Master's Thesis

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Institute of Computer Science
Chair of Communication Networks

Revised Version (as of June 5, 2024)

Modeling and Evaluation of Sustainable Mobility Concepts for the Campus Mobility

Modellierung und Evaluation nachhaltiger Mobilitätskonzepte für die Campus-Mobilität

Helena Fehler

Prof. Dr.-Ing. Marco Pruckner
Prof. Dr. Marie Schmidt
Supervisors

Submission
February 26, 2024

Deutsche Zusammenfassung

Aktuell fährt immer noch ein großer Teil der Studenten in Deutschland mit dem Auto zu ihrer Universität. Dies trifft vor allem auf Studenten zu, die nicht in der Nähe ihres Universitäts-Campus wohnen. Da solch ein tägliches Pendeln mit dem Auto einen großen Beitrag der gesamten jährlichen CO_2 Emissionen ausmacht, entwickelt diese Arbeit ein Framework für das Untersuchen und Vergleichen von alternativen Mobilitäts-Möglichkeiten durch die Auswertung dieser in einer Simulation. Konkret wird in dieser Arbeit untersucht, wie sich die jeweilige Anwendung von Ridesharing oder Ridepooling für die Universität Würzburg im Vergleich zueinander und zu einem Vergleichsszenario, in dem alle Studierenden mit einem eignen PKW fahren, verhält. Dafür entwickeln wir eigene Umsetzungen für Ridesharing und Ridepooling und werten diese für Agenten aus, die wir mit realistischen Heim-Positionen in und um Würzburg und Transport-Nachfrage versehen. Wir erfahren, dass Ridepooling nicht zur Reduktion der gesamten zurückgelegten Kilometer führt und die Nachhaltigkeit allein von den genutzten Ridepooling Fahrzeugen abhängt. Zudem zeigt sich in unseren Ergebnissen, dass Ridesharing zwar bessere Ergebnisse bezogen auf Emissionen- und Kostenreduktion hat, aber auch dazu führt, dass etwa 10% der Studierenden einen schlechten Service erfahren, da sie keine Ridesharing Rückfahrt finden und somit eine Alternative benötigen.

Contents

1 Introduction

In the wake of climate change, more sustainable mobility concepts are developed in order to reduce CO_2 emissions. Since daily commute by car makes up a large part of emissions [1], finding new ways for reducing the amount of cars on the street is a good starting point for alleviating emissions created by commuting. A large percentage of such daily commuters are university students. Although German students typically receive semester tickets from their university, about 25 percent of them commute in their private cars which can be seen in a study from 2018 [2]. This can be attributed to public transport not being convenient enough for a big portion of students, mainly students living farther away from their university campus [3]. In other areas, such as corporate mobility, methods like ridepooling are already being used for, e.g., bringing workers to their workplace and back home [4] or transporting them inside an employee campus [5]. But while some alternative models for campus mobility are offered and researched extensively [6, 7], there still is no popular solution for German students since, as of 2018, only six percent of them report about sharing a vehicle for commuting [2].

Developing alternative university campus mobility approaches comes with unique difficulties. Unlike most employees, university students do not have a single starting time for their classes which, additionally, differ every day of the week. Furthermore, their preferred commuting times do not have to correlate with their classes since students oftentimes stay at their campus for other activities such as studying, socializing and eating at the cafeteria. This leads to the conclusion that students' transportation needs are dynamic and ill-suited for public transport which comes with overloaded buses in peak times, empty buses in off-peak times, long rides due to detours and, especially in rural areas, infrequent arrival times.

This might result in students choosing commuting by car as a more comfortable option which also allows for a drive without more stops than needed - unlike taking the bus. Therefore, solutions for transporting students should be based on actual demand and consider having a sufficient amount of vehicles for peak times and not performing too many detours as not to have long rides. Additionally, creating an actual incentive for students to choose the dynamic alternative instead of driving their car as usual, such as offering low prices, is of importance. If these aspects are considered, dynamic solutions for grouping student rides to campus offer a compromise for balancing the quality of experience (e.g., costs for the students, ride times, waiting times) and sustainability.

Two mobility models that opt to assign passengers to groups that can efficiently drive to a destination together are ridepooling and ridesharing. Ridesharing is the method of people with similar destinations sharing a ride in a private car, grouping people for an already existing ride [8]. Then, the passengers contribute to paying the costs of the ride. Ridepooling, on the other hand, is a model in which a vehicle, in most cases owned by a ridepooling company, groups (pools) people with similar transportation needs together and then picks them up and drops them off at different stops [9].

In order to evaluate if alternative mobility approaches such as ridepooling and ridesharing have effects on making campus mobility more sustainable, they need to be compared in a realistic way. In this work we develop a framework for modeling and simulating different mobility modes under the same circumstances. With the help of this framework we can assess the opportunities and challenges for utilizing new mobility modes and can offer some decision support for universities trying to further the sustainability of their campus mobility.

For this we focus on students living outside of a specified radius around their campus since their transport connection is usually worse. For simplicity's and comparison's sake, a reference scenario in which all considered students drive their own private car to campus is created. We then compare this method, named "EverybodyDrives", to a ridesharing approach and a ridepooling approach which we develop as parameterized models.

Input are parameters such as accepted time-windows for arrival or departure at the student's campus or the distance a student is willing to walk. The algorithms used in the mobility models have to make decisions around grouping people for rides to and from multiple university campus positions. These decisions are made with the goal to have low ride times and costs for students as well as reducing CO_2 emissions. This can be done by grouping as many students as possible, which is limited by hard constraints like arrival or departure time-windows and maximum accepted ride times specified by students. In order to combat the problem's complexity, the developed algorithms for these approaches use heuristics like "least distance between student homes". We then compare the different mobility models for the output metrics "travelled kilometers", "travelled minutes", "emissions" and "costs".

