynamic Programming*. This theory is based on the *Bellmann equation*, stating that an optimal solution can be assembled from a set of optimal sub-solutions (Bellman 2013). In a nutshell, a complex problem is divided into smaller and simpler sub-problems, which can either be solved easily, or, if not, are split into smaller sub-problems. The basic idea is that the smaller the problems, the less time is necessary to solve them. Each time a solution for a sub-problem is found, the part is added to the partial solution, until finally the optimal solution can be assembled by combining all sub-problems. The partial solutions are stored, and can simply be reused if the same sub-problem occurs again. Dynamic programming hence builds a framework for analysing and solving a variety of problems. Following the formulation of Bradley et al. (1977), there are three main characteristics:

1. *Stages* The original problem is divided into multiple stages, which are solved sequentially as an ordinary optimization problem.
2. *States* Associated with each stage is the *state* of the process, reflecting the information used to assess the consequences of a certain decision.
3. *Recursive Optimization* This characteristic describes the approach of solving single one-stage problems and including one stage at a time.

Due to the breakdown principle, Dynamic Programming is often implemented in a recursive fashion. Furthermore, many routing algorithms, such as shortest path algorithms, are actually using the Dynamic Programming principle.

2.1.5 Set cover problem

A relevant and well-known sub-problem of combinatorial optimization is the so-called *set cover problem*, which has also been proven to be NP-hard (Karp 1972). The set cover problem can be described as follows. A set of elements $A = \{a_1,a_2,\ldots,a_n\}$ and multiple subsets $S_1,S_2\ldots,S_k \subseteq A$ are given, optionally each with an associated cost $c_1,c_2,\ldots,c_k$. The goal is to find a combination of subsets S such that all elements of A are contained in the union of the selected subsets. While, in the unweighted version, the main target is to minimise the number of selected subsets, the weighted version aims at minimising the costs c associated with each selected subset S.

The set cover problem has a variety of relevant real-world applications. A popular example is a company that needs a certain amount of different supplies, and various vendors that offer some supplies for a specific cost. Here, the set cover problem can be applied to figure out the best combination of vendors to minimize the cost.

Since the problem is very relevant to real-world applications, it is not surprising that a vast number of solution techniques exist. In the survey by Caprara et al. (2000), a large number of exact and heuristic approaches is provided, often applying linear programming or metaheuristics. A simple but well-known approach is the Greedy method, where the largest (or most cost-effective) subset is always chosen, until all elements are covered. This approximation algorithm does not deliver the optimum, but it can be shown that it achieves a certain quality of result (Chvatal 1979).

2.2 Vehicle routing problems (VRP)

The vehicle routing problem (VRP) can be regarded as one of the most extensively studied real-world combinatorial optimization problems (see Section 2.1.3). It asks for the optimal set of routes that a fleet of vehicles should take to serve a given set of customers. The problem was introduced and mathematically defined by Dantzig & Ramser (1959) by describing a real-world application on how to deliver petrol to service stations. A few years later, Clarke & Wright (1964) improved the approach by introducing a greedy heuristic called the *savings algorithm*. Since then, a large number of models and algorithms have been proposed for all different variations of the problem.

Probably the most common use case for vehicle routing problems is the distribution of *goods* between one or more *depots* and *customers*. The goods have to be collected from or delivered to the customers, and more constraints can be added, in terms of time and size restrictions, or specifying that certain goods have to be collected before they can be delivered. Typical applications in the real world are parcel delivery services, street cleaning, school bus routing, dial-a-ride systems, routing of salespeople and many more.

In order to transport the goods, a set of *vehicles*, initially located in the depots and operated by a set of *drivers*, has to travel to the customers using an appropriate *street network*. The goal of the VRP is to determine a set of *routes* using the given network, each performed by a single vehicle, such that all operational *constraints* are satisfied and the global *transportation costs* (sum of all route costs) are minimized.

2.2.1 The basic VRP

As noted by Toth & Vigo (2002), the basic VRP can be modelled as follows. Let $G = (V,E)$ be a graph representing the network of customers in a city. V is the vertex set, corresponding to the customers and the depot(s) (not the actual street network junctions). E is the edge set, representing the connections between the vertices. Each edge can be defined by its origin node i and the destination node j. All edges $(i,j) \in E$ are further associated with a non-negative *cost* c_{ij}, representing the travel costs between origin and destination node. The cost is usually the distance or travel time, which can be dependent on the time of traverse due to congestion or other obstructions (Demiryurek et al. 2010). The graph can further be *directed* or *undirected*, depending on whether the edges can be traversed only in one direction or bidirectionally. If G is undirected, $c_{ij} = c_{ji} \; \forall \, (i,j) \in E$ and the corresponding problem is called *symmetric VRP*, whereas if the graph is directed, the problem is referred to as *asymmetric VRP*. In the basic case, the graph G is complete, i.e. all vertices are connected with each other. In addition, some algorithms require that the graph should fulfil the *triangle inequality*, that states that a way via a third point can never be shorter than the direct way between two points:

$$c_{ik} + c_{kj} \geq c_{ij} \; \forall \, i,j,k \in E. \tag{2.5}$$

In the special case of vertices having coordinates on a plane and the edges connecting the vertices on the straight line, the resulting problem is called the *Euclidean VRP*, which implicitly satisfies the triangle inequality, and the *cost matrix* C is symmetric. Let K be further a set of (identical) vehicles available to serve the customers. In some constrained versions of the VRP, it is important that the size of K is feasible for serving all requests.

Following Laporte (1992), the VRP finally consists of designing a set of least-cost vehicle routes K in such a way that

1. each customer $i \in V$ is visited exactly once by exactly one vehicle;
2. all vehicle routes start and end at the depot and
3. all side constraints are satisfied.

Figure 2.3 visualizes, as an example, a VRP instance with three routes (circuits) which are sufficient to serve all customers.

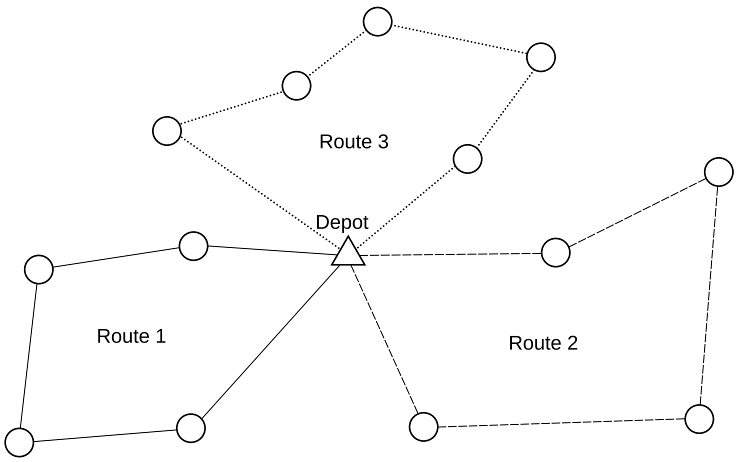


Figure 2.3: Vehicle Routing Problem with three routes.

The special case of $K = 1$ defines the well-known travelling salesman problem (TSP), that calls for a simple circuit visiting all customers once (*Hamiltonian circuit*). Thus, the VRP generalizes the TSP, which is known to be NP-hard. Likewise, the VRP and all its derivations are NP-hard (Toth & Vigo 2002).

According to Toth & Vigo (2002), the mathematic modelling of the VRP can be done in three different ways:

– **Vehicle flow formulation**
  The idea of the vehicle flow formulation is to insert an integer variable for every edge, indicating how often the edge is traversed by a vehicle. Additional flow constraints ensure that the edges can be formed to valid vehicle circuits. This is referred to as the standard model of the basic VRP.

– **Commodity flow formulation**
  In this type of model, the flow of goods is represented by integer variables instead of the flow of vehicles. Of course, if commodities are transported along an edge, it automatically implies that a vehicle must also drive there.

– **Set cover problem**
  If the VRP is modelled as a set cover problem (see Section 2.1.5), a binary decision variable is inserted for every feasible circuit. The arising combinatorial optimization problem aims for the optimal selection of circuits which cover all customers with minimum cost, while satisfying all constraints. The advantage of this method is that all types of route costs can be considered, since the whole circuit is known. The drawback is that an exponential number of variables is necessary to model all possible circuits, hence it is computationally very expensive to obtain optimal solutions.

2.2.1.1 Variations

Traditionally, the objective of the VRP is to minimize the global cost of all traversed edges, e.g. the total distance or total time. However, there are often other objectives that can be considered, such as minimization of the number of vehicles, balancing of the routes or the vehicle load, or minimization of certain penalties that apply. Furthermore, there are many derivations of the VRP defined by introducing different side constraints. According to Toth & Vigo (2002), the most common variants include:

– **Capacitated VRP (CVRP)**
  In the capacitated version of the VRP, a non-negative weight (or demand) d_i is attached to each customer (not the depot). A side constraint is introduced, such that no vehicle route K exceeds the vehicle capacity by the sum of the visited weights. A practical implication would be a bus that picks up passengers and has a maximum seating capacity. The constraint can also relate to other occurrences - e.g. a limitation of the maximum length (or time) of a vehicle tour, which is then called *Distance-Constrained VRP* (Toth & Vigo 2002) or DVRP (Laporte 1992).

– **VRP with time windows (VRPTW)**
  In this variant, each customer must be visited within a time interval, referred to as *time window*. Also, the depot can have a time window, which then limits the vehicle driving period allowed. If the vehicle arrives at a customer before the allowed time interval, waiting is usually allowed.

– **VRP with precedence relations**
  Precedence constraints are inserted between pairs of customers, indicating that customer i must be visited before customer j. A variant is the *VRP with Backhauls (VRPB)*, where the customers are

divided into two subsets, the *linehaul customers* L and the *backhaul customers* B. The goal is to create the routes such that, in each circuit, the linehaul customers precede all backhaul customers.

– **VRP with pickup and delivery (VRPPD)**
  In the VRPPD, each customer is assigned two quantities of the same commodities: a demand d_i that has to be delivered and a demand p_i that has to be picked up. A variant is to use a single commodity notation $d_i = d_i - p_i$, which can then be negative. Generally it is assumed that the delivery is performed before the pick-up, which can be important if vehicle capacity constraints are in place. Furthermore, precedence constraints are often imposed to model the transportation of a certain good between the pick-up and its associated drop-off stop (Desaulniers et al. 2002).

Figure 2.4 shows a classification of various vehicle routing problems. All variants can be combined and equipped with additional side constraints. A relevant version is the *capacitated VRP with pick-up and deliveries and time windows (CVRPPDTW)*. This is the basis for the so-called *Dial-a-Ride-Problem* (DARP), which is explained in more detail in section 2.2.2. The difference from the CVRPPDTW is that a human perspective has to be considered. If private vehicles replace the official vehicles of a transportation operator, it is called a *ride-sharing problem* (see Section 2.3).

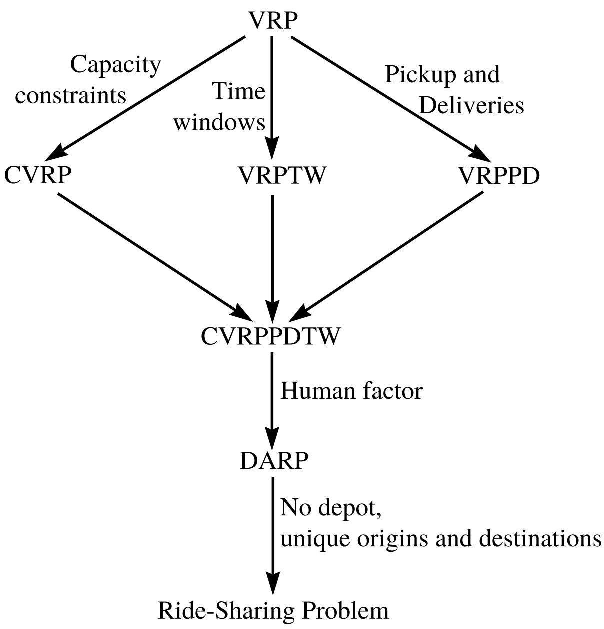


Figure 2.4: Classification of various vehicle routing problems, including ride-sharing.

2.2.1.2 Methods

Since the VRP is a very fundamental problem, with many variations and real-world applications, it is not surprising that many publications are available dealing with this problem class. In particular, enhancements in computational power have enabled the operations research community to also solve larger combinatorial optimization problems. Hence, the number of articles has begun to increase significantly since the 1980s and 1990s, and the number of articles continues to growing rapidly. According to a meta-study by Eksioglu et al. (2009), in the time period between 2000 and 2006 alone, more than 400 VRP articles were published in peer-reviewed journals. Due to the vast amount of literature, only a small fraction can be presented here.

Clarke & Wright (1964) were probably the first to incorporate more than one vehicle into the VRP formulation, improving the approach by Dantzig & Ramser (1959). Since then, many models and exact heuristics have been proposed for different versions of the VRP. Comprehensive, but slightly older reviews about various exact and heuristic approaches have been carried out by, for example, Laporte (1992); Christofides et al. (1981); Laporte & Nobert (1987); Laporte et al. (2000); a more recent study of the latest advances was published by Golden et al. (2008).

Since the general problem is so extensive, many reviews focus more on sub-problems of the VRP. For the VRP with time windows, several works that present, compare and review heuristics and exact methods are available, e.g. by Solomon (1987) and Desrochers et al. (1992); and more recently by Cordeau et al. (2002) and Baldacci et al. (2012). The same holds for the VRP with pick-up and deliveries (Savelsbergh & Sol 1995; Desaulniers et al. 2002; Nagy & Salhi 2005; Berbeglia et al. 2007).

Also, the dynamic case has gained much attention as a special focus. The first dynamic VRP studies date back to Powell (1986), having uncertain demand for vehicles to carry between the locations. Since then, many different approaches have been published to tackle the dynamic VRP (Pillac et al. 2013).

From a methodological point of view, a very popular approach is to use (meta-)heuristics employing local search methods such as simulated annealing (Osman 1993; Czech & Czarnas 2002), Tabu search (Gendreau et al. 1994; Taillard et al. 1997; Renaud et al. 1996; Cordeau et al. 1997) or ant colony optimization (Bell & McMullen 2004).

2.2.2 Dial-a-ride problem (DARP)

The dial-a-ride problem (DARP) calls for the creation of an efficient route plan to satisfy a set of customer transport requests. Each request consists of a set of human users, intending to travel from an origin to a destination within associated pick-up and drop-off time windows. In the basic version of the problem, a fleet of identical vehicles, equipped with a limited number of seats and based at a single depot, is used to accommodate the demand. The goal is to find vehicle routes that minimize the costs while, at the same time, handling as many requests as possible. The problem can, however, be much more complex, with several depots, a heterogeneous vehicle fleet, and special constraints such as buses designated only for wheelchairs. A service that offers a dial-a-ride service is also known as *Demand-responsive transportation* (DRT) system.

Historically, the DARP was mainly designed focusing on door-to-door transportation services for the eldery or disabled people. Early examples are the Telebus service Berlin (Borndörfer et al. 1999) or the Copenhagen Fire-Fighting Service (Madsen et al. 1995). A dial-a-ride system can be operated by companies or statutory authorities. In recent years, however, it has also attracted more attention for non-handicapped persons, due to emerging new mobility solutions and the shortcomings of conventional public transportation systems (Nelson et al. 2010; Kashani et al. 2016). The trend towards flexible dial-a-ride services for everybody has been further boosted by rapid developments in information and communication technologies which help to process requests and assign vehicles automatically.

From a technical perspective, the DARP generalizes many vehicle routing problems, namely Capacitated VRP, VRP with pick-up and deliveries and VRP with time windows (see Section 2.2.1.1). For the DARP, the customers have to be picked up during a time period and then dropped off at another node, ensuring compliance with time and capacity (seating) constraints. The basic difference from vehicle routing problems is the human perspective, since not only operating costs but also user inconvenience should be reduced

(Cordeau & Laporte 2003a). For example, it is not possible to transport a customer for hours through the city, just because it is more optimal in terms of mileage.

Dial-a-ride services can be offered in a *static* mode, where all requests are known in advance, or in a *dynamic* mode, where requests can appear ad hoc throughout the day, so that the vehicle routes have to be adjusted in real time. However, in most practical applications it will be a mixture of both modes, because at least a subset is often known in advance (Cordeau & Laporte 2003a).

Common objectives for the DARP are on the one hand minimizing costs and on the other hand maximizing satisfaction, subject to side constraints (Cordeau & Laporte 2003a). Hence, the operational cost parameters, such as total mileage, fleet size and driver wages have to be balanced with *quality of service* criteria, including customer waiting time, customer riding time and customer late arrival time. The criteria can be modelled as hard or soft constraints. A review of mathematical models can be found in Cordeau & Laporte (2007).

2.2.2.1 Variations

Following Cordeau & Laporte (2007), the DARP can be divided into four cases:

- Static single-vehicle DARP
- Dynamic single-vehicle DARP
- Static multi-vehicle DARP
- Dynamic multi-vehicle DARP

Furthermore, the DARP can be distinguished into the heterogeneous DARP (H-DARP) with non-uniform seating capabilities of the vehicles, or the Multi-Depot DARP (MD-DARP), which can also be combined (MD-H-DARP, see e.g. Braekers et al. (2014)). Since the DARP generalizes the vehicle routing problem, it is also NP-hard. As such, an exact solution can only be determined for small instances, or with very large computational effort. The alternative is to use heuristics to reach near-optimal solutions in reasonable processing times.

2.2.2.2 Methods

Since the dial-a-ride problem is in essence a generalization of the vehicle routing problem, the solution approaches are very similar.

Generally, the DARP can be divided into the assignment of customers to vehicles and the routing of vehicles. If only one vehicle is available, the problem complexity is significantly reduced, since the assignment can be neglected. One of the first to tackle this problem was Psaraftis (1980), who formulated and solved the problem with a dynamic programming approach. The objective function is a combination of vehicle travel time and customer satisfaction, with satisfaction defined as a weighted sum of waiting and riding time. The solution is optimal, but, typically for exact approaches, the size of the instances is limited (see Section 2.1.3). The approach was later extended to include time windows (Psaraftis 1983). In addition, dynamic requests were also considered. Later, Desrosiers et al. (1986) solved the problem as an integer program using dynamic programming with up to 40 customers.

Having multiple vehicles available to handle the demand, the problem becomes more complex. In the last few decades, numerous solution techniques have been developed to tackle the problem. While few works focus on exact solutions using methods like branch-and-cut (Cordeau 2006; Braekers et al. 2014) or graph conversion (Qian et al. 2017), the majority of the available literature applies heuristics to derive near-optimal solutions in reasonable time. Numerous different techniques have been used to solve the DARP, and sometimes the authors have combined them into multi-stage approaches.

A popular basic approach is the *insertion technique*, also known as *solution construction*, where requests are gradually or dynamically inserted into vehicle routes. Jaw et al. (1986) was one of the first to propose this technique, improved later by Madsen et al. (1995), who solved a real-life dynamic problem based on the transportation of disabled people in Copenhagen. More recently, new insertion-based algorithms have been developed (Diana & Dessouky 2004; Lu & Dessouky 2006; Wong & Bell 2006). For very dynamic problems with unexpected customers, Coslovich et al. (2006) has presented a two-stage dynamic insertion heuristic that is supposed to run in real time.

Another common approach to handling the scalability issue is to separate the problem into an assignment problem and a single-vehicle routing problem, also known as *cluster-first, route-second.* The idea has been applied by, for example, Bodin & Sexton (1986), who first clustered the customers, and then applied a single-vehicle routing algorithm to each cluster. In addition, swaps between the clusters were made to avoid possible local optima. Following this principle, several improvements have been proposed, such as approximating mini-clusters (Ioachim et al. 1995) or using genetic algorithms (Jorgensen et al. 2007).

Furthermore, particularly in the last decade, approaches based on neighbourhood search in various forms have become very popular for tackling the DARP. This includes Tabu search (Cordeau & Laporte 2003b; Kirchler & Calvo 2013), sometimes combined with other techniques such as constraint programming (Berbeglia et al. 2012). Parragh et al. (2010) propose a variable neighbourhood search heuristic, initially published as a two-step approach including a path relinking phase to determine alternative solutions (Parragh et al. 2009). Another hybrid algorithm by Parragh & Schmid (2013) uses large neighbourhood search (in the solution space) and column generation to reduce the scale of the optimisation problem. In addition, metaheuristics such as deterministic annealing have recently been applied to solve the DARP (Braekers et al. 2014).

2.3 Ride-Sharing

According to the definition by Furuhata et al. (2013), ride-sharing is a mode of transportation in which individual travellers share a vehicle for a trip. In essence, a driver offers spare seating capabilities in private (or organizational) vehicles to passengers with similar itineraries and time schedules (Agatz et al. 2011). Due to vehicle sharing, travel costs such as fuel, toll or parking fees can be distributed among the participants, making ride-sharing a very low-priced way of travelling, with the flexibility and the relatively high speed of private cars (Furuhata et al. 2013). The aim of the *ride-sharing problem* is to coordinate driver and passenger demand to achieve a particular goal, e.g. to maximize the matchings or distance savings, subject to various constraints such as travel time or vehicle capacity limitations. Ride-sharing differs from commercial taxicabs in terms of the financial motivation. While the ride-sharing payment is used to partially cover the expenses of the driver, it is not intended to earn substantial profit (Chan & Shaheen 2012).

The term *ride-sharing* is used very differently in the literature and within different societies. In the UK, it is widely known as *lift-sharing* and *car-sharing*, but this should not be confused with car-sharing in Europe and North America, which mostly refers to short-term rental cars (Chan & Shaheen 2012). Ride-sharing is also often interchanged with *carpooling*, which mostly refers to regular ride-sharing without unexpected changes of schedule. Carpooling is often used among co-workers for daily commuting (Ferguson 1997).

Differentiation

Chan & Shaheen (2012) suggest a categorization of ride-sharing as follows:

- *Acquaintance-based*, typically arranged among families, friends and co-workers without any third-party organization

- *Organization-based*, referring to ride-sharing that requires participants to join a service, either through formal membership (e.g. company carpooling) or by visiting a website (e.g. BlaBlaCar)

- *Ad hoc*, realized through self-organized casual ride-sharing without prearrangement or fixed schedules

Similarly, Furuhata et al. (2013) classifies ride-sharing into *unorganized* ride-sharing (covering the acquaintance-based and ad hoc versions) and *organized* ride-sharing, referring to the organization-based variant.

Ad hoc ride-sharing is also known as *flexible carpooling* or *slugging*. The shared rides are formed spontaneously at predetermined locations on a first-come-first-served basis (Furuhata et al. 2013). Often, users of flexible ride-sharing can benefit from reduced tolls or dedicated High-Occupancy Vehicle (HOV) lanes, which is a common incentive in some regions, especially in the US (Spielberg & Shapiro 2000; Burris & Winn 2006). An advantage of an ad hoc ride-sharing systems is that it is very simple and does not need any centralized ride-matching platform or organization. However, it requires a sufficiently large community of participants to work well, because otherwise long waiting times can occur. In addition, mixed forms of the different ride-sharing variations can be found. An example is the recently launched platform MatchRiderGO[3], offering organization-based ad hoc ride-sharing with fixed routes and timetables in the region of Stuttgart.

Note that the term *ad hoc* is used ambiguously in the literature. While Chan & Shaheen (2012) denominate ad hoc ride-sharing as flexible carpooling, it is often used as time-flexible ride-sharing, where requests occur spontaneously at short notice, in contrast with *static* ride-sharing, where all requests are known in advance (Winter & Nittel 2006).

Furthermore, service providers may be classified into *service operators*, offering a full-service solution, including vehicles and drivers, and *matching agencies*, focusing on matching between individual passengers and drivers (Furuhata et al. 2013). An example of a service operator is an airport shuttle transportation service. Generally, these services are more closely related to DRT/DARP systems (Section 2.2.2).

Ride-sharing can, moreover, be classified into positional elements. Here, Morency (2007) propose four different types of matching, based on findings from extensive surveys, refined later by Furuhata et al. (2013):

- *Identical ride-sharing*: Both driver and passenger origin and destination are identical.

- *Inclusive ride-sharing*: The passenger origin and destination are on the way of the original driver route.

- *Partial ride-sharing*: Passenger origin or destination is not on the way of the original driver route, but a pick-up or drop-off location is chosen along the original route.

- *Detour ride-sharing*: Either the pick-up or the drop-off location, or both, are not along the original driver route; hence, the driver has to take a detour.

[3] `http://www.matchridergo.de/`, accessed 11.10.2017

From a modelling point of view, Agatz et al. (2010) distinguishes four variants, as shown in table 2.1. If single drivers should be matched to single passengers, the task is to find the best matching pairs. If a driver is allowed/willing to accommodate multiple passengers, a routing is further necessary to determine the order of boardings and alightings. The single passenger/multiple driver case aims at finding an efficient route for the passenger using multiple different drivers, so that changeovers are possible. The case with multiple riders and drivers combines all possibilities and is very complex.

Table 2.1: Ride-sharing variants, from Agatz et al. (2010).

	Single passengers	**Multiple passengers**
Single driver	Matching of single pairs	Routing of drivers to accommodate multiple passengers
Multiple drivers	*Multi-hop* ride-sharing: passengers transfer between drivers	Complex routing of drivers and passengers

Another differentiation of ride-sharing can be done concerning target markets. Furuhata et al. (2013) classifies them as follows:

- *On-demand*: casual, one-off trips for relatively short distances, very short announcement time, usually intra-urban.
- *Commute*: based on regular work schedule and long-term relationships; in this work referred to as carpooling (Section 2.3.3).
- *Long-distance*: long inter-urban trips with advanced scheduling.
- *Event*: ride-sharing formed among travellers attending the same event, e.g. sport competitions or concerts.

2.3.1 Mathematical formulation

From an optimization perspective, the *ride-sharing problem* (RSP) is closely related to the dial-a-ride problem (see Figure 2.4): customers request rides from an origin to a destination location, and time constraints for the pick-up time window and maximum riding time have to be considered. The basic difference from the DARP is the role of the driver. While the DARP assumes that vehicles and drivers are dedicated only for the purpose of transporting people, the drivers in a RSP are usually commuters or people on private trips, offering one or multiple seats. Hence, there is no central depot, but instead different origins and destinations for the drivers, and they also define time windows. The difficulty with the RSP is determining which driver should pick up which passengers (*matching*) and minimizing the driver detour length or other objectives (*vehicle routing*). These two problems can be modelled individually or closely interconnected.

The basic ride-sharing problem can be modelled as follows: given a set of drivers $\psi \in \Psi$ and a set of passengers $\rho \in P$. The drivers define an origin location $\lambda^+(\psi)$, a destination location $\lambda^-(\psi)$, a time for the earliest possible departure $t^+(\psi)$ and the latest possible arrival $t^-(\psi)$, and the number of available seats q. Likewise, the passenger requests consist of origin and destination locations $\lambda^+(\rho)$ and $\lambda^-(\rho)$, and the corresponding time window $t^+(\rho)$ to $t^-(\rho)$. For each possible driver/passenger match, a cost $c_{\psi\rho}$ and a binary decision variable $x_{\psi\rho}$ is introduced, indicating whether the match is proposed (1) or not (0). Naturally, the time windows should make a trip possible.

Table 2.2: Notation of the mathematical notation.

Notation	Unit	Description
$\psi \in \Psi$	-	Driver request
$\rho \in P$	-	Passenger request
λ^+	-	Origin location
λ^-	-	Destination location
t^+	s	Earliest possible departure time
t^-	s	Latest possible arrival time
d	m	Distance
q	-	Vehicle capacity
c	-	Cost
x	{0,1}	Binary decision variable

For a ride-sharing system, a certain time flexibility is crucial: it has a significant impact on the performance of a ride-sharing system, so that even minor increases in flexibility can improve the matching rate (Stiglic et al. 2016). Table 2.2 provides an overview of the notation.

The single driver/single passenger case with fixed roles can be modelled as a matching-only problem, without the need for vehicle routing computations. It is usually formulated as a maximum weight bipartite matching model, which can be solved by binary integer programming or network flow approaches (Agatz et al. 2010, 2011; Najmi et al. 2017).

Care must be taken when formulating the objective function. A simple minimization of detour times or mileage would result in no matches at all. Hence, in order to stimulate matchings, incentives or penalties have to be included in the objective function, or in the constraints when using a lexicographical goal programming approach.

Najmi et al. (2017) list different strategies to define (partially conflicting) objectives:

- Maximizing net distance savings (see explanation below)
- Maximizing number of matches
- Maximizing distance proximity index (based on the proximity of driver and passenger initial trips)
- Maximizing adjusted distance proximity index (incorporates the length of the matched trip into the total trip length)

The first possibility, also used by Agatz et al. (2011), aims at maximizing *distance savings*, meaning the reduced total travel distance when a ride is shared, compared with both people driving alone. This setting corresponds to a scenario where the ride-share provider aims at minimizing the total system-wide vehicle mileage, including single rides by non-matched participants. The formulation is hence aligned with social and ecological aspects such as reducing emissions and traffic congestion. The calculation of a distance saving value $c_{\psi\rho}$ is shown in equation 2.6 using the denomination shown in Figure 2.5.

$$c_{\psi\rho} = (d_1 + d_3) - (d_2 + d_3 + d_4) \tag{2.6}$$

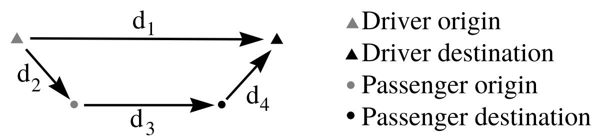


Figure 2.5: Direct path vs detour when sharing the ride.

Following Agatz et al. (2011), the model can be formulated as

$$\max \sum_{\psi \in \Psi} \sum_{\rho \in P} c_{\psi\rho} x_{ij} \tag{2.7}$$

subject to

$$\sum_{\psi \in \Psi} x_{\psi\rho} \leq 1 \qquad \forall\, \rho \in P, \tag{2.8}$$

$$\sum_{\rho \in P} x_{\psi\rho} \leq 1 \qquad \forall\, \psi \in \Psi, \tag{2.9}$$

$$x_{\psi\rho} = \{0{,}1\} \qquad \forall\, \psi{,}\rho \in \Psi \cup P. \tag{2.10}$$

If time windows are used, the following constraints should be added:

$$t^{+}(\psi) + t(d_2) + t(d_3) \leq t^{-}(\rho) \qquad \forall\, \psi{,}\rho \in \Psi \cup P | x_{\psi\rho} = 1 \tag{2.11}$$

$$t^{-}(\psi) - t(d_4) - t(d_3) \geq t^{+}(\rho) \qquad \forall\, \psi{,}\rho \in \Psi \cup P | x_{\psi\rho} = 1 \tag{2.12}$$

$$t^{+}(\psi) + t(d_2) + t(d_3) + t(d_4) \leq t^{-}(\psi) \qquad \forall\, \psi{,}\rho \in \Psi \cup P | x_{\psi\rho} = 1 \tag{2.13}$$

The constraints 2.8 and 2.9 ensure that every driver and passenger can be matched at most once. Constraint 2.10 limits the decision variables to binary values. Equation 2.11 ensures that the passenger reaches the destination on time, while equation 2.12 is responsible for a time-feasible pick-up. Finally, equation 2.13 ensures compliance with the driver time window. If the constraint of single passenger/single driver matches is relaxed, the problem becomes more complicated, since a routing is also necessary. Then, methods and solutions from the DARP must be adapted.

2.3.2 Methods

Numerous optimization strategies have been developed to solve the matching of drivers to passengers in different constellations. Agatz et al. (2012) provides a comprehensive review of optimization algorithms for the dynamic case of the ride-sharing problem. The single driver/single passenger problem can be solved efficiently by bipartite graph matching (Agatz et al. 2010, 2011). The single driver/multiple passenger problem is somewhat equivalent to the vehicle routing problem with pick-up and delivery (VRPPD, see

Section 2.2.1.1). If multiple drivers are allowed, passengers may change drivers during the trip. Gruebele (2008) investigates this issue by proposing an interactive system for real-time multi-hop ride-sharing.

A common approach for the dynamic version of the ride-sharing problem is to apply an *rolling horizon* method, where only requests known at execution time and not yet matched are included in the optimization, which is triggered with a certain frequency. Agatz et al. (2011) were able to show that less frequent optimization runs are sometimes better when using this approach, since more requests can accumulate in the meantime. Yousaf et al. (2012) use multi-objective path planning with a greedy algorithm, providing drivers with the flexibility to change the path according to personal requirements. Najmi et al. (2017) present a rolling horizon approach that is capable of solving a highly dynamic ride-sharing problem in real time. They embed a clustering algorithm based on k-Means into the rolling horizon framework, to split the problem into smaller sub-problems which can be solved quickly. A more spatial approach presented by Pelzer et al. (2015) divides the demand into distinct partitions based on the road network, which also significantly reduces the search space in the matching phase. The matching itself is then performed using an agent-based approach.

If running times are crucial, the algorithm of Schreieck et al. (2016) can be used as a simple but fast matching algorithm with a focus on performance, intended to enable real-time processing of requests. Furthermore, Geisberger et al. (2009) provide a method to compute the necessary detours for ride-sharing requests efficiently.

The ride-sharing problem is also often tackled by metaheuristics. Teodorovic & Orco (2005) propose to solve the ride-sharing problem with a *fuzzy bee system*, inspired by the foraging behaviour of bees in nature. The bees explore, step by step, the solution space, and after each step they try to convince other bees in the hive to follow their solution, if it is a promising one (Teodorovic & Orco 2008). Another approach from Herbawi & Weber (2012) uses a two-step approach, consisting of a genetic and an insertion heuristic, in order to solve the dynamic RSP with time windows.

The ride-sharing problem can also be solved in a decentralized manner, based on local negotiation between peers. Winter & Nittel (2006) investigates an urban ad hoc ride-sharing trip planning algorithm that is implemented as a mobile geosensor network of agents that interact locally through short-range communication. They were also able to show that a decentralized approach can reach near-optimal solutions, with the advantage of having a very scalable system. Similarly, Nourinejad & Roorda (2016) present a very flexible agent-based model, where single or multiple drivers can be matched with single or multiple passengers. The authors further claim that the proposed agent-based model is faster than conventional centralized optimization methods, while providing similar system objectives.

2.3.3 Carpooling

A variation of the ride-sharing system is the so called *Carpool Problem (CPP)*, which focuses more on daily commuting to a common work location and back. The main task is thereby to create schedules such that the driving task is fairly distributed among all carpooling members. Carpooling should be clearly distinguished from car-sharing, since in carpooling the vehicles are still privately owned (but may be driven by different people).

According to Calvo et al. (2004), the carpooling problem can be divided into two types: the *Daily Carpooling Problem (DCCP)*, where only one particular day is considered, and the *Long-term Carpooling Problem (LCCP)*. In the second type, all users can potentially be either driver or rider, and the objective is to minimize the total distance travelled while preserving long-time fairness among the members, so that no participant has to drive every day and others never. Baldacci et al. (2004) identifies two further variants of the carpooling problem: the *to-work Carpooling*, where the groups of people sharing rides do not change between outward bound and return trips, and the *return-from-work* problem, where group interchanges are possible. It requires the problems to be solved independently.

Many authors have developed various algorithms for this problem. Fagin & Williams (1983) propose several methods to determine a carpooling schedule, and examine them with respect to their fairness. The work by Coppersmith et al. (2011) provides lower and upper bounds for online algorithms. Baldacci et al. (2004) developed an exact method to solve the problem based on Lagrangian column generation. Calvo et al. (2004) proposed a heuristic method and a distributed GIS architecture, which was then tested in a real-life case study. A more recent work from Huang et al. (2016) applies Tabu search to solve the LCCP with multiple origins and a common destination.

3 Meeting points for shared rides: state of the art

This chapter provides an overview of the current state of the art in science concerning meeting points for shared rides, and highlights the research gap this work is attempting to close.

3.1 Meeting points

In the literature, the usage of meeting points has for a long time not gained much attention, in contrast with conventional ride-sharing or demand-responsive transportation systems, where usually a door-to-door service is assumed, i.e. a pick-up at the origin and a drop-off at the destination. However, interest in this research area is increasing, judging by the growing number of research papers that have been published in recent years.

When surveying the literature, it has to be stated that the naming of meeting points is not consistent. Denominations used include *meeting point*, *pick-up point* (sometimes abbreviated to PuP), *boarding point*, *stopping point*, *ride-access point*, *relay station* and *rendezvous point*; and correspondingly: *drop-off point*, *deboarding point* and *leaving point*. For the sake of consistency, the denominations *meeting point* (MP) and *divergence point* (DP) will be used throughout this work, to emphasise that people share the ride in between.

In this Section, the basics of meeting points as common destinations (3.1.1) as well as meeting points as intermediate locations are discussed (3.1.2).

3.1.1 Meeting points as destination

A meeting point, in its most basic interpretation, is, as the name suggests, a certain point where a set of at least two people who have a distinct origin meet each other. This Section deals with meeting points that are simultaneously the common destinations of the riders. The goal is then to determine a location in space that minimizes the necessary transportation cost among all riders. A real-world example is the determination of the most efficient conference location when the origin of all attendees is known, or the location for a tourist bus to pick up passengers, with the least effort for the tourists to reach it.

If the location can be placed anywhere in the Euclidean space, and the travel costs are based only on the Euclidean distance, the point that is searched is known as *Geometric Median*, indicating the location in the plane that minimizes the sum of distances to a set of n points. Note that this location may not be unique. For example, the optimal location for two people with different origins can be anywhere on the direct line between them, if the distance cost is linear.

There are many variations of this problem. The special case for a triangle (exactly three points in the plane) is known as *Fermat problem*, named after its inventor, Pierre de Fermat, in the 16th century. Correspondingly, the point is called a *Fermat point* or *Fermat-Torricelli point*. If the transportation costs per distance are allowed to be different for different points, the problem is known as *Weber problem*, named after Alfred Weber in 1909. These two problems can also be combined in the *Fermat-Weber problem*. The problem can be solved efficiently either numerically (Kulin & Kuenne 1962) or by a trigonometric approach (Tellier 1972). If weights are also allowed to be negative (meaning the greater the distance, the better), the problem is named *attraction-repulsion problem*. Another version of the problem searches for a set of m locations, minimizing the distance to n points. This problem is known as *multi-source Weber problem* or, probably more commonly, *location allocation*. It occurs in many real-world situations, e.g. when planning the location of warehouses for a specific product. Brimberg et al. (2000) reviews and compares several heuristics to solve the problem, including Tabu Search, p-Median Heuristic, Variable Neighbourhood Search, Genetic Algorithm and Relocation Heuristics. It was found that, on average, Variable Neighbourhood Search consistently yields the best results in moderate computing times. The umbrella term for all of the problems presented is *location theory*, which is a broad field in economic geography and spatial economics, addressing questions such as which economic activities are located where and why.

However, in most real-world scenarios, the possibilities for travelling and meeting are not continuous, since travelling is often restricted to a network (e.g. streets), and suitable meeting point candidate locations are also limited. Since the solution space is then discrete, the solutions can be enumerated, transforming the problem into a combinatorial optimization problem (see Section 2.1.3) which is NP-hard to solve optimally, as it can be reduced to the set cover problem (see Section 2.1.5). However, there are some specialised pruning algorithms available for this kind of problem. Yan et al. (2011) presents an optimal and a greedy heuristic to find a meeting location in a network efficiently. They distinguish between the *min-max* and the *min-sum* problem, depending on the type of objective function: while for the min-max case the maximum travel cost to a meeting point among all participants should be minimized, the min-sum problem aims at minimizing the sum of all travel costs (Figure 3.1). For solving the problem they apply a pruning of the search space based on convex hulls of point sets. Similarly, Xu & Jacobsen (2010) applies road network partitioning to reduce the necessary search space.

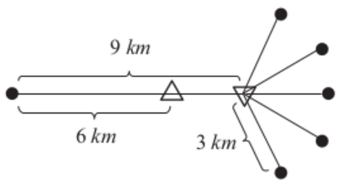


Figure 3.1: Optimal meeting point for the min-max (upward triangle) and the min-sum (downward triangle) problem. Source: Yan et al. (2011)

3.1.2 Intermediate meeting points

In contrast with meeting points as destinations, intermediate meeting points are locations where two or more people meet to travel further together to a common destination or another meeting point. In the literature, meeting points have been incorporated for various purposes and at different levels of detail. In this section, the literature is classified into meeting points for fixed routes, ride-sharing, carpooling, SDRT services, meeting areas, privacy protection and meeting point placement.

Meeting points at fixed routes

Probably the easiest situation for determining a meeting point occurs when the driver route is fixed, since the search can be limited to points along this route. O'Sullivan et al. (2000) proposes a GIS-based method to determine such boarding points along bus routes by using isochrones and accessibility analysis (see Figure

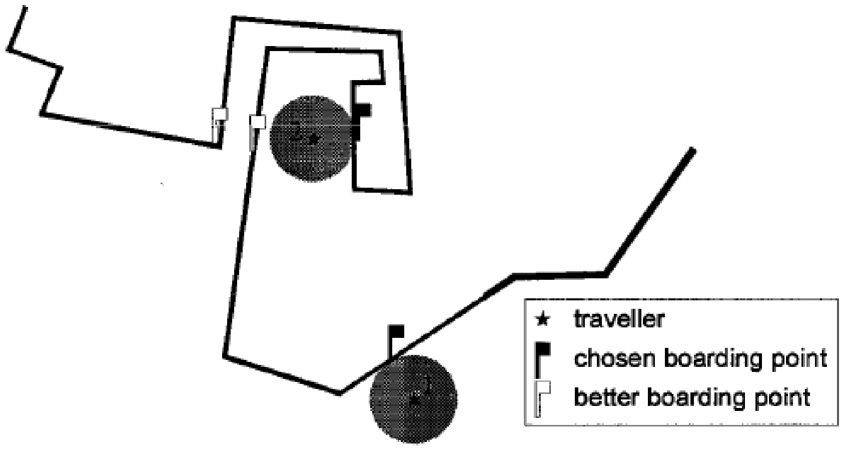


Figure 3.2: Bus boarding points along a fixed route. Source: O'Sullivan et al. (2000)

3.2). This technique allows the identification of reachable meeting locations by constructing space-time prisms based on space and time constraints. The results are visualised with isochrone maps. A similar approach, but more within the scope of ride-sharing with fixed driver routes, was presented by Rudnicki et al. (2008). They determined meeting and divergence points with the help of local communication. In an agent-based simulation they demonstrated that average time savings of 20 % compared to walking can be achieved with such a system.

Meeting points for ride-sharing

If the driver routes are somewhat flexible, the complexity of the problem increases. However, if at least the driver/passenger pairs are already fixed, the matching can be omitted and the problem is thus limited to the determination of meeting and divergence points for a given group.

Kamel Aissat and Ammar Oulamara have focused in depth on this kind of meeting point problem. In an early work, they propose an exact and two heuristic methods to identify meeting and divergence locations within a city, with the goal of minimizing the total travel costs (Aissat & Oulamara 2014a). All nodes in the street network are regarded as meeting point candidates. The exact method utilizes an enumeration of all possible locations based on multiple one-to-all Dijkstra routing requests. Due to the quadratic complexity, they also propose a heuristic that needs several one-to-all routings but prunes the search space. However, no assumptions are made regarding the mode of travel, since no walking constraints or public transport connections are introduced, but the ways to and from the meeting/divergence points can be arbitrarily long. Furthermore, it is not clear how much mileage is actually saved through this approach. The methods were later described with more details (Aissat & Oulamara 2014b). Moreover, they expanded this approach to allow flexible selection of drivers for a particular rider (Aissat & Oulamara 2015a). Given a set of drivers that have already placed a ride-sharing offer, they propose an exact method and a heuristic to select the most cost-efficient candidate for a specific passenger request. Additional constraints are also introduced, namely time windows, detour time and desired minimum user cost savings for a match. The driver offers are saved in so-called *buckets*, which store for each node all possible driver trips that do not violate the constraints. A fourth publication (Aissat & Oulamara 2015b) addresses more of a carpooling problem with outgoing and return trips, where the aim is to find appropriate meeting points, which they call *relay stations*. In this work, driver and passenger do not split up again during the ride; instead, the driver is supposed to drop off the passenger directly at the destination. In computational experiments they reached cost savings due to combined rides of 25 - 36 %.

The idea of walking to meeting points in the vicinity of passenger origins is discussed by Balardino & Santos (2016), who propose a greedy insertion heuristic and an iterated local neighbourhood search metaheuristic to assign passengers to drivers at meeting points, which they call *close enough points*. As in a carpooling scenario, the destination is common for all participants, and the maximum acceptable walking length of the passengers is limited by a Euclidean distance threshold. They use a real street network, where each node is a meeting point candidate, and limit further the possible detour length of the driver and the vehicle capacity. The problem is formulated as integer linear programming, with the primary objective of maximizing the passengers served and secondarily minimizing the total driver distance.

A comprehensive work with respect to meeting points was conducted by Stiglic et al. (2015), who investigated the benefits of using meeting points in a ride-sharing scenario. For their experiments, they used real-world demand data, but no street network; instead, all distances are based on the Euclidean distance, and the meeting point candidates are randomly placed. The ride-sharing setting has no routing phase, since only one pick-up and drop-off is allowed for each driver. Nevertheless, multiple passengers can join the ride between the meeting and divergence point with respect to time, walking and capacity constraints. Due to the omitted routing phase, they formulated the setting as a maximum weight bipartite matching problem. This approach allows the solving of even large instances exactly, by applying binary Integer Programming (Section 2.1.2). The objectives are, on the one hand, maximizing the system-wide driving distance savings, and, on the other hand, maximizing the matching rate. Since these objectives are sometimes competing, they apply a hierarchical approach called *lexicographical goal programming*. In a first optimization run, the optimal number of matched participants is determined and subsequently introduced as constraint in a second optimization run, where the goal is then to maximize distance savings. In computational experiments they were able to show that the introduction of meeting points can indeed improve the number of matched participants (up to 6.8 %) as well as mileage savings (up to 2.2 %), depending on the number of available meeting points. On the contrary, they notice an increase of average trip time for matched passengers due to additional walking of slightly more than 12 %, with an average walking time between 8 and 9 minutes. If the drivers are very flexible and accept large detours, the matching rate and mileage savings can be significantly increased. A similar effect can be observed when the maximum (walking) distance threshold to a meeting point is increased. Also, introducing more participants increases the matching rate further.

Meeting points for carpooling

For the carpooling case, Chen et al. (2016) developed a ride-sharing mechanism focusing on closed corporate communities. The objective is to reduce commuting costs, including mileage and time loss, for example due to transfer. As constraints, they limit the driver detour and introduce return restrictions, meaning that a commuter is guaranteed to return to their home location at the end of the day. The model is then formulated as mixed-integer linear program using binary decision variables, but separated into ride-sharing with meeting points and the extension considering return restrictions. Since the exact model becomes unfeasible when the number of customers is large, they propose a combination of a construction and a greedy improvement heuristic. They further distinguish between meeting point candidates (a node with potentially overlapping customer time windows) and car parking points (role of a person changes from driver to passenger). In a computational experiment, they use real-world carpool parking lot positions retrieved from a Dutch website[1], as well as real-world data from a company. Their results indicate that a reduction of vehicle miles of 7 - 25 % is achievable by using ride-sharing with meeting points. The savings are higher when more people participate and the demand is more concentrated. In addition, the matching rate increases with a higher transfer point density.

Meeting points for SDRT services

Another domain is the usage of meeting points for shared demand-responsive transportation (SDRT) services. Martínez et al. (2014) formulates an optimization problem for an urban semi-flexible SDRT system which they call *express minibus service*, using common meeting and divergence points for groups of clients. Their design methodology consists of four steps:

1. Selecting feasible passengers for the service
2. Hierarchical clustering of passenger requests
3. Merging of clusters into compatible services
4. Aggregating services into vehicle routes

The clustering of passengers takes into account normalized values of the distance between trip origins and destinations, and the difference in arrival time. It is furthermore constrained to a maximum distance radius and diameter, a maximum time, and a maximum cluster size. These clusters are then subsequently combined to vehicle routes by binary Integer Programming, subject to various constraints. As meeting and divergence points, the centroids of customer group locations are calculated, hence they are not necessarily aligned with a street network or restricted to a predefined subset of candidates. They also consider monetary values of the minibus system, e.g. by removing services that are not cost-covering. As a result, they claim that the service provides huge potential, since up to 53 % of private car trips longer than 5 km performed during the morning peak period could be replaced by the SDRT service, with each minibus replacing on average 9.15 private cars. For the experiment, they used minibuses of varying capacity (8, 16, or 24).

Another study within this scope from Häll et al. (2008) investigates a SDRT service as part of an integrated public transport system. Here, the meeting point candidates are predefined and uniformly placed in the operation area, based on a grid structure, but limited to places accessible by the road network using a distance threshold. For a simulation, they used real demand data from the Swedish town Gävle and tested an adaption of a dial-a-ride service, without timetables but with meeting points, and only those locations where customers have ordered a ride are visited by the bus. They discovered that the meeting point based operation does not seem to offer any major differences when compared with a door-to-door service, concerning efficiency. However, they also state that if the operator can select the MPs of the customers, the MP solution may be more beneficial. Furthermore, this result could also be due to a very dense network of meeting points, so that the distance between the doorstep of a customer and the closest meeting point is never far.

Meeting areas

While most approaches focus on recommending or assuming a single meeting point, other approaches also propose lines or areas as potential meeting point zones, leaving the exact meeting point choice to the travellers. Rigby et al. (2013) developed an opportunistic client user interface technique called *launch pads*,

[1] `http://carpoolplein.nl`

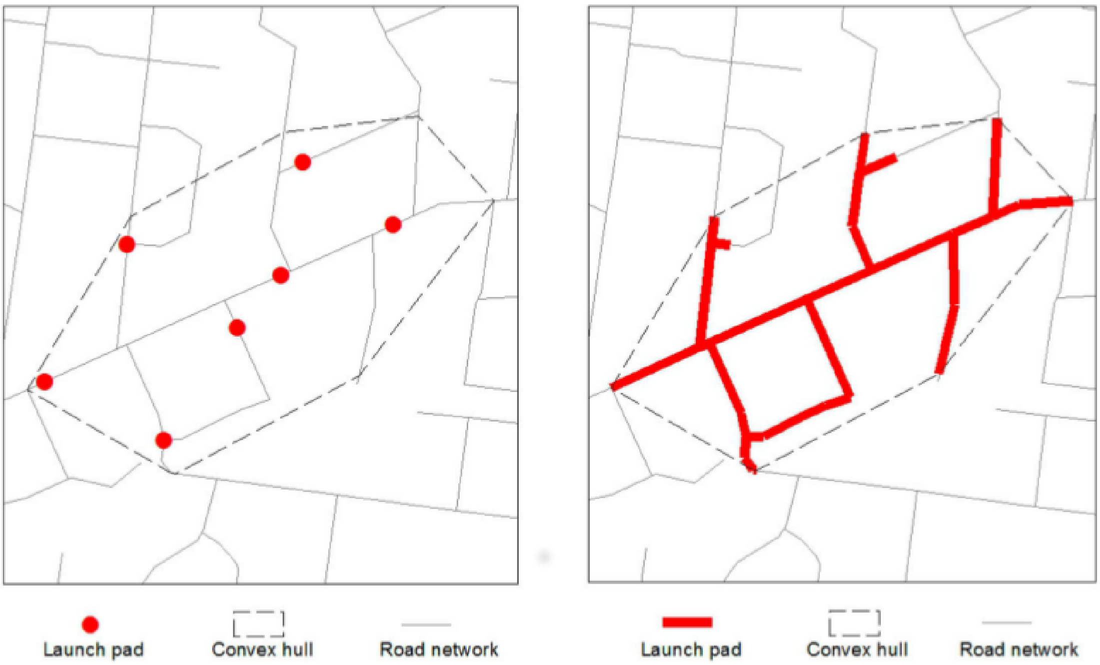


Figure 3.3: Launch pad variants: discrete (left) and continuous (right), created for a single vehicle. Source: Rigby & Winter (2016)

showing passengers the area in which they could potentially be picked up by a driver within a certain time interval. For the calculation of the launch pad areas they applied the theory of time geography to calculate space-time prisms (Miller 1991). The result is not a single point, but a few possible pick-up points or even lines (Figure 3.3). With increasing flexibility of the driver, the launch pads include more potential points. Furthermore, each meeting point suggestion is labelled with a grade of stability based on the number of overlapping vehicles, serving as an indicator of the pick-up probability. The launch pad technique was later enhanced by incorporating additional information, e.g. fares or departure times, into the launch pad visualization, such that the users can customise his/her interface according to individual preferences (Rigby & Winter 2015). For this, the authors implement a set of map algebra operations for combining the offers of multiple vehicles or ride properties. This improved visualization is stated to help the customers by enhancing their knowledge as to where to move within the environment. Later, the launch pads have been further extended to a continuous representation of vehicle accessibility (Rigby et al. 2016). In addition, they investigated human understanding and usage of different visualizations of launch pads, using a spatial cognitive engineering approach (Rigby & Winter 2016).

Privacy protection

As a side effect, the idea of launch pads provides the advantage of ensuring privacy for users. A contract can then be established without a need for revealing the actual location, since only the area of possible pick-up locations is communicated. However, there are further approaches in the literature concerning privacy protection in ride-sharing. Aïvodji et al. (2016) propose a distributed architecture for the determination of meeting points in ride-sharing, with the goal of preserving privacy. More precisely, they combine a cryptographic primitive called *private set intersection* (PSI) with multimodal routing algorithms to ensure privacy. In a nutshell, PSI is used to jointly compute the intersection of private input sets of two parties without leaking any additional information. Additionally, the meeting points are computed based on *isochrones* of the users, indicating the locations a user can reach after a certain amount of time. They propose a brute force and a heuristic approach to compute the potential sets. In the heuristic approach, a simple voting procedure is applied to select the final meeting and divergence location. They claim that their decentralised application does not significantly impact the quality of the ride-sharing solution, compared with an optimal, centralized service.

Another privacy-preserving approach was developed by Goel et al. (2016), providing a passenger matching and meeting point recommendation without revealing information about the actual starting nodes. They apply a recursive ellipse-based model to reduce the search space, based on the time constraints of the driver, and further propose a match-maker model to negotiate driver/passenger matches for meeting points. They consider meeting points as fixed locations on the road network, but for the computations they incorporate all network intersections as candidates. During the match making, the driver only gets hashed information about the meeting points that are within a route ellipse.

Placement of meeting points

A different but closely-related problem is to find the optimal number and placement of meeting points for ride-sharing in an urban environment. Goel et al. (2017) tackles this problem with the aim of maximizing coverage, while at the same time ensuring privacy for the customers. The selection of meeting points is computed using GRASP (*Greedy Randomised Adaptive Search Procedure*, a combination of greedy solution construction and neighbourhood search; see for example Feo & Resende (1995)). They model the optimal meeting point selection problem as a multi-objective problem, with two competing goals: maximizing the coverage versus minimizing the number of points due to privacy protection. Since these objectives are conflicting, they compute Pareto solution sets (the *Pareto front*) for different coverage solutions. In addition, the meeting points are weighted according to their closeness to additional options, such as public transport and entertainment. They also compare their meeting point selection with public transport stops and demonstrate that their model performs better in terms of privacy, coverage and occupancy. In a computational experiment they claim that the model is able to save 23 - 40 % of vehicle mileage if drivers are willing to take a detour and passengers are willing to travel to the meeting points.

3.2 Knowledge gap

Generally, it can be observed that the actual determination of eligible meeting points in a real city environment is often neglected. Nearly all existing models use either the Euclidean plane to simulate meeting points or use all nodes of the routing network as potential meeting point candidates. In reality, however, suitable locations for safe and convenient pick-up and drop-off are not ubiquitous. Just selecting any node will often result in either unreachable or inappropriate locations, such as those with heavy traffic, where a secure parking and/or boarding or de-boarding is almost impossible. In addition, there may be ambiguous cases, such as when a multi-lane junction is proposed to the customers, but the exact location on this junction is not clearly determined. Finally, there might be further, convenience-based evaluation criteria for the selection of good meeting points, not just travel time or mileage. As an example, parking possibilities or facilities to improve the waiting time (seating, shelter) can be of major importance for both driver and passenger. In summary, most currently available research approaches lack a more detailed determination and analysis of the meeting point situation in the real world. Hence, map-based approaches are necessary to push the meeting point research forward in a direction closer to reality. This holds both for ride-sharing and demand-responsive transportation systems.

In addition, most ride-sharing models focus on intra-urban rides covering shorter distances, where reachable meeting points for passengers are limited by a walking threshold. However, in long-distance inter-urban ride-sharing, more remote meeting points, e.g. close to motorway exits, may be beneficial for both driver and passenger, since unnecessary detours through inner city parts can be avoided. This is most relevant if the driver is intending just to pass the city. A recommendation for meeting points that are both conveniently reachable by public transportation and not far from a motorway exit can help to reduce the travel time for travellers, and also congestion in the inner city.

In order to close the gaps identified, three main research areas are utilised within this thesis:

1. **Meeting points for intra-urban ride-sharing**
   As stated above, most existing approaches neglect the usage of real-world meeting points, so in this work this knowledge gap is tackled for an intra-urban ride-sharing scenario. Furthermore, in most existing simulations regarding meeting points, travel time, mileage and matching rate are objectives that are mini- or maximized. However, it can be argued that convenience-based factors also play a role when selecting appropriate meeting points, namely the facilities of the meeting points, e.g. shelter, seating or illumination. To achieve a realistic model, the results from user surveys presented in chapter 4 are incorporated in the matching algorithm of an intra-urban ride-sharing model, and several simulation experiments are conducted to demonstrate the impacts (Chapter 6).

2. **Reaching meeting points by public transport**
   A location-based recommender system algorithm is presented, connecting meeting points, ride-sharing and multimodal transportation (chapter 7). The proposed method uses comprehensive precomputation of public transport connections to enable a real-time application with quick response times for long-distance ride-sharing. A simulation study demonstrates how the recommended meeting points and travel times change over time.

3. **Meeting points for SDRT services**
   The existing approaches in this field use only artificial meeting points, either as the centroid of customer locations (Martínez et al. 2014) or predefined but in a regular grid (Häll et al. 2008). Feasible meeting point candidates, such as public parking areas, are however usually unequally distributed within a city area, and dissimilarly reachable by vehicles and pedestrians. In addition, the road network may contain obstacles and one-way streets, requiring large detours to reach some meeting points. Hence, there is a lack of knowledge about the impact of using real-world meeting points in an SDRT system, and a need for algorithms to propose meeting points to the customers. This research area is discussed in chapter 8.

4 Real-world meeting points

As stated in chapter 3.2, one gap in research is the determination of meeting points in the real world. In this chapter, the question relates to how people judge meeting points, and how they can be assessed. Naturally, how people define a "good" meeting point is very subjective. Every individual person has a different opinion about the configuration of suitable locations. Hence the question of finding good meeting points can only be answered by investigating the statements and choices of individuals. In this chapter, two surveys are presented, one questionnaire-based (Section 4.1) and the other one map-based (Section 4.2).

4.1 Survey based on questionnaire

In order to obtain direct stated preferences, a questionnaire-based user survey was conducted by Thomas Reinicke in 2015, within the scope of his Bachelors thesis. The following figures have been created based on the data from this survey.

4.1.1 Setting

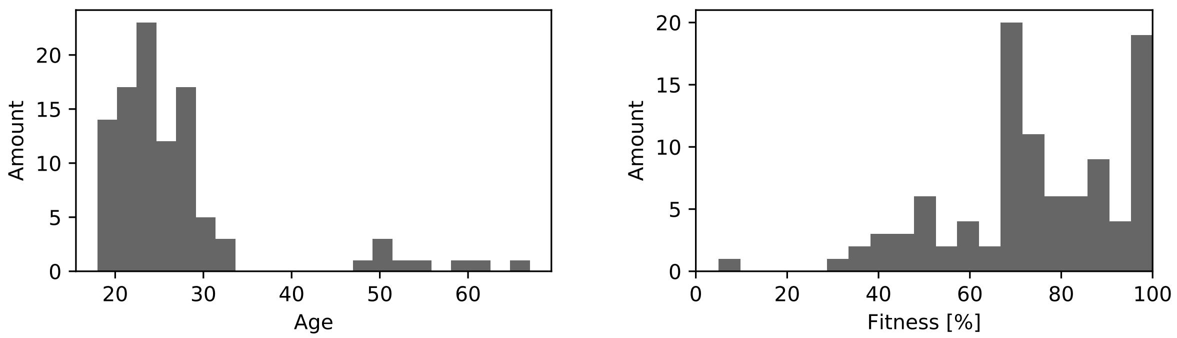
The survey was structured into four question groups:

- General questions about ride-sharing behaviour
- Questions about the suitability of meeting point types
- Questions about assessment criteria
- Personal questions

In total, 59 questions were asked, but the number of questions varies depending on personal ride-sharing behaviour. For example, a person is only asked about the suitability of a meeting point from a drivers perspective if he or she has ever picked up somebody using a car.

The web-based survey was created with LimeSurvey[1], hosted by Leibniz Universität Hannover. It was released on 12th January 2016 and closed on 14th February 2016. Altogether, 116 people participated, 100 of these completely and 16 partially. The group of 100 participants who completed the survey consisted of 52 females and 48 males, with an average age of 26 years. The youngest participant was 18, and the oldest 67. Figure 4.1a shows the histogram of the stated ages. Due to the personal addressing of potential participants, there is a strong focus on people between 20 and 30 (78 %). In addition, the participants have been asked to self-estimate their personal fitness as a percentage (Figure 4.1b). It can be seen that mainly young and fit people answered the questionnaire, hence the results are likely to be biased towards this user group. However, since young people between 25 and 34 years are the main target group of ride-sharing (Rayle et al. 2014), the survey results can still be of interest.

[1] `https://survey.uni-hannover.de/`, accessed 09.11.2017

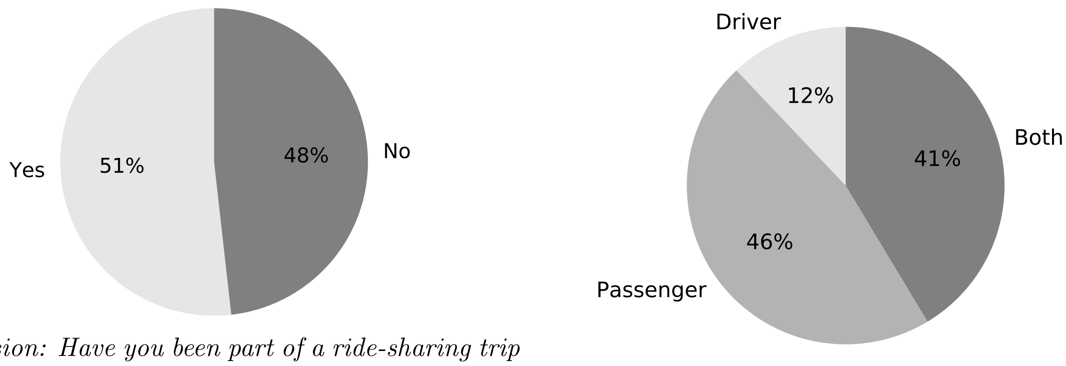


(a) Participants age.

(b) Participants fitness (self-estimated in %).

Figure 4.1: Histogram about personal details of the participants.

Figure 4.2 shows the ride-sharing activity of the participants. Approximately half of them have been taken part in ride-sharing in 2015, of which again almost half have used it as a passenger, 41 % as a driver and 12 % as both passenger and driver.



(a) Quesion: Have you been part of a ride-sharing trip in 2015?

(b) If yes: as which part?

Figure 4.2: General questions about ride-sharing activity. Difference to 100 %: Rounding.

4.1.2 Results

As a starting point, the participants were asked about their reasons for determining meeting points in the past (Figure 4.3). The most important aspect seems to be that the driver detour is kept small (46 %), in contrast with aiming at an equal journey time (9 %). This indicates a higher weighting of the driver travel time, when compared with the passenger travel time. Using well-known locations is the second most important factor, with 37 % of the votes, which corresponds to the findings of meeting point facilities (Figure 4.7). The facilities at the meeting points seems to play only a minor role (5 %). However, since only one answer was allowed, minor preferences are not considered.

Figure 4.4 shows frequently chosen meeting point locations for different scenarios. Most votes (nearly 200; multiple answers were allowed) designate meetings with known persons, e.g. when driving together to sports or social events. The most important meeting point for this scenario is in front of the doorstep (35 %), in other words picking up somebody at home. Other popular locations are parking places (13 %), bus/tram stops (12 %), train stations and street junctions (both 11 %). If the ride-sharing partner is not

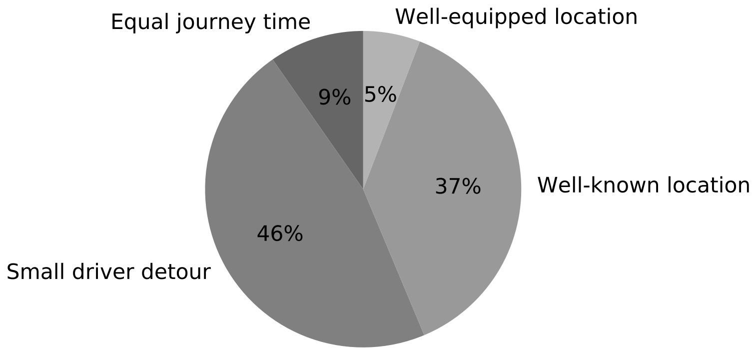


Figure 4.3: Importance of meeting point factors. Difference to 100 %: Rounding.

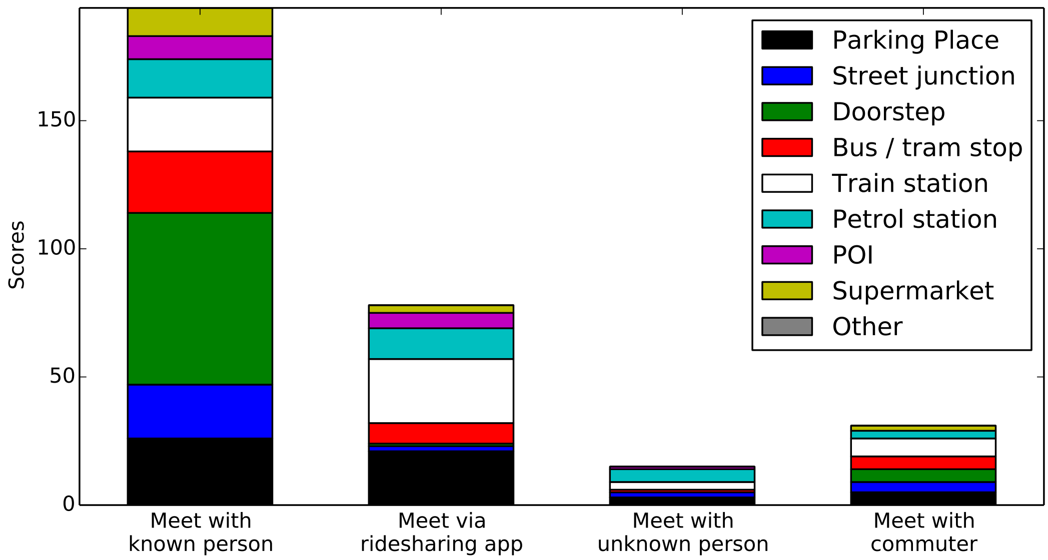


Figure 4.4: Common meeting points for different scenarios.

known, the result looks different. For rides that are booked via ride-sharing applications, the most popular meeting location is the train station and its surrounding (32 %), followed by parking places (27 %). The bar for meetings with unknown persons without booking by a ride-sharing application is mainly focussed on hitchhikers, but only a few participants have used this method of ride-sharing. However, in contrast with meetings with known persons, petrol stations play a more important role.

Figure 4.5 gives an impression of the suitability of different meeting point locations. This question was answered by 83 people, who are either only a driver or only a passenger, or did not participate in ride-sharing at all in 2015. All of the location types provided were mostly assessed as very well or well suited, especially at parking places and train stations. Only the location at street junctions was rated very critically; the majority regards this location as either not very suited or even not suited at all.

A more detailed breakdown is provided by Figure 4.6, showing only the answers from 23 participants, who acted as both passenger and driver (see Figure 4.2b), because they are assumed to know the difference in perspective. Figure 4.6a shows the rating for the driver's perspective, while Figure 4.6b shows the passenger's perspective. While there are only small differences in the opinion about street junctions, POIs and supermarket parking places, there are in fact some differences with other locations: parking places and petrol stations are more preferred by drivers, while bus/tram stops and train stations are more preferred

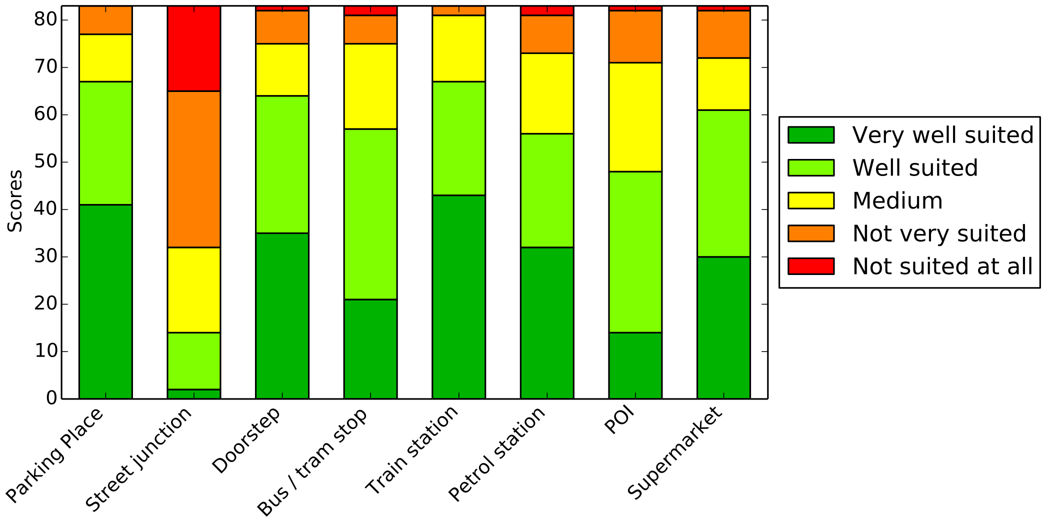


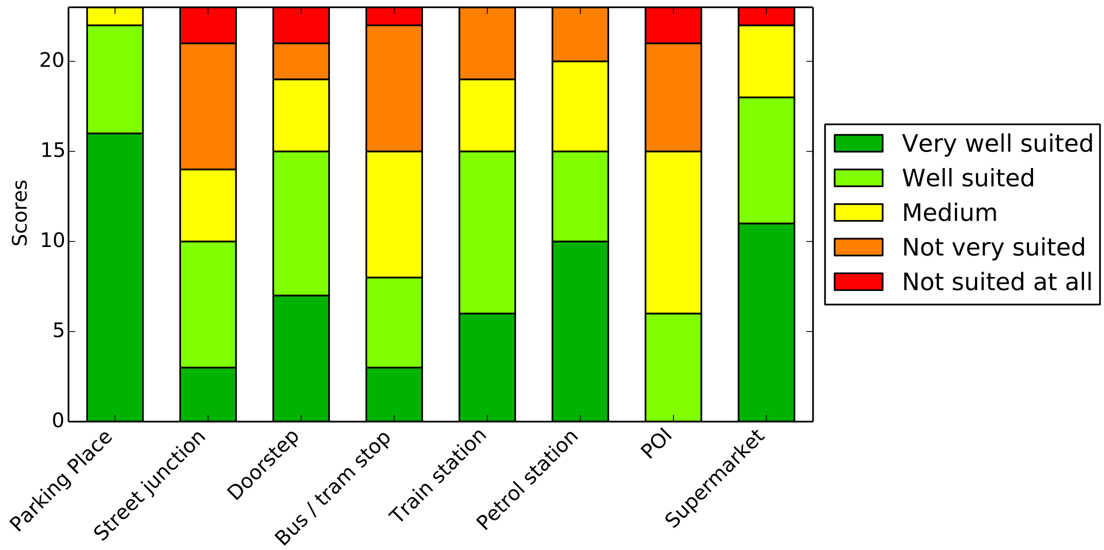
Figure 4.5: Suitability of meeting point locations. 83 people answered this question.

by passengers. The biggest difference occurs in relation to pick-up at the doorstep, which is very popular for passengers.

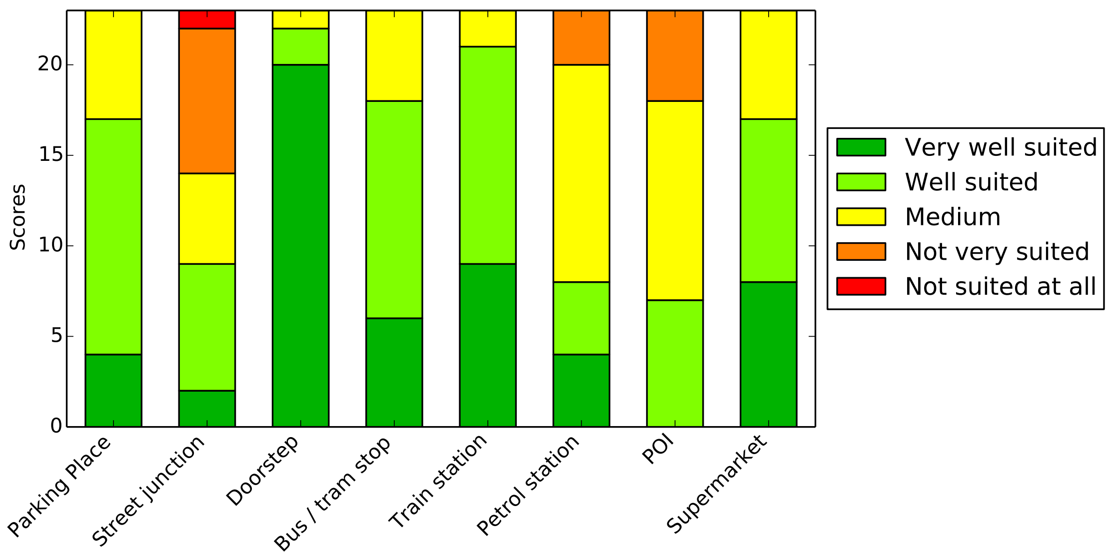
Figure 4.7 visualizes the stated importance of meeting point facilities, separated for summer (Figure 4.7a) and winter (Figure 4.7b), since there are some significant differences. Surprisingly, by far the most important aspect (it is not really a facility) is the unambiguousness of the location, which means that the place is well-defined and the ride-sharing partners can easily find each other. In summer time, only security (e.g. the ability to securely place a bicycle) and parking prices seem to have significant importance. Convenience facilities such as seating, shelter or toilets are not crucial. However, the picture looks different in winter time, where, in particular, illumination, shelter and warmth play a major role. Only toilets and seating possibilities are still not assessed as very important.

Finally, the participants were also asked about their preferences concerning maximum acceptable time and distance to access a meeting point (Figure 4.8). As expected, the acceptable distances are longer during summer time and with no luggage, and decrease for winter time and with luggage. The stated distances are in general very long, as almost one third of participants stated that they would walk 2 km or more in summer time. These values are surprising, since they are much higher than those from the literature: a common assumption in the scenario of walking to public transport stops is that people will walk on average 400 m without objection, with an acceptable maximum walking distance of 800 m (Hess 2012; Millward et al. 2013). One reason could be that it is hard to estimate distances without having a reference in mind. Therefore, a map-based second survey was additionally conducted (see Section 4.2). Another reason might be that mostly young and fit people participated in the survey (Figure 4.1a), hence the values are likely to be biased towards longer acceptable distances.

The walking time (Figure 4.8b) shows the cumulative acceptance of travel time (as a ratio of the total travel time) that would be acceptable for walking to a meeting point. Most participants state that they would accept 10 - 30 % of the time, corresponding to 6 - 18 minutes for a one-hour drive.

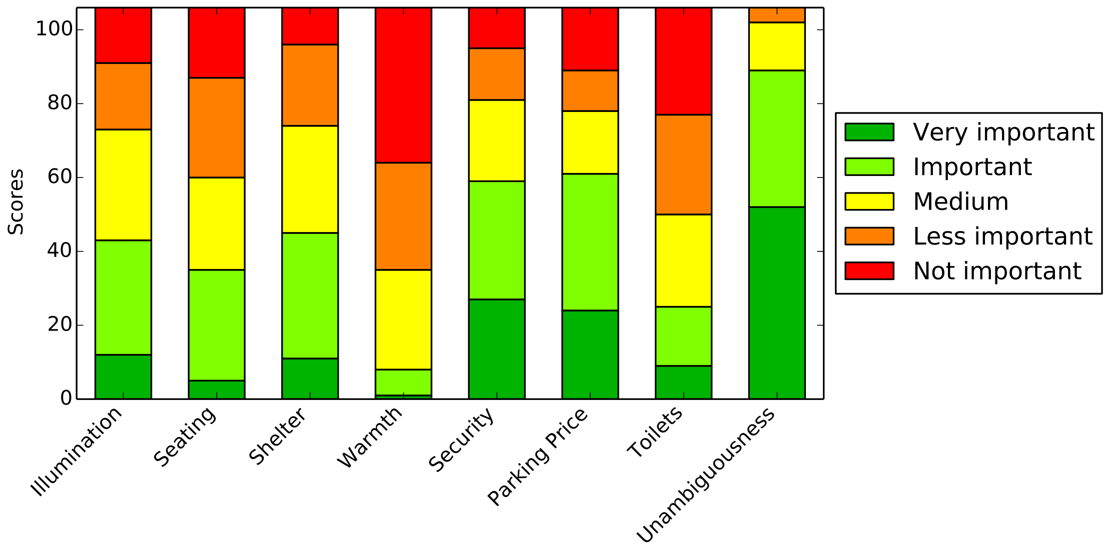


(a) For drivers

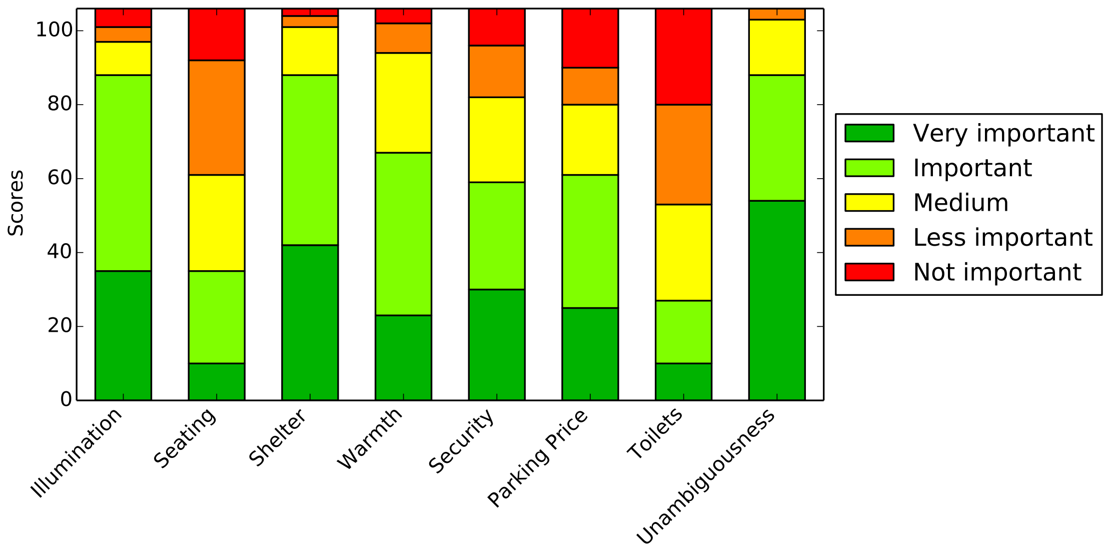


(b) For passengers

Figure 4.6: Suitability of meeting point locations, divided into driver and passenger ratings. 23 people answered this question.

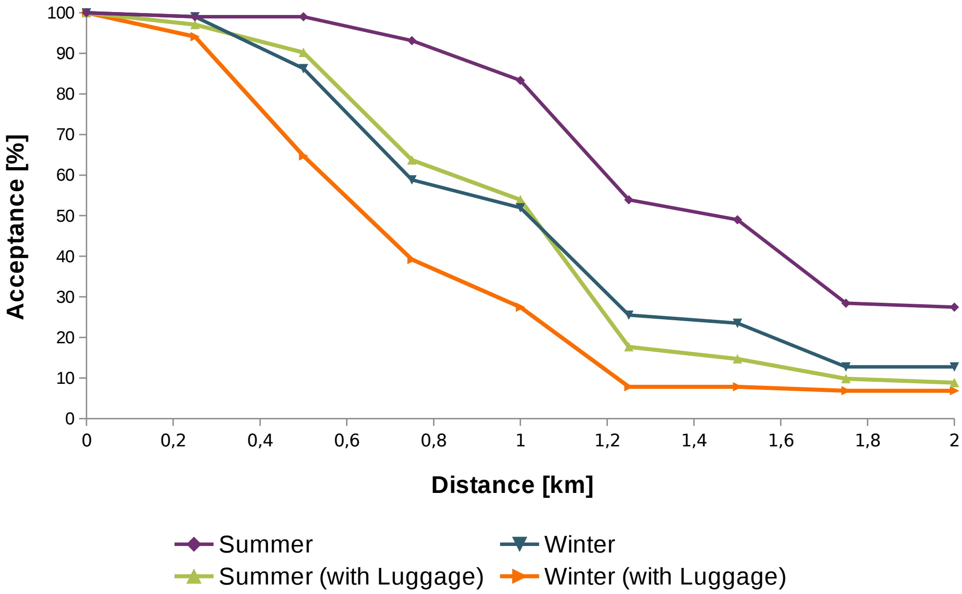


(a) In summer

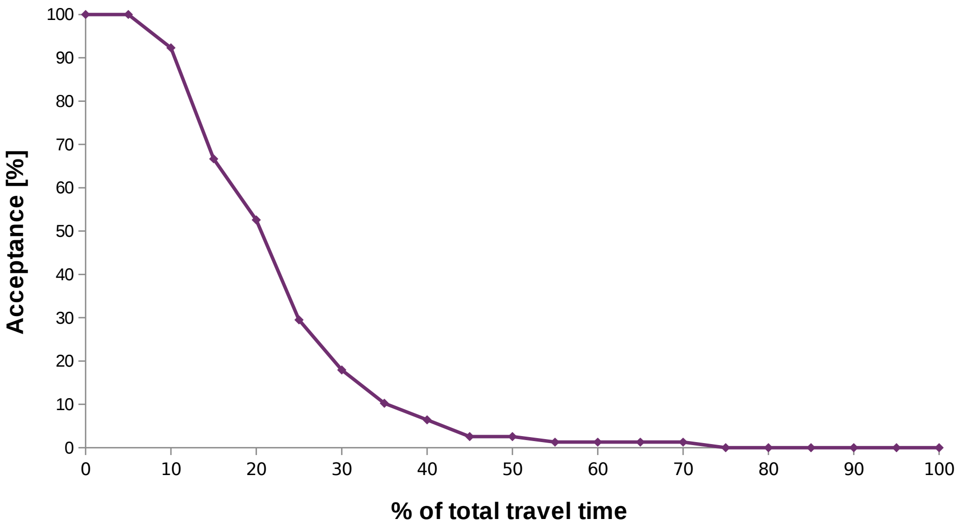


(b) In winter

Figure 4.7: Importance of meeting point facilities. 106 people answered this question.