In this work we utilize our framework to specifically examine the campus mobility of the University of Würzburg which does have ample public transport opportunities inside the city of Würzburg (Germany). Nevertheless, the aforementioned problems of either full or empty buses, infrequent bus schedules and long commutes for students living outside the city still persist. Therefore, we assess whether ridepooling and ridesharing could alleviate these problems for this setting. However, our framework is also applicable to other university campus mobility scenarios since we can configure campus locations, agents' local distribution and their travel demands.

In summary, this work presents, compares and evaluates the different mobility modes for a realistic scenario in order to answer our central research question: Are ridepooling and ridesharing more sustainable methods for campus mobility than our baseline scenario and to what extent? Furthermore, we assess which approach performs the best under which circumstances and test the stability of our models.

In chapter 2 we present the background and definition of ridesharing and ridepooling and give an overview of the related work for ridesharing and ridepooling approaches and campus mobility. Chapter 3 then describes the project's methodology, which entails the concepts, ideas and algorithms of this work. Additionally, the methodology contains more detailed explanations for the methods and processes used in our framework as well as a description of the specific scenario we use for evaluation. We present the results of the specified simulation scenario in chapter 4 which is followed by chapter 5 which entails sensitivity analyses that verify the stability of our models and examine which parameters have the biggest impact on the simulation's outcome. We conclude this work with a short summary, our most important findings and possible future work in chapter 6.

2 Background and Related Work

This chapter describes the background for ridesharing and ridepooling which entails a short summary of their history, most common usages, and new developments. Furthermore, we present other works in the literature for ridesharing and ridepooling as well as campus mobility as a whole.

2.1 Ridepooling

After presenting some background on ridepooling as a concept, we first address that ridepooling approaches can be very different and then expound on some of these approaches and their differences to our work.

2.1.1 A Brief Historical and Present Background for Ridepooling

Ridepooling, or Mobility on Demand (MOD), is a mobility concept in which people are grouped based on their transportation needs and transported without a fixed route. Nevertheless, the passengers do not have to have the same pickup and dropoff positions. They can either be picked up and dropped off at different stops or all driven to the same position (which is the basis for so-called feeder systems). For this, stops can either be made at any possible positions or have to be specified beforehand. [9]. The crucial difference to ridesharing is that a ridepooling organization supplies the vehicles and drivers for the grouped rides. Since the used vehicles are not privately owned in ridepooling, new mobility techniques such as autonomous driving and the utilization of electric vehicles can be tested to further maximize the effect of emission and cost reduction. On the other hand, more empty rides to depots or pickup positions are needed which could dampen the positive effects of grouping people for transport.

Pooling taxis or so-called “call buses” can be seen as the forerunners of ridepooling situations. But, organizing via phone turned out to be inefficient and pooling a lot of people made rides too long and chaotic.[10] A popular modern ridepooling service would be MOIA [11] which operates in Hamburg and Hanover and is booked by using a mobile phone app.

2.1.2 Ridepooling Categories

Ridepooling (as well as ridesharing) can be divided into categories. Vansteenwegen et al. [9], for example, differentiate between dynamic-online, dynamic-offline and static approaches. Generally, what constitutes a dynamic approach is that changes to the schedule can be made during the planning horizon, i.e. new demand information can affect the route planning while some services are already running. Furthermore, dynamic-online approaches allow changes to ongoing services (i.e. driving vehicles can change their route), whereas dynamic-offline methods only allow for services that have not yet started to be altered.

In static approaches, on the other hand, every service execution is planned beforehand and no changes can be made once operation starts. Thus, all considered demand data has to arrive before the planning horizon.

Vansteenwegen et al. further present additional categories for ridepooling approaches, such as many-to-many versus many-to-one (feeder systems), fully flexible vs semi flexible, and a category based on whose goals are optimized (passenger goals vs operator goals vs multiple objectives). Many-to-many versus many-to-one means whether passengers are all dropped off at the same position whereas fully flexible vs semi flexible addresses if time tables are either completely omitted or adapted based on demand.

The ridepooling approach used in this work falls into the category dynamic -offline as changes can be made during the planning horizon but ongoing vehicle rides can not be adjusted by new demand information. Furthermore, since no time tables are used and students of the same ride can be dropped off at different, but similarly located, stops (either campus locations or home positions) we consider this work's ridepooling approach many-to-few and fully flexible. Lastly, the considered objectives are from the passenger perspective.

2.1.3 More Ridepooling Papers

As previously presented, a ridepooling approach can be dynamic or static. In the literature, works that utilize static approaches try to solve complex situations that are fully known beforehand. For example, a static on-demand bus routing problem (ODBRP) is introduced by Melis and Sörensen [12]. The ODBRP involves a fleet of buses, bus stations, travel times and transportation requests with departure and arrival options and time-windows. The goal was to assign passengers to departure and arrival bus stations and create bus routes to minimize the passengers' travel times. For solving this complex problem, a large neighborhood search heuristic is presented. The results show that on-demand bus systems can significantly reduce total passenger ride times compared to traditional fixed-line and timetable-based systems.

On the other hand, works with dynamic approaches usually aim to solve a problem in which information is revealed bit by bit. One dynamic-online approach is by Gomes et al. [13] who developed a Demand Responsive Transportation (DRT) system in which passengers pick a start and end destination and are served by vehicles with the same capacity. Some of the transportation requests are assumed to be known beforehand which are solved as a static problem. Then, dynamically arriving requests are considered during operation. Gomes et al. use a heuristic method by utilizing a reactive greedy random approach with a local improvement phase afterward for building possible routes. They determine pick-up time-windows to be hard constraints and delivery time-windows to be soft constraints. In contrast, in this work's approach time-window constraints depend on the type of ride (either ride to campus or ride home) wherein either all pickups or all drop-offs entail a hard time-window constraint. Another difference is that, in the work of Gomes et al., a stop can be visited multiple times during one ride since passengers may have different time-windows for this stop whereas in this work, only passengers with overlapping time-windows for the same stop are grouped together in order to only visit a stop once during a ride.