(a) Accumulated acceptance of walking distances to a meeting point.



(b) Accumulated acceptance of walking times as ratio of the total travel time.

Figure 4.8: Stated preferences about acceptable time and distance to a meeting point. 84 people answered both questions.

4.2 Map-based survey

In order to overcome the shortcomings of the first survey, a second, map-based survey was conducted in 2016. The map provided a reference for the survey participants to better estimate locations and distances. The idea is that people are more comfortable in well-known environments, so every participant could choose the investigation area on his/her own, from the whole of Europe (limited only by the coverage of the routing API used).

4.2.1 Setting

Generally, the survey is divided into three steps:

1. Enter origin.
2. Enter (several) meeting points within walking range.
3. Enter (several) meeting points for long-distance ride-sharing, reachable by public transport.

In step 1, every participant is asked to place a marker on a map at his/her current (or fictive) origin. The location does not need to be highly accurate, so that a certain amount of privacy is preserved, and the emphasis is more on being generally familiar with the environment.

After confirming the chosen location, users proceed to step 2. Based on the origin entered, participants were asked to enter meeting point locations for a fictive ride-sharing meeting where they are supposed to walk to the meeting point (step 2). After clicking on the map, a marker popup with a JavaScript dialogue appears, asking the users to enter a rating and various properties of the location. For each meeting point, a type had to be selected from a list of suggestions (see Figure 4.12). If there were additional aspects to be considered, participants had the chance to enter comments. The marker is saved when the popup window is closed. In addition, existing markers could also be removed again from the map.

As soon as the participants are satisfied with the meeting points entered, they proceed to step 3. In this scenario, four routes are shown that a fictive driver could take while passing the city of the origin location. The four driver routes are built based on a routing with Graphhopper Routing API[2], passing the city of interest. The fictive driver's origin and destination is created by adding and/or subtracting a distance of 50 km to the x and y coordinates of the chosen user origin. In total, four passing routes are created: one horizontally, one vertically, and two diagonally (Figure 4.9). Because of the long way for the driver, the routes mostly use motorways or other major roads passing the city (Figure 4.10). Participants are requested to place one or more meeting points for each route on the map, with the aim of limiting detours for the driver. Again, after putting a marker on the map, a property dialogue opens, requesting some details about the location.

The survey was built using PHP, JavaScript including the leaflet map API[3] and a MySQL database.

[2] https://graphhopper.com/api/1/docs/routing/
[3] http://leafletjs.com/

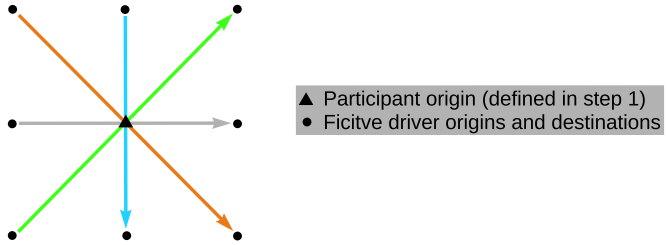


Figure 4.9: Fictive driver routes passing the participants location.

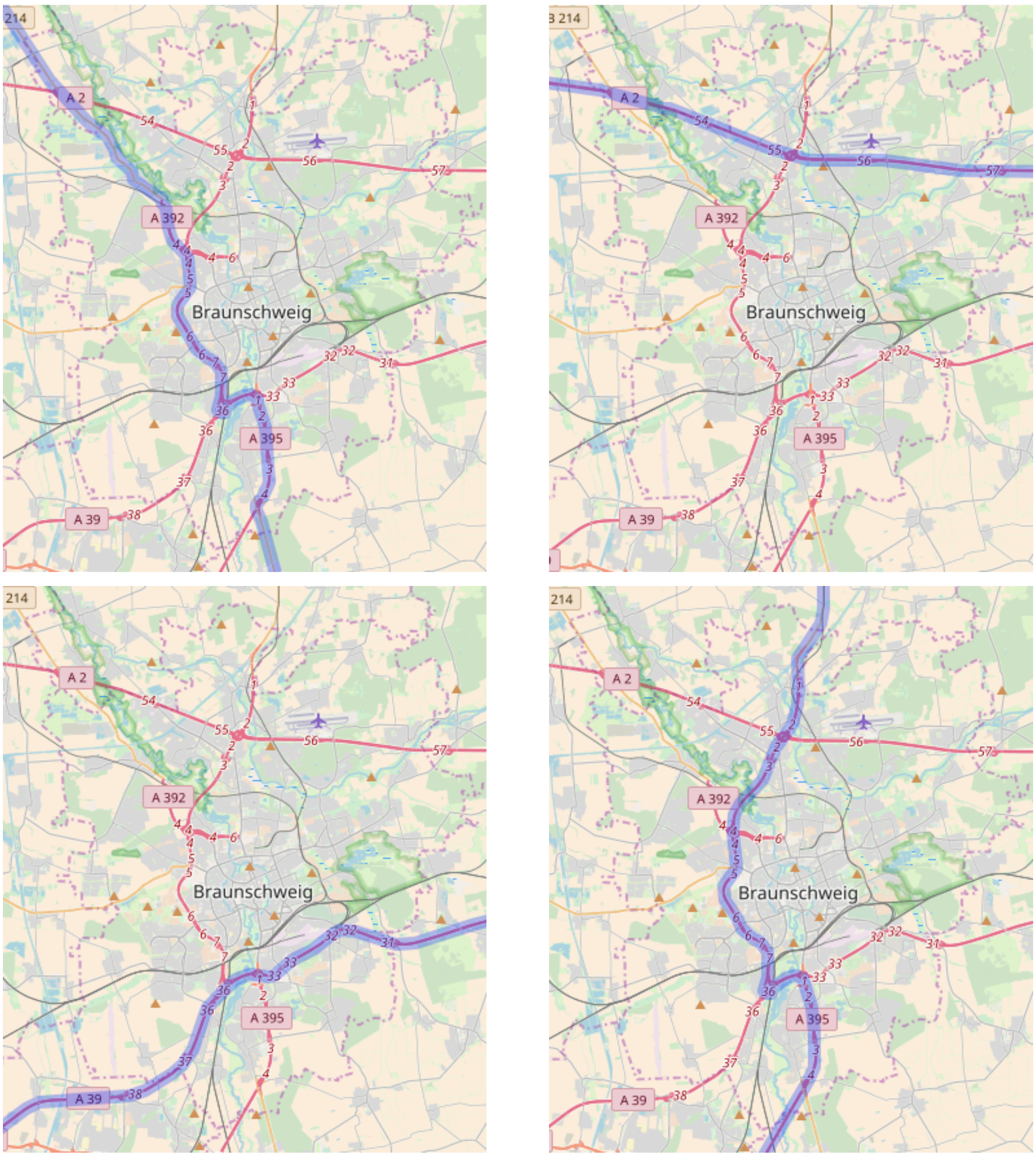


Figure 4.10: The four driver routes, created for the participant origin Braunschweig.

4.2.2 Results

In total, 76 people from Germany and Switzerland participated, recording a total of 302 meeting point locations. Figure 4.11 shows an example extract for the Hannover city region including all reported origins and meeting points.

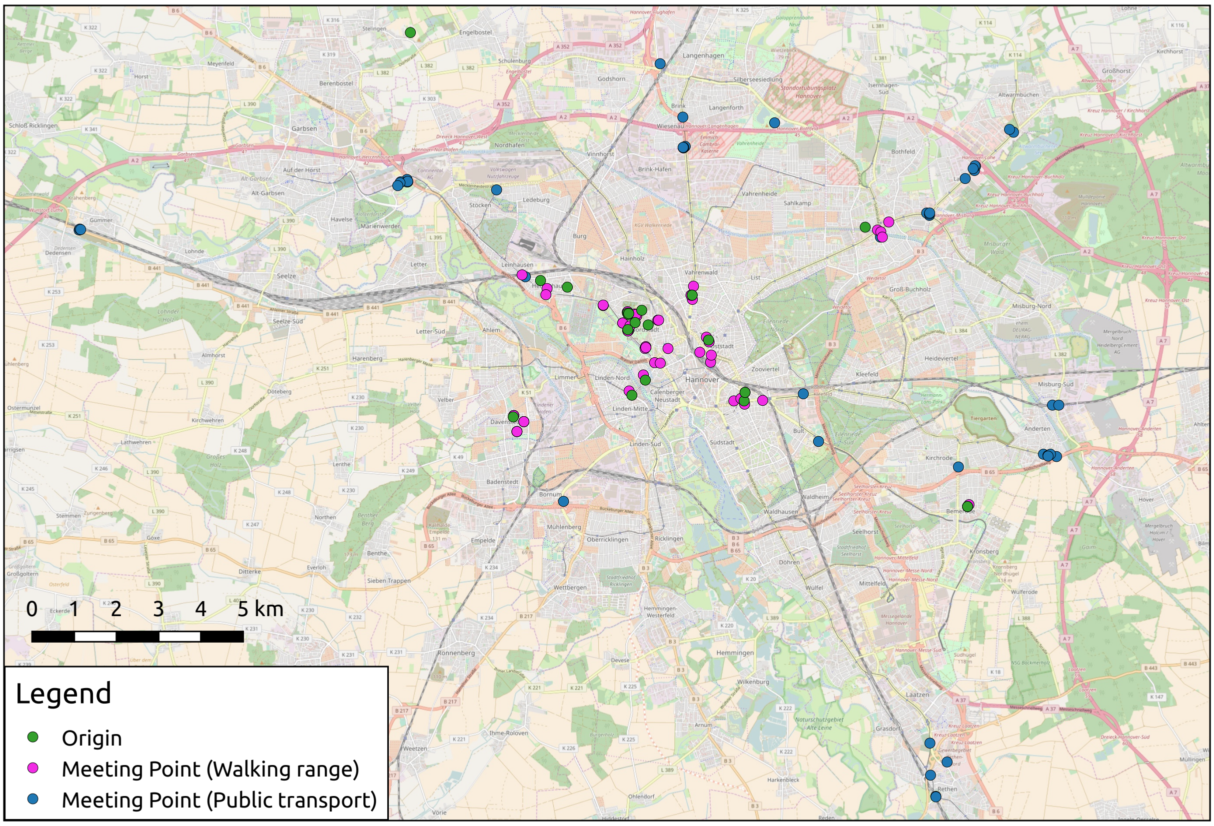
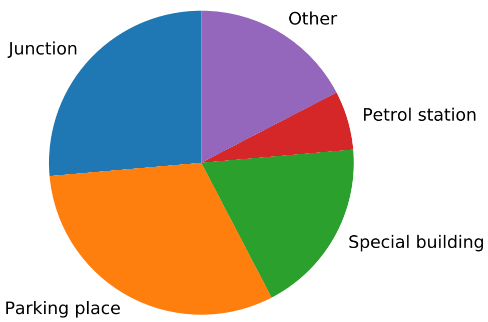


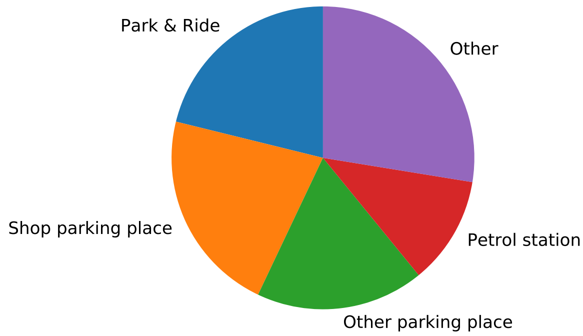
Figure 4.11: Stated origins and meeting point suggestions in the Hannover region.

Figure 4.12 shows the distribution of selections by the participants, for MPs within walking range (Figure 4.12a) and MPs along the driver paths, reachable via public transport (Figure 4.12b). Types other than those listed are mostly bus stops, tram or train stations, or POIs such as a stadium or a university. Not surprisingly, petrol stations and larger parking places play a much bigger role in the long-distance version.

Figures 4.13 and 4.14 visualize the participant input of the meeting point dialogues that pop up after clicking on the map, split into meeting points within walking range (Figure 4.13) and those for long-distance trips (Figure 4.14). Note that the questions about meeting point facilities (illumination, shelter, seating and heating) are represented using binary variables, since the participants were able to select only available or not available (or unknown). The user ratings are mostly positive, but this is not surprising, since people will hardly mark highly unsuitable locations. Illumination is available in most cases, shelter and seating rather more infrequently, and heating is available only in very few cases. Between step 2 and step 3 meeting points, there is a large difference in the number of users who were not able to answer the questions about the meeting point facilities. An explanation is that people are more aware of their neighbourhood, compared with more remote locations. In general, saliency and parking availability is much better rated in the public transport case. It may be rooted in the fact that meeting points in the vicinity of motorway exits are more



(a) *Walking range MP*



(b) *Public transport MP*

Figure 4.12: Stated types of meeting points.

often located at bigger parking places in front of restaurants, stores or at Park&Ride places. In inner city parts, large (and free) parking places are more scarce.

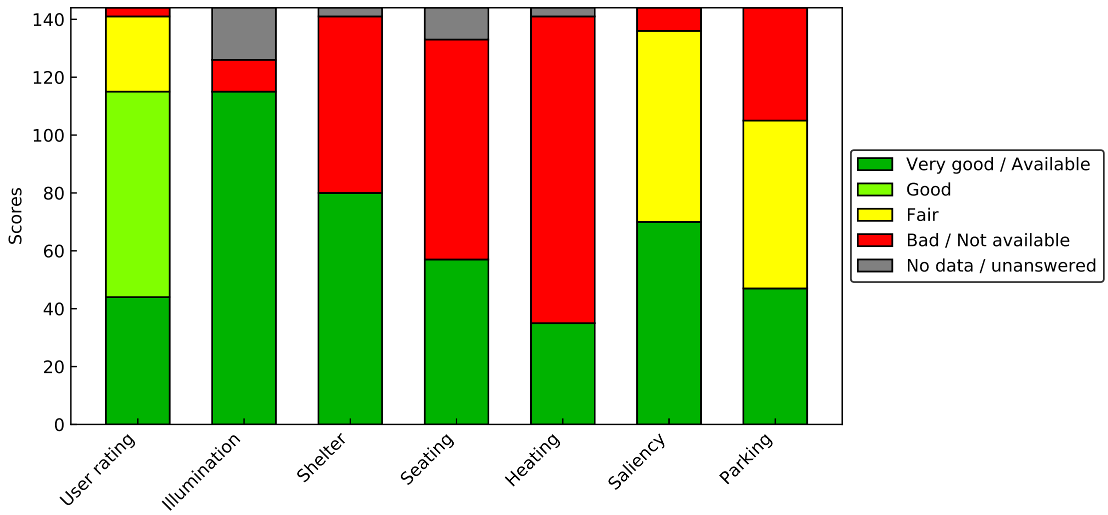


Figure 4.13: Stated properties of meeting points within walking range.

An interesting question is whether the walking distances to the meeting points vary significantly from those stated in the first, questionnaire-based survey (Section 4.1.2). Figure 4.15a shows the histogram of walking distances from the origins to the meeting points, based on passenger routing using Graphhopper Routing API. It can be clearly seen that there is a difference between the two surveys, with the map-based one being much closer to the results in the literature. The mode is at 400-500 m, and there are only few meeting points with a walking distance of more than 800 m. One meeting point having a walking distance of more than 4 km is probably an outlier, e.g. based on they assumption that the distance could also be covered by bike. Figure 4.15b shows a histogram of the detour times for the drivers as the difference between the direct route and the route including a meeting point. Again, Graphhopper Routing API was used to determine the driving times. The routes for detour calculation are the same as those that were shown to the participants in step 3. It turns out that most participants propose meeting points very close to the original driver route, with the majority of locations resulting in only 2-3 minutes detour. Only very few meeting points call for a detour of more than 8 minutes.

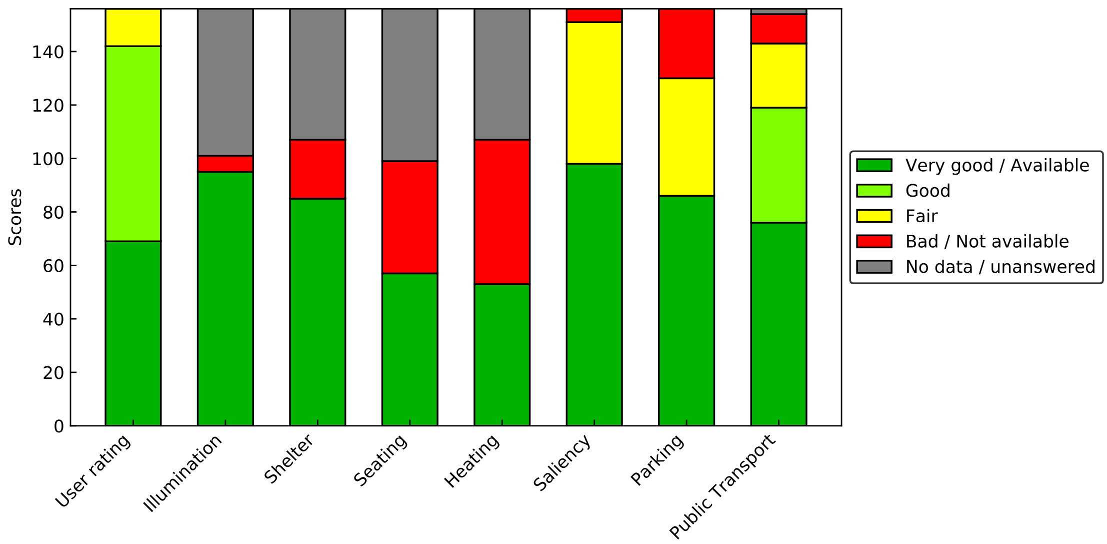
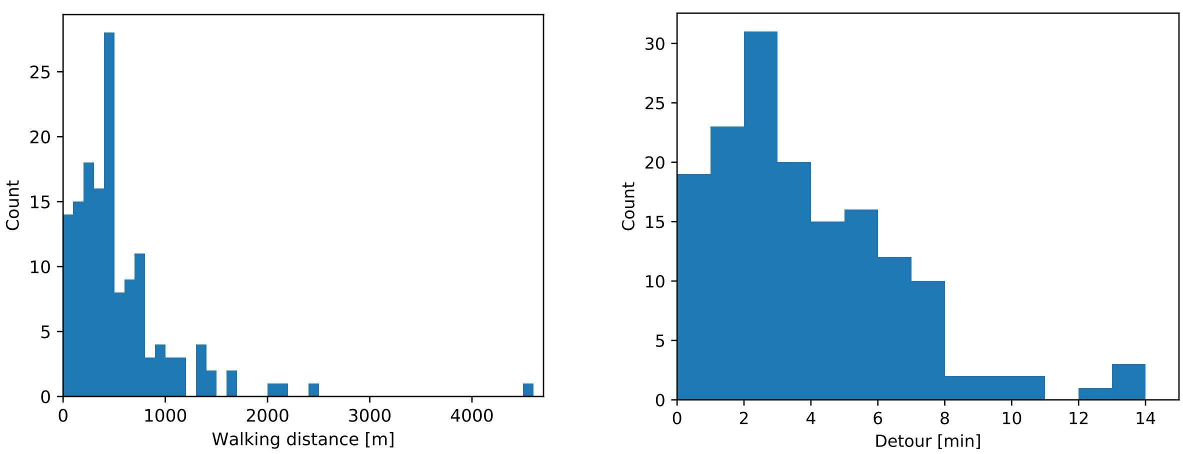


Figure 4.14: Stated properties of meeting points reachable by public transportation.



(a) Walking distances to meeting points. *(b) Driver detour times to meeting points.*

Figure 4.15: Histograms of revealed meeting point statistics.

5 Study area and data

For the research projects in Sections 6, 7 and 8, a common database is used, including a routing-enabled street network with meeting points, public transport connections and a set of artificial demands. To this end, the city of Braunschweig, Germany, is used as a spatial template for all the following experiments.

Braunschweig is a medium-sized city with $\sim 250\,000$ inhabitants and a typical European city structure: the centre is dominated by its historical core with an irregular street network and pedestrian precincts, surrounded by a ring road and some densely populated areas with a more regular street network. On the outskirts, the population density is lower, and there are some industrial areas. For vehicles, there is an outer ring formed by five motorways, with no motorway on the eastern side of the city. Figure 5.1 provides an overview of the city with its motorways, motorway exits, major roads and stations.

Figure 5.1: Overview of the investigation area Braunschweig city. Source: Municipality of Braunschweig (https://www.braunschweig.de/leben/stadtplanung_bauen/geoinformationen/rbe4_uk.html)

Nearly all geodata used in this work (streets, parking places, amenities, etc.) has been obtained from OpenStreetMap[1], a free, crowdsourced world map. There are two exceptions: a Level-of-Detail 1 (LoD1) building model of the city of Braunschweig, used for demand creation, and a point dataset containing street lamps in the city district. These datasets were obtained from the municipality of Braunschweig[2].

This chapter provides an overview of the data sources and data processing concerning the street network (Section 5.1), meeting points (Section 5.2), public transport (Section 5.3) and artificial demand (Section 5.4).

5.1 Street network

For the basic street network, all appropriate line geometries have been extracted from OpenStreetMap, transformed into a routing-enabled directed graph and stored in a PostGIS database[3]. In the following, the graph is denominated as $G = (V,E)$, consisting of a set of vertices $v \in V$ and a set of edges $(u,v) \in E$. Each edge $(u,v) \in E$ has an associated non-negative length $d(u,v)$ and further informations such as the street name and speed limits. Vehicle driving times $t^{driv}(u,v)$ are derived from the speed limits and the length of the edge, multiplied by a factor of 0.9 to simulate the retarding effect of traffic lights and congestion. One-way streets have an infinite high cost for driving in the opposite direction. Passenger walking times $t^{walk}(u,v)$ are based only on the edge lengths, using a constant walking speed of 4.8 km/h, corresponding to the findings of Millward et al. (2013) for active-transport walking trip speed. The traversing of footpaths, cycle ways and stairways is prohibited for vehicles. Likewise, pedestrians are not allowed to walk on major roads or motorways.

The street network graph is further extended by meeting points (Section 5.2) and demand connections (Section 5.4). The final graph has a total of 88 381 nodes and 99 497 edges. For routing requests on the graph, the open source routing engine pgRouting[4], an extension of the PostGIS geodatabase, is used.

5.2 Meeting point candidates

In addition to the street network, a set of meeting point candidates (MPC) $\mu \in M$ is included. They also serve as divergence point candidates $\delta \in M$. The meeting point candidates are automatically extracted from OpenStreetMap by a GIS workflow, using the command line tools osmconvert[5] and osmfilter[6]. In order to ensure safety and convenience aspects, for example boarding places with reduced traffic, possibilities for parking and easily recognizable places, the candidate locations are limited to the following selection:

- Publicly accessible parking places without parking fees,
- Side road intersections (with all adjacent roads having a maximal speed of $\leq$ 30 km/h),
- Turning areas (mostly at the end of a cul-de-sac),
- Petrol stations.

[1]`http://www.openstreetmap.org`
[2]`https://www.braunschweig.de/leben/stadtplanung_bauen/geoinformationen/geodaten.html` (01.07.2017)
[3]`http://postgis.net/` (09.11.2017)
[4]`http://pgrouting.org/` (01.07.2017)
[5]`http://wiki.openstreetmap.org/wiki/Osmconvert` (01.07.2017)
[6]`http://wiki.openstreetmap.org/wiki/Osmfilter` (01.07.2017)

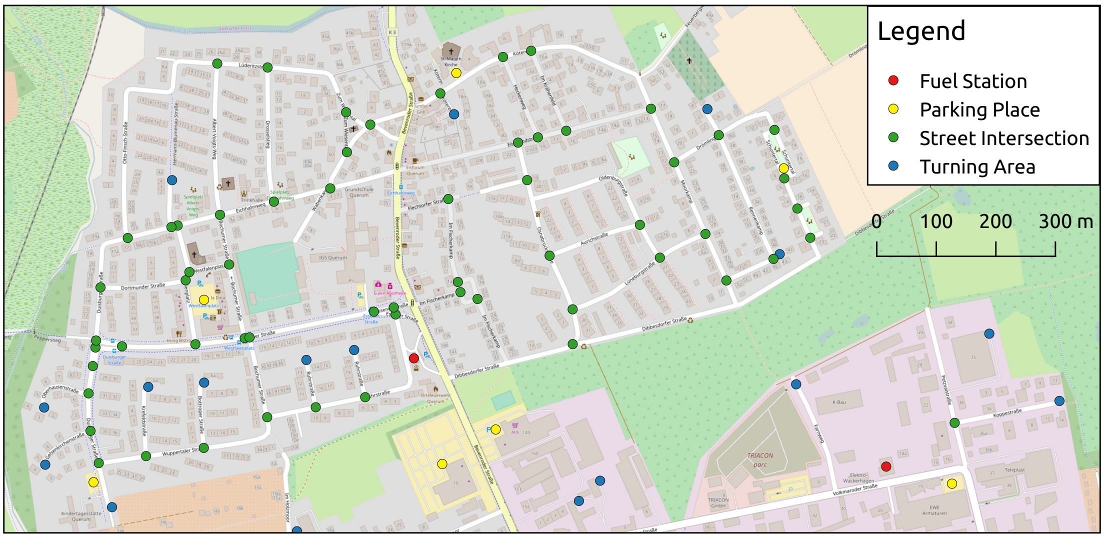


Figure 5.2: Map extract with meeting point candidates in Querum, a typical suburb of Braunschweig.

In practice, frequently used meeting places surely include also the curb or public transport stops. However, this possibility is intentionally disregarded, as it would be irresponsible to encourage people to meet in this manner. Further, it is not advisable according to road traffic regulations in most countries.

If parking areas and petrol station areas are originally mapped as polygon features in OpenStreetMap, they are first converted to point features using the centroid. Each candidate location is connected to the street network G with an edge between the meeting point location and the closest point on the closest edge. If the closest edge is not reachable by vehicles (such as a footpath), a second edge is inserted at the closest drivable edge. The same procedure is applied for edges not accessible on foot. Figure 5.2 shows a map extract with some meeting points from the four categories. In addition, the meeting point candidates are checked concerning their reachability for drivers and passengers, determined by a one-to-all routing from a central location. As an example, parking places located on private property may not be accessible for drivers, hence they are removed.

In total, 3475 meeting points have been extracted within the investigation area. In relation to the total investigation area size of 193 km^2 (area of Braunschweig municipality), the MPC density is approximately 18 per km^2. In dense urban areas the density can be up to 40 MP per km^2. The observed mean distance to the nearest neighbouring MP is approximately 70 m. For experiment 7, only parking places and petrol stations are used, resulting in a total of 705 meeting point candidates. The simulation in Section 6 requires further knowledge about meeting point properties. Hence, the vicinity of each meeting point is scanned for the following facilities:

– Illumination

– Shelter

– Seating

The occurrence of these facilities is represented as a binary value $b^{facl}(\mu) \in \{-1,1\}$, stating if a corresponding object is within a threshold of 25 m (1) or not (-1). For illumination, the street lamp dataset was used to determine the closest light source. Objects that give shelter are, for example, canopies, telephone booths, bus/tram shelters or various other types appropriately tagged with the attribute *shelter* in OpenStreetMap.

Similarly, seating objects are selected by the OSM tag *seating*. Here, frequently identified objects are park benches and bus/tram stops. The parking quality $b^{park}(\mu)$ is assessed by a manual assignment. Parking places and turning areas have the best rating ($b^{park}(\mu) = 3$), fuel stations are intermediate ($b^{park}(\mu) = 2$) and street intersections are inferior ($b^{park}(\mu) = 1$).

5.3 Public transport network

The public transport system of Braunschweig, operated by *Braunschweiger Verkehrs GmbH*[7], currently includes a network of 5 tram lines and 37 bus lines. The main public transport lines are in operation all day long, with a night break from approx. 2 am to approx. 4 am.
The public transport data is provided by the Connect GmbH[8]. Following the notation of Müller-Hannemann et al. (2007), the data is represented as a set of stops $\mathcal{S}$, a set of vehicle lines $\mathcal{Z}$ (e.g. a tram line) and a set of elementary connections $\mathcal{C}$. A connection element $c \in \mathcal{C}$ is then a 5-tuple $c = (z{,}s^+{,}s^-{,}t^+{,}t^-)$ that can be interpreted as vehicle z leaving stop s^+ at time t^+ and arriving stop s^- at time t^-.

Routing requests for public transport connections are processed by an instance of the open source multi-modal routing engine OpenTripPlanner[9], based on the OpenStreetMap street graph and timetable data of Braunschweig.

5.4 Demand

In addition to the routing base, a set of 40 000 artificial customer requests Θ was created as simulation input for various experiments. The customers were generated randomly, based on the pool of available origin and destination buildings (see below). Each customer $\theta \in \Theta$ is assigned an origin node $v^+(\theta)$, a desired destination node $v^-(\theta)$ and a time of earliest departure time at the origin $t^+_\uparrow(\theta)$, yielding a triplet $\theta = \left(v^+{,}v^-{,}t^+_\uparrow\right)$ as a basic property.

Spatial distribution

Potential origins are all residential buildings from the LoD1 building model with a size of more than 100 m^2. The size threshold is applied to prevent small huts (e.g. in allotment gardens) being chosen. Likewise, workplace buildings from the LoD1 dataset have been used as potential customer destinations. The probability of a building being chosen as origin or destination depends on its volume, i.e. big buildings have a higher chance of being selected than small buildings, following the assumption that there is a correlation between the volume and the available living and working spaces. To prevent huge factory buildings being chosen disproportionately often, the volume for destination buildings was capped at a threshold of 10 000 m^3. Furthermore, trip requests with a bee line between origin and destination of less than 2 000 m have been removed, since the passengers are assumed to walk or cycle the whole path. All buildings considered are connected to the street network, to model the whole path of the user (Figure 5.3).

In total, 26 845 potential home and 2 615 potential work locations have been added to the network. Figure 5.4 visualizes the spatial distribution of customers within the investigation area. Customers without a

[7] `http://www.verkehr-bs.de/`
[8] `http://www.connect-fahrplanauskunft.de`
[9] `http://www.opentripplanner.org/` (01.07.2017)

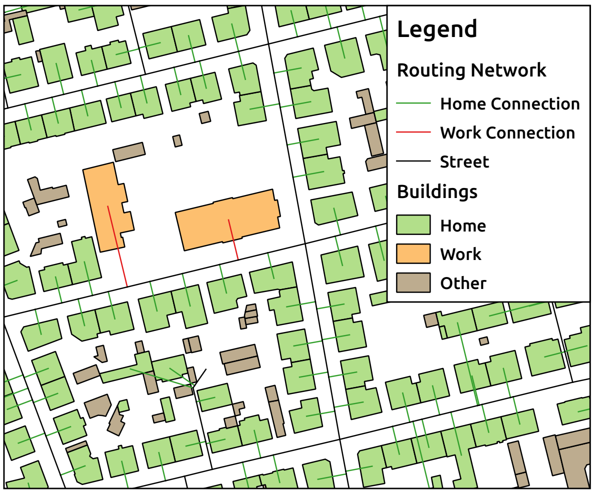
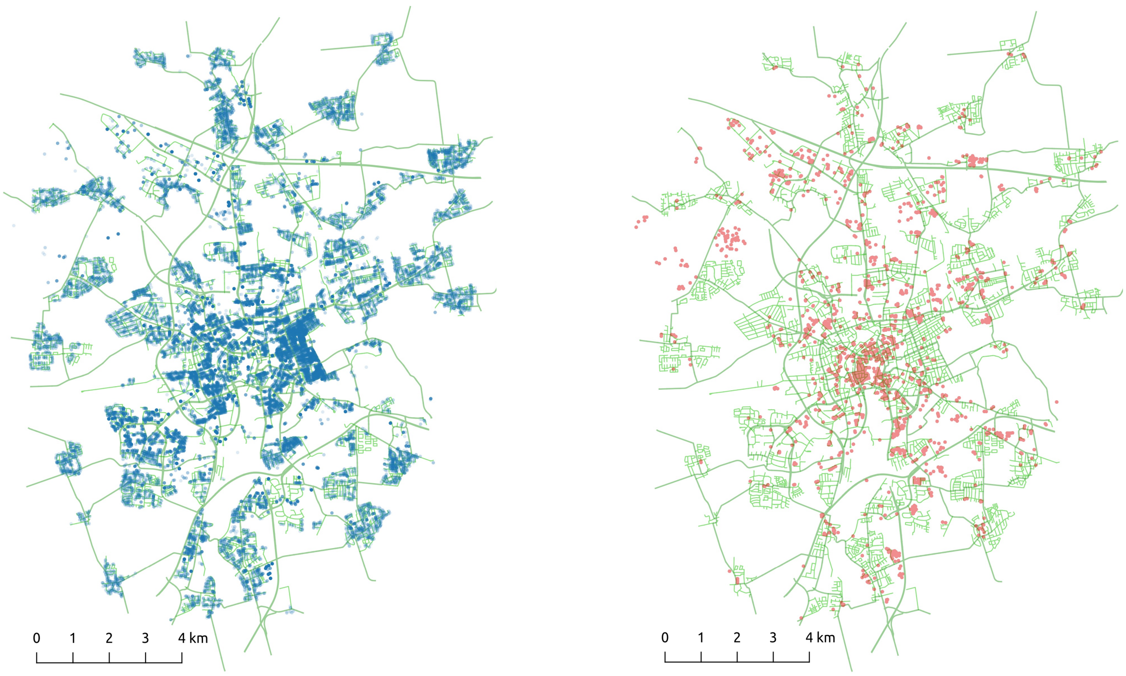


Figure 5.3: Map extract with customer origins and destinations in a typical suburb of Braunschweig.

vehicle are modelled to begin (and end) their trip directly at the corresponding building, while drivers begin (and end) at the closest node accessible by vehicles.

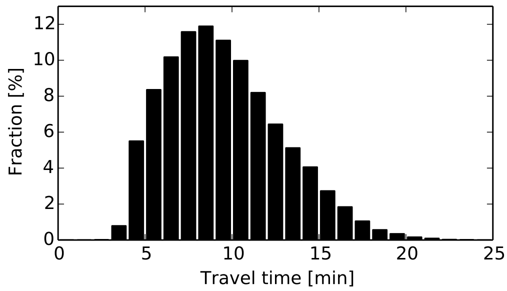


(a) Customer origin locations *(b) Customer destination locations*

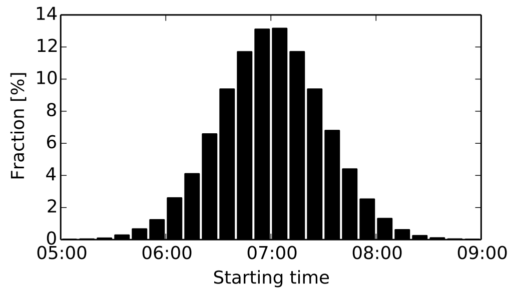
Figure 5.4: Distribution of customer origins and destinations within the investigation area.

Temporal distribution

The temporal distribution of the requests follows a Gaussian distribution centred at 07:00 am with a standard deviation of 30 minutes to simulate a busy morning commute peak (Figure 5.5b). Figure 5.5a visualizes the distribution of direct travel times with a vehicle from an origin to a destination. Since Braunschweig is a relatively small city, the travel times are also relatively short. The average is 9:33 minutes; the fastest



(a) *Distribution of direct vehicle travel times*



(b) *Distribution of starting times*

Figure 5.5: Statistics about randomly generated customer demand.

customer can reach the destination within 3.07 minutes, and the customer with the longest travel time needs 22:39 minutes.

Personal preferences

Each customer is further equipped with a set of personal preferences regarding the importance of meeting point facilities, such as shelter, seating or illumination. The distribution of importance values is based on the questionnaire-based survey (chapter 4.1). The survey participants could assign to various meeting point facilities such as seats, shelter, security or illumination an importance value between 0 (*Not important*) and 4 (*Very important*). In this work, only the meeting point facilities illumination, shelter and seating during summer times are considered for the demand, since warmth and security are not easy to identify in map data. A unique location is stated as most important to the customers, but since the meeting points used are estimated as clearly distinguishable locations, this aspect can be regarded as fulfilled. The used distribution is listed in table 5.1. Each passenger is hence assigned three concern values τ^{seat},τ^{shlt},$\tau^{\text{illu}} \in \{0,0.25,0.5,0.75,1\}$ with zero indicating no concern and one indicating a high concern about the feature. The concern values are assigned with a weighted random selection based on the distribution in table 5.1.

Table 5.1: Pattern for the distribution of meeting point facility concern values.

Concern value	Illumination	Seating	Shelter
0	14,4 %	18,5 %	10,3 %
0.25	17,5 %	24,7 %	20,6 %
0.5	25,7 %	23,7 %	26,8 %
0.75	31,9 %	27,8 %	32,9 %
1	10,3 %	5,1 %	9,2 %

6 Meeting points for intra-urban ride-sharing

In this chapter, the impact of meeting points for intra-urban ride-sharing is investigated through several computational experiments. It focuses particularly on the differences between conventional door-to-door ride-sharing and ride-sharing using meeting points (Section 6.3.2), the influence of convenience facilities (Section 6.3.3) and the number of necessary meeting points in a city (Section 6.3.4). Parts of the results of this chapter are published in Czioska et al. (2017).

6.1 Motivation

As already stated in Section 3.2, the actual determination of eligible meeting points in a real city environment is often neglected in simulation experiments. For example, the work of Stiglic et al. (2015) highlights the benefits of meeting points in a ride-sharing scenario, but the locations are randomly placed on the Euclidean plane. In the experiments from Häll et al. (2008), the meeting points are based on a regular grid, which in a second step is matched on the street network. Another approach is to consider all nodes in the street network as potential meeting point candidates, regardless of their feasibility (Aissat & Oulamara 2014a, 2015a; Balardino & Santos 2016; Rigby et al. 2013). An exception is the work by Chen et al. (2016), who consider a carpooling scenario and use real-world carpool parking lot positions as meeting point candidates, but focus more on long-distance commuting trips. Hence, there is a lack of realistic ride-sharing simulations considering real-world meeting points, which this chapter is aiming to close.

In most existing approaches, the most common objectives for the matching are travel time, distance savings and/or the matching rate. However, it can be argued that, for customers, not only time and distance savings play a role, but also a specific level of convenience during the ride. Certainly, comfort during the actual ride is of major importance. Common factors that influence the convenience of a ride include safety, driving skills, the friendliness of the driver, permission to smoke on board, or preferences in terms of gender, chattiness or music. In the literature, social networks are often used to find *similar* ride-sharing partners, with the goal of enhancing trust and improving personal satisfaction (Chaube et al. 2010; Yousaf et al. 2014). However, these aspects are not within the main focus of this research, and so are not considered in this work. On the other hand, convenience in terms of meeting point quality also plays a role. This includes parking quality (accessibility, safety while boarding and alighting, fees) as well as facilities for the passengers while waiting in bad weather or during darkness. Then, important facilities are shelter, seating possibilities and illumination. In the user survey (chapter 4), participants were asked about the importance of these facilities to them personally, and the results are incorporated into the random customer creation (chapter 5.4). This dataset is used in this chapter within a computational experiment, in order to investigate how the incorporation of user preferences influences the ride-sharing metrics.

In addition, the question of how ride-sharing usage is affected by a changing number of available meeting points is considered in this chapter. For this, the number of meeting point candidates is artificially limited, representing the case that not all locations are available as a meeting place. The results are especially interesting for traffic management authorities, to estimate how many official meeting points would actually be necessary to satisfy the demand in a city.

6.2 Basic matching problem

The basic matching procedure is common for all experiments in this chapter, and involves the assignment of passengers to drivers at meeting and divergence points. An important restriction is that only one meeting and divergence point is allowed for each driver. This reduces the problem to a pure matching problem; the vehicle routing part can be neglected, similarly to the approach by Stiglic et al. (2015). A driver can accommodate multiple passengers, but in that case all passengers must board at the same meeting point and alight at the same divergence point. Furthermore, all requests are assumed to be known in advance (*static scenario*), and the driver/passenger roles are fixed, which further reduces the complexity of the problem.

The city of Braunschweig is used as a template for the experiments, including meeting points, street network and customer demand (Chapter 5). The set of customers Θ is split into equally-sized sets of drivers Ψ and passengers P, so that a 1:1 rate is always given. All the following notations of the customer amount refer to the combined requests of drivers and passengers; a customer amount of e.g. 6 000 indicates 3 000 drivers and 3 000 passengers. The notations used in this chapter are summarized in table 6.1.

Notation	**Unit**	**Description**
$\mu \in M$	-	Meeting point candidates
$\delta \in D$	-	Divergence point candidates
$\theta \in \Theta$	-	Set of requests (driver and passenger)
$\psi \in \Psi$	-	Driver requests
$\rho \in P$	-	Passenger requests
$\xi \subset P$	-	Set of multiple passenger requests
$\zeta \in Z$	-	Customer trip / match
v^{+}	-	Origin node
v^{-}	-	Destination node
t^{+}	s	Departure time
t^{-}	s	Arrival time
$t_{\uparrow}$	s	Earliest / minimum time
$t_{\downarrow}$	s	Latest / maximum time
Δt^{drive}	s	Driving time
$\Delta t^{\text{drive}}_{\leftrightarrow}$	s	Direct travel time from origin to destination
Δt^{walk}	s	Walking time
Δt^{total}	s	Total travel time
$\Delta t^{\text{wait}}_{*}$	s	Passenger early arrival time (precautionary)
$\Delta t^{\text{serve}}_{*}$	s	Service time (for boarding and alighting)
$\Delta t^{\text{flex}}_{*}$	s	Departure time flexibility
$\Delta t^{\text{detr}}_{*}$	s	Acceptable detour time
r^{detr}_{*}	-	Acceptable detour time ratio
s^{walk}_{*}	m/s	Walking speed
d^{walk}_{*}	m	Maximum passenger walking distance
q^{vehi}_{*}	-	Vehicle capacity
$e \in E$	-	Edges of the bipartite graph
x_e	-	Binary decision variable
w_e	-	Weight
ν_e	-	Number of included passengers
γ^{penl}_{*}	-	Penalty for non-matched passengers

Table 6.1: Notation used in this chapter.