We concluded that our ridepooling approach should use a dynamic approach as we find that this resembles the real-world commuting behavior of students, who oftentimes spontaneously decide about transportation needs, more closely.

While further deciding how to structure and develop our approach, we came across many works in the literature that use precise simulations that can model traffic behavior such as MATSim for evaluating their ridepooling approach. For example, Liyanage and Dia [14] developed an agent-based simulation in which they compare on-demand public transport to scheduled bus services in Melbourne, Australia. For this, they develop a microscopic traffic simulation approach. In order to compare the on-demand approach to the scheduled bus services, they used the same passenger demands, which they modelled in a origin-destination matrix, for both scenarios.

Another ridepooling approach using a microscopic traffic simulation model (MATSim) was developed by Narayan et al. [15] in order to evaluate different traffic modes, namely fixed public transport, private cars and flexible on-demand transport for which they used the algorithm by Hörl [16]. These traffic modes exist in competition with each other and are chosen by the agents whose behavior was then analyzed. The authors considered different fleet sizes for the flexible mode and cost ratios between the fixed and flexible mode.

Since we are not interested in the influence of traffic on ridepooling (and ridesharing) and in turn the resulting influence on traffic in this work, we do not use a microscopic traffic model. Instead, we focus on the feasibility and the effects of different mobility modes on metrics like emissions and costs under the same circumstances. Therefore, we also do not consider agents' choice behavior since the different scenarios are evaluated separately.

Many works examine ridepooling opportunities inside of cities as an alternative to, e.g., taxis. We, however, deal with students who do not live inside their university city and thus commute long distance trips in contrast to short distance trips inside a city. One work that, like this work, focuses on people commuting from outside a city is by Liu et al. [17] who investigate a ridepooling service (which they call bus ridesharing) in which the passengers request long distance trips and wait until enough people gather for the ride. The authors focus on optimizing ride-matching by solving the capacitated clustering problem of travel demand and the location-allocation problem for pick-ups and drop-offs as well as pruning with constraints afterward. For this, exact and approximate algorithms are developed. A real-life dataset from Shanghai taxis is used to demonstrate the efficiency of the proposed service which is shown to be cost-effective and energy-efficient according to the results.

Further considerations have to be made about the vehicle fleet. Most works about ridepooling assume that the utilized vehicles are autonomous so as to not consider operational constraints. Since autonomous vehicles are not yet commonly used as of now, examining ridepooling with employed professional drivers would result in more realistic evaluations. Zwick et al. [18] simulate and evaluate a non-autonomous ridepooling approach and compare its results to autonomous ridepooling. They demonstrate that non-autonomous ridepooling is severely limited by operational restrictions like driver shifts regarding the amount of served rides. Furthermore, they find that shift plans have an important impact on these results.

Since we assess alternative mobility modes that could improve campus sustainability in the future, we optimistically choose autonomous vehicles for fewer limitations from the ridepooling provider side.

While most works in the literature focus on developing and evaluating algorithms for efficiently grouping people together, machine learning approaches are emerging more and more for solving and helping with ridepooling problems, especially in the form of reinforcement learning. Si et al. [19], for example, develop a two-level framework for executing online ridepooling with the upper level using a reinforcement learning model. They specifically investigate inter-city ridepooling and use the reinforcement learning model of the framework's upper level for assigning vehicles to an intercity line whereas the framework's lower level utilizes a large neighbourhood search heuristic for dynamic vehicle routing. Their results show that their approach successfully balances the supply for the incoming demand.

Meneses-Cime et al. [20] also utilize reinforcement learning for on-demand ridepooling and specifically use it for creating a dispatcher of shared autonomous vehicles with the goal of serving as many passengers as possible while minimizing their wait times. They also find that their reinforcement learning based approach improves the results evaluated for a realistic ridepooling problem in comparison to a simple vehicle dispatcher.

We however, opt for an approach without machine learning in order to have easily explicable results where each simulation step result can be traced back to our algorithms' instructions.

2.2 Ridesharing

In this section we first present background information about ridesharing's history and present usages. Then, we introduce different categories for ridesharing that are found in the literature and describe different ridesharing works afterward.

2.2.1 A Brief Historical and Present Background for Ridesharing

Ridesharing is a method in which a driver of a private car takes people along to a common travel goal in exchange for financial compensation. This way, no additional vehicles have to be on the road and no empty rides have to be carried out. A downside is that the driver, who provides their car, can decide whether to taking somebody with them (considering the possible additional time of picking up and dropping off the extra passenger). If the driver, e.g. decides that they do not want to drive an additional distance, less groupings of students can be found or passengers have to be willing to walk from their starting position before the ride or to their end position afterward.

In order to arrange ridesharing, information about transportation demands of people without a car and the supply of people with a car announcing their ride has to be gathered and/or distributed. This can be done by people communicating with neighbours, companies grouping employees who live close to one another or via an application that uses the internet.

Naturally, ridesharing has existed since people are able to share transportation devices. Travelling together has always had the advantage that less resources (e.g. wagons, gas) were needed for the same amount of people and company can be shared. In the past some governments or companies have tried to get people to share a ride in order to incentivize saving gas or reducing pollution and congestion. However, the amount of organizations whose goal is to group people for ridesharing exploded with the rise of the internet since the grouping of people who do not know each other to share a ride is especially easy now, seeing as the internet allows for the fast exchange of strangers' data.[8]

Online platforms like BlaBlaCar [21] can let people advertise their planned ride for other people to join them for a fee. Other platforms automatically set the groupings based on algorithms for maximizing efficiency.[22]

2.2.2 Ridesharing Categories and Challenges

Identifying obstacles for persons choosing to share a ride with someone is crucial for developing a ridesharing approach that could realistically be accepted by students. Furuhata et al. [23] present a categorization for different ridesharing approaches and examine the challenges that stand in the way of further acceptance and utilization of ridesharing opportunities. The authors' developed a framework for recognizing the crucial challenges and addressing them.

The presented categories for ridesharing are "identical ridesharing", where the passengers have the same start and end positions as the driver, "inclusive ridesharing" where the start and end locations of passengers are on the way of the driver's path, as well as "partial ridesharing" and "detour ridesharing". Partial ridesharing is when either the goal or the start position of passengers is not on the driver's way and passengers complete this portion of the way themselves. Finally, detour ridesharing is when the pickup

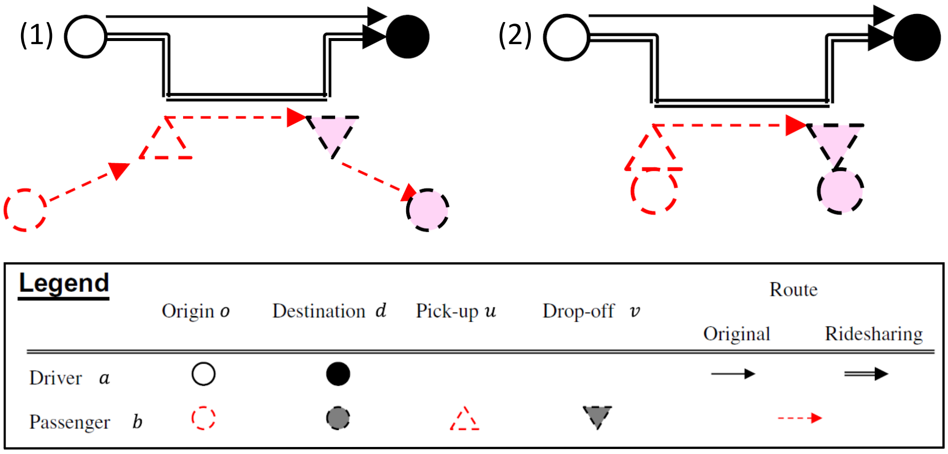


Figure 2.1: An image excerpt from [23] for the visualization of detour ridesharing.

or drop-off location of passengers is not on the driver's way and thus the driver makes a detour. The case of the pickup or drop-off points not being the passenger's origin or goal destination is also included in this scenario. Detour ridesharing is further exemplified in Figure 2.1 which is a simplified excerpt of a Figure from [23] where (1) displays the case of pickup and drop-off not being the passenger's origin and destination and (2) shows the other case.

Our ridesharing approach can be categorized as detour ridesharing since we model drivers to be willing to drive a detour for picking up or dropping off other students at their campus. However, as students are not picked up or dropped off at their home location, they still have to complete a section of their path alone; thus we use a case of detour ridesharing (1). Therefore, we examined the challenges for detour ridesharing presented by Furuhata et al. and addressed them in our approach. For example, they mention the difficulty of cost splitting for detour ridesharing since detours lead to additional costs for the driver and the challenge of detours that are not beneficial for all passengers where all agents need to accept the additional travel. Moreover, in the case of detour ridesharing (1) where passengers need to traverse a section of their journey themselves, they need to find another mode of travel for this section.

We tackled these challenges by limiting the accepted detour time for every agent individually and by also limiting the distance for traversing part of the journey alone which makes it possible to walk by foot to the pickup or from the drop-off location. This way the alternative mode for passengers is settled and does not produce any further costs or emissions. Since all campuses of the University of Würzburg are located in the same city and either the start or end location of the ride is the same for everybody, the detours and thus the additional costs are also limited to a degree. Therefore, we split the gas costs for the whole ride evenly between all agents which then compensates the driver for the undertaken inconveniences.

2.2.3 More Ridesharing Papers

Similarly to ridepooling, ridesharing approaches can also be categorized into static and dynamic. Since requests can arrive at short notice in many real-world scenarios and dynamic approaches offer more complex and thus interesting problems, most works that we encountered in our research use dynamic ridesharing. However, there are some that assume all information to be known beforehand. Tafreshian and Masoud [24], for example, work with reoccurring pre-determined trips, e.g. for commuting to work, and use a mixed integer program for solving their static problem.

As we are examining students who may have varying demands every day and can decide to change their mobility needs spontaneously, we also use a dynamic approach for both ridesharing and ridepooling.

A few research papers try to match people with the help of additional criteria such as social aspects. Wang et al. [25] introduce a new ridesharing approach named collaborative activity-based ridesharing (CAR) that combines social-network-based and activity-based ridesharing. The authors take advantage of users' social networks and their space-time flexibility in order to increase the overall number of matches and further the number of matches with friends. Evaluating CAR with an agent-based simulation with realistic demand data showed no significant difference regarding detour costs compared to pure network-based ridesharing.

In our work we do not use social criteria due to our focus being on the sustainability of our approaches. While we find travelling with friends to be a good incentive for agents to use ridesharing, we deem quality of service constraints to be of more importance.