6.2.1 Mathematical model

The mathematical model is constructed as follows. A basic customer request θ consists of a triplet $\left(v^{+}, v^{-}, t_{\uparrow}^{+}\right)$, as defined in chapter 5.4. In addition, a time of latest arrival at the destination $t_{\downarrow}^{-}$ is attached to each customer, to provide a closed time window. This time is computed differently for drivers and for passengers.

For drivers $\psi \in \Psi$, the maximum travel time $\Delta t_{\downarrow}^{\text{total}}(\psi)$ is dependent on the *direct travel time* $\Delta t_{\leftrightarrow}^{\text{drive}}(\psi)$, representing the necessary driving time from origin to destination with a car. In order to pick up and drop off passengers, each driver is assumed to accept a particular detour. The extra detour time $\Delta t_{\downarrow}^{detr}(\psi)$ is limited by a ratio r_{*}^{detr}, defining the maximal travel time in relation to the direct travel time. A value of $r_{*}^{\text{detr}} = 1.5$ for example would allow the driver a 5 minute detour if the direct travel time is 10 minutes. To avoid unreasonably short or long additional travel times, the detour extra time has lower and upper bounds. On the one hand, a minimal detour time of $\Delta t_{*\uparrow}^{\text{detr}}$ has to be accepted by every driver, regardless of the actual direct travel time. On the other hand, the extra detour time can not exceed $\Delta t_{*\downarrow}^{\text{detr}}$. The maximal detour time is thus defined as:

$$\Delta t_{\downarrow}^{detr}(\psi) = \begin{cases} \Delta t_{*\uparrow}^{\text{detr}} & \text{if } t_{\leftrightarrow}^{\text{drive}}(\psi) \cdot r_{*}^{\text{detr}} \leq \Delta t_{*\uparrow}^{\text{detr}}, \\ \Delta t_{*\downarrow}^{\text{detr}} & \text{if } t_{\leftrightarrow}^{\text{drive}}(\psi) \cdot r_{*}^{\text{detr}} \geq \Delta t_{*\downarrow}^{\text{detr}}, \\ t_{\leftrightarrow}^{\text{drive}}(\psi) \cdot r_{*}^{\text{detr}} & \text{else} \end{cases} \tag{6.1}$$

The maximum travel time also includes twice the service time to pick up and drop off a passenger:

$$\Delta t_{\downarrow}^{\text{total}}(\psi) = \Delta t_{\leftrightarrow}^{\text{drive}}(\psi) + \Delta t_{\downarrow}^{detr}(\psi) + 2 \cdot \Delta t_{*}^{\text{serve}} \tag{6.2}$$

For passengers $\rho \in P$, the maximum travel time $\Delta t_{\downarrow}^{\text{total}}(\rho)$ is naturally limited by the maximum acceptable walking time to and from a meeting point d_{*}^{walk}. The longest possible travel time includes twice the longest possible walking time, twice the service time for boarding and alighting, and the direct travel time between origin and destination:

$$\Delta t_{\downarrow}^{\text{total}}(\rho) = 2 \cdot d_{*}^{\text{walk}} \cdot s_{*}^{\text{walk}} + 2 \cdot \Delta t_{*}^{\text{serve}} + \Delta t_{\leftrightarrow}^{\text{drive}}(\rho) \tag{6.3}$$

Although this is not completely exact, since the driving time between meeting and divergence point can be longer than the direct travel time from origin to destination, it still yields a meaningful upper bound of acceptable travel time.

To allow a further degree of flexibility, all customers (drivers and passengers) are assumed to allow departure within a certain time window, determined by a *departure time flexibility* parameter $\Delta t_{*}^{\text{flex}}$. This value specifies the time span within which the departure can be shifted. This allows the determining of the latest arrival time for each customer at the desired destination:

$$t_{\downarrow}^{-}(\theta) = t_{\uparrow}^{+}(\theta) + \Delta t_{*}^{\text{flex}} + \Delta t_{\downarrow}^{\text{total}}(\theta) \tag{6.4}$$

The computation of the latest acceptable arrival time is also visualized in Figure 6.1.

A *match* consists of a a single driver $\psi \in \Psi$, a set of passengers $\xi = \{\rho_1, \rho_2, \cdots\} \subset P$, a meeting point $\mu \in M$ as well as a divergence point $\delta \in D$. This is hereinafter called a *trip candidate* $\zeta = (\psi, \xi, \mu, \delta)$. While a trip is limited to one driver, it can include multiple passengers, as long as they do not exceed the vehicle capacity ($|\xi| \leq q_{*}^{\text{vehi}}$).

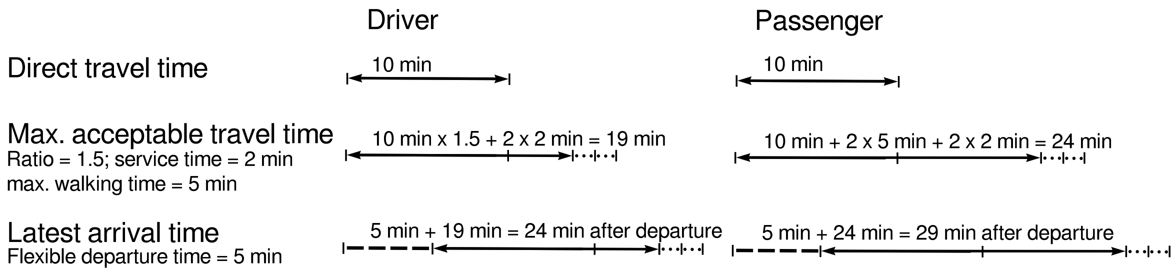


Figure 6.1: Time budget of drivers and passengers with exemplary times.

A ride-sharing procedure has the following sequence: firstly, the passengers arrive at the meeting point prior to the expected arrival time of the drivers. The time that the passengers arrive earlier is called *waiting time* and is defined in the settings ($\Delta t_*^{\mathrm{wait}}$). The waiting time should imitate reality, where passengers usually arrive earlier at a meeting point than the driver, just as at bus stops. Then they board the vehicle, drive to the divergence point, alight, and continue to their destinations. A trip is only considered feasible if all time constraints are satisfied. This means that the driver and all passengers can arrive the destination prior to their latest acceptable arrival time.

For both the boarding and alighting procedure, a *service time* $\Delta t_*^{\mathrm{serve}}$ is added. Figure 6.2 illustrates the sequence and the time constraints.

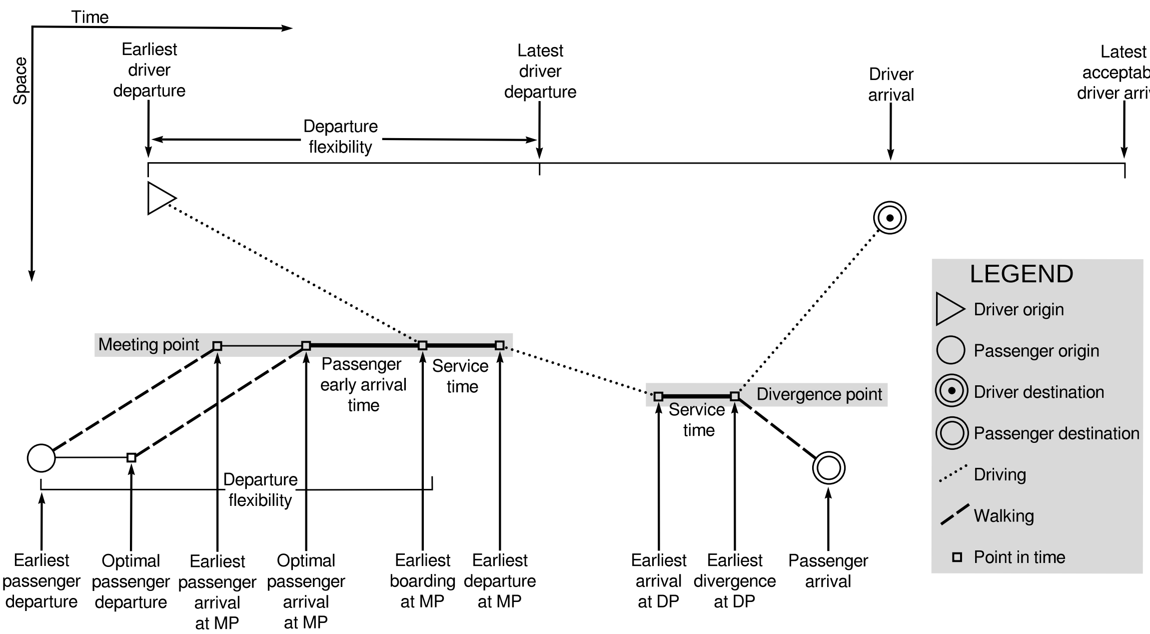


Figure 6.2: Sequence of a shared ride with time notations.

In order to ensure time feasibility, every trip candidate ζ is checked for time validity. Firstly, the earliest possible common passenger boarding time at the meeting point is determined:

$$t_{\uparrow}^{+}(\mu,\xi) = \max\left(t_{\uparrow}^{+}(\rho) + \Delta t^{\mathrm{walk}}\left(v^{+} \to \mu\right) + \Delta t_*^{\mathrm{wait}}\ \forall\, \rho \in \xi\right) \tag{6.5}$$

Secondly, the earliest common boarding time at the meeting point for all, including the driver, is calculated:

$$t^{+}_{\uparrow}(\mu,\psi,\xi) = \max\left(t^{+}_{\uparrow}(\psi) + \Delta t^{\text{drive}}\left(v^{+} \to \mu\right), t^{+}_{\uparrow}(\mu,\xi)\right) \tag{6.6}$$

Next, the travel and service times are added, to calculate the earliest possible divergence time at the divergence point:

$$t^{-}_{\uparrow}(\delta,\psi,\xi) = t^{+}_{\uparrow}(\mu,\psi,\xi) + \Delta t^{\text{serve}}_{*} + \Delta t^{\text{drive}}(\mu \to \delta) + \Delta t^{\text{serve}}_{*} \tag{6.7}$$

Lastly, the latest acceptable common arrival time at the divergence point for all passengers and the driver is determined by:

$$t^{-}_{\downarrow}(\delta,\psi,\xi) = \min\left(t^{-}_{\downarrow}(\psi) - \Delta t^{\text{drive}}\left(\delta \to v^{-}\right), \min\left(t^{-}_{\downarrow}(\rho) - \Delta t^{\text{walk}}\left(\delta \to v^{-}\right) \forall \rho \in \xi\right)\right) \tag{6.8}$$

If $t^{-}_{\uparrow}(\delta,\psi,\xi) \leq t^{-}_{\downarrow}(\delta,\psi,\xi)$, all customers can reach the destination on time, and the total travel times can be calculated for the driver and the passengers:

$$\Delta t^{\text{total}}(\psi) = \left(t^{-}_{\uparrow}(\delta,\psi,\xi) + \Delta t^{\text{drive}}\left(\delta \to v^{-}\right)\right) - \left(t^{+}_{\uparrow}(\mu,\psi,\xi) - \Delta t^{\text{drive}}\left(v^{+} \to \mu\right)\right) \tag{6.9}$$

$$\Delta t^{\text{total}}(\rho) = \left(t^{-}_{\uparrow}(\delta,\psi,\xi) + \Delta t^{\text{walk}}\left(\delta \to v^{-}\right)\right) - \left(t^{+}_{\uparrow}(\mu,\psi,\xi) - \Delta t^{\text{wait}}_{*} - \Delta t^{\text{walk}}\left(v^{+} \to \mu\right)\right) \tag{6.10}$$

If the travel time for the driver is less than the maximum detour riding time $\left(\Delta t^{\text{total}}(\psi) \leq \Delta t^{detr}_{\downarrow}(\psi)\right)$, the trip is considered feasible.

6.2.2 Matching problem

The algorithm itself is formulated as a single driver, multiple passengers ride-share matching problem, similar to the work of Agatz et al. (2011) and Stiglic et al. (2015). This problem can be solved as a maximum weigh bipartite matching problem with side constraints (chapter 2.3.1). The bipartite graph is constructed as follows. For each driver $\psi \in \Psi$ and each passenger $\rho \in P$, a node is created. Subsequently, nodes are inserted for each possible set of passengers ξ who could travel together. Then, edges $e \in E$ are inserted for every feasible trip $\zeta = (\psi,\xi,\mu,\delta)$, each connecting a driver with a passenger or a set of passengers. Each edge has three values associated with it: a binary decision variable x_e, indicating if the edge is included in the final optimal solution ($x_e = 1$) or not ($x_e = 0$), a weight w_e, which provides a measure of the solution quality, and the number of passengers included ν_e.

The weight w_e is a normalized value between zero and one, with zero indicating a perfect match and one indicating a very poor match. It is composed of the single weights of the customers involved in a trip.

In the basic version, the customer weight is dependent only on the necessary additional time, compared with the direct travel time $\Delta t^{\text{drive}}_{\leftrightarrow}$. For the drivers, the additional time consists only of the detour driving time. For passengers, the additional time is composed of the detour driving time, the waiting time for fellow travellers, and the walking time. The personal customer weight $w(\theta)$ is calculated as:

$$w(\theta) = \frac{\Delta t^{\text{total}}(\theta) - \Delta t^{\text{drive}}_{\leftrightarrow}(\theta)}{\Delta t^{\text{total}}_{\downarrow}(\theta) - \Delta t^{\text{drive}}_{\leftrightarrow}(\theta)} \tag{6.11}$$

Apparently, the weight is zero for no additional time and one if the travel time is equal to the maximum allowed travel time. Values above one are, in theory, possible, but are capped to one. The final edge weight for a trip is then assembled as an average of the personal customer weights of all trip participants:

$$w_e = \frac{w(\psi) + \sum_{\rho \in \xi} w(\rho)}{1 + |\xi|} \tag{6.12}$$

The single driver, multiple rider ride-share matching problem can then be formulated as

$$\min \left(\sum_{e \in E} x_e \cdot w_e \right) + \left(\gamma^{\text{penl}}_{*} \cdot \left(|P| - \sum_{e \in E} x_e \cdot \nu_e \right) \right) \tag{6.13}$$

subject to

$$\sum_{e \in E_\psi} x_e \leq 1 \qquad \forall \psi \in \Psi \tag{6.14}$$

$$\sum_{e \in E_\rho} x_e \leq 1 \qquad \forall \rho \in P \tag{6.15}$$

$$x_e \in \{0{,}1\} \qquad \forall e \in E \tag{6.16}$$

The objective function (6.13) minimizes a value composed of two parts: the weights of the accepted trips and the number of unmatched passengers. The variable γ^{penl}_{*} is a penalty term that can be adjusted for optimization. With any value above one, the cost for not matching a passenger is higher than matching a passenger with a bad weight; hence, the primary target is to maximize the number of participants. With a value between zero and one, the acceptance of matching a passenger can be made dependent on the trip quality, defined by its weight. This concept is different from the approach of Stiglic et al. (2015), who apply a two-step optimization with different objective functions (lexicographical goal programming). The constraints (6.14) and (6.15) ensure that each driver and passenger is included in at most one trip, and constraint (6.16) defines the binary decision values.

The procedure for constructing the set of feasible trips is outlined as high-level Algorithm 1. The N-COMBINATIONS function mentioned yields all possible combinations of a given set (the *powerset*) until the size of n, e.g. N-COMBINATIONS(2,[a,b,c]) = [a], [b], [c], [ab], [ac], [bc]. Since every driver is processed separately, the algorithm can be executed in parallel.

The dominating factor in this algorithm concerning the complexity is the size of the possible solutions yielded by the *n-Combinations*-function. The number of combinations C can be determined by the following formula, with n as the total number of items, in this case all passengers, and r as the sample size, in this case the vehicle capacity q^{vehi}_{*}:

$$C(n{,}r) = \frac{n!}{(n-r)! \cdot r!} = \binom{n}{r} \tag{6.17}$$

Algorithm 1 Feasible trip generation

```
Initialise trip set: 𝒵 ← {}
for ψ ∈ Ψ do
    for ξ ∈ N-COMBINATIONS(q_*^vehi, P) do
        if Combination (ψ,ξ) is obviously infeasible regarding time then    ▷ Quick check
            Skip and continue with next combination
        else
            Initialise temporary result set: 𝒯 ← {}
            for μ ∈ M_ξ do                                          ▷ All common meeting points
                for δ ∈ D_ξ do                                      ▷ All common divergence points
                    Compute t_↑^-(δ,ψ,ξ)                            ▷ Earliest possible arrival at DP
                    Compute t_↓^-(δ,ψ,ξ)                            ▷ Latest possible arrival at DP
                    if t_↑^-(δ,ψ,ξ) ≤ t_↓^-(δ,ψ,ξ) then             ▷ Trip is feasible
                        Compute weight w of the solution
                        𝒯 ← 𝒯 ∪ (w,ψ,ξ,μ,δ)
                    end if
                end for
            end for
        end if
        𝒵 ← 𝒵 ∪ min(𝒯)                                             ▷ Add trip with minimum weight
    end for
end for
return 𝒵
```

This term can be rewritten as

$$C(n,r) = \frac{1}{\prod_{i=1}^{r} i} \prod_{i=0}^{r-1} (n-i) \tag{6.18}$$

This term can, in turn, be expanded to a polynomial of degree r. Hence, the complexity of this part of the algorithm can be expressed as $\mathcal{O}(n^r)$, considering only the term with the highest exponent. In total, the complexity of the whole algorithm is $\mathcal{O}(kn^r)$, with k as the number of drivers, n as the number of passengers and r as the vehicle capacity. The algorithm has a polynomial complexity, but the exponent is dependent on the maximum vehicle capacity. As an example, for one driver with three spare seats and 100 passengers, 166 750 different combinations are possible.

In practice, this makes it necessary to skip non-feasible combinations, for example because of non-overlapping time windows, at an early stage, in order to limit the computation time to a reasonable extent. In the loop, the time windows and overlapping meeting points are therefore checked initially, so that non-relevant combinations can be skipped before they are investigated in more detail. The algorithm running time is also sensitive to customer time flexibility, since with greater flexibility the time windows allow more options, which have to be checked.

6.3 Simulation experiments

In this section, the previously described algorithm is executed and modified to allow a view on the impacts of using meeting points for shared rides. In the first Section (6.3.1), a *baseline scenario* is presented, showing the results from a simulation with a standard parameter setting. These results are then compared with the results from a door-to-door service (Section 6.3.2) and a convenience-based matching (Section 6.3.3). In addition, a simulation with a reduced set of meeting points is performed (Section 6.3.4) to show how the number and selection of meeting points influences ride-sharing.

The algorithms for deriving the following results are implemented in Python 2.7.13[1] using Numpy extension[2]. For the MIP optimization, Google Optimization Tools[3] is used, which in turn wraps the *Coin-or branch-and-cut (CBC)*[4] solver, an open-source mixed integer programming solver written in C++.

6.3.1 Baseline scenario

In the baseline scenario, the proposed matching algorithm with meeting points (Section 6.2) is executed with a standard parameter setting (Table 6.2). The algorithm is applied to different demand instances, ranging from 400 (200 drivers and 200 passengers) to 6 000 (3 000 drivers and 3 000 passengers).

Table 6.2: Basic parameter setting.

Notation	**Unit**	**Description**	**Value used for simulation**
d_*^{walk}	m	Maximum passenger walking distance	800 m
$\Delta t_{*\uparrow}^{\text{detr}}$	s	Minimum detour time	4 minutes
$\Delta t_{*\downarrow}^{\text{detr}}$	s	Maximum detour time	30 minutes
r_*^{detr}	-	Maximum driver detour time ratio	1.25 (25 %)
$\Delta t_*^{\text{serve}}$	s	Vehicle service time (for boarding/alighting procedure)	1 minute
Δt_*^{wait}	s	Passenger early arrival time (waiting for driver)	2 minutes
Δt_*^{flex}	s	Departure time flexibility	15 minutes
s_*^{walk}	m/s	Walking speed	1.3 m/s
q_*^{vehi}	-	Maximum vehicle capacity	3 passengers
γ_*^{penl}	-	Objective function penalty for unmatched passengers	High value (999)

Results

With more customers participating in ride-sharing, the opportunity is higher for the algorithm to find appropriate and time-feasible matches, which increases the overall matching rate (Figure 6.3a). Starting with 65 % successful matches for 200 drivers and passengers, the rate increases up to 93.3 %, when 3 000 drivers and 3 000 passengers can be matched. Concurrently, the average passenger occupancy in matched vehicles increases, since the chances are higher for matching multiple passengers to a single driver when the demand is high. Figure 6.3b shows the trend of passenger occupancy. A value of one indicates that all vehicles involved in a match accommodate one passenger. According to the settings, up to three passengers are possible per vehicle. However, in most matchings only a single passenger is involved (Figure 6.4d on page 70).

Figure 6.4 on page 70 shows several histograms based on a simulation run with 6 000 customers. The Figures 6.4a and 6.4b visualize the actual detour time as the difference between actual travel time and direct travel

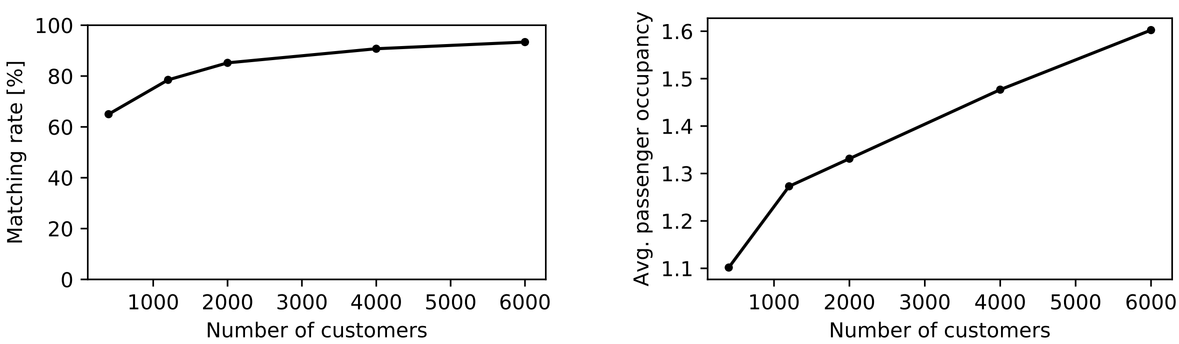
[1] `https://www.python.org/downloads/release/python-2713/`, accessed 19.09.2017
[2] `http://www.numpy.org/`, accessed 19.09.2017
[3] `https://developers.google.com/optimization/`, accessed 19.09.2017
[4] `https://projects.coin-or.org/Cbc`, accessed 19.09.2017

time $\left(\Delta t^{\text{total}}\left(\theta\right) - \Delta t^{\text{drive}}_{\leftrightarrow}\left(\theta\right)\right)$ in seconds for drivers and passengers, respectively. It can be clearly seen that the detour time for the driver is often very low. The minimum value of 120 seconds is due to the service time for boarding and alighting. If this is subtracted, there are some drivers who are close to no detour at all. This indicates that meeting points are selected that are more or less on the direct route of the driver.

In contrast, the histogram for the passenger detour times shows a different distribution, with the majority having a detour time between 5 and 20 minutes. The minimum value (240 seconds) is fixed for boarding (120 seconds) and waiting (120 seconds), so that every passenger has a minimum detour time of four minutes. The rest of the detour time can be explained in relation to the walking times to and from the meeting points, which correspond to an average walking distance of almost 400 metres (800 metres in total, creating approx. 10 minutes of walking time). If the walking time is subtracted, only the driving time from the meeting to the divergence point is left. If this *common riding time* is compared with the direct travel time, it can be seen that these connections are usually shorter than a direct ride (Figure 6.4e). Hence, the meeting points seem to be located mostly very efficiently, in terms of reducing the actual driving time.

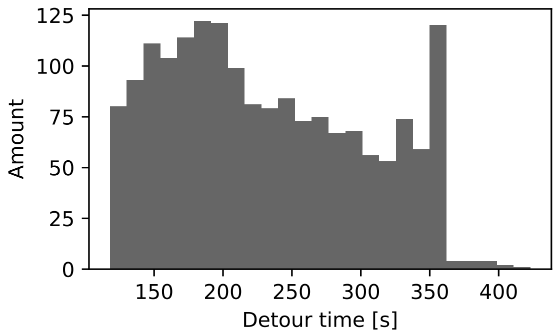
Figure 6.4c shows the histogram of assigned walking distances for customers to the meeting point. They are distributed over the full range of the allowed 800 metres, with shorter distances being assigned slightly more often than longer distances. However, there are also quite a few passengers who have to walk up to 800 metres.

Figure 6.4f shows the distribution of chosen meeting point types. Since no weighting of meeting point facilities was applied, the distribution follows roughly the overall meeting point type distribution, with street intersections as the most common type. An interesting outcome is that far fewer turning circles are used, but in turn more fuel stations, compared with the overall meeting point type distribution in the dataset. One reason could be that turning circles are mostly located at the end of small side roads, probably inducing a large detour. On the contrary, fuel stations are often placed at comparatively good positions, close to major roads.

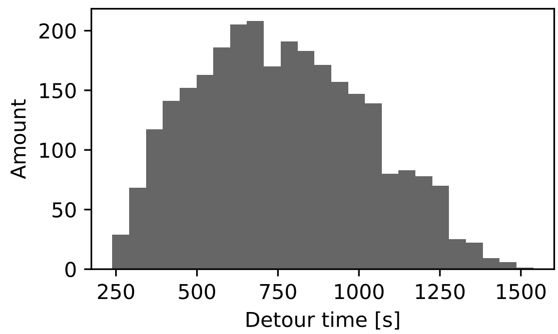


(a) Matching rate. *(b) Average passenger occupancy in matched vehicles.*

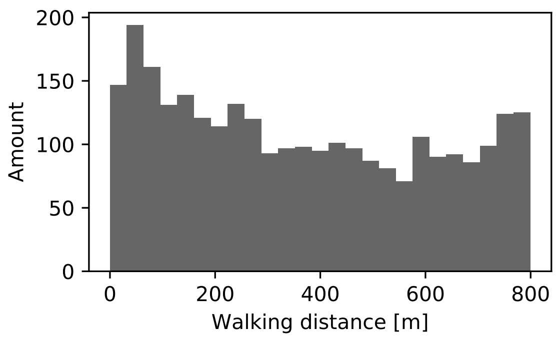
Figure 6.3: Results for different ride-sharing demand.



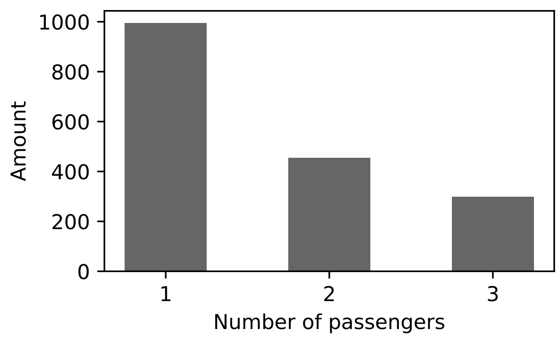
(a) Driver detour time



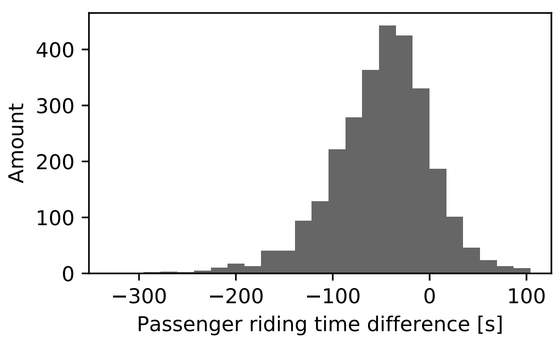
(b) Passenger detour time



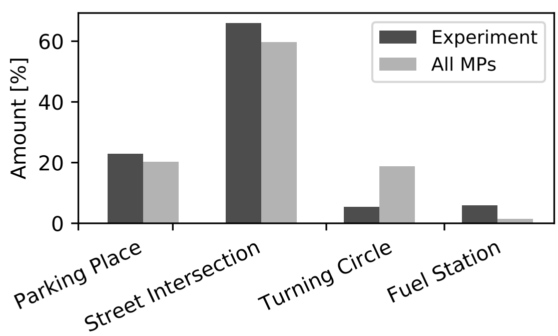
(c) Passenger walking distance (to a meeting point)



(d) Passenger occupancy in matched vehicles



(e) Passenger detour time without walking



(f) Used meeting point types

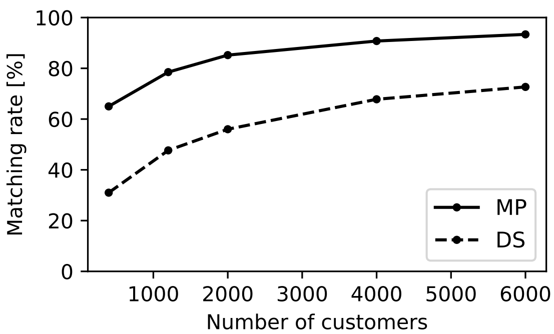
Figure 6.4: Histograms of various metrics, based on a simulation run of 6 000 customers.

6.3.2 Door-to-door service

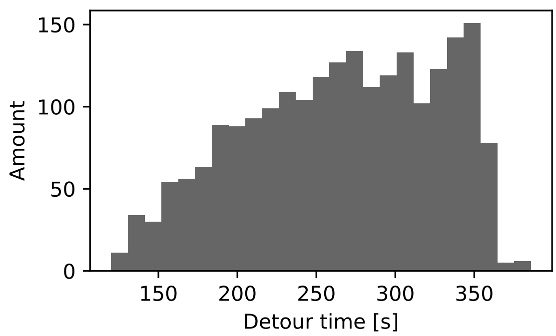
In this section, the previous results are compared with a conventional door-to-door based ride-sharing, with a pick-up and a drop-off at the doorstep (consequently referred to as DS). The DS mode needs only minor modifications: the maximum walking distance for each passenger is limited to zero, so that meeting and divergence points can no longer be reached. In exchange, one meeting and one divergence point is added (the *doorstep*), represented by the closest node that is reachable by a vehicle. This implies that a match can theoretically include only one passenger, with the only exception being people with exactly the same origin and destination node. Hence, a trip can technically be shortened to $\zeta = (\psi,\rho)$. For the simulation experiment, all other parameters have been kept as stated in table 6.2.

Figure 6.5a shows the differences in the matching rate (the share of matched passengers). As can be seen, the meeting point based mode allows more matchings, in particular when the demand is low. For higher demands, the matching rates increase. The higher matching rates occur due to the possibility of having multiple passengers in a vehicle, and more flexibility for the driver to satisfy detour constraints when meeting points can be used to shorten the necessary detour.

Also the detour times for the driver (Figure 6.5b) show a different distribution compared to the detour times when using meeting points (Figure 6.4a). While in the MP case, most drivers have a relatively low detour time, the drivers in the DS case most often have to take longer detours. Very likely, this shift can be explained by longer driving times through residential districts and potential detours because of one-way streets.



(a) Matching rate MP vs DS mode



(b) Histogram of driver detour times (DS mode)

Figure 6.5: Results of the doorstep (DS) based simulation results.

Figure 6.6 provides an overview of the average detour times for drivers and passengers for different demand instances. Note that all values include already a 120 second detour due to the service time. The passengers have moreover also 120 seconds waiting time involved, resulting in a constant detour of 240 seconds for passengers in the DS case.

The highest detour times occur for passengers using meeting points. This is clearly caused by the walking times to and from meeting and divergence points, respectively. The detour times for drivers are, on average, higher in the DS mode compared with the MP mode, althought this effect is only minor. With increasing customer demand, the average detour times for drivers in the MP case decrease slightly. This effect can be explained by higher availability, and options to match passengers at meeting points on the way.

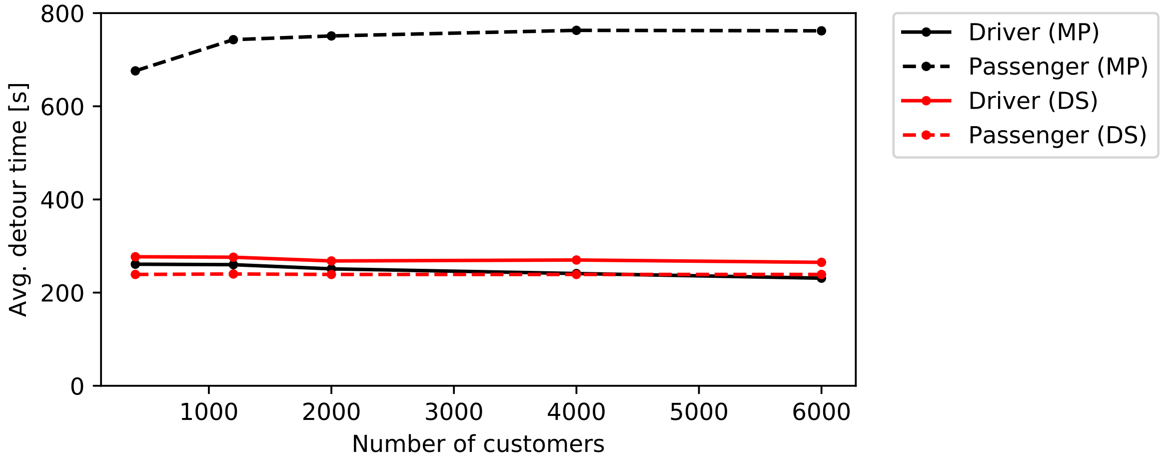


Figure 6.6: Detour times (additional time compared to direct drive) for drivers and passengers.

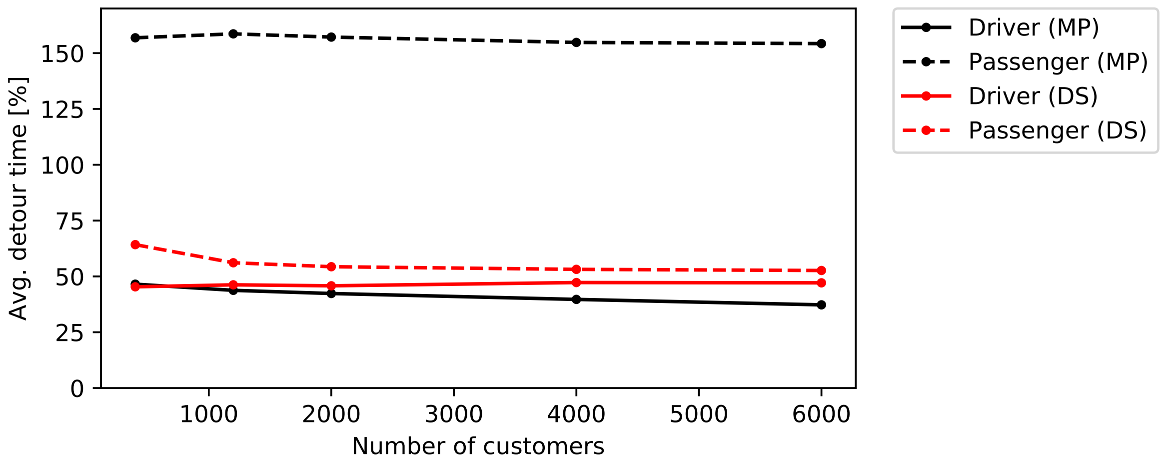


Figure 6.7: Detour times (as percentage to the direct travel time $\Delta t_{\leftrightarrow}^{drive}$) for drivers and passengers.

Figure 6.7 shows the same aspect, but as the average percentage of the direct travel time $\Delta t_{\leftrightarrow}^{\text{drive}}$ instead of the detour time directly. For drivers, the average detour is higher than the allowed driver detour time ratio r_{*}^{detr} (set to 1.25, see table 6.2), which can be explained by the fact that the service time adds up to the detour time. The passenger detour ratios are constantly very high (above 150 %), indicating more than a doubling of travel time when compared with the direct travel time. The reason is, beside the service and waiting time, the high impact of walking times within the travel times, since the direct travel times are relatively short in the investigation area (9:33 minutes on average, see Figure 5.5a on page 60). Hence, the detour time composed of walking, waiting, boarding, alighting and again walking can easily be higher than the direct travel time.

6.3.3 Convenience-based matching

As mentioned in the motivation, the individual rating of meeting points can also play a major role. To investigate possible impacts of considering meeting point quality, a personal satisfaction value was calculated and included in the matching. For drivers, parking quality of the meeting and divergence points is considered, whereas for passengers, facilities at the meeting point play a role. It is assumed that passengers do not care about facilities at the divergence point, since they do not have to wait there.

All satisfaction values are normalised between zero and one, with zero indicating a perfect fit regarding the requirements, and one a very poor fit. Firstly, parking quality is assessed by a normalization of parking availability (see Section 5.2). As a recall: parking places and turning areas have the best rating $\left(b^{\text{park}}(\mu) = 3\right)$, fuel stations are intermediate $\left(b^{\text{park}}(\mu) = 2\right)$ and street intersections are inferior $\left(b^{\text{park}}(\mu) = 1\right)$. The satisfaction value for a driver stopping at meeting point μ and divergence point δ is calculated by:

$$w^{\text{facl}}(\psi,\mu,\delta) = 1 - \frac{b^{\text{park}}(\mu) + b^{\text{park}}(\delta) - 2}{4} \tag{6.19}$$

The satisfaction value for passengers is based on a combination of personal preferences (see Section 5.4) and available meeting point facilities. Recall: each passenger is assigned three concern values $\tau^{\text{seat}},\tau^{\text{shlt}},\tau^{\text{illu}} \in \{0,0.25,0.5,0.75,1\}$, with zero indicating no concern and one indicating a high concern about the features specified: seating, shelter and illumination. These values are derived from the results of the user survey (Section 4.1.2). In addition, each meeting point is equipped with a binary value $b^{\text{facl}}(\mu) \in \{-1,1\}$, stating if a corresponding object (seating, shelter, light) is within a threshold (1) or not (-1, see Section 5.2).

The personal satisfaction value of a passenger concerning a particular meeting point is then calculated as follows:

$$w^{\text{facl}}_{\text{tmp}}(\rho,\mu) = \frac{\tau^{\text{seat}} \cdot b^{seat}(\mu) + \tau^{\text{shlt}} \cdot b^{\text{shlt}}(\mu) + \tau^{\text{illu}} \cdot b^{\text{illu}}(\mu) + 1.5}{3} \tag{6.20}$$

This results in a range of possible values from -0.5 (very poor fit, all important facilities) to 1.5 (perfect fit, all important facilities). Since values in the range below zero and above one are very rare, the valid value range is capped and inverted:

$$w^{\text{facl}}(\rho,\mu) = 1 - \max\left(\min\left(w^{\text{facl}}_{\text{tmp}}(\rho,\mu),1\right) 0\right) \tag{6.21}$$

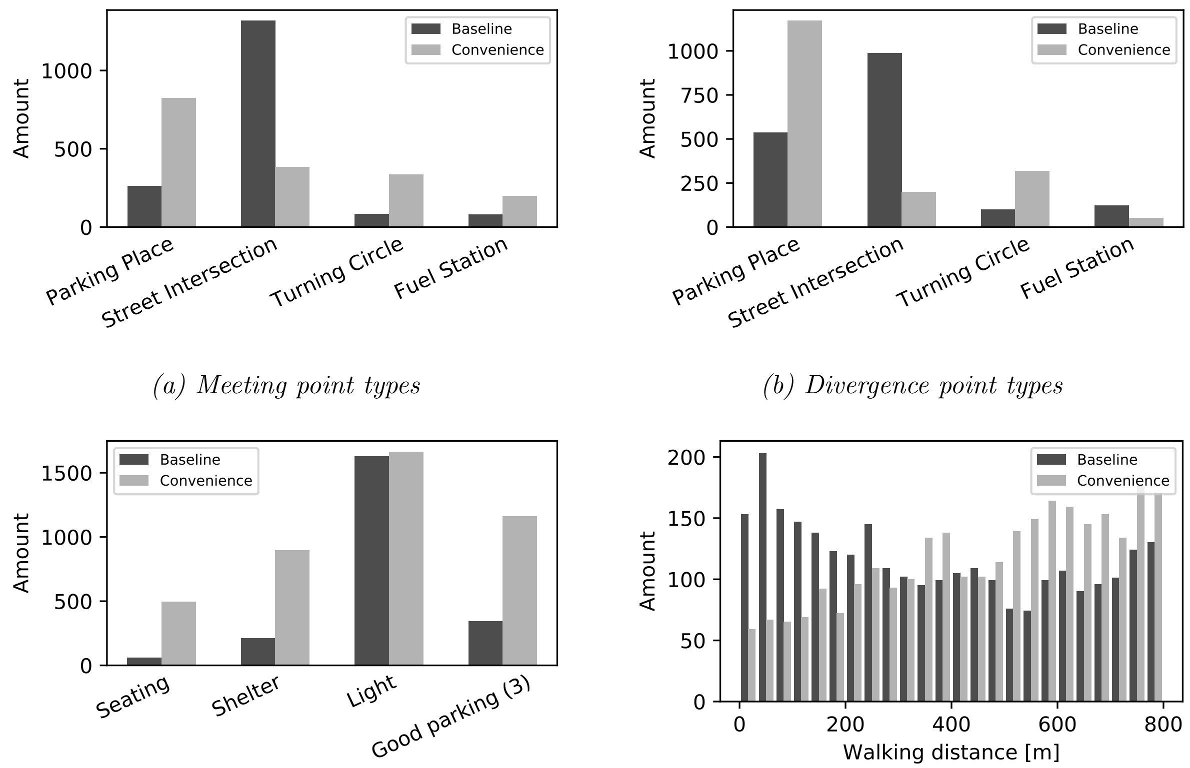
Subsequently, the meeting point satisfaction is combined with the detour time satisfaction to compute a personal, final weight. This is computed for drivers and passengers:

$$w(\theta,\mu,\delta) = \frac{w^{\text{detr}}(\theta,\mu,\delta) + w^{\text{facl}}(\theta,\mu,\delta)}{2} \tag{6.22}$$

Theoretically, the formulation allows an optional weighting of all influencing parameters, if specific aspects needs to be emphasised, but this option was not applied for this study. The final bipartite edge weight of a trip is again a combination of the individual weights of the participants:

$$w_e = \frac{w(\psi) + \sum_{\rho\in\xi} w(\rho)}{1 + |\xi|} \tag{6.23}$$

In order to assess the impacts of a meeting point rating and personal satisfaction, a simulation run with 6 000 fictive participants was executed, with the previously mentioned adaptation. Since only the weights of the edges change, the amount of matchings is equal to the baseline scenario. However, the selection of meeting points changes. Figure 6.8 compares the meeting point types from the baseline scenario with those from the modified scenario. It can be seen that parking places and turning areas are much preferred, and street intersections far less, if convenience plays a role (Figures 6.8a and 6.8b). The results are now more



(a) Meeting point types

(b) Divergence point types

(c) Facilities of the used meeting points

(d) Walking distance to the meeting points

Figure 6.8: Differences between the baseline and the convenience scenario.