Yet again as for ridepooling, many ridesharing approaches consider the effects of traffic on ridesharing and the influence of ridesharing on traffic. Traffic criteria can be analyzed with the help of simulation frameworks like MATSim. Such is done by Wang et al. [26] who aim to examine the effect ridesharing has on the transportation system in MATSim and want to use this as a basis for possible policy suggestions.

Another approach that takes traffic influences into account is by Long et al. [27] who use stochastic travel times between nodes in order to approximate the instability of traffic of the real world and examine the effect of travel-time uncertainty on cost-saving from ridesharing with the help of a Monte Carlo simulation method.

As with ridepooling, we do not consider traffic for evaluating ridesharing since our focus lies on the comparison between our ridesharing and ridepooling algorithms.

Similarly to papers that investigate mode choice, some ridesharing works examine which adoption rate is necessary for their ridesharing approaches to have a significantly better performance than alternative scenarios. In the work of Bistaffa et al. [28], for example, multiple adoption rates for ridesharing are assessed regarding their results for quality of service and sustainability metrics. They find that a reduction of 10% of CO_2 emissions requires a 20% ridesharing adoption rate and that an adoption rate of 80% is necessary for achieving a reduction of 50% of emissions.

While we do not consider adoption rate for the main evaluation of our approaches since we assume all students to take part, we do examine the effects of lower adoption rates in our sensitivity analysis.

As is the case for ridepooling, there also exists literature about using machine learning for finding ridesharing solutions. Haliem et al. [29] utilize deep reinforcement learning for a ridesharing approach that allows users to influence matching decisions. Their model-free framework lets drivers plan their rides based on expected rewards and allows for matches to be accepted or rejected by possible passengers based on their preferences. The framework furthermore influences the ridesharing matching with the help of reinforcement learning, based on pricing information. The results show that the framework greatly improved upon the acceptance rates of matches and significantly reduced the covered distances.

As previously described, we do not use machine learning for either ridesharing or ridepooling and instead create whitebox-approaches for evaluating the strengths and weaknesses of our algorithms in regard to matching decisions.

2.3 Comparison of Different Mobility Concepts

Other mobility concepts that often times get confused for ridesharing or ridepooling are ridehailing and carsharing.

Ridehailing means the use of taxis or uber-like services which can be booked very spontaneously and are used without additional passengers, i.e. by one person or one group that is travelling together. However, this is not a sustainable mobility concept since the ride is not shared and, additionally, empty rides have to be carried out to pick up the passenger.[11]

Carsharing is a concept in which different people share the use of a car, i.e. use the car at different times of day or days of the week. This is mainly used by people who do not need a car often. This way, the emissions produced by the manufacturing of cars can be reduced. Nevertheless, the rides themselves are not more sustainable than the conventional single use of cars. Furthermore, additional rides to and from the dropoff position of the car may have to be made.[11]

To the best of our abilities, we could not find any works in the literature that compare ridesharing and ridepooling separately (without mode choice).

With this work we contribute a framework for modeling and evaluating different mobility modes for the campus mobility through simulation under the same circumstances and hard constraints. Our framework makes it possible to easily and dynamically set different input parameters and add new mobility modes. We further develop and evaluate our own approaches for ridesharing and ridepooling based on heuristics and assess which input parameters are crucial for the results of each mobility mode.

3 Methodology

In this chapter we first present this work's problem statement and objective. We further describe the utilized concepts and methods as well as our approaches for the mobility modes.

3.1 Problem Statement and Objective

While students living near their university campus typically utilize public transport, students who live outside their university town usually have fewer options for transportation and are thus more likely to commute by car [2]. As commuting by car contributes greatly to CO_2 emissions [1], universities could help reduce the strain on the environment by exploring alternative more sustainable modes of transport for the campus mobility of their more remotely located students.

When developing a more sustainable mode of commute to and from campus, care must be taken to ensure that students would be accepting of said mode. This can be done by establishing quality of service constraints and creating an incentive for choosing this mode, such as low costs.

In this work we therefore develop a framework that utilizes (hard) quality of service constraints for evaluating alternative mobility modes for students who are more likely to commute by car due to their distance to the university campus. We further evaluate whether our implemented approaches for the examined modes are able to perform reasonably well under the hard quality of service constraints. We judge said performance by output metrics such as CO_2-emissions and resulting gas costs. The chosen quality of service constraints for ensuring acceptance among students are the distance a student is willing to walk by foot, the buffer time before and after a student's preferred arrival and departure time, and how long they would be willing to ride in a vehicle given how long they travel alone.

In our work we choose the mobility modes ridesharing and ridepooling which have been shown in the literature to have a beneficial impact on reducing CO_2 emissions while still being convenient as alternatives to our baseline mobility mode "everybodyDrives".

In order to then make statements about the preferable mobility mode for campus mobility, we evaluate and compare our mobility mode approaches for the campus mobility of the University of Würzburg in Germany. While these results are not fully applicable to other universities, we can deduce which factors are crucial for the success of ridesharing and ridepooling and thus make statements for different initial situations.

To summarize, this work's central research question is: How do ridesharing and ridepooling perform in comparison to each other and to the baseline scenario everybodyDrives regarding specified output metrics for the case of the campus mobility in Würzburg?

This can be divided into multiple sub-questions:

- Are ridesharing and ridepooling feasible under the specified hard constraints?
- Do ridesharing and ridepooling perform well enough in a realistic scenario for significant reductions in CO_2 emissions and student costs to occur compared to our baseline scenario of driving alone by car?
- Does ridesharing or ridepooling perform better and does this depend on the circumstances?
- How sensitive are the outcomes of of each mobility mode on assumptions on e.g. the distance a student is willing to walk by foot?

In order to answer these questions, the steps conceptualization, implementation and evaluation as well as a sensitivity analysis have to be completed. In the step conceptualization we worked to ensure the comparability of the different approaches and thus modeled a mobility demand for all agents (students) which was used as input for all approaches and entails the spatial distribution of agents as well their planned arrival times at and departure times from their university campus. Additionally, the step conceptualization also involved the design of the baseline scenario and the ridesharing and ridepooling approaches and algorithms.

For the step implementation we modeled and programmed the different mobility approaches in Java and used R for data generation. In order to set input parameters that specify the simulation model's attributes and sequence of events, the simulation was developed to read a configuration file for every simulation run.

The step evaluation entails the simulation runs with a chosen configuration and assessing the output metrics in order to answer the research question. Additionally, we executed a sensitivity analysis for ensuring the models' stability and for assessing which input parameters have the biggest impact on simulation results.

3.2 Tools and Methods

Alongside the programming languages Java and R we used multiple tools and methods which are presented in this section.

3.2.1 Simulation

This sub-section is based on Averill M. Law's book "Simulation Modeling and Analysis" [30].

When a system is to be emulated, for example in order to assess the result of changes to said system inexpensively, it can be modeled as an interplay of its components. If this interplay is known entirely, the system's result can be calculated exactly. However, many systems that are of interest in the real world are very complex and often contain non-deterministic behavior. In this case, a simulation of the system can be more of use since a mathematical model of the components' interplay can be used to approximate the behavior of the real system.

There are different categories of models and simulations. One categorization is static vs dynamic. A static model does not have a concept of time whereas a dynamic model's internal state changes over time. Another category is deterministic vs stochastic. If a system is modeled without probabilistic elements, it is called deterministic. A model that does contain probabilistic elements is considered stochastic. The next category is discrete vs continuous. A continuous model's state changes continuously over time whereas a discrete model's state only changes at specific instances, such as time markers like seconds or when events occur. This leads to the categorization of discrete-time simulations vs discrete-events simulation. While discrete-time simulations can change the system's state at every defined timestep, discrete-event simulations only do so every time a predefined event takes place during the simulation's run. A variation of discrete-event simulation is the agent-based simulation in which agents with attributes and behaviors are able to make decisions during the simulation run and hereby affect the outcome.

For our study we develop a dynamic, discrete-event simulation model (that is also an agent-based model) where the model's state changes over time but only events such as incoming requests or ride starts lead to changes to the system state. Additionally, while this is not the focus of our work, we model our agents to perform decisions about using alternative mobility modes or travelling alone based on specified hard quality of service constraints. Furthermore, the model which we develop is deterministic since we generate and save all random variants, i.e. the agents' home locations and campus assignments, as well as all request data, separately before a simulation run in order to easily input the same starting conditions for all mobility approaches. This way, we establish the comparability of the approaches. However, changing our simulation's characteristic to stochastic would be fairly easy by generating the request data after the simulation start which could present a goal for future work.

3.2.2 OMOD

As the simulation's agents should resemble actual students of the University of Würzburg, we ensure a realistic spacial distribution of their home locations. For this we use the tool OMOD[1] developed by Strobel and Pruckner [31] which can generate mobility-demand for a specified area.

[1] https://github.com/L-Strobel/omod/

Given the number of agents as well as OpenStreetMap (OSM) and geographical data in the form of a osm .pbf and .geojson file for the region the user wants to create demand-data for, OMOD returns a json output file containing agent-objects with home locations and activities for each evaluated day of the week and the corresponding locations. For this, OMOD uses real buildings, extracted from the OSM data, and estimates the possible use of the building (like residential or commercial space) based on an OSM building tag. This way, OMOD distributes agents' homes to realistic positions. As only these home positions are needed for our purpose, we utilize the generated home positions and discard the rest of the results output by OMOD.

3.2.3 Graphhopper

Finding an actually usable path for commuting between students' homes and their university campus is important for evaluating a realistic scenario of students using different mobility modes. Graphhopper[2] is an open-source routing engine which we include as a Java library in this work. Graphhopper utilizes OpenStreetMap data and allows for the use of different routing algorithms like Dijkstra or A*. Furthermore, it provides a speed mode which makes very fast responses without using heuristics to routing requests possible. Still, a longer preparation time for the speed mode is necessary. However, it only has to be executed once for the same OSM data since the preparation result is stored afterward. Therefore, we use the speed mode for the two vehicle profiles "foot" and "car" to realize fast routing for each mobility mode.

3.2.4 Jsprit

Another open-source tool that we use in this work is jsprit[3]. Its purpose is solving Traveling Salesman Problems (TSP) and Vehicle Routing Problems (VRP) in a lightweight manner. Additionally, it is further able to solve multiple variations of these problems such as the VRP with Time-Windows which we utilize since our problem deals with hard constraint time-windows. Furthermore, as jsprit is a flexible tool, adding more constraints and adjusting algorithms is easy and allows for solving even more complex scenarios which is why we also utilize this feature. For example, we create an additional constraint for not allowing waiting times at pick-up or drop-off stops. For calculating the different costs of a possible VRP solution we use the path distances given by Graphhopper and dynamically build a distance and time matrix.

[2]https://github.com/graphhopper/graphhopper
[3]https://github.com/graphhopper/jsprit

```
{
  "postcode pairs": [
    {
      "distance": 28.35710889822725,
      "postcode pair": "97816-97070",
      "student count": 12
    },
    {
      "distance": 10.128943058015576,
      "postcode pair": "97199-97080",
      "student count": 4
    },
    {
      "distance": 42.36392116987996,
      "postcode pair": "97437-97074A",
      "student count": 8
    },
```


Figure 3.1: An excerpt from the zip code pair file.