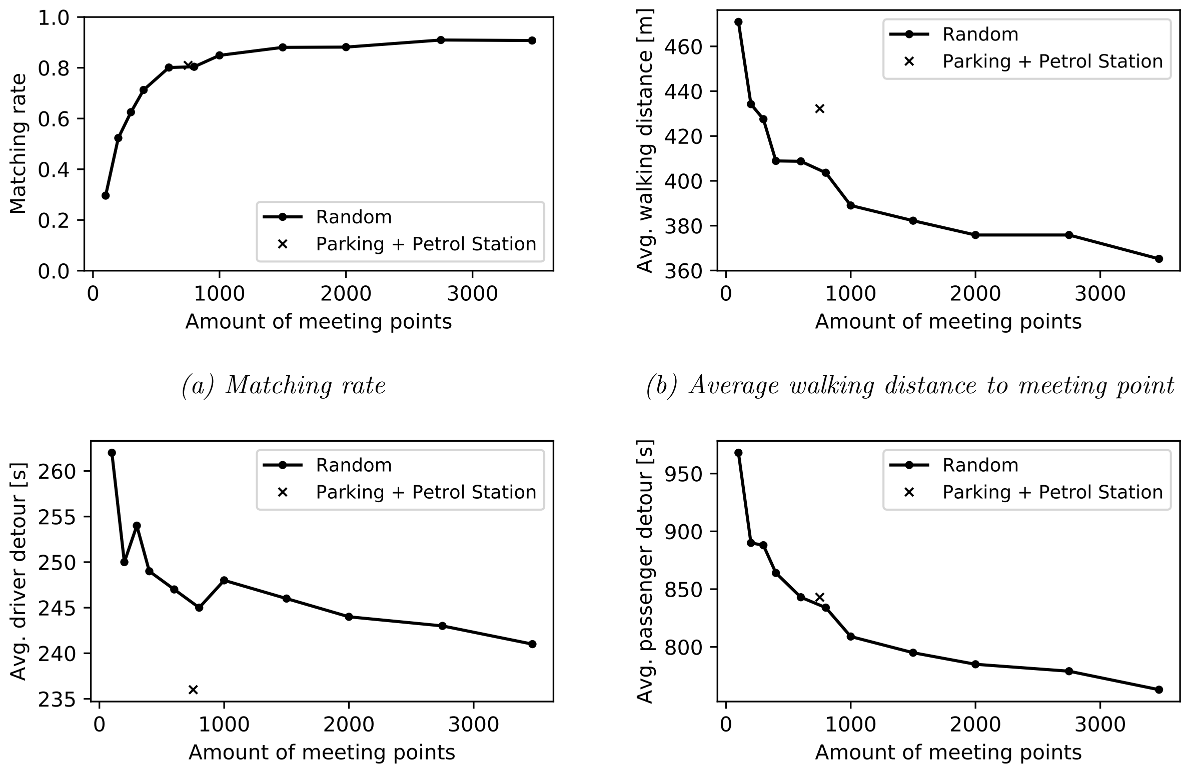
aligned with the results of the map-based survey (Section 4.2.2), where also parking places are used more frequently than street intersections (Figure 4.12a).

Figure 6.8c shows that many more locations are chosen that have seating and shelter facilities. The difference in illumination is merely marginal, since most meeting points are already equipped with some light source. In addition, meeting points that offer a good parking situation ($b^{\text{park}}(\mu) = 3$) are much more preferred. However, this comes at the cost of longer walking distances (Figure 6.8d). More passengers walk longer distances, and there are far fewer acceptable meeting points in the direct vicinity of the passenger origins.

6.3.4 Meeting point reduction

Further questions of interest are how many meeting points are necessary in a city to enable convenient ride-sharing with meeting points, and what are the implications of a reduced set of available meeting point locations. To answer these questions, a simulation with 4 000 customers was executed, where the set of meeting points was subsequently reduced by a random selection. In addition, the case of using only *good* meeting points is considered (only parking places and petrol stations; 752 points out of 3 471).

The results are visualized in Figure 6.9. The first Figure 6.9a shows the matching rate. It increases very quickly, and already with 200 meeting points, half of the passengers can be matched. With 600 meeting points available, more than 80 % of passengers find a driver. Then, the matching rate increases more slowly, and a saturation effect occurs; inserting more meeting points does not increase the matching rate substantially. With 1 000 meeting points available, the difference to the full set of meeting points (3 471) is



(a) Matching rate

(b) Average walking distance to meeting point

(c) Average driver detour time

(d) Average passenger detour time

Figure 6.9: Results from simulations with a reduced set of meeting points.

merely marginal. Figure 6.9b visualizes the average walking distance of passengers to the meeting point. As might be expected, the walking distance decreases with an increasing amount of available meeting points, since the opportunity is higher that a meeting point is available in the vicinity. A similar trend can be seen for driver detour times (Figure 6.9c) and passenger detour times (Figure 6.9d). A higher number of available meeting point locations improves the chance of reducing a necessary detour.

If only parking places and petrol stations are used, the results are somewhat different. While the matching rate and average passenger detour times are very similar to results using random selection, there is a difference in driver detour time and walking distance (Figures 6.9c and 6.9b). In essence, passengers have to walk longer distances, but in contrast drivers have a reduced detour time. A conclusion might be that parking places and petrol stations are located most often close to arterial roads, so drivers do not have to take longer detours through residential areas with lower speed limits.

6.4 Discussion

The simulation experiments allow a differentiated view on the impacts of using meeting points for intra-urban ride-sharing. Three main aspects have been investigated: a comparison with a door-to-door based mode, a consideration of convenience facilities at meeting points, and a varying number of meeting points in a city. In essence, the following statements can be made, based on the three experiments:

- With meeting points, more passengers can be matched, but those who are matched have a substantially longer travel time, mainly due to the walking time (compared with a pick-up at the doorstep).
- With meeting points, the travel times for drivers can be slightly reduced (compared with a pick-up at the doorstep).
- Incorporating convenience into the matching leads to a selection of less street intersections and more parking places.
- The share of meeting points with seating, shelter and good parking positions increases when considering convenience in the matching.
- With a higher availability of meeting points, he matching rate increases, and travel and detour times decreases.
- Using only parking places and petrol stations results in slightly longer walking distances for passengers, but also slightly less driving times.

An assumption is that the incorporation of convenience seems to yield a more realistic distribution of meeting point types. A comparison with the results of the map-based survey (Section 4.2.2) shows that parking places are chosen more frequently than street intersections, which is also the case when using the convenience-based mode (Figures 4.12a and 6.8a).

Stiglic et al. (2015) investigated a very similar setting to analyze and quantify the benefits that meeting points can bring to ride-sharing, with the main difference that they do the simulation on the Euclidean plane without real street networks. Hence, it is worth to comparing the results.

Similarly to the presented results (Section 6.3.4), they also show that the matching rate increases with increasing meeting point numbers. The matching rates range from 68 % to 74 %, which is quite comparable to the results from this simulation (65 % to 93 %). Note that they also allow a pick-up at the doorstep, so the matching rate does not decrease to zero when there is no meeting point available. On the other hand, the value does not rise so high since they allow at most four meeting points per TAZ (*Travel analysis zone*), where each TAZ has a size of 4.1 mi^2 on average. Hence, the MPC density is, in the most dense case, approximately 0.39 per km^2. Compared with the Braunschweig dataset used, with on average 18 meeting points per km^2, in this simulation almost 50 times more meeting points are available, which indicates a significant difference from the study by Stiglic et al. (2015). As a result, the walking distances are also much higher in their study, with on average 643 metres (0.4 miles), compared with 346 metres in the present simulation.

The average relative trip time increasing in their study is about 26 % for drivers as well as for passengers (in the most dense MPC scenario). This value is aligned with the results of the present simulation concerning the driver detour, if the service time is subtracted (25 % for 200 drivers, 17.5 % for 3 000 drivers, Figure 6.7). For passengers, however, there is a large gap to the percentaged detour times in the present simulation (Figure 6.7), which has two main explanations. On the one hand, the average direct travel times are much higher in their study, which lowers the detour ratio. On the other hand, Stiglic et al. (2015) allow, in their simulation, a pick-up at the doorstep and do not enforce a pick-up at a meeting point, and include home pick-ups in the average trip time increase, which again substantially lowers the average passenger trip time ratio.

An important aspect that should be discussed concerning the experiment where customer convenience is involved (Section 6.3.3) is that the results are highly dependent on the quality of the provided map data. In the case of Braunschweig, the quality of the OpenStreetMap data is very good, with many objects mapped, and corresponding details tagged appropriately. This is certainly not the case in many other regions. Furthermore, even with many objects accurately tagged, it is not always possible to determine whether a location really offers the desired facilities such as shelter, seating and light, or if they can be identified in the map. Nevertheless, with more and more map data available by crowd-sourcing, the quality of meeting point assessment will improve gradually. User ratings, as proposed by Hansen et al. (2010), could provide further valuable input to involve customer convenience in meeting point selection.

7 Meeting point recommendations for long-distance ride-sharing

In this chapter, a location-based approach is presented, aiming at enabling fast and automatic recommendation of suitable meeting (and divergence) points for long-distance ride-sharing trip participants. The section is divided into a motivation, a description of the proposed method, a simulation experiment and a concluding discussion. The work presented in this chapter is published in the Journal of Location Based Services (Czioska et al. 2018).

7.1 Motivation

People offering spare seats in long-distance rides on ride-sharing platforms such as BlaBlaCar often find passengers who need to be picked up or dropped off in cities en route. As an example, if a driver wants to travel from Hamburg to Munich, it is not unlikely that he/she might pick up a passenger in Hannover who wants to de-board in Nürnberg. Hence, it is necessary to negotiate a meeting (and/or divergence) point in these cities. For this purpose, common locations that are well-known and easily reachable by public transport are frequently chosen, e.g. the central train or bus station in the city. However, such locations are usually located in city centres, causing unnecessary detours and time loss for the drivers due to congestion in the inner-city parts. The use of meeting points that are close to motorways and arterial roads, and furthermore easily reachable by public transport could reduce the driving time, driving distance and congestion in urban areas. A recommendation of such points is especially helpful when a driver or a passenger is not familiar with the environment. In the future, ride-sharing or navigation applications could even already contain a set of predefined meeting point locations for this use case. It is also worth considering that some points could be designated and equipped (e.g. with shelter) by the municipal traffic management, to ease the establishing of shared rides, similar to bus stops.

Most existing approaches focus on intra-urban rides covering shorter distances, where reachable meeting points for passengers are limited by a walking threshold (e.g. Balardino & Santos (2016); Stiglic et al. (2015); Rigby et al. (2013)). The novelty of the current approach is to extend the meeting point search by including public transportation, allowing passengers to reach more remote meeting points, e.g. close to motorway exits, which is most relevant if the driver is intending to just pass the city. In a prospective recommender system application, the results should be available in real time. Since every meeting point recommendation is based on an optimization procedure with increasing complexity for an increasing number of participants, response time is a significant factor. Hence, an extensive precomputation of shortest paths is applied to substantially reduce the query time. This technique is commonly used in various route-planning algorithms, e.g. for distance tables in hierarchical routing networks (Sanders & Schultes 2005) or precomputed cluster distances (Maue et al. 2009).

In this chapter, a location-based method is described that recommends real-world meeting points to long-distance ride-sharing customers. The scenario comprises a driver passing the city on major roads, having to pick up one or multiple passengers at exactly one point in the city. While the driver moves on the street network, the passengers are supposed to walk and use public transport to reach the meeting point. The goal is to determine the location among a set of predefined candidate locations that maximize the satisfaction of the users. The optimal location is thereby dependent on the current spatio-temporal location of driver and passengers, and should minimize the total or maximum time consumption. In addition, other factors influencing individual satisfaction should be able to be included, as personal preferences. Divergence points are determined in the same way, therefore the workflow is described only for meeting points. Since the method should be able to serve requests in real time, short response times are crucial.

The algorithm is explained in section 7.2. Subsequently, an experimental simulation (Section 7.3) demonstrates the impact of the proposed algorithm. The notation used in this chapter is provided in Table 7.1.

7.2 Proposed method

This Section describes the workflow of the proposed method, which is divided into three parts:

1. **Preparation phase**
2. **Precomputation phase**
3. **Operation phase**

The preparation phase (Section 7.2.1) has to be executed only once in the beginning and includes a GIS workflow to process the geodata and reduce the number of necessary meeting points to be considered. The precomputation phase (Section 7.2.2) involves comprehensive shortest-path calculations for drivers (using the street network) and passengers (using the public transport network). The precomputation phase has to be executed in the beginning, and whenever the public transport timetable changes. Finally, the operation phase (Section 7.2.3) describes the method for real time processing of customer requests. This part is designed as a service that waits for incoming customer requests.

In a nutshell, the algorithm works as follows. Given is a fixed group who have already negotiated to travel together. The group consists of a single driver, and one or multiple passengers in a city. Known is moreover the current driver location, the planned driver path, and all passenger origins in the city. The objective is to recommend a single meeting point in the city of the passengers that is best suited to the needs of the group members, in terms of travel time, walking distances, and possible further aspects.

When a request arrives, travel time costs are iteratively computed for every feasible meeting point candidate in the city. To speed up the checking procedure, the estimated travel times for the driver and for the passengers are stored in a matrix.

The driving times are computed from so called *inlet points* on the motorway (which are not the motorway exits). On the passenger side, it is computationally inapplicable to precompute the travel times from everywhere in the city. Hence, representative *public transport entry* (PTE) nodes $\pi \in \Pi$ are created as fictive origins in the vicinity of public transport stops, since the passengers are expected to use the public transportation system. A request from a passenger thus first requires a reachability analysis of PTE nodes in the vicinity of the starting location. Subsequently, the precomputed public transport connections from the PTE nodes can be used to estimate the arrival time at the meeting point candidates. Figure 7.1 visualizes the basic principle.

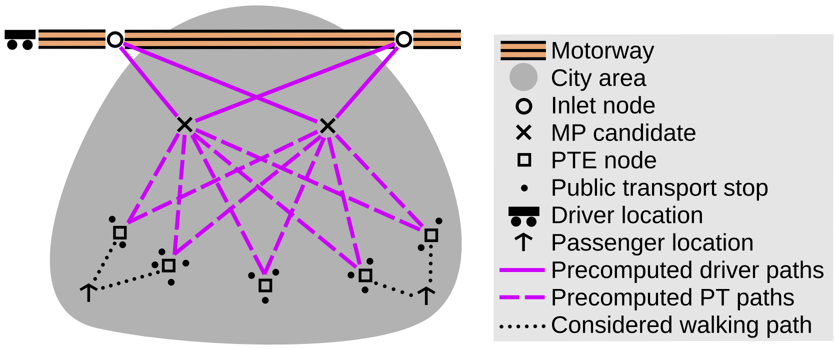


Figure 7.1: Schematic visualization of the precomputed paths.

Notation	**Unit/Size**	**Description**
$\psi \in \Psi$	-	Drivers
$\rho \in P$	-	Passengers
$\mu \in M$	-	Meeting point candidates
$\mu' \in M'$	-	Reduced set of meeting point candidates
$s \in \mathcal{S}$	-	Public transport stop locations
$s' \in \mathcal{S}'$	-	Clustered public transport stops
$z \in \mathcal{Z}$	-	Public transport vehicle lines
$c \in \mathcal{C}$	-	Public transport connections
$\pi \in \Pi$	-	Public transport entry (PTE) nodes
$\gamma \in \Gamma$	-	Representative passenger origin locations
$i \in I$	-	Inlet points
$i^+ \in I^+$	-	Inbound inlet node
$i^- \in I^-$	-	Outbound inlet node
λ	-	Current or planned location (of driver or passenger)
ϵ^{PT}	m	DBSCAN distance threshold for public transport stops
ϵ^{MP}	m	DBSCAN distance threshold for meeting point candidates
d_*^{walk}	m	Maximum passenger walking distance
d_*^{repr}	m	Maximum distance between γ and π for k-Means clustering
$\Delta t^{\mathrm{drive}}$	s	Driving time
Δt^{walk}	s	Walking time
t^+	s	Departure time
t^-	s	Arrival time
$\Delta t^{\mathrm{total}}$	s	Total travel time
$\Delta t_*^{\mathrm{detr}}$	s	Maximum driver detour time
$\Delta t_*^{\mathrm{wait}}$	s	Passenger waiting time tolerance
$\Delta t_*^{\mathrm{dgap}}$	s	Minimum departure gap
d_*^{walk}	m	Maximum passenger walking distance
$A_\Psi^{\mathrm{inbound}}$	$\lvert I^+\rvert \times \lvert M'\rvert$	Travel time matrix (driving times inbound)
$A_\Psi^{\mathrm{outbound}}$	$\lvert M'\rvert \times \lvert I^-\rvert$	Travel time matrix (driving times outbound)
A_P	$\lvert\Pi\rvert \times \lvert M'\rvert$	Travel time matrix (public transport and walking)

Table 7.1: Notation used in this chapter.

7.2.1 Preparation phase

In the preparation phase, the raw data is processed to prepare the precomputation. The following data is necessary:

- A street network $G = (V,E)$ of the service area, including vehicle driving times and passenger walking times. See section 5.1 for details about the notation and the model of the street network.
- Public transport (PT) data, consisting of a set of stops $\mathcal{S}$, a set of vehicle lines $\mathcal{Z}$ and a set of elementary connections $\mathcal{C}$. See section 5.3 for details about the PT model.
- A set of meeting point candidate locations $\mu \in M$. See section 5.2 for an exemplary dataset.
- A set of *Inlet points* I. These points should be located on motorways or other high-level roads and indicate the entry (and exit, respectively) of the service area such that the important inter-urban connections pass an inlet point inbound and another inlet point outbound (see Figure 7.2). Note that the inlet points should explicitly *not* be located on motorway exits, but rather between them.

In the following, the generation of public transport entry (PTE) nodes (step 7.2.1.1) and the preparation of meeting point candidates (step 7.2.1.2) is described in more detail.

7.2.1.1 Generation of public transport entry (PTE) nodes

In the precomputation step (Section 7.2.2), the public transport connections from PTE nodes to meeting point candidates will be determined. Since the passengers are supposed to use the public transportation system and the stops are where they change over from walking to public transport, it is reasonable to place the PTE nodes close to the public transport stops $\mathcal{S}$. A simple way is to simply create a PTE node π at every stop s. However, since every PTE node invokes a precomputation and storage of connections in step 2, it is reasonable to reduce the number of PTE nodes beforehand.

Firstly, the stop positions $\mathcal{S}$ are clustered into $\mathcal{S}'$, since mostly a stop (e.g. 'Main Street') consists of several discrete stopping positions, e.g. for different directions or bus lines. The goal of the clustering is to unify these stopping places, either based on the stop name, or, if the naming is not consistent, by a density-based clustering such as DBSCAN (Ester et al. 1996). Figure 7.3a shows the result of such a clustering. The DBSCAN distance threshold ϵ^{PT} should be chosen such that each group of stopping positions covers

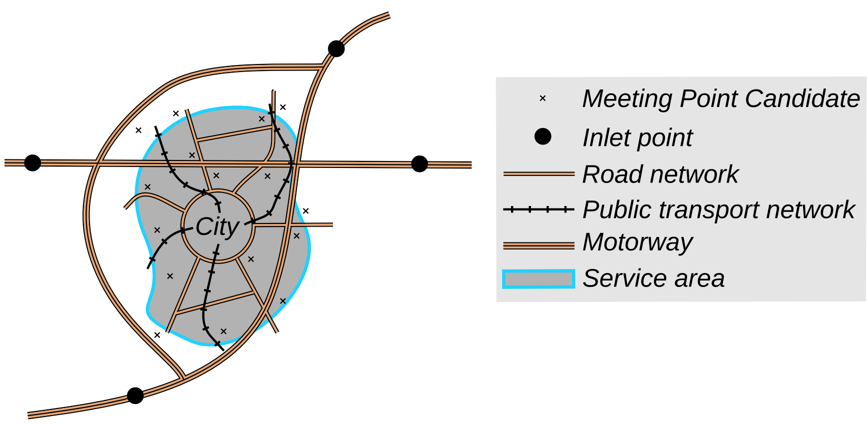
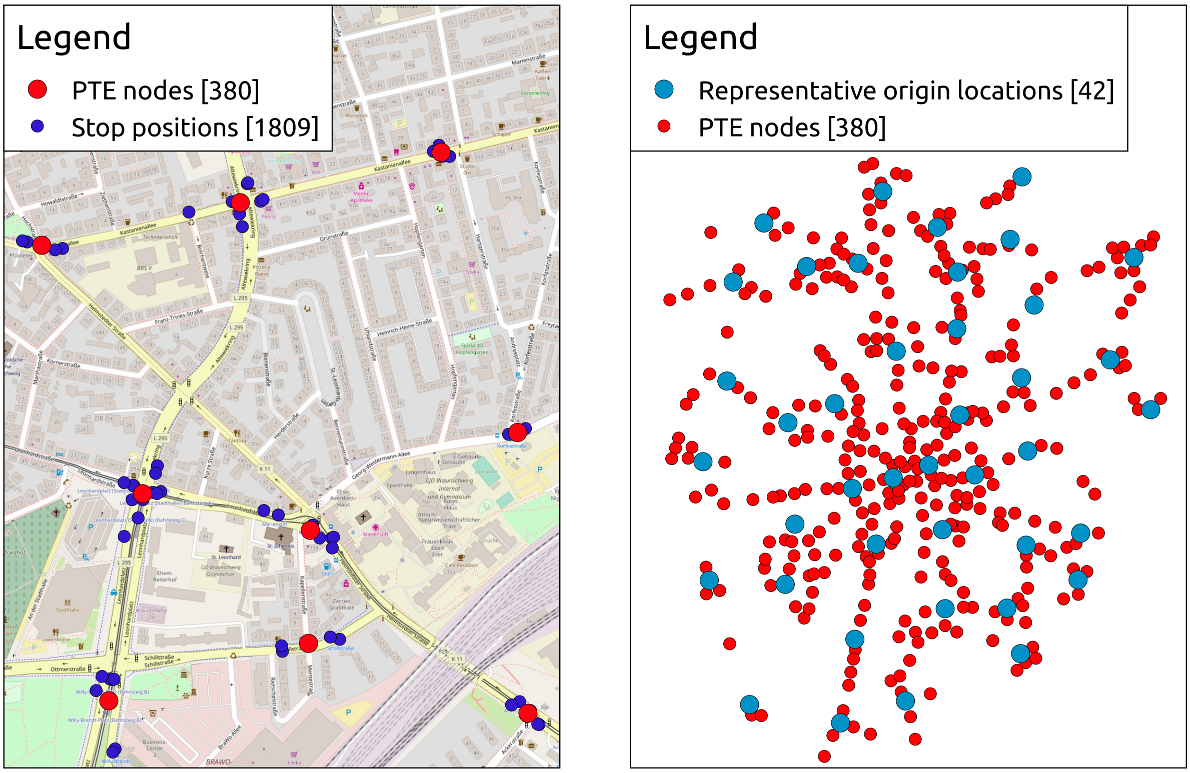


Figure 7.2: Schematic representation of the service area and location of inlet points.



(a) Clustering of stop positions. *(b) Sampling among stop positions.*

Figure 7.3: Stop preparation.

exactly one stop. In the simulation experiments (Section 7.3), 100 m was determined as an appropriate value for this purpose. However, if the stop positions of two different stops are closer than ϵ^{PT}, the DBSCAN clustering will group them together. Hence, a post-processing check is necessary to ensure that all stop positions are covered by a PTE node $\pi \in \Pi$ within a reasonable distance.

7.2.1.2 Meeting point candidates preparation

Since many of the meeting point candidate locations will be situated at locations that are not very useful for the purpose of long-distance ride-sharing, it is advisable to filter them before the precomputing phase. The fewer meeting points are left, the faster the precomputation algorithm will finish. Hence, especially for larger urban areas or weak computation infrastructures, it is advisable to execute this preparation step. On the other hand, if the city is small or an extensive computation infrastructure is available, this step can also be skipped.

As a first step, all meeting point candidates should be removed if they are not accessible via public transportation, i.e. have no stop position $s' \in \mathcal{S}'$ within a predefined walking threshold d_*^{walk}.

All points of the remaining set are in theory feasible for being considered. However, a majority of these points is not useful to consider during the operational phase, since it is very unlikely that they ever will be used, e.g. due to a location far from motorway exits or other major roads. Hence, they can be removed, to speed up the precomputation phase.

For this, a simulation run is proposed to determine the usage frequency of the meeting points, and only the most promising locations should be kept. Initially, some representative passenger origin locations $\gamma \in \Gamma$ need to be sampled as fictive passenger starting points. They should be distributed evenly across the investigation area, e.g. at least one location in every suburb, to ensure that every possible passenger origin in the city can roughly be represented by a fictive passenger origin γ. On the other hand, the number should be limited, to keep the computation time short. A quality measure can optionally be applied, such that all PTE nodes from the previous step are covered within a certain distance d_*^{repr}. The consideration of the population density is not necessarily appropriate here, since then the meeting point selection is likely to be biased towards very densely populated areas.

For creating a representative origin location set Γ, an iterative k-Means clustering of the PTE nodes ($\pi \in \Pi$) is proposed, as outlined in Algorithm 1. The basic assumption of this idea is that public transport stops are already relatively equally sampled along the populated areas of a city. The k-Means clustering technique allows the creation of a homogeneous distribution of a certain number of representative origin locations. Since the number of necessary origin locations is not clear, the k-Means function is called iteratively, with an increasing number of origin locations. If a solution is found with the quality criterion satisfied (distance to any passenger origin is below d_*^{repr} for all PTE nodes), the iteration stops. Figure 7.3b exemplifies a result of such an iterative k-Means clustering.

Algorithm 2 Algorithm used to sample stop positions across service area

```
Given: List of PTE nodes Π, Threshold d*^repr
for i ∈ {0,1, ··· ,|Π|} do
    Γ ← K-MEANS(data=Π, clusterCount=i)          ▷ Get cluster center positions
    for π ∈ Π do                                 ▷ Iterate through all PTE nodes
        if ∄ γ ∈ Γ | DIST(π,γ) ≤ d*^repr then
            Break inner loop and continue outer loop
        end if
    end for
    return Γ                      ▷ All PTE nodes are covered within certain distance
end for
```

Subsequently, the travel times from all representative passenger origins Γ to the meeting point set M are precomputed (see Section 7.2.2). Then, fictive meetings of random driver/passenger groups in the service area are simulated, and the recommended meeting points recorded. More precisely, a list of n tuples, each containing a random driver inbound inlet node i^+, a random driver outbound inlet node i^-, a random group of representative passenger entry nodes Γ, and a random time of day, is created and iteratively used as simulation input.

Figure 7.4 shows a typical frequency of meeting points being selected in the simulation. As can be seen, the meeting point candidates are chosen with a very different frequency - some very often, others never. The meeting points with low scores are then removed by a threshold selection, yielding a reduced set $M' \subset M$.

However, the remaining meeting points $\mu' \in M'$ can be located very close to each other. In these situations, only one of the adjacent meeting points would be sufficient, so the set of candidate points can further be downsized. A refined approach based on the one previously described is outlined in Algorithm 2 and works as follows. Initially, a DBSCAN clustering is applied to the meeting points, with a distance threshold ϵ^{MP}, indicating the minimum distance between two meeting points in the resulting set. If two meeting points are located closer to each other than this threshold, only one of them will be kept in the final selection.

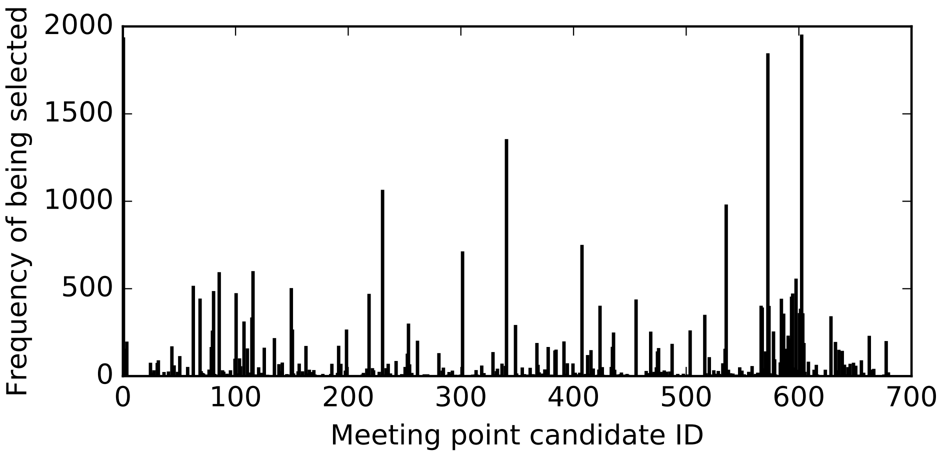


Figure 7.4: Frequency of meeting point candidates being selected.

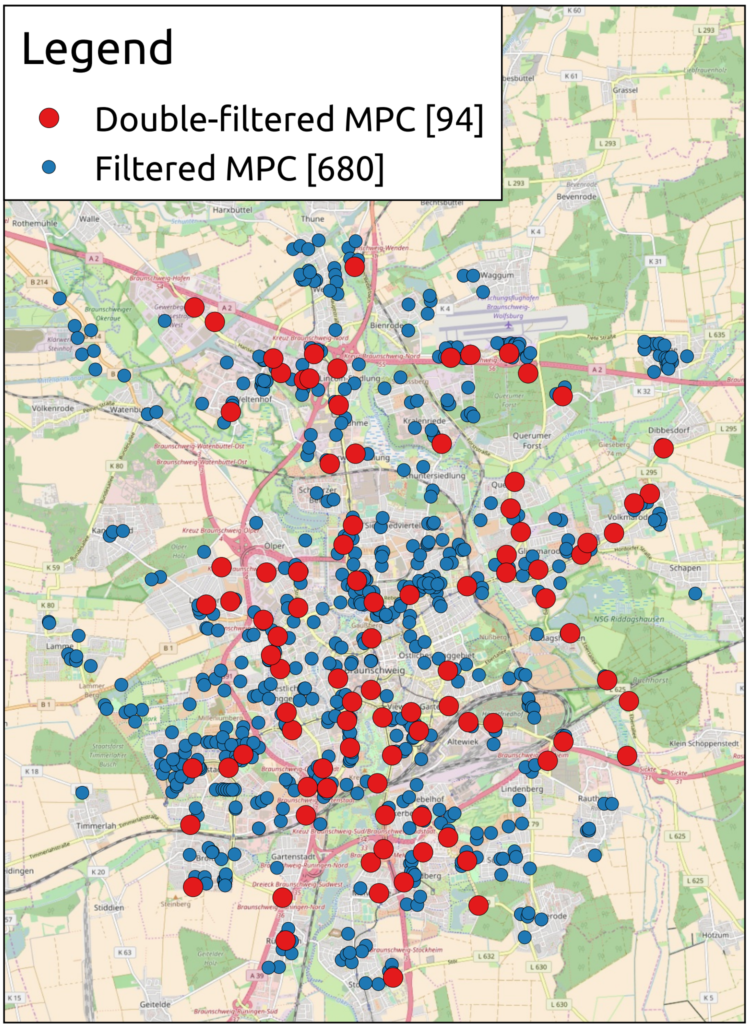


Figure 7.5: Filtering of unused meeting points after a simulation run.

Subsequently, the meeting point candidates are processed iteratively in descending order of their selection frequency. If none of the meeting point neighbours is already in the final set, the meeting point is added. Otherwise, the neighbouring meeting points that are already in the final set are checked if the distance is greater than ϵ^{MP}. If this applies to any of them, the meeting point is discarded.

Figure 7.5 illustrates the final reduction of meeting point candidates.

Algorithm 3 Meeting point selection with clustering

```
Given: Meeting point candiate (MPC) set M, Threshold ε^MP
Initialize result set Φ ← {}
Initialize cluster check set Θ ← {}
C(M) ←DBSCAN(data=M, threshold=ε^MP)                ▷ MPC assignments to clusters
U(M) ←DetermineUsage(M)                             ▷ Frequency of MPCs being selected
for m ∈ Sort(M,U) do                  ▷ Sort MPCs by descending selection frequency
    if C(m) ∈ Θ then
        if (Dist(m,n) ≥ ε^MP) ∀ n ∈ {Φ | C(n) = C(m)} then
            Φ ← Φ ∪ m
        end if
    else
        Φ ← Φ ∪ m
        Θ ← Θ ∪ C(m)
    end if
end for
return Φ
```

7.2.2 Precomputing phase

In this phase, travel times are precomputed and stored.

Firstly, the driving times $\Delta t^{\mathrm{drive}}$ from all inlet points I to all meeting point candidates M' and back are stored in matrices A_{Ψ}. Equation 7.1 shows the matrix structure for static driving times that do not change over time. This is the easiest assumption that requires only one value for each connection. However, since driving times change dynamically during the day, they are not static in reality. A straightforward approach is to apply a constant congestion factor which is dependent on the time of day, so that at rush hour the speed used for the calculations is reduced on all streets Lin et al. (2016). For improved precision, dynamic travel times, either path- or link-based, can be used to estimate realistic driving times depending on the chosen route and time. Such values can for example be derived from historical traffic data (see e.g. Chien & Kuchipudi (2003)). In this case, a three-dimensional matrix is necessary, with a driving time value for each combination of inlet point i^{+}, meeting point candidate μ, and time of day.

Secondly, a matrix A_P is created, containing possible multimodal public transport connections throughout the day from all PTE nodes in Π to all meeting point candidates in M'. In this matrix, each entry is a list of connections, represented as a tuple of departure time t^{+}, corresponding arrival time t^{-} and necessary walking distance d^{walk}, sorted by departure time (see Equation 7.2). If multiple arrivals are possible for the same departure time, only the fastest connection is stored. Slow connections that are being overtaken by other connections are consequently neglected. If it is possible to walk from a PTE node π to a MPC μ, only the walking time is stored. Furthermore, the walking distance is attached, to exclude meeting points which exceed possible walking distance thresholds of passengers.

In order to limit the number of stored connections for each π - μ pair, a *minimum departure gap* parameter Δt_*^{dgap} is introduced, defining the minimum time gap between two departures. The number of connections for one day is hence limited to $24 \cdot 60/\Delta t_*^{\text{dgap}}$, given that Δt_*^{dgap} is provided in minutes. This is especially useful if the public transport connection is so good that a departure is theoretically possible every few minutes.

$$A_\Psi^{\text{inbound}} = \begin{array}{c} \\ i_1^+ \\ i_2^+ \end{array} \begin{array}{c} \begin{array}{cc} \mu_1 & \mu_2 \end{array} \\ \begin{pmatrix} \Delta t^{\text{drive}} & \Delta t^{\text{drive}} \\ \Delta t^{\text{drive}} & \Delta t^{\text{drive}} \end{pmatrix} \end{array}, \qquad A_\Psi^{\text{outbound}} = \begin{array}{c} \\ \mu_1 \\ \mu_2 \end{array} \begin{array}{c} \begin{array}{cc} i_1^- & i_2^- \end{array} \\ \begin{pmatrix} \Delta t^{\text{drive}} & \Delta t^{\text{drive}} \\ \Delta t^{\text{drive}} & \Delta t^{\text{drive}} \end{pmatrix} \end{array} \tag{7.1}$$

$$A_P = \begin{array}{c} \\ \pi_1 \\ \pi_2 \end{array} \begin{array}{c} \begin{array}{cc} \mu_1 & \mu_2 \end{array} \\ \begin{pmatrix} \left[\left(t_1^+, t_1^-, d_1^{walk}\right), \left(t_2^+, t_2^-, d_2^{walk}\right), \ldots\right] & \left[\left(t_1^+, t_1^-, d_1^{walk}\right), \ldots\right] \\ \left[\left(t_1^+, t_1^-, d_1^{walk}\right), \left(t_2^+, t_2^-, d_2^{walk}\right), \ldots\right] & \Delta t^{\text{walk}}, d^{walk} \end{pmatrix} \end{array} \tag{7.2}$$

The matrix A_P can be constructed in polynomial time $\mathcal{O}\left(|\Pi||M'|\right)$, since for each π - μ combination a limited number of connections has to be computed. Adding meeting points while holding the number of PTE nodes constant can be established in linear time, and vice versa. However, the preparation phase is still recommended to shorten computation times.

7.2.3 Operational phase

The operational module can be regarded as a service interface waiting for incoming requests of a driver/passenger group and returning a recommendation of one (or more) meeting points. The necessary components of a request are:

- Planned driver inlet node (inbound) i^+
- Planned driver inlet node (outbound) i^-
- Current location of the driver $\lambda\left(\psi\right)$
- Current (or planned) location of one or multiple passengers $\lambda\left(\rho\right)$
- Maximum driver detour time Δt_*^{detr}
- Passenger waiting time tolerance Δt_*^{wait}
- Maximum passenger walking distance d_*^{walk}

The maximum driver detour parameter Δt_*^{detr} controls the feasible meeting point candidates, i.e. a smaller value of Δt_*^{detr} leads to selection of meeting points that are closer to the motorway exits. The parameter allows drivers to specify their time budget for picking up (or dropping off) passengers. Furthermore, the parameter can be used by traffic management entities to influence how far vehicles should penetrate the city, e.g. for pollution reduction.

The waiting time tolerance parameter Δt_*^{wait} defines the flexibility of arrival times at the meeting point. A negative value of -5 minutes indicates that all passengers must arrive at the meeting point at least 5 minutes prior to the driver arrival. In contrast, a positive value of 5 minutes allows the passengers to arrive up to 5 minutes later than the driver. In a real-world application, this parameter could be chosen individually by the driver.

The workflow of request processing contains the following steps:

1. Estimate driver arrival times at meeting point candidates
2. Determine reachable PTE nodes for the passengers
3. Estimate passenger arrival times at meeting point candidates
4. Compute total travel times
5. Voting

Since the meeting point determination works very fast, it can be repeated several times after the first run. So, if the traffic situation changes, the meeting point selection can be adapted dynamically. Naturally, this is only possible until the first passenger has left his/her home, since then it will be difficult for a passenger to change plans spontaneously.

7.2.3.1 Estimate driver arrival times at meeting point candidates

In order to calculate all arrival and departure times, the only unknown value is the expected time of the driver passing the inlet node (inbound) $t^-_\psi(i^+)$. It can be estimated based on the current location of the driver $\lambda(\psi)$, which can automatically be transmitted from any GPS sensor. An arbitrary (third-party) routing service may further be applied to estimate the remaining journey time until the inlet node is reached. Using Equation 7.3 and the precomputed driving time matrix A^{inbound}_Ψ, the arrival times of the driver at all meeting point candidates can then be estimated instantly:

$$t^-_\psi(\mu) = t^-_\psi(i^+) + A^{\text{inbound}}_\Psi(i^+ \to \mu) \tag{7.3}$$

If the threshold Δt^{detr}_* is set, all meeting point candidates that require a driver detour time exceeding the threshold can be disregarded for this request.

7.2.3.2 Determine reachable stop nodes for the passengers

Since the algorithm is designed as a location-based service, the meeting point recommended depends on the current (or planned) position of the passengers $\lambda(\rho)$. The location can automatically be transmitted from any GPS sensor, e.g. a smartphone, or manually entered by the customer, which might be useful if the planned location at time of departure is already known beforehand. Since the public transport connections are precomputed only from the PTE nodes, as described in section 7.2.2, first all reachable PTE nodes Π_ρ have to be determined with respect to the maximum walking distance d^{walk}_*. Further, for each PTE node $\pi \in \Pi_\rho$, the corresponding walking time t^{walk} has to be calculated.

If, due to privacy reasons, the actual location of the passenger should be obfuscated, this step can also be outsourced to a trusted third-party that returns the closest PTE nodes, or the passengers can choose the PTE nodes directly and estimate the walking time themselves.