3.3 Creation of Agents and Request Data

In order to be able to execute a simulation for the campus mobility of Würzburg, we need agents that represent the university students in and around the city. For the purpose of evaluating a possible real-world scenario, these agents should resemble actual students in their mobility behaviour regarding their home and campus locations as well as arrival and departure times at campus.

3.3.1 Agent Home Zip Code Distribution

We start with distributing the students' home positions in a way that reflects the reality of the university students in Würzburg. For this task, we are provided by the University of Würzburg with the results of a survey about its students' faculty affiliations and zip codes of their home locations.

With this information we discover the necessary amount of students per home zip code for each campus zip code. We further calculate the smallest distance (i.e. nearest points distance) between each zip code pair. We then save this "home zip code to campus zip code student amount" data along with the respective smallest distances in a json file, called "all_postcode_pairs", of which an excerpt is shown in Figure 3.1.

However, some students questioned in the study offered home zip codes whose distance to Würzburg is not realistic for daily commute. A possible explanation is that these students answered with their parents' home zip code. To address this problem, we set a maximum radius of 55km as a cutoff point since more than 90% of German commuters travel less than this distance daily [32].

Therefore, we only consider students with zip codes within the specified radius by excluding home zip codes whose smallest distance between this home zip code region and the campus zip code region is above the maximum radius.

As we use the smallest distance between zip codes for limiting the amount of agents of our base population, there might be some included agents that actually live farther away than 55km because they are located in the middle or the outer edge of their home zip code region. However, this strategy ensures that the edge case of an agent living on the nearest point to their campus zip code will also be included when the campus location is also on the corresponding nearest point of the campus zip code. Moreover, these agents chosen by the zip code distance only build a base set for agents to be pooled out of. The agent selection for a simulation run is then based on actual distances between agent homes and campus locations and pooled out of this base set.

3.3.2 Generating Agent Home Positions with OMOD

These aforementioned selected "home zip code to campus zip code" pairs with their according amount of necessary students therefore serve as base for creating a wide range of agents whose commuting behavior is of interest and can still be called realistic.

We create these agents with the help of the tool OMOD [31] which is explained in more detail in Subsection 3.2.2. Since a radius of 55km around Würzburg entails the federal states of Bavaria, Hessen and Baden-Württemberg, we merge the OSM data of the three states and output the merged result into an osm.pbf file. For the geographical data we create a geojson file with the radius of 60km. Afterwards, these two files and the amount of agents to be generated is input into OMOD.

We choose three million as the number of created agents as this would roughly be the amount of people living in the region covered in the geojson file (based on the average German population density). This high number of generated agents ensures that it is very likely that enough agents with the necessary home zip codes are created. Otherwise, OMOD runs with smaller numbers of agents could lead to some zip code regions not being filled with enough agents which did indeed happen for our first experimental OMOD runs with, e.g., one million agents.

Out of these generated three million, we choose agents with homes in the zip code regions mentioned in the base "all_postcode_pairs.json" file until the necessary amount of agents for each zip code is covered. Then, we assign a university campus zip code to each agent which is also based on the zip code pair information.

Figure 3.2 displays a possible distribution, generated by OMOD, of agent home locations in the 55km zip code radius where each agent is shown in a color that is based on their campus affiliation.

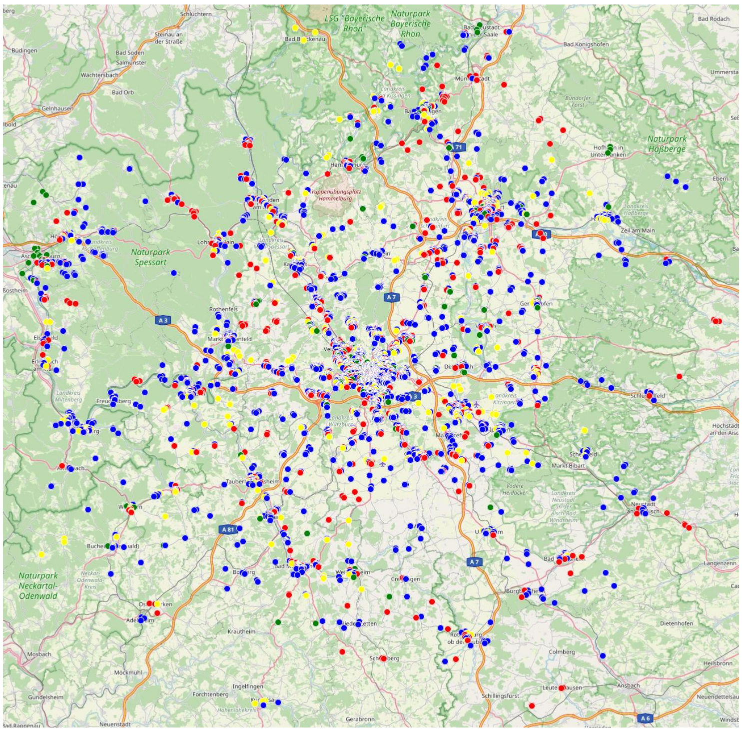


Figure 3.2: An exemplary agent home distribution generated by OMOD with four different campus affiliations, limited by a zip code radius of 55km.

3.3.3 Request Data Creation

Since our created agents still lack transportation needs, we begin the step of creating and storing such demand in the form of transportation requests which can be used as input for the different mobility modes.

Basis for this request generation is a survey by the German Federal Ministry for Digital and Transport (BMDV)[4] who examined the mobility in Germany. This survey contains information about the questioned university students' travel start times and end times to their university as well as the duration of their stay which we use to additionally calculate the students' departure times.