7.2.3.3 Estimate passenger arrival times at meeting point candidates

The previously derived driver arrival times at meeting points determine the relevant connections for the passengers. For every reachable PTE node of the passengers and every meeting point, the corresponding time of departure for the passenger is calculated such that the waiting time constraint is met. For this, the item $(t^{+}(\pi),t^{-}(\mu))$ closest to the arrival time is fetched from the sorted list of arrival times in the passenger connection matrix A^{P}, which can be done efficiently using binary search. Note that the waiting time tolerance $\Delta t_{*}^{\mathrm{wait}}$ has to be applied, as shown in constraint 7.4.

$$t_{\psi}^{-}(\mu) + \Delta t_{*}^{\mathrm{wait}} \geq t_{\rho}^{-}(\mu) \,\forall\, \rho \in P \tag{7.4}$$

Consequently, the departure time for this passenger can be determined by Equation 7.5, including the initial walking time from the current location to PTE node.

$$t_{\rho}^{+}\left(\lambda\left(\rho\right)\right) = t^{+}(\pi) - t^{walk}\left(\lambda\left(\rho\right) \rightarrow \pi\right) \tag{7.5}$$

7.2.3.4 Compute total travel times for all meeting point candidates

Having the travel times for the driver and all passengers to the meeting point at hand, the time of passing the outbound inlet node can be determined using equation 7.6. This enables the deriving of the total travel time for the whole group of driver and passengers by summing up the individual travel times (Equation 7.7).

$$t_{\psi}^{-}(i^{-}) = \max\Big(t_{\psi}^{-}\left(\mu\right), \max\left(\left\{t_{\rho}^{-}\left(\mu\right) \mid \rho \in P\right\}\right)\Big) + A_{\Psi}^{\mathrm{outbound}}(\mu \rightarrow i^{-}) \tag{7.6}$$

$$\Delta t^{\mathrm{total}} = t_{\psi}^{-}\left(i^{-}\right) - t_{\psi}^{+}\left(i^{+}\right) + \sum_{\rho \in P}\Big(t_{\psi}^{-}\left(i^{-}\right) - t_{\rho}^{+}\left(\lambda\left(\rho\right)\right)\Big) \tag{7.7}$$

7.2.3.5 Voting

The last step of the workflow is to choose an appropriate meeting point among the set of candidates. Considering several persons who need to agree on a meeting point, those persons will probably have different preferences regarding the meeting point candidates, not just because of their different distances from the meeting points, but also because of other properties such as shelter, illumination or seating, or because of different time pressures. A common approach for determining a commonly acceptable agreement based on different individual preferences is a vote. Dennisen & Müller (2015) discuss several different voting rules for decision-making in traffic applications. In the present scenario, a voting rule that yields exactly one winner is necessary, also called social choice function (Rothe et al. 2012).

In the present approach, two different voting rules are proposed. The first is a *range voting* rule, where each voter scores all candidates on a range ballot, based on the individual travel time. The scores are summed up, and the candidate with the lowest value is selected. If time is the only factor to be represented in the voting, the resulting meeting point is the one with the lowest value of $\Delta t^{\mathrm{total}}$, corresponding to a utilitarian approach.

However, this rule does not necessarily yield the most socially acceptable solution. Consider a situation with three riders A, B and C, and two meeting point candidates α and β. Let further interpret the travel times as dissatisfaction values. Meeting point α gets a dissatisfaction score of 10 by each rider, and meeting point β is scored with a dissatisfaction of 2 by riders A and B. Rider C scores meeting point β with a dissatisfaction of 25. According to the range voting rule, β wins because $\Delta t^{\mathrm{total}}(\beta) = 2+2+25 \leq \Delta t^{\mathrm{total}}(\alpha) = 10+10+10$.

Now, it could be argued that the selection discriminates against rider C. The example highlights that the range voting rule is prone to imbalanced travel times among the riders.

For this reason, a second voting rule based on the *minimax* principle is introduced. Here, the meeting point that minimizes the maximum travel time (or detour time of the driver, respectively) among the riders is chosen, leading to a more balanced distribution of travel times. The travel times are regarded as dissatisfaction values, and the winner of the vote is the meeting point with the lowest maximum dissatisfaction value. This corresponds to optimizing the travel time according to an egalitarian approach. In this basic example, meeting point α would have been recommended because $10 \leq 25$. In the simulation experiment (Section 7.3.2), both voting rules are compared.

In the basic parameter setting, all travel times are considered as being equally weighted to enable a fair meeting point choice among all participants. However, it may be the case that one or multiple customers are under time pressure. In this case, the travel times can be weighted by a factor f, according to the individual time budgets, which influences the voting result. The dissatisfaction score S of a meeting point μ is then composed as:

$$S = f(\psi) \cdot \Delta t^{\text{total}}(\psi) + \sum_{\rho \in P} f(\rho) \cdot \Delta t^{\text{total}}(\rho) \tag{7.8}$$

In addition, passengers may have different maximum acceptable walking distances, due to luggage or impaired mobility. Hence, individual walking thresholds can be applied, by excluding all meeting points in the operational phase where the connection requires walking further than the threshold allows. As a result, only suitable meeting points are available in the vote.

7.3 Simulation experiment

In order to demonstrate the effect of the proposed algorithm, a simulation experiment is conducted.

7.3.1 Simulation setting

For the street network, the public transport infrastructure, the meeting points and the demand basis, the Braunschweig scenario is used, as explained in detail in chapter 5. Note that only petrol stations and parking places are used as meeting point candidates (705 in total), since these locations are most frequently used and rated most positively in ride-sharing (see survey in Section 4.2.2).

After all refinement steps (Section 7.2.1.2), only 94 meeting point candidates remain (see Figure 7.6). As can be seen, the meeting point candidates in more remote outskirts are mostly removed - this is due to the filtering by usage, since meeting points in these remote areas have been selected too infrequently in the simulation. As inbound and outbound nodes, six locations on the motorways have been manually selected, visualized as black triangles in Figure 7.6.

In the simulation, meeting points are determined for groups consisting of one driver and one to three passengers (randomly selected). The driver route is randomly chosen among the six available inlet nodes. U-turns (inbound and outbound inlet nodes are equal) are not allowed. The time of driver arrival at the inbound inlet node is randomly chosen between 6 am and 11 pm to avoid the night break. The passenger origin locations are randomly sampled within the investigation area, as explained in Section 5.4. Figure 7.7 shows an example of meeting point recommendation involving three passengers.

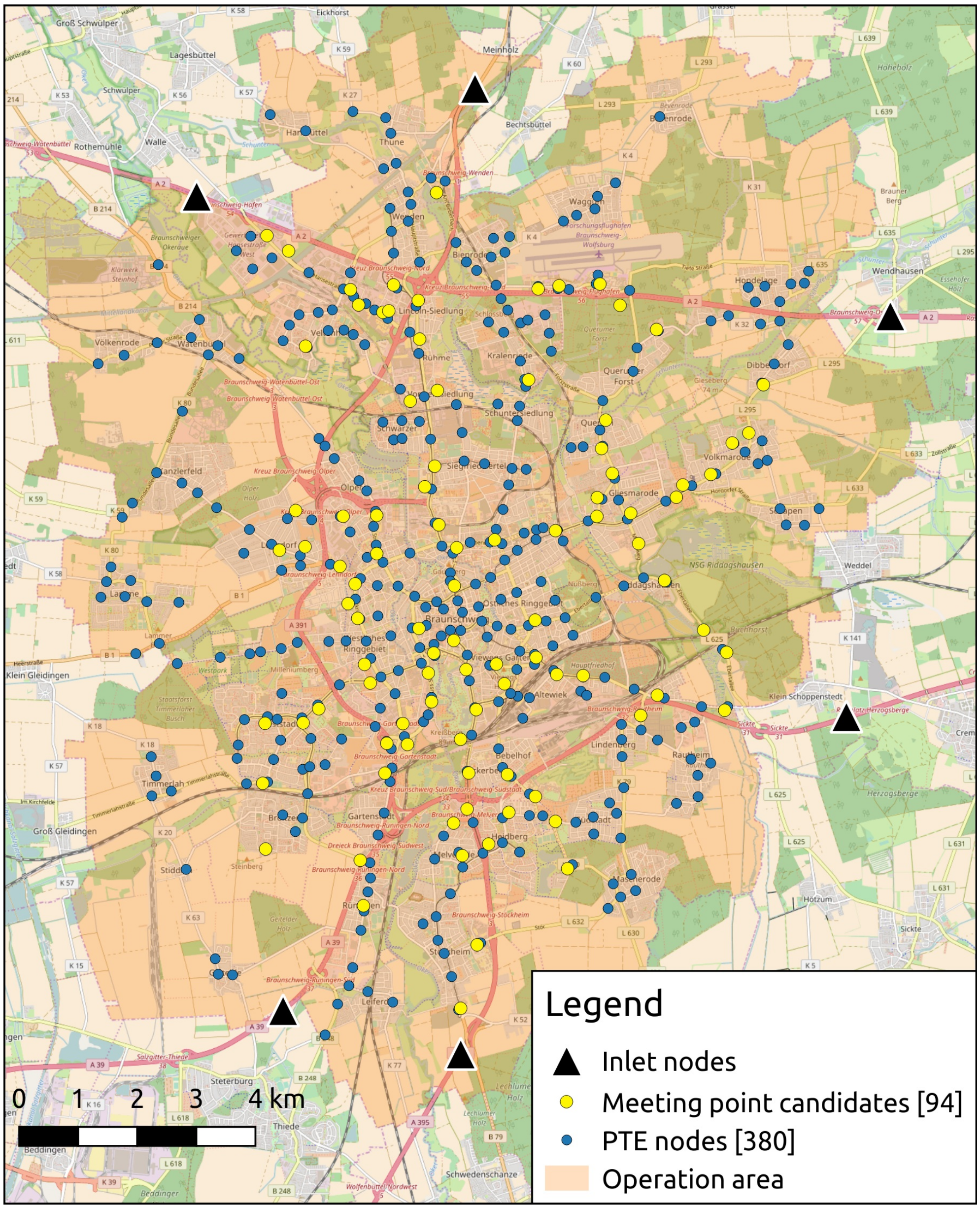


Figure 7.6: Operation area (city of Braunschweig) with inlet nodes, meeting point candidates and PTE nodes. Background: OpenStreetMap.

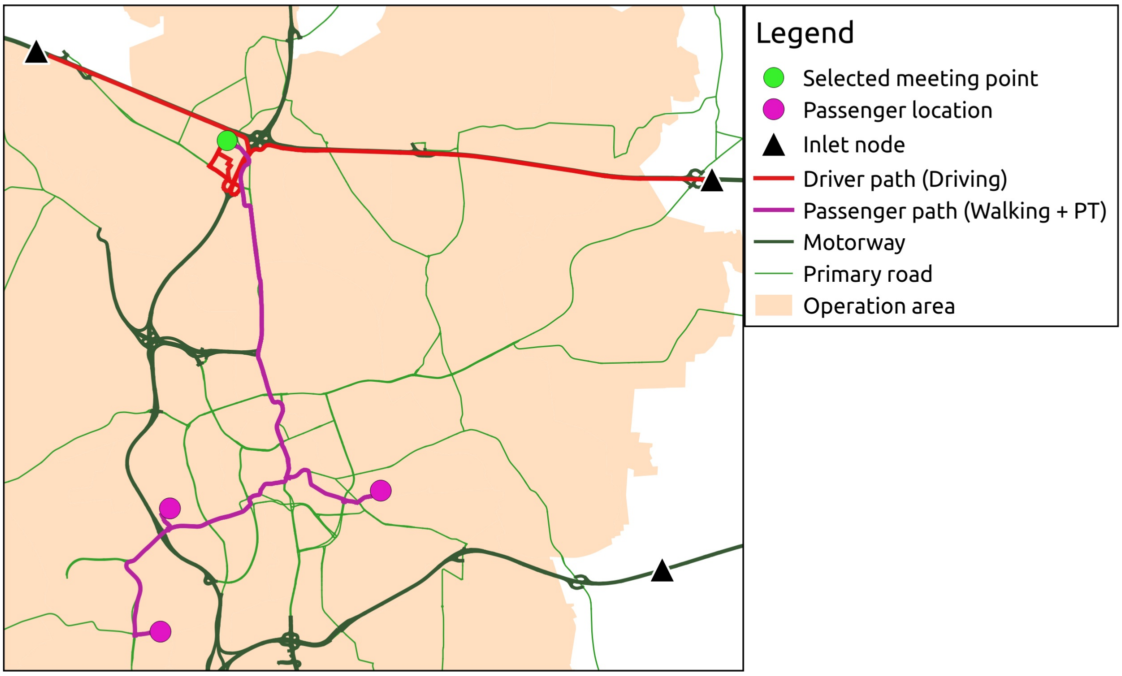


Figure 7.7: Example meeting point selection with three passengers involved.

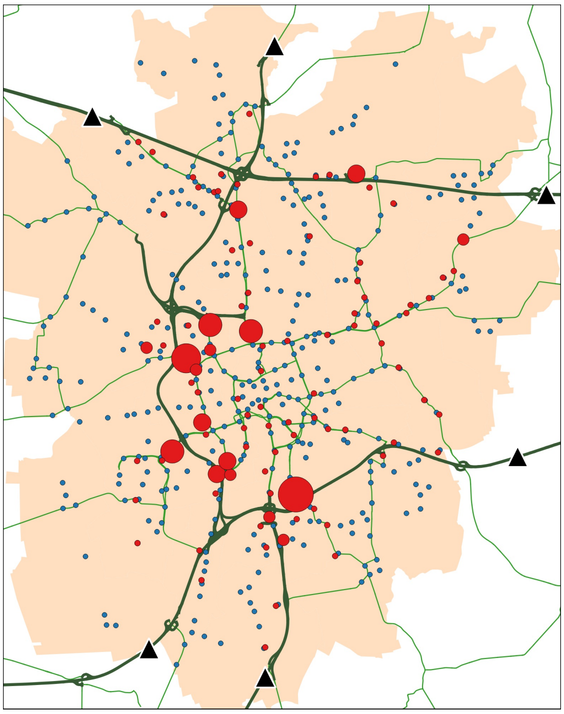
To keep the simulation simple, the voting is based only on time values. Individual preferences are not considered.

7.3.2 Results

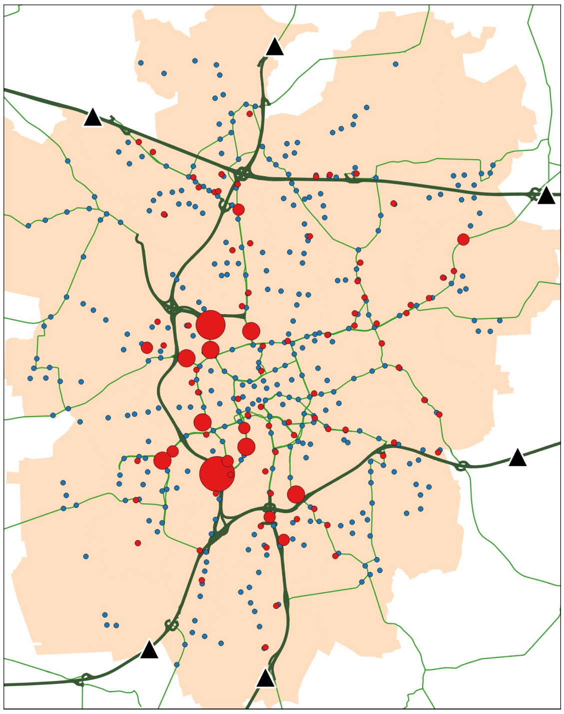
The first experiment (Figure 7.8) shows patterns for meeting point recommendations at noon at 12 pm (Figure 7.8a), in the later evening at 9 pm (Figure 7.8b), and in total over the whole day (Figure 7.8c). The maximum driver detour time was set to 30 minutes. The differences are only minor. Most changes between noon and the evening hours can be explained by a reduced service level of public transportation.

In a second experiment, the maximum allowed detour time parameter $\Delta t_*^{\mathrm{detr}}$ is varied. This parameter controls how far the drivers are willing or allowed to deviate from the direct route to reach a meeting point. Figure 7.9a shows the recommended meeting points for a tight 5 minute threshold, Figure 7.9b for 10 minutes. Not surprisingly, the most frequently recommended meeting points are located very close to a motorway exit. In the 5 minute case, the usage is very condensed into a few points. However, some meeting points are still selected far from a motorway in the eastern part of the city. These points have been selected in cases where the driver was taking the route from north east to south east (or reverse), and since there is no motorway, the route through the city is the shortest path anyway.

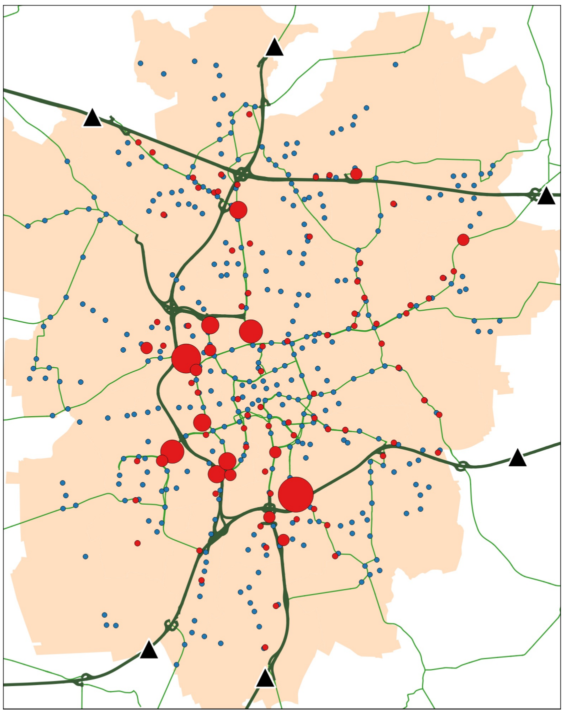
Figure 7.10 shows various statistics when using different values for $\Delta t_*^{\mathrm{detr}}$. Naturally, the average driver detour increases when the maximum allowed detour is increased, but not linearly (Figure 7.10a). The average driver detour to reach a meeting point converges to a value of 9.1 minutes (no detour time restrictions). Compared to the values revealed in the map-based user survey (Section 4.2.2), this value is rather large, since the majority of detour times in the survey is between two and three minutes. If the drivers are allowed or willing to deviate more from the direct route, the passengers have to travel less (Figure 7.10b).



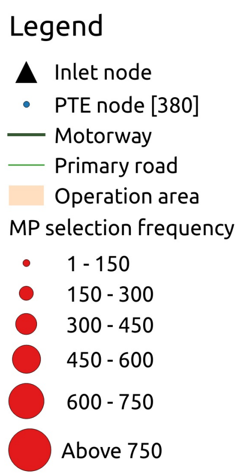
(a) MP selection frequency (Noon)



(b) MP selection frequency (Evening)

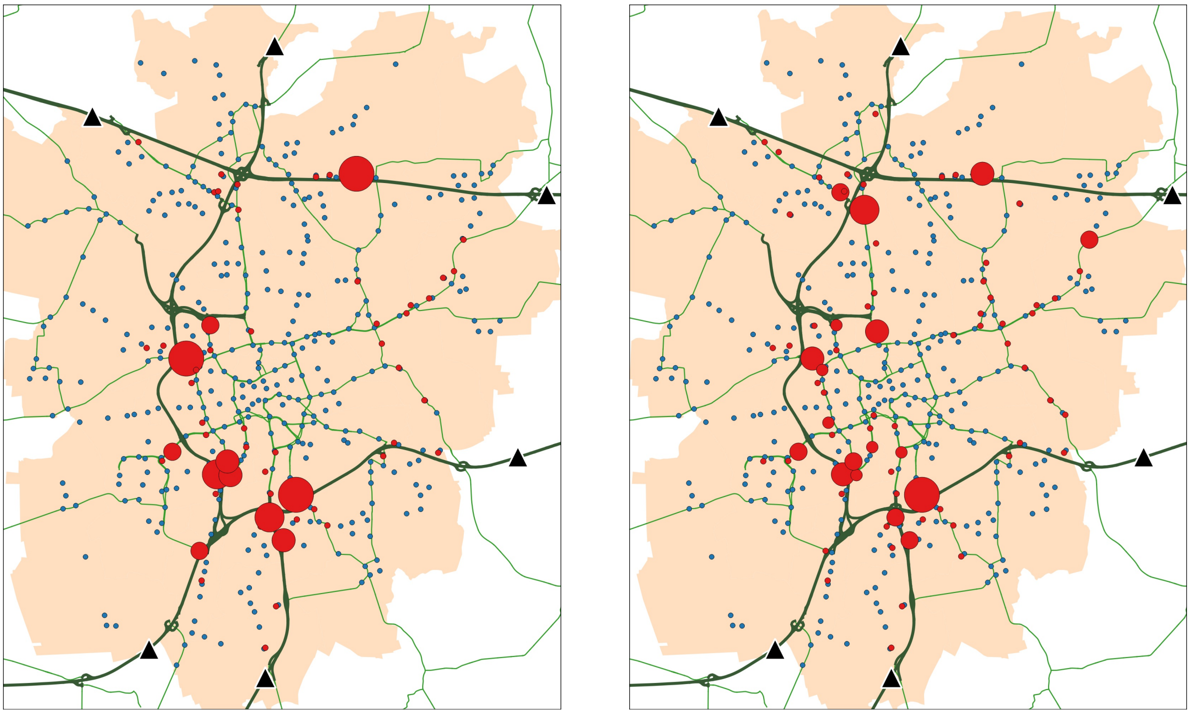


(c) MP selection frequency (Total)



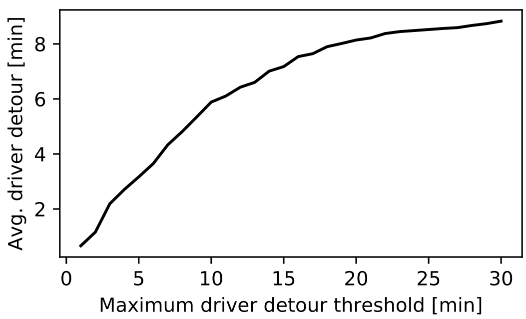
(d) Legend

Figure 7.8: Time-dependent meeting point selection frequency (10 000 simulation runs, each resulting in a single selection).

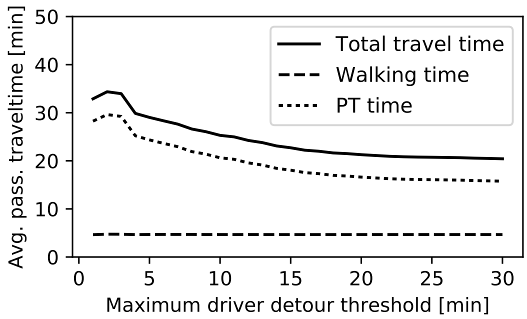


(a) MP selection frequency (5 minutes maximum driver detour) *(b) MP selection frequency (10 minutes maximum driver detour)*

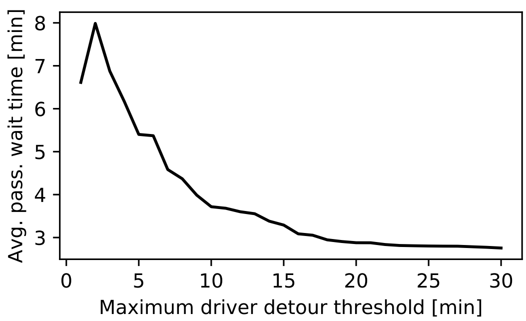
Figure 7.9: Detour-dependent meeting point selection frequency (10 000 simulation runs).



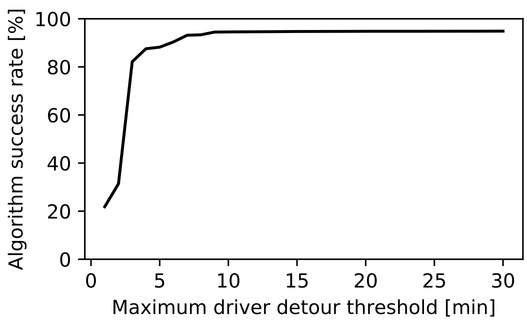
(a) Average driver detour time



(b) Average passenger travel time



(c) Average passenger waiting time



(d) Algorithm success rate

Figure 7.10: Simulation run with varying maximum detour threshold (10 000 runs each).

The passenger waiting time is defined as the time that a customer has to wait at a meeting point for the driver because of an early arrival. Note that the waiting time is not included in the average passenger travel time. In Figure 7.10c it can be seen that the waiting time also decreases with more flexibility in the meeting point selection. Finally, Figure 7.10d shows the algorithm success rate, indicating how often the algorithm was able to find a valid solution satisfying all constraints. As expected, with unreasonably strict detour constraints (e.g. one minute), the algorithm is only able to find a solution in 20 % of the requests, since no applicable meeting points can be reached. With 3 minutes of allowed detour time however, 80 % of the requests can be handled successfully.

As can also be seen, for 4.5 % of the requests it is not possible to find a common meeting point satisfying the constraints at all, regardless of the $\Delta t_*^{\mathrm{detr}}$ value. This happens because of requests from remote locations, where the public transport system does not offer rides frequently enough to reach the destination in time. In this case, the travellers would have to manually negotiate a meeting point.

A further experiment investigates the differences between using range voting and minimax voting, as described in Section 7.2.3.5. While the range voting leads to a minimized overall travel time for the group, the minimax voting aims at selecting a meeting point that minimizes the maximum travel time of the participants. Table 7.2 shows the differences, based on different group sizes (driver and one passenger, driver and two passengers and so forth). The results show that the selection of a voting rule has a significant impact on the travel time and meeting point choice.

The first row shows the percentage for how often the algorithm returned the same meeting point result for the two different voting rules. If the driver meets only one passenger, the same meeting point is recommended in 59 % of the cases. With larger groups, this value decreases. The following rows all refer to the case when the range voting returned a different vote from the minimax voting. In the second row, the average delay using minimax voting is given. Delay in this scope means the additional time that the group needs until reaching the outbound inlet node when using minimax voting rule. While the difference is relatively large for small groups (12.5 minutes for one passenger), the difference is only minor for large groups (below 4 minutes for 4 passengers). On the other hand, the travel time of the participant with the maximum travel time can be clearly reduced, as rows three and four show. With only one passenger involved, the travel time of the passenger is on average approximately halved (from 19 to 9 minutes), if minimax voting is used instead of range voting. Of course, this implies also an increased total travel time (12.5 minutes). With more passengers involved, the maximum travel times increase in general, and the differences between the voting rules decrease.

Table 7.2: Range voting vs. minimax voting (10 000 simulation runs).

Voting between driver and ...	1 passenger	2 passengers	3 passengers	4 passengers
Equal results of both voting rules	59 %	31 %	30 %	29 %
Different results: Average delay using minimax voting	12:29 min	5:32 min	4:30 min	3:57 min
Different results: Average maximum travel time using range voting	19:33 min	34:04 min	38:25 min	41:14 min
Different results: Average maximum travel time using minimax voting	9:20 min	26:16 min	31:11 min	34:37 min

7.4 Discussion

The simulation shows that the proposed algorithm is, in theory, capable of handling many requests within a short time, and also the suggested locations look reasonable from manual inspection. Due to the precomputation, average processing response times of 5 to 10 milliseconds per request could be reached (Python 3.5 on Ubuntu Linux, running on AMD FX-6100 using a single core and 4 GB RAM), which makes real-time applications with a high request frequency conceivable. Of course, this comes at the cost of long precomputation times. For the Braunschweig example, the precomputation took approximately 10 hours (Python 3.5 and OpenTripPlanner instance on FreeBSD, running on Intel Xeon E5410 with 32 GB RAM). For a fully equipped real-world operation, the precomputation phase would have to be computed six times: standard weekday, Saturday and Sunday, each for meeting and divergence points. The storage space requirements are manageable however, with approximately 15 - 20 MB for an A_P matrix in the Braunschweig example using NumPy[1] binary format.

A limitation is clearly that the algorithm can handle requests only from drivers whose planned route passes an inlet node inbound and an inlet node outbound. Hence, drivers approaching or leaving on smaller streets cannot be considered. However, it is often not a problem to insert more inlet nodes, as the driver time precomputation is usually much quicker compared with the public transport precomputation. Furthermore, introducing new inlet points scales linearly if the number of meeting points remains constant.

Note further that the algorithm can recommend only one single meeting point, regardless of passenger numbers. In some cases it would be more reasonable to recommend two or more separate meeting points that have to be approached by the driver successively, e.g. if the first passenger is located close to a motorway exit on the west side, and the second passenger is located close to a motorway exit on the eastern side of the city. A simple adaptation to tackle this issue is to apply the algorithm iteratively on the single passengers, if the necessary travel times for a common meeting point are assessed as unreasonably (or unacceptably) large, resulting in a recommendation of multiple different meeting points. The drawback is that the recommended meeting points are then not time-optimal, since the driver travel times between the meeting points are not known.

Certainly, Braunschweig is a rather small city. However, implementing it for larger cities is possible due to the polynomial complexity. When limiting the number of meeting points to a specific number, the time and space needed grows only linearly with more PTE nodes available. For metropolitan areas consisting of distinguishable cities or suburbs, the investigation area can be split up into distinct regions. The drawback is that each passenger then has to be picked up in the region of origin; a change of regions is not possible with the approach presented.

Another advantage of the approach is that the algorithm is capable of reacting in real time to congestion and disturbances in the street network. If such a change in driving times occurs, it is sufficient to add extra time to the corresponding values in the matrices $A_{\Psi}^{\text{inbound}}$ and $A_{\Psi}^{\text{outbound}}$, and the algorithm will automatically adapt to the modified situation. On the other hand, a drawback is that changes in the public transport network are more difficult to include. That is, if the route network or the timetable changes, the passenger travel time matrix A^{P} has to be recalculated, at least partly.

[1] `http://www.numpy.org/`)

8 Meeting points for demand-responsive transportation

In this chapter, the usage of meeting points for demand-responsive transportation is investigated and discussed. The chapter is divided into a motivation (Section 8.1), a description of the proposed method (Section 8.2), a simulation experiment (Section 8.3) and a concluding discussion (Section 8.4). Parts of the workflow have been implemented by Aleksandar Trifunović (TU Braunschweig) and Ronny Kutadinata (University of Melbourne); these parts are appropriately marked to emphasize their contribution, which is highly acknowledged. The work presented in this chapter is publicly available as arXiv e-print (Czioska et al. 2017).

8.1 Motivation

Shared demand-responsive transportation (SDRT) services, also known as dial-a-ride services, provide a mobility solution based on door-to-door transportation on request (see chapter 2.2.2). However, there is a lack of research about how meeting points can be determined for SDRT services, and the impact of using meeting points (Section 3.2). The recent extension of services towards a meeting point based operation by large companies such as Uber, Bridj or Via shows the relevance of research in this area.

However, most of the service providers currently just use nearby street junctions as meeting point recommendations, although suitable locations for a safe and convenient pick-up and drop-off are not ubiquitous. For buses, it may, for instance, not be possible to stop at a major junction. Moreover, feasible meeting point candidates, such as public parking areas, are often unequally distributed within the city area and dissimilarly reachable by vehicles and pedestrians. The impacts of these limitations need to be investigated regarding a SDRT scenario.

The chapter is roughly split into two parts. Firstly, a workflow is presented to solve the problem of SDRT with meeting points (in the following abbreviated as SDRT-MP). The SDRT-MP extends the conventional SDRT problem by introducing a constraint that all customers have to board and alight at a meeting and divergence point from a set of predefined locations. Customers are supposed to walk to the meeting points, but they have a maximum walking distance and time window constraints. Each time a vehicle stops at a meeting point, one or multiple customers can board or de-board.

Secondly, the proposed method is applied to a real-world scenario, based on the data presented in chapter 4, to demonstrate the impacts of using meeting points, in contrast with a door-to-door service. The notation used in this chapter is listed in Table 8.1.

Notation	Unit	Description
$\rho \in P$	-	Passenger requests
$\mu \in M$	-	Meeting point candidates
$\delta \in D$	-	Divergence point candidates
$\zeta \in Z$	-	Trip (group of passengers)
λ^+	-	Origin location
λ^-	-	Destination location
v^+	-	Origin node
v^-	-	Destination node
t^+	s	Departure time
t^-	s	Arrival time
$t_\uparrow$	s	Earliest time
$t_\downarrow$	s	Latest time
Δt^{drive}	s	Driving time
Δt^{walk}	s	Walking time
Δt_*^{wait}	s	Maximum passenger waiting time
$\Delta t_*^{\text{serve}}$	s	Service time (for boarding and alighting)
Δt_*^{detr}	s	Maximum allowed vehicle detour time
r_*^{detr}	-	Maximum allowed vehicle detour time ratio
d_*^{walk}	m	Maximum passenger walking distance
σ	-	Shortcut ratio (for alternative MPs)
k_1	-	Maximum cluster size for the initial clustering
k_2	-	Maximum cluster size for the re-clustering
q_*	-	Maximum vehicle capacity
c_*^{dist}	km^{-1}	Vehicle distance dependent cost
c_*^{vehi}	-	Vehicle capital cost
c_*^{wait}	s^{-1}	Passenger wait time cost
c_*^{late}	s^{-1}	Passenger late time cost
α	-	Passenger wait time cost growth
β	-	Passenger late time cost growth

Table 8.1: Notation used in this chapter.

8.2 Proposed method

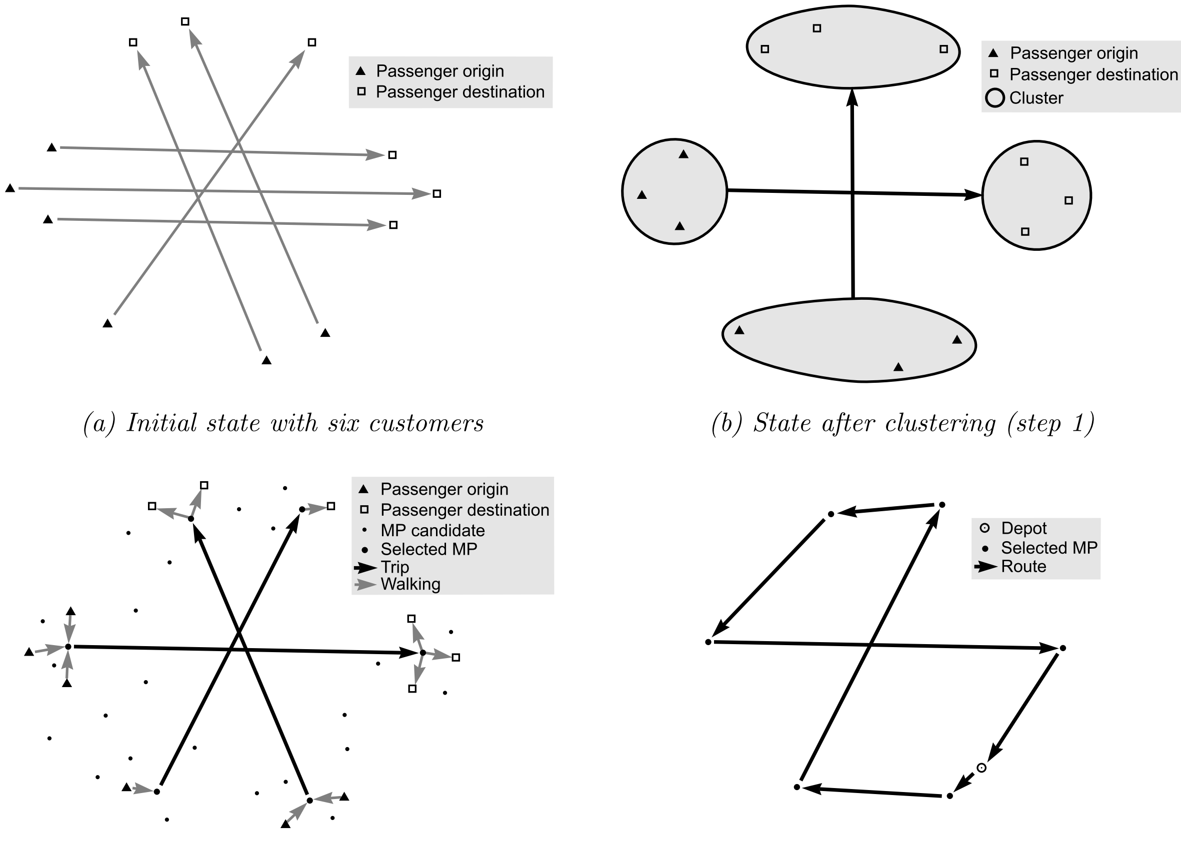
The SDRT-MP problem is tackled by a multi-step approach, basically consisting of three discrete steps:

1. **Clustering**

2. **MP candidates selection**

3. **Routing Optimisation with final MP selection**

Figure 8.1 visualizes the processing chain. In a nutshell, the demand (Figure 8.1a) is initially clustered into groups of equal size, with similar itineraries and time schedules (Figure 8.1b). Secondly, each group is separately split into *trips*, with each trip having a common meeting and divergence point and potential alternative meeting points (Figure 8.1c). Finally, the vehicle routing problem is solved, to construct routes and fix the meeting point selection (Figure 8.1d). All steps are explained in detail in the course of this Section.



(a) Initial state with six customers

(b) State after clustering (step 1)

(c) State after MP candidates selection (step 2)

(d) State after vehicle optimization (step 3)

Figure 8.1: Basic workflow with clustering, MP selection and route optimization.

The vehicle routing optimization (step 3) can, in theory, be solved with any (modified) optimization method that is able to solve vehicle routing problems. In this project, a neighbourhood search approach is used. However, since this step is very time-consuming when being performed on large input data, the workflow is slightly modified to a five-step version, to enable parallel processing. For this, the trips resulting from step 2 are again clustered (*Re-Clustering* phase) such that the trip data is grouped into equally large instances. For the re-clustering, the same technique as for the initial clustering (step 1) is used. The clustered trips are then processed separately and in parallel by the route optimization solver. Since the partitioned instances are then much smaller, the computation time can be significantly reduced. Finally, a *Concatenation* step is necessary to merge the vehicle routes again.

8.2.1 Clustering

In this first step, similar customer transportation requests are clustered into groups of limited size. Although spatial and temporal constraints are often imposed already in the clustering phase, the proposed workflow applies the constraint checking later, during the MP candidates selection step (Section 8.2.2). The reason is that then the actual distances based on the street network can be applied, instead of Euclidean distances.

Clustering techniques are a common heuristic to solve and combat the computational complexity of dial-a-ride problems (*cluster-first, route-second* approach, see section 2.2.2.2). In essence, it means that, at first, the customers are clustered based on their itineraries and time schedules, and subsequently a single-vehicle routing is applied on each cluster separately (Bodin & Sexton 1986). The clusters can also be treated as single request in a global optimization, as done by Martínez et al. (2014). With this approach, it is possible to serve two distinct clusters using a single vehicle at the same time. The clustering technique used in this project also works in this way. It was mainly developed and implemented by Aleksandar Trifunović[1].

In a nutshell, it works as follows. Each customer request is represented by a vector of features, in this case the origin and destination coordinates, as well as the desired departure time. The function to compute the distance between two customers is defined as the Euclidean distance. Hence, also the time is considered as a spatial distance. The clusters are processed iteratively. At first, the first (empty) cluster is processed, and the closest customer is appended, which is, in this case, a random selection. Then, the next customer to be appended is the one with the lowest sum of distance costs to all customers already in the cluster. This step is repeated until the cluster has reached its maximum size, then the next cluster is processed.

8.2.2 Meeting Point Candidates Selection

In this step, the previously determined clusters are investigated for feasible meeting and divergence points. This is the main part of this section. Since the previous clustering step uses Euclidean distance instead of the street network for the distance calculations, the walking and time thresholds have to be checked within this step to satisfy the constraints of real-world conditions.

Most likely, not all customers of a cluster will be feasible for a single meeting and divergence point, so the group has to be split up into subgroups which can reach one (or more) common meeting and divergence point(s). Such a subgroup is called a *trip* ζ. It is defined as a tuple consisting of one or more passengers with one or more meeting and divergence point candidates, each associated with a corresponding time window, indicating the valid boarding and de-boarding times:

$$\begin{aligned}\zeta_i = \Big(& \{\rho_{i,1}, \quad \rho_{i,2}, \cdots\}, \\ & \{(\mu_{i,1}, t_\uparrow(\mu_{i,1}), t_\downarrow(\mu_{i,1})), (\mu_{i,2}, t_\uparrow(\mu_{i,2}), t_\downarrow(\mu_{i,2})), \cdots\}, \\ & \qquad \{(\delta_{i,1}, t_\uparrow(\delta_{i,1}), t_\downarrow(\delta_{i,1})), (\delta_{i,2}, t_\uparrow(\delta_{i,2}), t_\downarrow(\delta_{i,2})), \cdots\}\Big)\end{aligned} \tag{8.1}$$

A trip is considered as feasible if the time windows and walking distances are within the thresholds for all customers of the trip. The earliest and latest pick-up times for a meeting point μ are determined by

$$t_\uparrow(\mu) = t^+ + \Delta t^{\text{walk}}(v^+ \to \mu) \tag{8.2}$$

and

$$t_\downarrow(\mu) = t^+ + \Delta t^{\text{walk}}(v^+ \to \mu) + \Delta t_*^{\text{wait}}, \tag{8.3}$$

whereas the earliest and latest drop-off times for a divergence point δ are calculated by

$$t_\uparrow(\delta) = t^+ + \Delta t^{\text{drive}}(v^+ \to v^-) - \Delta t^{\text{walk}}(v^- \to \delta) - \Delta t_*^{\text{serve}} \tag{8.4}$$

[1]Institute of Transportation and Urban Engineering, TU Braunschweig, Germany. a.trifunovic@tu-braunschweig.de

and

$$t_{\downarrow}(\delta) = t^{+} + \Delta t_{*}^{\text{wait}} + \min\left(r_{*}^{\text{detr}} \cdot \Delta t^{\text{drive}}\left(v^{+} \to v^{-}\right),\right.$$
$$\left.\Delta t^{\text{drive}}\left(v^{+} \to v^{-}\right) + t_{*}^{\text{detr}} - \Delta t^{\text{walk}}\left(v^{-} \to \delta\right) - \Delta t_{*}^{\text{serve}}\right), \quad (8.5)$$

respectively.

Single MP / DP selection

The task is therefore to split up the initial group resulting from the clustering as efficiently as possible into subgroups, such that the group size of the subgroups remains as large as possible. This issue can be interpreted as a *set covering* problem (see Section 2.1.5), with the goal of finding the smallest number of subgroups that satisfies the walking and time constraints of the passengers from a given cluster. Since the set covering problem is known to be NP complete, the input size is naturally limited. However, due to the limited cluster size resulting from step 1, a set covering algorithm to determine the optimal solution can still be applied. For the approach presented, a recursive dynamic programming algorithm is proposed.

Algorithm 4 describes the procedure for the meeting point determination for a given cluster. The *2-Combinations* function mentioned yields all possible paired combinations of a given set, for example: *2-Combinations (a,b,c,d) = [a-bcd, b-acd, c-abd, d-abc, ab-cd, ac-bd, ad-bc]*. The algorithm returns the optimal combination with a minimum number of subgroups.

Initially, the algorithm attempts to place all customers of a cluster into one single trip, with a common meeting and divergence point. If this is spatially and/or temporally infeasible, the group is split into all possible subgroup combinations, and their feasibility is checked likewise. This is done recursively for each subgroup until a feasible solution is found. The main objective is to split the initial cluster into as few separate trips as possible. If there are multiple different combinations with the same number of necessary trips, the combination with the least sum of squared walking distances of all customers is chosen. The squaring is necessary to penalize longer walking distances more than short distances, so that the walking distances among the passengers are equally distributed.

The theoretical complexity of the algorithm can be expressed as $\mathcal{O}(2^n)$, since the power set of all passengers in a group has to be processed. To speed up the computation, the intermediate results are stored during the recursive process. If the same request occurs again during the computation, the result can be fetched from a lookup table (Dynamic Programming principle, see Section 2.1.4). With this technique it is possible to achieve optimal results for small to medium sized clusters, still within a reasonable time.

The maximum cluster size for the initial clustering is a tuning parameter that balances the trade-off between the quality of the result and the computation time. Larger clusters lead to better results in terms of lowering the necessary boarding stops, since the meeting point determination method yields the optimal solution for a given group. On the other hand, the computation time grows exponentially with increasing cluster size. Figure 8.2 shows an experimental determination of the resulting quality and running time, depending on the cluster size, based on 5 000 customers. The maximum cluster size varies from 2 to 12; the resulting number of trips and the computation times are recorded using parallel processing on 4 cores. As can be seen, a larger input cluster size leads to a fewer total number of trips (which is better). On the other hand, the computation time grows very quickly. In the simulation experiments, a maximum cluster size of 11 was chosen, to obtain good results but still with a reasonable computation time.

Algorithm 4 Meeting Point Determination

procedure FINDCOMBINATION(c) ▷ Procedure is called for every cluster $c \in C$
    Given: Passenger group $c \leftarrow \{\rho_1, \rho_2, \ldots\}$
    $\mathcal{M} \leftarrow \mathcal{M}_{\rho_1} \cap \mathcal{M}_{\rho_2} \cap \ldots$ ▷ Find common meeting points
    $\mathcal{D} \leftarrow \mathcal{D}_{\rho_1} \cap \mathcal{D}_{\rho_2} \cap \ldots$ ▷ Find common divergence points
    $\mathcal{M}_{\mathrm{TF}}, \mathcal{D}_{\mathrm{TF}} \leftarrow \{\}$
    if $\mid \mathcal{M} \mid \geq 1$ & $\mid \mathcal{D} \mid \geq 1$ **then** ▷ Check if at least one common MP and DP exists
        for $\mu \in \mathcal{M}$ **do** ▷ Check temporal feasibility for common meeting points
            $t_{\uparrow}(\mu) \leftarrow \max_{\rho \in c}(t_{\uparrow}(\rho, \mu))$ ▷ Earliest possible departure time at MP
            $t_{\downarrow}(\mu) \leftarrow \min_{\rho \in c}(t_{\downarrow}(\rho, \mu))$ ▷ Latest possible departure time at MP
            if $t_{\uparrow}(\mu) \leq t_{\downarrow}(\mu)$ **then** ▷ Check time feasibility for the meeting point
                $\gamma(\mu) \leftarrow \sum_{\rho \in c} d^{walk}(v^{+}(\rho) \rightarrow \mu)^2$ ▷ Calculate cost for this meeting point
                $\mathcal{M}_{\mathrm{TF}} \leftarrow \mathcal{M}_{\mathrm{TF}} \cup (\gamma(\mu), \mu)$ ▷ Time feasible - add to set
            end if
        end for
        Compute $\mathcal{D}_{\mathrm{TF}}$ likewise for divergence points
        if $\mid \mathcal{M} \mid \geq 1$ & $\mid \mathcal{D} \mid \geq 1$ **then** ▷ Check if time feasible common MP and DP exist
            $\gamma^*(\mu), \mu^* \leftarrow \min(\mathcal{M}_{\mathrm{TF}})$ ▷ Find μ with minimal cost
            $\gamma^*(\delta), \delta^* \leftarrow \min(\mathcal{D}_{\mathrm{TF}})$ ▷ Find δ with minimal cost
            return $\mathcal{S} \leftarrow (1, \gamma^*(\mu) + \gamma^*(\delta), \mu^*, \delta^*)$ ▷ Return combined cost, μ and δ
        end if
    else ▷ No common meeting and divergence point exists
        $\mathcal{S}_{\mathrm{Cur}} \leftarrow (|c|, \infty, \{\}, \{\})$ ▷ Initialize current best solution
        for $c_1, c_2 \in$ 2-COMBINATIONS(c) **do** ▷ Iterate through possible pairwise combinations
            $\mathcal{S}_1 \leftarrow$ FINDCOMBINATION(c_1)
            $\mathcal{S}_2 \leftarrow$ FINDCOMBINATION(c_2)
            $\mathcal{S}_{\mathrm{New}} \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2$
            if $\mathcal{S}_{\mathrm{New}}$ better than $\mathcal{S}_{\mathrm{Cur}}$ **then** ▷ Better = Less separate groups
                $\mathcal{S}_{\mathrm{Cur}} \leftarrow \mathcal{S}_{\mathrm{New}}$
            end if
        end for
    end if
    return $\mathcal{S}_{\mathrm{Cur}}$
end procedure

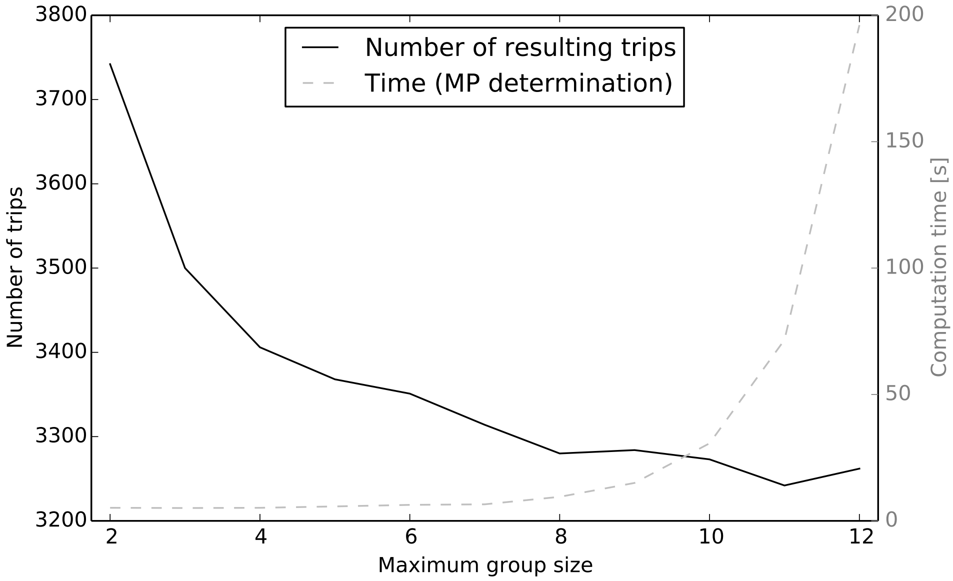


Figure 8.2: The number of trips and the computation time for different maximum cluster size settings.

Alternative MP / DP selection

The previously described algorithm returns exactly one meeting and one divergence point for each trip. However, there can be other meeting points worth considering from the operator's perspective. For instance, consider the scenario shown in Figure 8.3. In this case, the closest meeting point from the passenger origin (Point A) is chosen, which is located north of the motorway. Two other meeting point candidates, namely B and C, would also be feasible for the trip, but they have not been chosen because of longer walking distances. Assume further that the vehicle is currently located south of the highway. For the operator, it could then be advantageous in the routing phase to also consider meeting point candidate C, since it offers the possibility of approaching the passenger from the south, without having to take a large detour around the motorway. From the passenger's perspective, it is only a minor extension of the walking path via the footbridge.

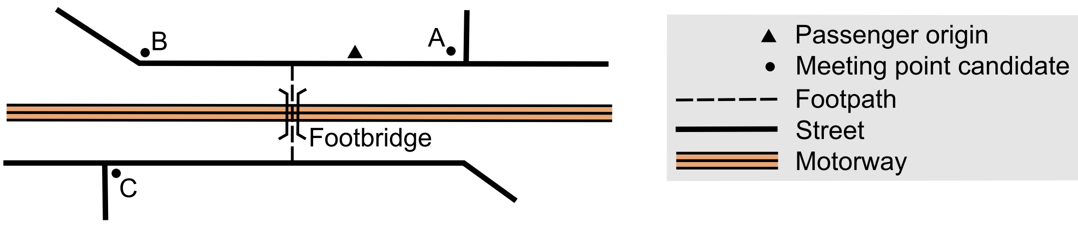


Figure 8.3: Schematic drawing of a situation with useful alternative meeting point search.

To combat this, a second algorithm is proposed to identify alternative meeting points, which can be considered during the route optimization phase (Section 8.2.3). Note that only one meeting point is still served by the vehicle, but the alternative meeting points provide more options for creating shorter routes.

More formally, the relevance of a meeting point alternative candidate μ_a with respect to an already considered meeting point μ_c is expressed by a *shortcut ratio*, indicating the ratio between driving time and walking time between those two meeting points (see Equation 8.6). In the example above (Figure 8.3), the ratio with respect to point A (already considered) would be low for point B and high for point C. To limit the number of meeting points to be considered, a threshold σ needs to be applied. Alternative meeting point candidates are therefore considered if

$$\frac{\Delta t^{\text{drive}}(\mu_c \rightarrow \mu_a)}{\Delta t^{\text{walk}}(\mu_c \rightarrow \mu_a)} \geq \sigma, \tag{8.6}$$

where μ is the initially identified meeting point candidate.

The proposed approach is outlined in Algorithm 5. For each trip resulting from Algorithm 4, multiple meeting points are available, and one meeting point is already selected. The algorithm recursively checks whether there are alternative meeting points available where the shortcut ratio is above the threshold value for all meeting points already considered. Note that also the corresponding time windows of the alternative meeting points are checked, in order to prevent time constraint violations.

Algorithm 5 Recursive Alternative Meeting Point Search

Input
$\mathcal{M}$ ▷ Full set of meeting points that all passengers of a trip can reach
$\mathcal{M}_{\mathrm{C}}$ ▷ Set of meeting points to be considered (initially one item)
A ▷ Travel time ratio matrix
σ^* ▷ Travel time ratio threshold
procedure FINDMPALTERNATIVES($\mathcal{M}$,$\mathcal{M}_{\mathrm{C}}$)
  $\mathcal{R} \leftarrow \emptyset$ ▷ Initialize empty result set
  for $\mu_1 \in \mathcal{M} \setminus \mathcal{M}_{\mathrm{C}}$ **do** ▷ Iterate through all non-considered meeting points
    $\mathcal{S} \leftarrow \emptyset$ ▷ Initialize an empty temporary set
    for $\mu_2 \in \mathcal{M}_{\mathrm{C}}$ **do** ▷ Iterate through already considered meeting points
      $\mathcal{S} \leftarrow \mathcal{S} \cup [\mathrm{A}[\mu_1][\mu_2],\mu_1]$ ▷ Add ratio value and meeting point to temporary set
    end for
    $\mathcal{R} \leftarrow \mathcal{R} \cup \min(\mathcal{S})$ ▷ MP with minimum value among already considered MPs
  end for
  $\gamma \leftarrow \max(\mathcal{R})$ ▷ Currently not considered meeting point with the maximum ratio
  if $\gamma[0] \geq \sigma^*$ **then** ▷ Check if value is above the threshold
    $\mathcal{M}_{\mathrm{C}} \leftarrow \mathcal{M}_{\mathrm{C}} \cup \gamma[1]$ ▷ Add this point to the set of considered points
    return FINDMPALTERNATIVES($\mathcal{M}$,$\mathcal{M}_{\mathrm{C}}$) ▷ Search for more points
  end if
  return $\mathcal{M}_{\mathrm{C}}$ ▷ Return set of meeting points to be considered
end procedure

8.2.3 Route Optimization with Final Meeting Points Selection

In this step, trips having one or more meeting point alternatives resulting from the previous unit are combined and concatenated to vehicle routes. In order to speed up the process, a second clustering is initially applied to the trips, in order to form equally sized trip bunches with similar itineraries, which can then be solved in parallel. For this, the clustering method already described in section 8.2.1 is reused, but with a different threshold. This makes it necessary to append a post-processing step after the vehicle routing, to combine the results of the simultaneously derived vehicle routes.

The proposed vehicle route optimization was developed and implemented by Ronny Kutadinata[2]. For further details, the reader is kindly referred to Kutadinata et al. (2017).

The resulting route optimisation problem differs from those in the literature mainly because of the alternative meeting points, determined by Algorithm 5. The proposed method to tackle the route optimisation problem with meeting points is derived from the approach presented by Kutadinata et al. (2017) and uses a two-layer neighbourhood search approach. The top layer optimises the trip allocation to vehicles, and the bottom layer optimises the route of each vehicle, including the selection of meeting and divergence points.

To start the algorithm, an initial assignment of trips to vehicles is created in the top layer. Using this assignment, the bottom layer is called for every vehicle to optimize its route. An initial route is created and subsequently improved by choosing a set of *neighbours* in the solution space. A neighbour is created by removing and appropriately reinserting a stop within the route. Then, the optimal selection of meeting and divergence points is determined, using a Dynamic Programming approach. Finally, the algorithm *moves* to the best neighbour, and the process is repeated. The route optimization finishes after a certain number of iterations or if the solution does not change significantly. Then the solution is returned to the top layer, which operates in a similar manner. Here, a neighbour is obtained by removing a random trip from the allocated vehicle, and reinserting it into another randomly chosen vehicle. Also, the top layer optimization procedure terminates after a predefined number of iterations, or if certain quality requirements are fulfilled.

The optimization function includes a number of terms, such as the *service level cost*, which takes into account the passenger's late time, pick-up wait time, and detour time. The penalization of the parameters can be adjusted by using various polynomial forms of penalty terms. For instance, a quadratic term can be used to penalize longer wait/late times more than short wait/late times. Note that the time windows are treated as soft constraints, which is different from typical formulations in the literature (Cordeau & Laporte 2007; Baldacci et al. 2012). As an example, it allows the optimisation algorithm to choose a solution that has late services, but which may be justified by savings in the number of necessary vehicles. Typically, higher penalty weight parameters are used to avoid an unreasonably high number of late arrivals.

The output of the optimization process is a group of routes, each route performed by a vehicle. Since the previous optimization is performed in parallel for each cluster of trips, some of the routes can now again be concatenated to reduce the total number of routes (and consequently the total number of vehicles used). Thus, this step can be described as a problem of maximizing the number of concatenations by using a Linear Programming (LP) approach (see Section 2.1.1). To ensure that a concatenated route can still be feasibly

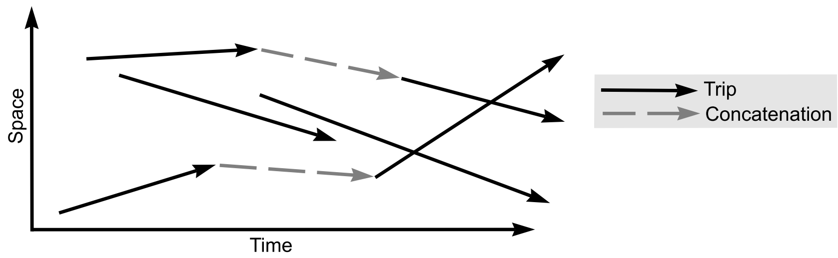


Figure 8.4: Concatenation of routes to form longer ones.

[2]Department of Infrastructure Engineering, The University of Melbourne, Parkville, Victoria 3010 Australia. ronny.kutadinata@unimelb.edu.au

served by a vehicle, a constraint is applied so that there is enough time to travel from the last stop of a preceding route to the first stop of the subsequent one. The proposed concatenation approach is visualized as a simplified 2D version in Figure 8.4.

8.3 Simulation experiment

To evaluate the potential benefits of the use of meeting points as proposed by the workflow, a simulation experiment is carried out, comparing the performance of a meeting point based service (MP) with a conventional door-to-door service (DS). While in the DS case, all customers are picked up and dropped off separately at their respective origins and destinations, the MP case allows for a grouping of requests and expects the customers to walk to meeting points. A waiting time is possible due to different arrival times of other passengers using the same meeting point. Note that in both cases (MP and DS) the vehicles can accommodate multiple passengers at a time, but only in the MP case is it possible for multiple passengers to board and alight at the same location.

Technically, in the MP service, all steps of the proposed workflow are executed, whereas the DS service uses only the vehicle routing optimization, applied on the raw demand data. As a result, this simulation focuses on highlighting the benefits of clustering and meeting point selection rather than the route optimization itself. For the simulation experiments, the parallel version of the route optimization is used to solve the instances within a reasonable processing time.

8.3.1 Simulation setting

The simulation experiment investigates the impact of using meeting points for various demand densities. The parameters used in the optimisation and meeting point algorithm are shown in Table 8.2. It is expected that, as the demand density increases, the difference between the MP and the DS case becomes more significant.

Table 8.2: Workflow parameters.

Notation	Unit	Description	Value used for simulation
Δt_*^{wait}	s	Maximum passenger waiting time	1200 s
d_*^{walk}	m	Maximum passenger walking distance	800 m
t_*^{detr}	s	Maximum allowed vehicle detour time	1200 s
r_*^{detr}	%	Maximum allowed vehicle detour time percentage	25 %
$\Delta t_*^{\text{serve}}$	s	Vehicle service time (for boarding/alighting procedure)	120 s
σ	%	Shortcut ratio threshold (see Section 8.2.2)	50 %
k_1	-	Maximum cluster size for the initial clustering	11 passengers
k_2	-	Maximum cluster size for the re-clustering	10 trips
q_*	-	Maximum vehicle capacity	9 passengers
c_*^{dist}	km^{-1}	Vehicle distance dependent cost	1 / km
c_*^{vehi}	-	Vehicle capital cost	2000
c_*^{wait}	s^{-1}	Passenger wait time cost	0.5 / second
c_*^{late}	s^{-1}	Passenger late time cost	5 / second
α	-	Passenger wait time cost growth	0.5
β	-	Passenger late time cost growth	2

To this end, a total of 40 000 randomly generated passenger requests are used as input (see Section 5.4). They are further subdivided into seven demand instances (1 000, 4 000, 7 000, 10 000, 20 000, 30 000 and 40 000), which are used for different purposes. To simplify the experiment, only a static problem is considered, i.e. all trips are assumed to be known in advance.

8.3.2 Results

Due to the missing walking distance and time constraint checking, the initial clustering phase yields (almost) exclusively groups having the maximum allowed size (in our case 11, see Figure 8.2). Then, the meeting point selection step splits these groups into feasible subgroups that satisfy the constraints. Figure 8.5 visualises the group size histogram after the MP candidates selection step for four cases with different total numbers of customers (10 000 to 40 000).

As can be seen, the majority of customers boards alone or with one, two or three other passengers. Larger groups are more uncommon, and only groups with the maximal size (here: 11) are slightly more frequent, as they cover cases where the cluster of customers with similar itineraries could have been even larger, e.g. commuters from a densely populated residential district to the city centre. Note that the group size can be higher than the actual maximum vehicle capacity, which is set to 9 to imitate minibuses. The actual assignment of passengers to minibuses is part of the vehicle routing phase, since only there can the actual vehicle occupancy be handled. As an example, a group with size 11 could be transported by three different vehicles: one with 5 spare seats, one with 3 spare seats and another one with at least 3 spare seats.

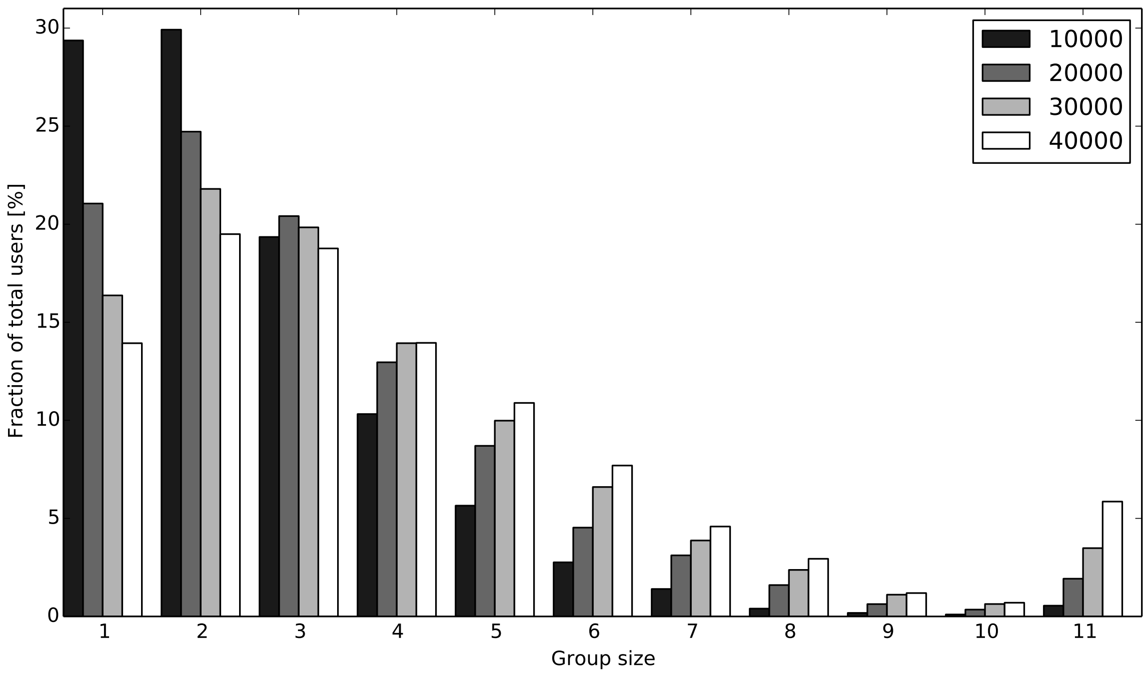
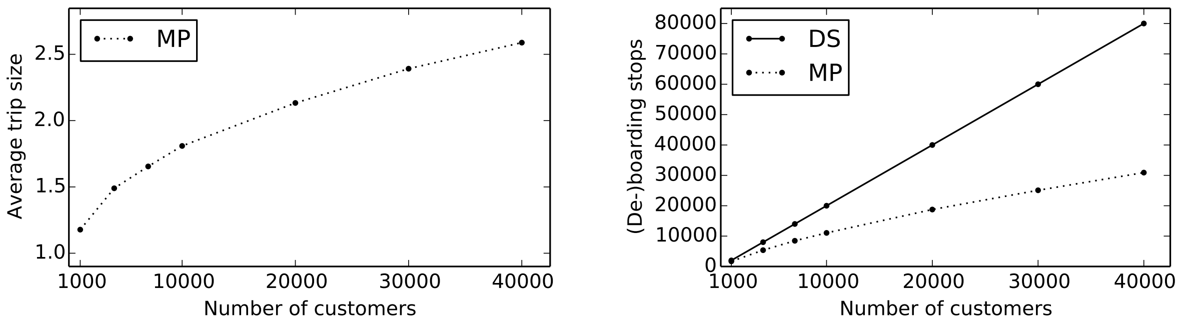


Figure 8.5: Fraction of users per group size after MP candidates selection for different total demand scenarios.

The group size shown in Figure 8.5 naturally correlates with the average number of customers per pick-up (Figure 8.6a). Generally, it can be stated that, with an increasing total number of customers, the portion of bigger groups increases, as more people with similar itineraries and time schedules can be grouped together.



(a) Average customers per pick-up *(b) Amount of boarding and alighting stops*

Figure 8.6: Comparison between meeting point and doorstep simulation concerning trip size. DS = Doorstep, MP = Meeting Points.

On the other hand, in the study area of Braunschweig, 4 000 passengers are already sufficient for a majority of people to share their rides. This also inherently reduces the number of necessary boarding and deboarding service stops for the vehicles (Figure 8.6b), since they have to stop only once for a group, instead of stopping for every single customer. Naturally, the savings are higher when the demand is dense. With 5 000 customers, the number of necessary stops is reduced by 33 %, while it is halved at about 15 000 customers.

The following statistics about vehicle usage and trip times are the results of the vehicle routing optimization phase. All experiments have been conducted based on four demand instances: 1 000, 4 000, 7 000 and 10 000 (only morning commute to work, see Section 5.4). Table 8.3 lists the results for both the doorstep case (DS) and the meeting point case (MP) about vehicle trip statistics and passenger waiting, detour and walking times, so that they can be compared. Obviously, the passenger walking times are zero for all DS case scenarios.

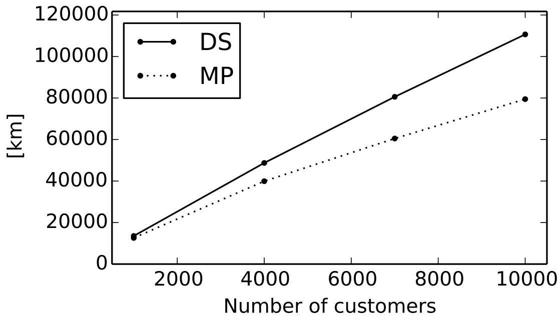
Table 8.3: Vehicle and passenger statistics.

No. of riders	1 000		4 000		7 000		10 000	
Case	DS	MP	DS	MP	DS	MP	DS	MP
Vehicle mileage [km]	13 536	12 594	48 723	39 915	80 570	60 485	110 655	79 470
Vehicle hours [h]	379	344	1 392	1 065	2 324	1 634	3 208	2 133
No. of vehicles used	198	184	794	602	1 296	931	1 871	1 248
Dead mileage [km]	6 603	6 357	24 032	20 668	40 016	31 222	54 806	41 118
Idle hours [h]	32	36	91	91	136	134	160	161
Passenger average walk time [min]	0	5.87	0	7.74	0	8.24	0	8.51
Group average pick-up waiting time [min]	0	2.05	0	1.73	0	1.54	0	1.39
Passenger average pick-up waiting time [min]	2.50	3.49	2.64	4.83	2.70	5.05	2.77	5.32
Passenger average detour time [min]	5.75	11.26	7.09	14.51	7.62	15.32	7.95	15.75

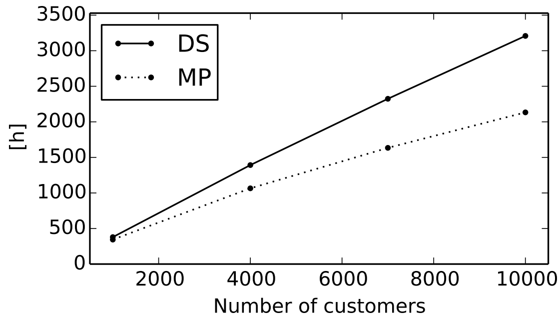
The total vehicle mileage is larger in the doorstep case than in the meeting point case for all four instances, with an increasing gap when the demand is higher (Figure 8.7a). The same applies for the total vehicle hours (Figure 8.7b), which naturally has a very similar trend. The savings in the 10 000 customers case are approximately 30 %. Also, the number of vehicles in use can be reduced (Figure 8.7c), offering a huge savings potential for the operator. In addition, the *dead mileage* (vehicle mileage without a passenger) is reduced in the MP case (Figure 8.7d). The total idle hours (time of a vehicle without a passenger and movement) however shows no clear difference or trend between the DS and the MP case.

The last four rows of Table 8.3 focus on the impact on the passengers. The average walking time (including the walking time from the alighting point to the destination) is obviously zero for the DS case, and increases in the MP case with more total customers (Figure 8.7e). This effect can be explained by the fact that with more customers more common meeting points with fellow travellers are selected, which are further away than single passenger meeting points.
The waiting times are separated into waiting times for passengers (Figure 8.7f) and waiting times for the group (Figure 8.7g). The group waiting time describes the time that the whole group waits for the vehicle, while the passenger waiting time represents the full waiting time at the pick-up point (MP or DS) from arriving until boarding. The passenger MP waiting time therefore includes the group pick-up waiting time, and in the DS case the group waiting time is obviously zero. When comparing the waiting time between groups and passengers it can be noticed that the group waiting times are much lower, indicating an overall better service quality. The higher total passenger waiting time in the MP case can be explained by the waiting time for fellow travellers of the group.

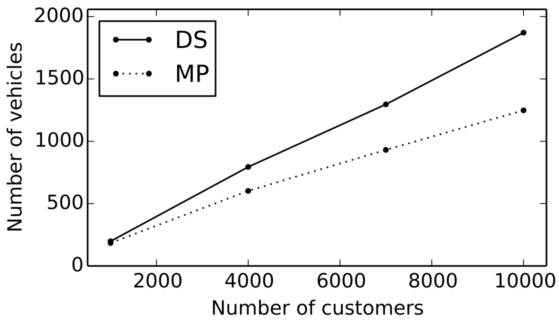
Finally, the average detour time represents the difference between a fictive direct travel time from origin to destination and the actual travel time, including walking, waiting and vehicle detour times (Figure 8.7h). The detour times for the DS case are composed of waiting and extensive vehicle cruising to pick up and drop off passengers en route. Naturally, the detour times are higher for the MP case because of the walking time. In contrast, the additional time due to vehicle cruising is comparatively low.



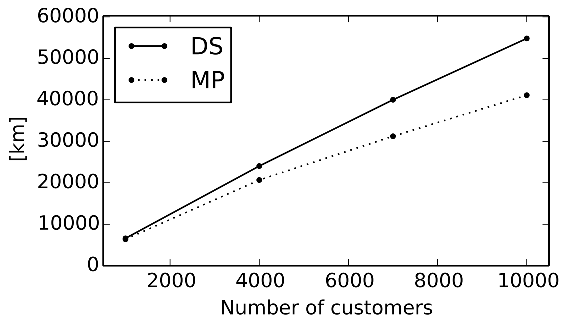
(a) Total vehicle mileage



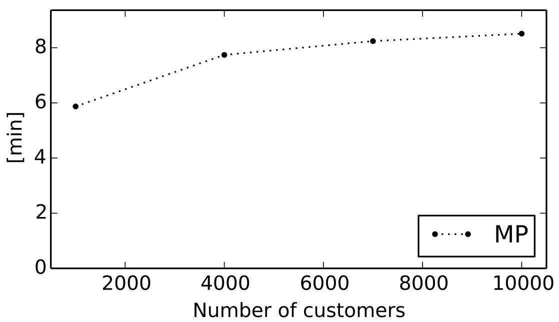
(b) Total vehicle hours



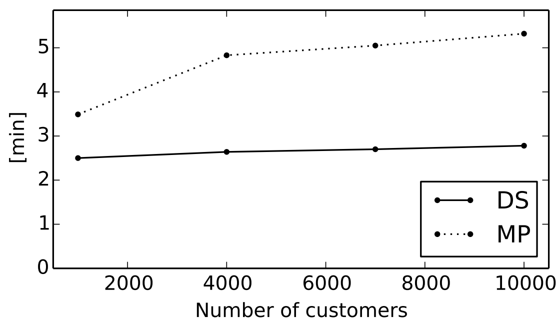
(c) Number of used vehicles



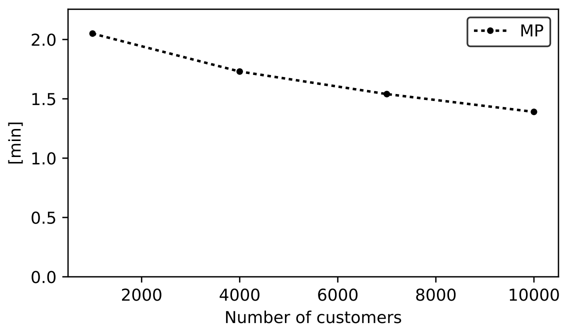
(d) Vehicle dead kilometres



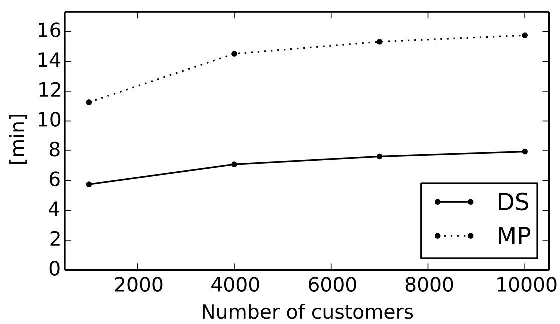
(e) Passenger average walking time



(f) Passenger average pickup wait time



(g) Group average waiting time



(h) Passenger average detour time

Figure 8.7: Comparison between doorstep (DS) and meeting point (MP) based simulation results.

8.4 Discussion

In general, the results show particular benefits for operators of SDRT systems when switching to a meeting point based mode. The overall operational costs can be significantly reduced, by using fewer vehicles, driving fewer kilometres and reducing boarding and de-boarding stops. The higher the demand, the more appealing are the benefits.

On the other hand, these benefits for the operator come at the cost of a certain inconvenience for the customers, as they have to spend additional time walking and waiting for fellow travellers. However, the reductions in operational costs may translate to reduced travel pricing, so that there can be a monetary benefit for the customers. The price can be expected to decrease if more customers participate. In cases with a relatively high demand, the waiting and walking times do not increase much as the demand increases. In contrast, the operator's saving is consistently improving. Furthermore, the point-to-point transfer in the MP case seems to be faster than in the DS case, which is probably caused by the reduced total service time and less cruising around to reach all customers.

The simulation experiment by Häll et al. (2008) states that, in general, the DS case offers a better service for customers, which can be confirmed by the results of this experiment regarding the passenger time statistics. However, according to their results, they found no major differences in the results between the MP and the DS case and state that a door-to-door service can be offered without any noticeable loss in efficiency. This contradicts the findings of this experiment, where an improved operational efficiency could be demonstrated.

Although the simulation experiment by Stiglic et al. (2015) does not include a routing phase, and focuses on ride-sharing, which allows only one boarding and alighting per vehicle, it is nevertheless interesting to draw a comparison. They conclude that the introduction of meeting points can improve a number of metrics, such as mileage savings and an increase in the number of matched participants, and that the average trip time for matched riders increases by approximately 12% due to walking to a meeting point. The difference to the results in this chapter are quite large, since the travel time increase is up to 44%. However, the difference can be explained by the relatively low total travel times (17.5 minutes for the DS case, 25.3 minutes for the MP case on average). Since Braunschweig is a small city and congestion is not modelled, all nodes of the city network can be reached within a relatively short time. Hence, meeting and waiting times have a high impact on the average total trip time.

In essence, it could be demonstrated that the usage of meeting points in SDRT services can be beneficial for the operator in terms of vehicle usage, operation hours and mileage at the cost of increased walking and waiting times for the passengers. Future prospects in this area include the design of an improved efficient solver which can be used for real-time operation, since the current version of the route planning part is not capable of delivering solutions within a short time.

9 Conclusion

In this thesis, the topic of using meeting points for a ride-sharing or demand-responsive transportation system is examined and discussed through user studies and computational experiments solving optimization problems. The novelty of this work is its findings concerning human preferences about meeting points, benefits and downsides of using meeting points, and new algorithms to cope with them in ride-sharing systems. A main focus and the major difference from other scientific works is the usage of real-world meeting points and city structures at the same time, bringing the topic closer to reality. In essence, four main aspects have been examined, which is aligned with the scientific knowledge gaps discovered in Section 3.2:

- Determination of customer preferences (Chapter 4)
- Meeting points for intra-urban ride-sharing (Chapter 6)
- Reaching meeting points by public transport for long-distance ride-sharing (Chapter 7)
- Meeting points for intra-urban demand-responsive transportation (Chapter 8)

As a general conclusion, it can be stated that meeting points offer a wide range of possibilities for easing the intra- and inter-urban transportation of customers, and making boarding and alighting procedures easier and safer. All experiments show that the usage of meeting points offers benefits, in particular for the drivers (or service providers), since driver detour mileage or time can be reduced. Also, on average more customers can be accommodated when using meeting points. In addition, the grouping of multiple passengers at meeting points allows further savings on driving time and the number of necessary vehicles. On the downside, this implies longer travel times for passengers, due to the walking distance to and from the meeting and divergence points and potentially also waiting times for fellow travellers. However, the additional walking activity can also be seen as a contribution to human health due to the active movement (Giles-Corti et al. 2016).

The research questions from Section 1.3 can at this point be answered briefly:

Which properties and facilities are important for drivers and passengers concerning a meeting point, and how can personal preferences about meeting points be incorporated into the selection?

In the user surveys from Chapter 4, participants stated that parking places including supermarket parking, train stations and petrol stations are generally most suited as a meeting point, together with a pick-up at the doorstep (Figure 4.5). Points of interest (POI) and bus or tram stops are rated only intermediate, while street junctions seem to be the worst choice. While drivers clearly prefer parking places and petrol stations, passengers prefer a pick-up at the doorstep or, slightly less popular, at public transport stops and parking places (Figure 4.6). Regarding facilities, the most important aspect is the unambiguousness of the location, followed by security and the parking price. In winter time, illumination, shelter and warmth also play a significant role (Figure 4.7).

These findings have subsequently been used in a ride-sharing model that incorporates meeting point quality, in order to answer the second part of the research question (Section 6.3.3). The results of this convenience-based matching show that, if meeting point properties and facilities are included in the matching, more parking places, turning areas and petrol stations are used, and far fewer street intersections (Figure 6.8a). This distribution is similar to the revealed types from the user survey (Figure 4.12a), indicating a more realistic outcome than from the non-convenience matching. Also, the number of meeting points with seating, shelter and a good parking availability increases (Figure 6.8). However, this comes at the cost of, on average, longer walking distances for the passengers.

What is the impact (benefits and downsides) of using real-world meeting points for intra-urban ride-sharing?

Compared to a conventional door-to-door service, more passengers can be matched due to the greater flexibility and the possibility to accommodate more than one passenger in a shared ride (Figure 6.5a). Also, the driving times are on average lower since the meeting points can be chosen such that they are more or less on the direct way of the drivers. This effect is more significant when the demand is high. Again, this comes at the cost of high travel times for the passengers, since they have to walk (Figure 6.6). Besides a high demand, it is essential for a well working intra-urban ride-sharing that enough meeting points are available (if pick-ups at the doorsteps are prohibited). If not enough meeting points are available, the matching rate is clearly reduced, and walking distances and detour times of the matched customers are increased (Figure 6.9).

How can appropriate meeting points be automatically recommended to ride-sharing customers, particularly for long-distance trips?

In long-distance ride-sharing, the drivers are often just passing the city where a passenger wants to join the ride. In this case, it can be advantageous to negotiate a meeting point close to the drivers path, e.g. near a motorway exit, and the passenger uses public transport to reach this point. The problem is that there are many meeting points to consider, and a computation of all public transport connections is not possible in real-time. Hence, a method was developed to reasonably filter the meeting point candidates and precompute and store public transport connections, so that the travel times can be determined by a simple lookup (Chapter 7). The presented solution enables a real-time operation, offering drivers and passengers a range of appropriate meeting points, from which a final meeting point is chosen based on a voting procedure. In a simulation study, it could be shown that in fact most of the suggested meeting points are located very close to motorway exits (Figure 7.8).

What is the impact (benefits and downsides) of using real-world meeting points for intra-urban demand-responsive transportation systems?

A three-step algorithm was developed and consequently used for a simulation to answer this question (Chapter 8). It turns out that the demand size has a strong influence on the metrics of the ride-sharing system. With a high demand, more passengers can be grouped together to join a common ride (Figure 8.5). This helps to significantly reduce the amount of necessary boarding and alighting stops: with 15 000 customers, the number of stops can already be halved (Figure 8.6b). The grouping and common meeting has further advantages for the service operators: the number of vehicles, the amount of dead kilometers, vehicle operating hours as well as vehicle mileage can be reduced, each up to 30 % for 10 000 customers

(Figure 8.7). On the downside, the passengers have to travel 6 - 8 minutes more due to waiting and walking times (Figure 8.7).

How can meeting points be used by municipal traffic management?

Traffic management authorities can influence the city traffic by meeting points in various ways. On the one hand, by limiting the set of available meeting points. Whole areas can theoretically be excluded, for example the city centre. Particularly for long-distance ride-sharing, this offers the chance to control how far drivers should enter into the city. Given a maximum time or distance threshold, only the meeting points that fulfil the requirements are then proposed, so that passengers are guided to take the public transport system to reach meeting points in more remote areas. The simulation with long-distance ride-sharing trips (Chapter 7) shows that this method is very effective (Figure 7.9). However, the driver detour threshold should be set to at least three minutes, to be able to compute valid meeting point recommendations for at least 80 % of the requests (Figure 7.10). According to the map-based user survey, many appropriate meeting points can already be reached within a few minutes detour (Figure 4.15b).

On the other hand, traffic management can also support meeting points in the city and promote ride-sharing or even offer a shared demand-responsive transportation system. Since these systems operate most efficiently when the demand is high (see, for instance, Section 8.3), it is most beneficial if many customers switch from their private car to such services. Well-equipped meeting points can further enhance the user experience.

Consideration of algorithm complexity

The complexity of the proposed algorithms varies. The algorithm to generate feasible trips for intra-urban ride-sharing (Chapter 6) has a polynomial time complexity of $\mathcal{O}(kn^m)$, with k as the number of drivers, n as the number of passengers and m as the maximum vehicle capacity. The running time is therefore highly dependent on the number of available seats in the vehicles. In practice, the running time is also very sensitive to time flexibility, since with greater flexibility, more options have to be checked. It is thus best applicable to vehicles with few spare seats and customers with a relatively tight schedule, and not suited to vehicles with a higher capacity, such as minibuses. Here, the algorithm proposed in Chapter 8 is more appropriate. Due to the clustering phase, the group size is always limited, so that the Dynamic Programming algorithm finishes in a reasonable time, although it has an exponential complexity ($\mathcal{O}(2^n)$).

The method for determining meeting points for long-distance ride-sharing (Chapter 7) focuses inherently on real-time applications. The operational phase consists only of value lookups, hence the result can be obtained in linear time of $\mathcal{O}(m)$, with m as the number of considered meeting points, which are processed iteratively. The precomputation phase is more interesting regarding scalability, since the processing times are much higher. Here, the complexity is polynomial ($\mathcal{O}(nm)$), with n as the number of PTE (Public Transport Entry) nodes and m as the number of considered meeting points.

Further research

Further research considering meeting points for shared rides is particularly necessary within the scope of validation. This thesis provides mainly theoretical approaches and algorithms, and also the simulation studies are based on artificial demand. To validate the results, a comparison with real-world data is desirable. For example, user- and trip-related data from ride-sharing operators such as BlaBlaCar could be analyzed to find patterns of real-world meeting point usage. The ideal case would be to implement the proposed algorithms for public usage and see how the usage of meeting points actually changes, so that results are not only available from theoretical considerations.

Another important enhancement of the methods in this thesis is the capability of dealing with requests at short notice (*dynamic ride-sharing*). So far, the simulations are mainly based on a static demand scenario, meaning that all requests are known in advance. In reality, however, requests can occur spontaneously. Especially for demand-responsive transportation services, short announcement times make a service more appealing to customers. Incorporating dynamic requests would also bring the simulations closer to the reality. In further research projects, *rolling horizon* strategies could, for example, be used to cope with spontaneous requests and the uncertainty of future demand (Agatz et al. 2010, 2011; Kleiner et al. 2011; Najmi et al. 2017).

Finally, it is desirable that the proposed methods and simulations are also applied to other datasets from different cities, with varying size and street structures. A drawback of the presented simulations is that they are all based on the Braunschweig scenario, which represents a mid-sized European city. It is not entirely clear how the results change when the investigation city is very large, very small, or has a different city structure. A comprehensive study based on various city models from different continents could help to solve this issue.

Bibliography

Agatz et al. 2010
AGATZ, Niels ; ERERA, Alan ; SAVELSBERGH, Martin ; WANG, Xing: *Sustainable passenger transportation: Dynamic ride-sharing*. `https://ssrn.com/abstract=1568676`. Version: 2010

Agatz et al. 2012
AGATZ, Niels ; ERERA, Alan ; SAVELSBERGH, Martin ; WANG, Xing: Optimization for dynamic ride-sharing: A review. In: *European Journal of Operational Research* 223 (2012), Nr. 2, 295 - 303. `https://doi.org/10.1016/j.ejor.2012.05.028`

Agatz et al. 2011
AGATZ, Niels A. ; ERERA, Alan L. ; SAVELSBERGH, Martin W. ; WANG, Xing: Dynamic ride-sharing: A simulation study in metro Atlanta. In: *Transportation Research Part B: Methodological* 45 (2011), Nr. 9, 1450 - 1464. `https://doi.org/10.1016/j.trb.2011.05.017`. – Select Papers from the 19th {ISTTT}

Aissat & Oulamara 2015a
AISSAT, K ; OULAMARA, A: Meeting Locations in Real-Time Ridesharing Problem: A Buckets Approach. In: *Operations Research and Enterprise Systems*. Springer, 2015, S. 71–92

Aissat & Oulamara 2014a
AISSAT, Kamel ; OULAMARA, Ammar: Dynamic ridesharing with intermediate locations. In: *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2014 IEEE Symposium on* IEEE, 2014, S. 36–42

Aissat & Oulamara 2014b
AISSAT, Kamel ; OULAMARA, Ammar: A priori approach of real-time ridesharing problem with intermediate meeting locations. In: *Journal of Artificial Intelligence and Soft Computing Research* 4 (2014), Nr. 4, S. 287–299

Aissat & Oulamara 2015b
AISSAT, Kamel ; OULAMARA, Ammar: The Round-Trip Ridesharing Problem with Relay Stations. In: *International Conference on Computational Logistics* Springer, 2015, S. 16–30

Aïvodji et al. 2016
AÏVODJI, Ulrich M. ; GAMBS, Sébastien ; HUGUET, Marie-José ; KILLIJIAN, Marc-Olivier: Meeting points in ridesharing: A privacy-preserving approach. In: *Transportation Research Part C: Emerging Technologies* 72 (2016), S. 239–253

Balardino & Santos 2016
BALARDINO, Allan F. ; SANTOS, André G: Heuristic and Exact Approach for the Close Enough Ridematching Problem. In: *Hybrid Intelligent Systems*. Springer, 2016, S. 281–293

Baldacci et al. 2004
BALDACCI, Roberto ; MANIEZZO, Vittorio ; MINGOZZI, Aristide: An exact method for the car pooling problem based on lagrangean column generation. In: *Operations Research* 52 (2004), Nr. 3, S. 422–439

Baldacci et al. 2012
BALDACCI, Roberto ; MINGOZZI, Aristide ; ROBERTI, Roberto: Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. In: *European Journal of Operational Research* 218 (2012), Nr. 1, S. 1–6

Bell & McMullen 2004
BELL, John E. ; MCMULLEN, Patrick R.: Ant colony optimization techniques for the vehicle routing problem. In: *Advanced engineering informatics* 18 (2004), Nr. 1, S. 41–48

Bellman 2013
BELLMAN, Richard: *Dynamic programming.* Courier Corporation, 2013

Berbeglia et al. 2007
BERBEGLIA, Gerardo ; CORDEAU, Jean-François ; GRIBKOVSKAIA, Irina ; LAPORTE, Gilbert: Static pickup and delivery problems: A classification scheme and survey. In: *TOP* 15 (2007), Nr. 1, 1–31. http://dx.doi.org/10.1007/s11750-007-0009-0

Berbeglia et al. 2012
BERBEGLIA, Gerardo ; CORDEAU, Jean-François ; LAPORTE, Gilbert: A Hybrid Tabu Search and Constraint Programming Algorithm for the Dynamic Dial-a-Ride Problem. In: *INFORMS Journal on Computing* 24 (2012), Nr. 3, 343-355. http://dx.doi.org/10.1287/ijoc.1110.0454

Bicocchi et al. 2015
BICOCCHI, N. ; MAMEI, M. ; SASSI, A. ; ZAMBONELLI, F.: Opportunistic Ride Sharing via Whereabouts Analysis. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, S. 875–881

Bodin & Sexton 1986
BODIN, Lawrence D. ; SEXTON, T: The multi-vehicle subscriber dial-a-ride problem. In: *TIMS studies in Management Science* 26 (1986), S. 73–86

Borndörfer et al. 1999
BORNDÖRFER, R. ; GRÖTSCHEL, M. ; KLOSTERMEIER, F. ; KÜTTNER, C. ; WILSON, Nigel H. M. (Hrsg.): *Telebus Berlin: Vehicle Scheduling in a Dial-a-Ride System.* http://dx.doi.org/10.1007/978-3-642-85970-0_19. Version: 1999

Bradley et al. 1977
BRADLEY, Stephen P. ; HAX, Arnoldo C. ; MAGNATI, Thomas L.: *Applied Mathematical Programming.* Addison-Wesley, 1977

Braekers et al. 2014
BRAEKERS, Kris ; CARIS, An ; JANSSENS, Gerrit K.: Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. In: *Transportation Research Part B: Methodological* 67 (2014), 166 - 186. https://doi.org/10.1016/j.trb.2014.05.007

Brimberg et al. 2000
BRIMBERG, Jack ; HANSEN, Pierre ; MLADENOVIC, Nenad ; TAILLARD, Eric D.: Improvements and Comparison of Heuristics for Solving the Uncapacitated Multisource Weber Problem. In: *Operations Research* 48 (2000), Nr. 3, 444-460. https://doi.org/10.1287/opre.48.3.444.12431

Burris & Winn 2006
BURRIS, Mark W. ; WINN, Justin R.: Slugging in Houston - casual carpool passenger characteristics. In: *Journal of Public Transportation* 9 (2006), Nr. 5, S. 2

Calvo et al. 2004
CALVO, Roberto W. ; LUIGI, Fabio de ; HAASTRUP, Palle ; MANIEZZO, Vittorio: A distributed geographic information system for the daily car pooling problem. In: *Computers & Operations Research* 31 (2004), Nr. 13, 2263 - 2278. https://doi.org/10.1016/S0305-0548(03)00186-2

Caprara et al. 2000
CAPRARA, Alberto ; TOTH, Paolo ; FISCHETTI, Matteo: Algorithms for the Set Covering Problem. In: *Annals of Operations Research* 98 (2000), Nr. 1, 353–371. https://doi.org/10.1023/A:1019225027893

Caulfield 2009
CAULFIELD, Brian: Estimating the environmental benefits of ride-sharing: A case study of Dublin. In: *Transportation Research Part D: Transport and Environment* 14 (2009), Nr. 7, 527 - 531. https://doi.org/10.1016/j.trd.2009.07.008

Chan & Shaheen 2012
Chan, Nelson D. ; Shaheen, Susan A.: Ridesharing in North America: Past, Present, and Future. In: *Transport Reviews* 32 (2012), Nr. 1, 93-112. http://dx.doi.org/10.1080/01441647.2011.621557

Chaube et al. 2010
Chaube, V. ; Kavanaugh, A. L. ; Perez-Quinones, M. A.: Leveraging Social Networks to Embed Trust in Rideshare Programs. In: *2010 43rd Hawaii International Conference on System Sciences*, 2010, S. 1–8

Chen et al. 2016
Chen, Wenyi ; Mes, MRK ; Schutten, JMJ ; Quint, Job: A ride-sharing problem with meeting points and return restrictions. In: *Beta Working Paper series* 516 (2016). http://doc.utwente.nl/101813/1/wp_516.pdf

Chien & Kuchipudi 2003
Chien, Steven I-Jy ; Kuchipudi, Chandra M.: Dynamic Travel Time Prediction with Real-Time and Historic Data. In: *Journal of Transportation Engineering* 129 (2003), Nr. 6, S. 608–616. http://dx.doi.org/10.1061/(ASCE)0733-947X(2003)129:6(608). – DOI 10.1061/(ASCE)0733–947X(2003)129:6(608)

Christofides et al. 1981
Christofides, N. ; Mingozzi, A. ; Toth, P.: Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. In: *Mathematical Programming* 20 (1981), Nr. 1, 255–282. http://dx.doi.org/10.1007/BF01589353

Chvatal 1979
Chvatal, Vasek: A greedy heuristic for the set-covering problem. In: *Mathematics of operations research* 4 (1979), Nr. 3, S. 233–235

Clarke & Wright 1964
Clarke, Geoff ; Wright, John W.: Scheduling of vehicles from a central depot to a number of delivery points. In: *Operations research* 12 (1964), Nr. 4, S. 568–581

Coppersmith et al. 2011
Coppersmith, Don ; Nowicki, Tomasz ; Paleologo, Giuseppe ; Tresser, Charles ; Wu, Chai W.: The optimality of the online greedy algorithm in carpool and chairman assignment problems. In: *ACM Transactions on Algorithms (TALG)* 7 (2011), Nr. 3, S. 37

Cordeau et al. 2002
Chapter VRP with Time Windows. In: Cordeau, J.-F. ; Desaulniers, G. ; Desrosiers, J. ; Solomon, M. ; Soumis, F.: *The vehicle routing problem.* SIAM, 2002, S. 157 – 194

Cordeau 2006
Cordeau, Jean-François: A branch-and-cut algorithm for the dial-a-ride problem. In: *Operations Research* 54 (2006), Nr. 3, S. 573–586

Cordeau et al. 1997
Cordeau, Jean-François ; Gendreau, Michel ; Laporte, Gilbert: A tabu search heuristic for periodic and multi-depot vehicle routing problems. In: *Networks* 30 (1997), Nr. 2, S. 105–119. – ISSN 1097–0037

Cordeau & Laporte 2003a
Cordeau, Jean-François ; Laporte, Gilbert: The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. In: *4OR: A Quarterly Journal of Operations Research* 1 (2003), Nr. 2, S. 89–101

Cordeau & Laporte 2003b
Cordeau, Jean-François ; Laporte, Gilbert: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. In: *Transportation Research Part B: Methodological* 37 (2003), Nr. 6, S. 579–594

Cordeau & Laporte 2007
Cordeau, Jean-François ; Laporte, Gilbert: The dial-a-ride problem: models and algorithms. In: *Annals of operations research* 153 (2007), Nr. 1, S. 29

Coslovich et al. 2006
Coslovich, Luca ; Pesenti, Raffaele ; Ukovich, Walter: A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. In: *European Journal of Operational Research* 175 (2006), Nr. 3, 1605 - 1615. https://doi.org/10.1016/j.ejor.2005.02.038

Czech & Czarnas 2002
Czech, Zbigniew J. ; Czarnas, Piotr: Parallel simulated annealing for the vehicle routing problem with time windows. In: *Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on* IEEE, 2002, S. 376–383

Czioska et al. 2017
Czioska, P. ; Kutadinata, R. ; Trifunović, A. ; Winter, S. ; Sester, M. ; Friedrich, B.: Real-world Meeting Points for Shared Demand-Responsive Transportation Systems. In: *ArXiv e-prints* (2017). https://arxiv.org/abs/1709.08488

Czioska et al. 2017
Czioska, Paul ; Mattfeld, Dirk C. ; Sester, Monika: GIS-based identification and assessment of suitable meeting point locations for ride-sharing. In: *Transportation Research Procedia* 22 (2017), 314 - 324. https://doi.org/10.1016/j.trpro.2017.03.038

Czioska et al. 2018
Czioska, Paul ; Trifunovic, Aleksandar ; Dennisen, Sophie ; Sester, Monika: Location- and time-dependent meeting point recommendations for shared interurban rides. In: *Journal of Location Based Services* 0 (2018), Nr. 0, 1-23. http://dx.doi.org/10.1080/17489725.2017.1421779. – DOI 10.1080/17489725.2017.1421779

Dantzig & Ramser 1959
Dantzig, George B. ; Ramser, John H.: The truck dispatching problem. In: *Management science* 6 (1959), Nr. 1, S. 80–91

Demiryurek et al. 2010
Demiryurek, Ugur ; Banaei-Kashani, Farnoush ; Shahabi, Cyrus: A case for time-dependent shortest path computation in spatial networks. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* ACM, 2010, S. 474–477

Dennisen & Müller 2015
Dennisen, Sophie L. ; Müller, Jörg P: Agent-Based Voting Architecture for Traffic Applications. In: *German Conference on Multiagent System Technologies* Springer, 2015, S. 200–217

Desaulniers et al. 2002
Chapter VRP with Pickup and Delivery. In: Desaulniers, Guy ; Desrosiers, Jacques ; Erdmann, Andreas ; Solomon, Marius ; Soumis, Francois: *The vehicle routing problem.* SIAM, 2002, S. 225 – 242

Desrochers et al. 1992
Desrochers, Martin ; Desrosiers, Jacques ; Solomon, Marius: A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. In: *Operations Research* 40 (1992), Nr. 2, 342-354. http://dx.doi.org/10.1287/opre.40.2.342

Desrosiers et al. 1986
Desrosiers, Jacques ; Dumas, Yvan ; Soumis, François: A Dynamic Programming Solution of the Large-Scale Single-Vehicle Dial-A-Ride Problem with Time Windows. In: *American Journal of Mathematical and Management Sciences* 6 (1986), Nr. 3-4, 301-325. http://dx.doi.org/10.1080/01966324.1986.10737198

Diana & Dessouky 2004
Diana, Marco ; Dessouky, Maged M.: A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. In: *Transportation Research Part B: Methodological* 38 (2004), Nr. 6, 539 - 557. https://doi.org/10.1016/j.trb.2003.07.001

Eksioglu et al. 2009
Eksioglu, Burak ; Vural, Arif V. ; Reisman, Arnold: The vehicle routing problem: A taxonomic review. In: *Computers & Industrial Engineering* 57 (2009), Nr. 4, 1472 - 1483. `https://doi.org/10.1016/j.cie.2009.05.009`

Ester et al. 1996
Ester, Martin ; Kriegel, Hans-Peter ; Sander, Jörg ; Xu, Xiaowei et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd* Bd. 96, 1996, S. 226–231

Ewing et al. 2014
Ewing, Reid ; Meakins, Gail ; Hamidi, Shima ; Nelson, Arthur C.: Relationship between urban sprawl and physical activity, obesity, and morbidity–Update and refinement. In: *Health & place* 26 (2014), S. 118–126

Ewing et al. 2003
Ewing, Reid ; Schmid, Tom ; Killingsworth, Richard ; Zlot, Amy ; Raudenbush, Stephen: Relationship between urban sprawl and physical activity, obesity, and morbidity. In: *American journal of health promotion* 18 (2003), Nr. 1, S. 47–57

Fagin & Williams 1983
Fagin, Ronald ; Williams, John H.: A fair carpool scheduling algorithm. In: *IBM Journal of Research and development* 27 (1983), Nr. 2, S. 133–139

Fellows & Pitfield 2000
Fellows, N.T. ; Pitfield, D.E.: An economic and operational evaluation of urban car-sharing. In: *Transportation Research Part D: Transport and Environment* 5 (2000), Nr. 1, 1 - 10. `http://www.sciencedirect.com/science/article/pii/S1361920999000164`

Feo & Resende 1995
Feo, Thomas A. ; Resende, Mauricio G. C.: Greedy Randomized Adaptive Search Procedures. In: *Journal of Global Optimization* 6 (1995), Nr. 2, 109–133. `http://dx.doi.org/10.1007/BF01096763`

Ferguson 1997
Ferguson, Erik: The rise and fall of the American carpool: 1970–1990. In: *Transportation* 24 (1997), Nr. 4, 349–376. `http://dx.doi.org/10.1023/A:1004928012320`

Follmer et al. 2010
Follmer, Robert ; Gruschwitz, Dana ; Jesske, Birgit ; Quandt, Sylvia ; Lenz, Barbara ; Nobis, Claudia ; Köhler, Katja ; Mehlin, M: Mobilität in Deutschland 2008. In: *Methodenbericht. Bonn und Berlin* (2010). `www,mobilitaet-in-deutschland.de`

Furuhata et al. 2013
Furuhata, Masabumi ; Dessouky, Maged ; Ordonez, Fernando ; Brunet, Marc-Etienne ; Wang, Xiaoqing ; Koenig, Sven: Ridesharing: The state-of-the-art and future directions. In: *Transportation Research Part B: Methodological* 57 (2013), 28 - 46. `https://doi.org/10.1016/j.trb.2013.08.012`

Geisberger et al. 2009
Geisberger, R. ; Luxen, D. ; Neubauer, S. ; Sanders, P. ; Volker, L.: Fast Detour Computation for Ride Sharing. In: *ArXiv e-prints* (2009)

Gendreau et al. 1994
Gendreau, Michel ; Hertz, Alain ; Laporte, Gilbert: A Tabu Search Heuristic for the Vehicle Routing Problem. In: *Management Science* 40 (1994), Nr. 10, 1276-1290. `http://dx.doi.org/10.1287/mnsc.40.10.1276`

Gendreau & Potvin 2010
Gendreau, Michel ; Potvin, Jean-Yves: *Handbook of metaheuristics*. Bd. 2. Springer, 2010

Giles-Corti et al. 2016
GILES-CORTI, Billie ; VERNEZ-MOUDON, Anne ; REIS, Rodrigo ; TURRELL, Gavin ; DANNENBERG, Andrew L. ; BADLAND, Hannah ; FOSTER, Sarah ; LOWE, Melanie ; SALLIS, James F. ; STEVENSON, Mark et al.: City planning and population health: a global challenge. In: *The Lancet* (2016). http://dx.doi.org/10.1016/S0140-6736(16)30066-6

Goel et al. 2017
GOEL, P. ; KULIK, L. ; RAMAMOHANARAO, K.: Optimal Pick up Point Selection for Effective Ride Sharing. In: *IEEE Transactions on Big Data* 3 (2017), Nr. 2, S. 154–168. http://dx.doi.org/10.1109/TBDATA.2016.2599936. – DOI 10.1109/TBDATA.2016.2599936

Goel et al. 2016
GOEL, Preeti ; KULIK, Lars ; RAMAMOHANARAO, Kotagiri: Privacy-Aware Dynamic Ride Sharing. In: *ACM Trans. Spatial Algorithms Syst.* 2 (2016), Nr. 1, 4:1–4:41. http://doi.acm.org/10.1145/2845080

Golden et al. 2008
GOLDEN, Bruce L. ; RAGHAVAN, Subramanian ; WASIL, Edward A.: *The vehicle routing problem: latest advances and new challenges*. Bd. 43. Springer Science & Business Media, 2008

Gruebele 2008
GRUEBELE, P: Interactive system for real time dynamic multi-hop carpooling. In: *Global Transport Knowledge Partnership* (2008)

Häll et al. 2008
HÄLL, Carl H. ; LUNDGREN, Jan T. ; VÄRBRAND, Peter: Evaluation of an integrated public transport system: a simulation approach. In: *Archives of Transport* 20 (2008), Nr. 1-2, S. 29–46

Hansen et al. 2010
HANSEN, Erik G. ; GOMM, Moritz L. ; BULLINGER, Angelika C. ; MOSLEIN, Kathrin M.: A community–based toolkit for designing ride–sharing services: the case of a virtual network of ride access points in Germany. In: *International Journal of Innovation and Sustainable Development* 5 (2010), Nr. 1, S. 80–99

Herbawi & Weber 2012
HERBAWI, Wesam M. ; WEBER, Michael: A Genetic and Insertion Heuristic Algorithm for Solving the Dynamic Ridematching Problem with Time Windows. In: *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA : ACM, 2012 (GECCO '12), 385–392

Hess 2012
HESS, Daniel B.: Walking to the bus: perceived versus actual walking distance to bus stops for older adults. In: *Transportation* 39 (2012), Nr. 2, S. 247–266

Huang et al. 2016
HUANG, Can ; ZHANG, Defu ; SI, Yain-Whar ; LEUNG, Stephen C. H.: Tabu search for the real-world carpooling problem. In: *Journal of Combinatorial Optimization* 32 (2016), Nr. 2, 492–512. http://dx.doi.org/10.1007/s10878-016-0015-y

Ioachim et al. 1995
IOACHIM, Irina ; DESROSIERS, Jaqces ; DUMAS, Yvan ; SOLOMON, Marius M. ; VILLENEU, Daniel: A Request Clustering Algorithm for Door-to-Door Handicapped Transportation. In: *Transportation Science* 29 (1995), Nr. 1, 63-78. http://www.jstor.org/stable/25768670

Jacobson & King 2009
JACOBSON, Sheldon H. ; KING, Douglas M.: Fuel saving and ridesharing in the US: Motivations, limitations, and opportunities. In: *Transportation Research Part D: Transport and Environment* 14 (2009), Nr. 1, 14 - 21. https://doi.org/10.1016/j.trd.2008.10.001

Jaw et al. 1986
JAW, Jang-Jei ; ODONI, Amedeo R. ; PSARAFTIS, Harilaos N. ; WILSON, Nigel H.: A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. In: *Transportation Research Part B: Methodological* 20 (1986), Nr. 3, S. 243–257

Jorgensen et al. 2007
JORGENSEN, Rene M. ; LARSEN, Jesper ; BERGVINSDOTTIR, Kristin B.: Solving the dial-a-ride problem using genetic algorithms. In: *Journal of the operational research society* 58 (2007), Nr. 10, S. 1321–1331

Karp 1972
KARP, Richard M.: Reducibility among combinatorial problems. In: *Complexity of computer computations*. Springer, 1972, S. 85–103

Kashani et al. 2016
KASHANI, Z. N. ; RONALD, N. ; WINTER, S.: Comparing demand responsive and conventional public transport in a low demand context. In: *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2016, S. 1–6

Kelley 1960
KELLEY, James E. Jr: The cutting-plane method for solving convex programs. In: *Journal of the society for Industrial and Applied Mathematics* 8 (1960), Nr. 4, S. 703–712

Kirchler & Calvo 2013
KIRCHLER, Dominik ; CALVO, Roberto W.: A Granular Tabu Search algorithm for the Dial-a-Ride Problem. In: *Transportation Research Part B: Methodological* 56 (2013), 120 - 135. `https://doi.org/10.1016/j.trb.2013.07.014`

Kleiner et al. 2011
KLEINER, Alexander ; NEBEL, Bernhard ; ZIPARO, Vittorio A.: A mechanism for dynamic ride sharing based on parallel auctions. In: *IJCAI* Bd. 11, 2011, S. 266–272

Korte & Vygen 2012
KORTE, Bernhard ; VYGEN, Jens: *Combinatorial Optimization - Theory and Algorithms*. Bd. 21. Fifth edition. Springer, 2012

Kulin & Kuenne 1962
KULIN, Harold W. ; KUENNE, Robert E.: An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics. In: *Journal of Regional Science* 4 (1962), Nr. 2, S. 21–33

Kutadinata et al. 2017
KUTADINATA, R. ; THOMPSON, R. ; WINTER, S.: Passenger-Freight Demand Responsive Transport Services: A Dynamic Optimisation Approach. In: *Transportation Science* (2017). – under review

Laporte et al. 2000
LAPORTE, G. ; GENDREAU, M. ; POTVIN, J-Y. ; SEMET, F.: Classical and modern heuristics for the vehicle routing problem. In: *International Transactions in Operational Research* 7 (2000), Nr. 4-5, 285–300. `http://dx.doi.org/10.1111/j.1475-3995.2000.tb00200.x`

Laporte 1992
LAPORTE, Gilbert: The vehicle routing problem: An overview of exact and approximate algorithms. In: *European Journal of Operational Research* 59 (1992), Nr. 3, 345 - 358. `http://dx.doi.org/10.1016/0377-2217(92)90192-C`

Laporte & Nobert 1987
LAPORTE, Gilbert ; NOBERT, Yves: Exact Algorithms for the Vehicle Routing Problem. In: *North-Holland Mathematics Studies* 132 (1987), 147 - 184. `http://dx.doi.org/10.1016/S0304-0208(08)73235-3`

Lawler & Wood 1966
Lawler, Eugene L. ; Wood, David E.: Branch-and-bound methods: A survey. In: *Operations research* 14 (1966), Nr. 4, S. 699–719

Lin et al. 2016
Lin, J. ; Sasidharan, S. ; Ma, S. ; Wolfson, O.: A Model of Multimodal Ridesharing and Its Analysis. In: *2016 17th IEEE International Conference on Mobile Data Management (MDM)* Bd. 1, 2016, S. 164–173

Lu & Dessouky 2006
Lu, Q. ; Dessouky, M. M.: A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. In: *European Journal of Operational Research* 175 (2006), Nr. 2, S. 672–687

Madsen et al. 1995
Madsen, Oli B. ; Ravn, Hans F. ; Rygaard, Jens M.: A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. In: *Annals of operations Research* 60 (1995), Nr. 1, S. 193–208

Martínez et al. 2014
Martínez, L M. ; Viegas, José M. ; Eiró, Tomás: Formulating a new Express Minibus service design problem as a clustering problem. In: *Transportation Science* 49 (2014), Nr. 1, S. 85–98

Maue et al. 2009
Maue, Jens ; Sanders, Peter ; Matijevic, Domagoj: Goal-directed shortest-path queries using precomputed cluster distances. In: *Journal of Experimental Algorithmics (JEA)* 14 (2009), S. 2

Miller 1991
Miller, Harvey J.: Modelling accessibility using space-time prism concepts within geographical information systems. In: *International Journal of Geographical Information Systems* 5 (1991), Nr. 3, 287-301. `http://dx.doi.org/10.1080/02693799108927856`

Millward et al. 2013
Millward, Hugh ; Spinney, Jamie ; Scott, Darren: Active-transport walking behavior: destinations, durations, distances. In: *Journal of Transport Geography* 28 (2013), S. 101–110

Morency 2007
Morency, Catherine: The ambivalence of ridesharing. In: *Transportation* 34 (2007), Nr. 2, 239–253. `http://dx.doi.org/10.1007/s11116-006-9101-9`

Müller-Hannemann et al. 2007
Müller-Hannemann, Matthias ; Schulz, Frank ; Wagner, Dorothea ; Zaroliagis, Christos: Timetable information: Models and algorithms. In: *Algorithmic Methods for Railway Optimization*. Springer, 2007, S. 67–90

Nagy & Salhi 2005
Nagy, Gabor ; Salhi, Saıd: Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. In: *European Journal of Operational Research* 162 (2005), Nr. 1, 126 - 141. `https://doi.org/10.1016/j.ejor.2002.11.003`

Najmi et al. 2017
Najmi, Ali ; Rey, David ; Rashidi, Taha H.: Novel dynamic formulations for real-time ride-sharing systems. In: *Transportation Research Part E: Logistics and Transportation Review* 108 (2017), 122 - 140. `https://doi.org/10.1016/j.tre.2017.10.009`

Navidi et al. 2016
Navidi, Z. ; Ronald, N. ; Winter, S.: Comparing demand responsive and conventional public transport in a low demand context. In: *Proceedings of IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2016

Nelder & Mead 1965
NELDER, John A. ; MEAD, Roger: A simplex method for function minimization. In: *The computer journal* 7 (1965), Nr. 4, S. 308–313

Nelson et al. 2010
NELSON, John D. ; WRIGHT, Steve ; MASSON, Brian ; AMBROSINO, Giorgio ; NANIOPOULOS, Aristotelis: Recent developments in Flexible Transport Services. In: *Research in Transportation Economics* 29 (2010), Nr. 1, 243 - 248. `https://doi.org/10.1016/j.retrec.2010.07.030`. – Reforming Public Transport throughout the World

Nourinejad & Roorda 2016
NOURINEJAD, Mehdi ; ROORDA, Matthew J.: Agent based model for dynamic ridesharing. In: *Transportation Research Part C: Emerging Technologies* 64 (2016), 117 - 132. `http://dx.doi.org/10.1016/j.trc.2015.07.016`

OECD 2014
OECD: *The cost of air pollution: Health Impacts of Road Transport.* `http://dx.doi.org/10.1787/9789264210448-en`. Version: 2014

Osman 1993
OSMAN, Ibrahim H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. In: *Annals of Operations Research* 41 (1993), Nr. 4, 421–451. `http://dx.doi.org/10.1007/BF02023004`

O'Sullivan et al. 2000
O'SULLIVAN, David ; MORRISON, Alastair ; SHEARER, John: Using desktop GIS for the investigation of accessibility by public transport: an isochrone approach. In: *International Journal of Geographical Information Science* 14 (2000), Nr. 1, S. 85–104

Parragh et al. 2010
PARRAGH, Sophie N. ; DOERNER, Karl F. ; HARTL, Richard F.: Variable neighborhood search for the dial-a-ride problem. In: *Computers & Operations Research* 37 (2010), Nr. 6, 1129 - 1138. `https://doi.org/10.1016/j.cor.2009.10.003`

Parragh et al. 2009
PARRAGH, Sophie N. ; DOERNER, Karl F. ; HARTL, Richard F. ; GANDIBLEUX, Xavier: A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. In: *Networks* 54 (2009), Nr. 4, 227–242. `http://dx.doi.org/10.1002/net.20335`

Parragh & Schmid 2013
PARRAGH, Sophie N. ; SCHMID, Verena: Hybrid column generation and large neighborhood search for the dial-a-ride problem. In: *Computers & Operations Research* 40 (2013), Nr. 1, 490 - 497. `https://doi.org/10.1016/j.cor.2012.08.004`

Pelzer et al. 2015
PELZER, D. ; XIAO, J. ; ZEHE, D. ; LEES, M. H. ; KNOLL, A. C. ; AYDT, H.: A Partition-Based Match Making Algorithm for Dynamic Ridesharing. In: *IEEE Transactions on Intelligent Transportation Systems* 16 (2015), Oct, Nr. 5, S. 2587–2598. `http://dx.doi.org/10.1109/TITS.2015.2413453`. – DOI 10.1109/TITS.2015.2413453

Pillac et al. 2013
PILLAC, Victor ; GENDREAU, Michel ; GUÉRET, Christelle ; MEDAGLIA, Andrés L: A review of dynamic vehicle routing problems. In: *European Journal of Operational Research* 225 (2013), Nr. 1, 1 - 11. `https://doi.org/10.1016/j.ejor.2012.08.015`

Powell 1986
POWELL, Warren B.: A stochastic model of the dynamic vehicle allocation problem. In: *Transportation science* 20 (1986), Nr. 2, S. 117–129

Psaraftis 1980
Psaraftis, Harilaos N.: A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. In: *Transportation Science* 14 (1980), Nr. 2, S. 130–154

Psaraftis 1983
Psaraftis, Harilaos N.: An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. In: *Transportation science* 17 (1983), Nr. 3, S. 351–357

Qian et al. 2017
Qian, Xinwu ; Zhang, Wenbo ; Ukkusuri, Satish V. ; Yang, Chao: Optimal assignment and incentive design in the taxi group ride problem. In: *Transportation Research Part B: Methodological* (2017), -. https://doi.org/10.1016/j.trb.2017.03.001

Rayle et al. 2014
Rayle, Lisa ; Shaheen, Susan ; Chan, Nelson ; Dai, Danielle ; Cervero, Robert: *App-based, on-demand ride services: Comparing taxi and ridesourcing trips and user characteristics in san francisco university of california transportation center (uctc).* 2014

Renaud et al. 1996
Renaud, Jacques ; Laporte, Gilbert ; Boctor, Fayez F.: A tabu search heuristic for the multi-depot vehicle routing problem. In: *Computers & Operations Research* 23 (1996), Nr. 3, 229 - 235. http://dx.doi.org/10.1016/0305-0548(95)00026-P

Ribeiro et al. 2007
Ribeiro, Suzana K. ; Kobayashi, Shigeki ; Beuthe, Michel ; Gasca, Jorge ; Greene, David ; Lee, David S. ; Muromachi, Yasunori ; Newton, Peter J. ; Plotkin, Steven ; Sperling, Daniel et al.: Transportation and its infrastructure. In: *Institute of Transportation Studies* (2007)

Rigby et al. 2016
Rigby, M. ; Winter, S. ; Krüger, A.: A Continuous Representation of Ad Hoc Ridesharing Potential. In: *IEEE Transactions on Intelligent Transportation Systems* 17 (2016), Nr. 10, S. 2832–2842. http://dx.doi.org/10.1109/TITS.2016.2527052. – DOI 10.1109/TITS.2016.2527052

Rigby et al. 2013
Rigby, Michael ; Krüger, Antonio ; Winter, Stephan: An Opportunistic Client User Interface to Support Centralized Ride Share Planning. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 2013, S. 34–43

Rigby & Winter 2015
Rigby, Michael ; Winter, Stephan: Enhancing launch pads for decision-making in intelligent mobility on-demand. In: *Journal of Location Based Services* 9 (2015), Nr. 2, S. 77–92

Rigby & Winter 2016
Rigby, Michael ; Winter, Stephan: Usability of an opportunistic interface concept for ad hoc ride-sharing. In: *International Journal of Cartography* 2 (2016), Nr. 2, 115-147. 10.1080/23729333.2016.1145040

Rothe et al. 2012
Rothe, Jörg ; Baumeister, Dorothea ; Lindner, Claudia ; Rothe, Irene: *Einführung in Computational Social Choice: Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen.* Springer-Verlag, 2012

Rudnicki et al. 2008
Rudnicki, Radoslaw ; Anders, Karl-Heinrich ; Sester, Monika: Rendezvous-Problem in Local Shared-Ride Trip Planning. In: *International archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (2008)

Sanders & Schultes 2005
Sanders, Peter ; Schultes, Dominik: Highway hierarchies hasten exact shortest path queries. In: *European Symposium on Algorithms* Springer, 2005, S. 568–579

Savelsbergh & Sol 1995
Savelsbergh, M. W. P. ; Sol, M.: The General Pickup and Delivery Problem. In: *Transportation Science* 29 (1995), Nr. 1, 17-29. http://dx.doi.org/10.1287/trsc.29.1.17

Schreieck et al. 2016
Schreieck, Maximilian ; Safetli, Hazem ; Siddiqui, Sajjad A. ; Pflügler, Christoph ; Wiesche, Manuel ; Krcmar, Helmut: A Matching Algorithm for Dynamic Ridesharing. In: *Transportation Research Procedia* 19 (2016), 272 - 285. http://dx.doi.org/10.1016/j.trpro.2016.12.087

Solomon 1987
Solomon, Marius M.: Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. In: *Operations Research* 35 (1987), Nr. 2, 254-265. http://dx.doi.org/10.1287/opre.35.2.254

Spielberg & Shapiro 2000
Spielberg, Frank ; Shapiro, Phillip: Mating habits of slugs: Dynamic carpool formation in the I-95/I-395 corridor of Northern Virginia. In: *Transportation Research Record: Journal of the Transportation Research Board* (2000), Nr. 1711, S. 31–38

Stiglic et al. 2015
Stiglic, Mitja ; Agatz, Niels ; Savelsbergh, Martin ; Gradisar, Mirko: The benefits of meeting points in ride-sharing systems. In: *Transportation Research Part B: Methodological* 82 (2015), 36 - 53. http://dx.doi.org/10.1016/j.trb.2015.07.025

Stiglic et al. 2016
Stiglic, Mitja ; Agatz, Niels ; Savelsbergh, Martin ; Gradisar, Mirko: Making dynamic ride-sharing work: The impact of driver and rider flexibility. In: *Transportation Research Part E: Logistics and Transportation Review* 91 (2016), 190 - 207. http://dx.doi.org/10.1016/j.tre.2016.04.010

Taillard et al. 1997
Taillard, Eric ; Badeau, Philippe ; Gendreau, Michel ; Guertin, Francois ; Potvin, Jean-Yves: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. In: *Transportation Science* 31 (1997), Nr. 2, 170-186. http://dx.doi.org/10.1287/trsc.31.2.170

Teal 1987
Teal, Roger F.: Carpooling: Who, how and why. In: *Transportation Research Part A: General* 21 (1987), Nr. 3, 203 - 214. http://dx.doi.org/10.1016/0191-2607(87)90014-8

Tellier 1972
Tellier, Luc-Normand: The Weber problem: solution and interpretation. In: *Geographical Analysis* 4 (1972), Nr. 3, S. 215–233

Teodorovic & Orco 2005
Teodorovic, Dusan ; Orco, Mauro: Bee colony optimization–a cooperative learning approach to complex transportation problems. In: *Advanced OR and AI methods in transportation* (2005), S. 51–60

Teodorovic & Orco 2008
Teodorovic, Dusan ; Orco, Mauro: Mitigating Traffic Congestion: Solving the Ride-Matching Problem by Bee Colony Optimization. In: *Transportation Planning and Technology* 31 (2008), Nr. 2, 135-152. http://dx.doi.org/10.1080/03081060801948027

Toth & Vigo 2002
Toth, Paolo ; Vigo, Daniele: *The vehicle routing problem.* SIAM, 2002

Trivedi 2017
Trivedi, Ronak: *A Street Smart POOL Experience for Manhattan.* newsroom.uber.com/manhattanpool, 2017. – Accessed: 2017-06-07

US Census Bureau 2007
US Census Bureau: *Statistical abstract of the United States.* US Government Printing Office, 2007

U.S. Department of Transportation 2015
U.S. Department of Transportation: National transportation statistics. In: *Bureau of Transportation Statistics, Washington, DC* (2015)

Winter & Nittel 2006
Winter, Stephan ; Nittel, Silvia: Ad hoc shared-ride trip planning by mobile geosensor networks. In: *International Journal of Geographical Information Science* 20 (2006), Nr. 8, S. 899–916

Wong & Bell 2006
Wong, K. I. ; Bell, Michael G. H.: Solution of the Dial-a-Ride Problem with multi-dimensional capacity constraints. In: *International Transactions in Operational Research* 13 (2006), Nr. 3, 195–208. http://dx.doi.org/10.1111/j.1475-3995.2006.00544.x

Xu & Jacobsen 2010
Xu, Zhengdao ; Jacobsen, Hans-Arno: Processing Proximity Relations in Road Networks. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, ACM, 2010 (SIGMOD '10), 243–254

Yan et al. 2011
Yan, Da ; Zhao, Zhou ; Ng, Wilfred: Efficient algorithms for finding optimal meeting point on road networks. In: *Proceedings of the VLDB Endowment* 4 (2011), Nr. 11

Yousaf et al. 2014
Yousaf, Jamal ; Li, Juanzi ; Chen, Lu ; Tang, Jie ; Dai, Xiaowen: Generalized multipath planning model for ride-sharing systems. In: *Frontiers of Computer Science* 8 (2014), Nr. 1, 100–118. https://doi.org/10.1007/s11704-013-3021-6

Yousaf et al. 2012
Yousaf, Jamal ; Li, Juanzi ; Chen, Lu ; Tang, Jie ; Dai, Xiaowen ; Du, John: Ride-Sharing: A Multi Source-Destination Path Planning Approach. In: Thielscher, Michael (Hrsg.) ; Zhang, Dongmo (Hrsg.): *AI 2012: Advances in Artificial Intelligence: 25th Australasian Joint Conference, Sydney, Australia, December 4-7, 2012. Proceedings*, 2012, 815–826