In order to create enough request data for students, probability distribution functions must be extracted from the study data about arrival and departure times. We do this in the programming language R which provides a `density` function for estimating kernel densities and an `approxfun` function which can interpolate data points. With this, we calculate the probability for a preferred arrival time for every 5 full minutes of a day. Then, we adjust the resulting values in such a way that they form an actual probability distribution and add up to 1. With this probability distribution we are able to generate samples for the arrival times in a 5 minute tact.

[4]https://www.mobilitaet-in-deutschland.de/archive/index.html

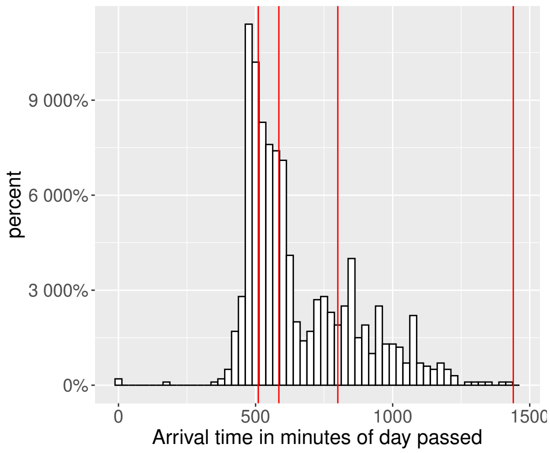


Figure 3.3: The distribution of arrival times with the quartiles drawn in.

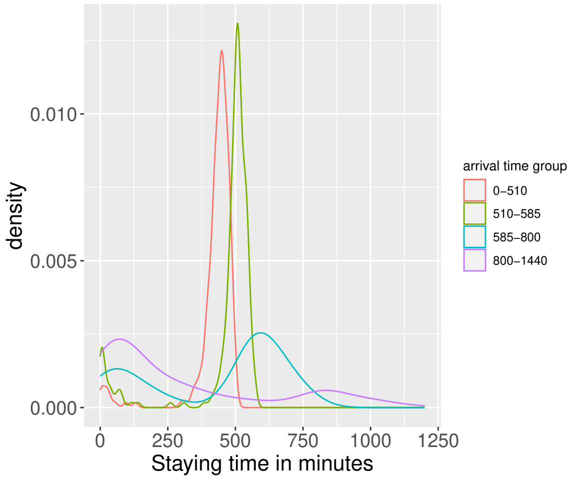


Figure 3.4: The staying time kernel densities for the different arrival time categories.

Since the arrival time influences the time that students spend at their university (e.g., a student who arrives at 16:00 at campus will likely not stay as long as a student who arrives at 7:00), we split the arrival times and their corresponding staying times at campus into four categories, namely the quartiles. For each quartile we then separately determine a probability distribution for the staying times. As exemplified in Figure 3.3, which displays the arrival time distribution derived from the study, the first quartile of people arrive at their campus until 8:30 (510 minutes of day passed), the second quartile of students arrive until 9:40 (580 minutes of day passed), the third quartile until 14:10 and the last quartile contains all students who arrive until the end of the day.

Figure 3.4 shows the different staying time kernel densities for each category. With these `density` function results and the function `approxfun` we determine for each category a different probabilities for the staying times for every 5 full minutes. We then yet again adjust the resulting values for each category to add up to 1 and thus obtain four valid probability distributions where the arrival time determines which probability distribution for the staying time is used. With this we then generate an accompanying staying time for every arrival time and save this as a departure time by adding the staying time to the arrival time.

Additionally, an arrival time for the actual request has to be generated for every student request as this time is needed for the simulation runs of ridesharing and ridepooling. This is done by generating a decision time before the arrival time (i.e. the amount of minutes that the request is sent out before students want to be at their campus) with which the request's arrival time can be determined by subtracting the decision time from the arrival time at campus. For generating the decision time we chose a continuous distribution between 60 and 360 minutes.

These request information tuples (arrival time, departure time, and request arrival time) are then randomly assigned to each of the previously generated students. This results in every request now containing the data of student, home position, campus location, arrival time, departure time and request arrival time. We generate this information for 12780 students that are the total amount of students created for the chosen maximum radius of 55km (i.e. 55km as the smallest distance between zip codes).

This ensures that changing the outer radius for a new simulation run still preserves previously regarded agents since a smaller radius means that a subset of the agents is now considered. For example, expanding the outer radius to 40km after a simulation run with an outer radius of 30km still retains the agents and their data living in the 30km range. Consequently, the outer radius should not be set above 55km.

3.4 Framework for the Simulation

In this section we describe how we model different components of the simulation and which parameters can be input for configuration.

3.4.1 Modeling the Simulation Components

In order to execute simulations for multiple mobility scenarios, a framework in which the same circumstances can be evaluated is needed. This framework is implemented in the object-oriented programming language Java and contains modules for helping with the agent and request data generation which is more closely described in Section 3.3. Furthermore, the framework entails multiple Java classes that model important elements for the simulation runs. Figure 3.5 depicts the structure and connections of these classes which we will highlight in the following text by capitalizing their names.

The input data for the simulation is modeled as the Java classes Agent, Request, Coordinate and Vehicle. An Agent is specified by an id, a request, a car, a home position and attributes about their willingness to accept alternatives or inconveniences. These attributes consist of how long they are willing to ride in a vehicle (which can be set to different values based on the mobility mode), how far they would walk (e.g. to another home position) and whether they would accept using alternative mobility modes at all. Furthermore, agents are also specified with a time interval which determines the time-window they accept for arrival or departure at campus. For the sake of simplicity and comparability of the different mobility models we further model each agent to own a private car.

A Request belongs to exactly one agent, entails an id, a Requesttype, request time, arrival time and departure time, as well as a home zip code, campus zip code, home position and drop-off position. Additionally, the corresponding agent's accepted time interval value is used to calculate the time-windows for arrival and departure which are

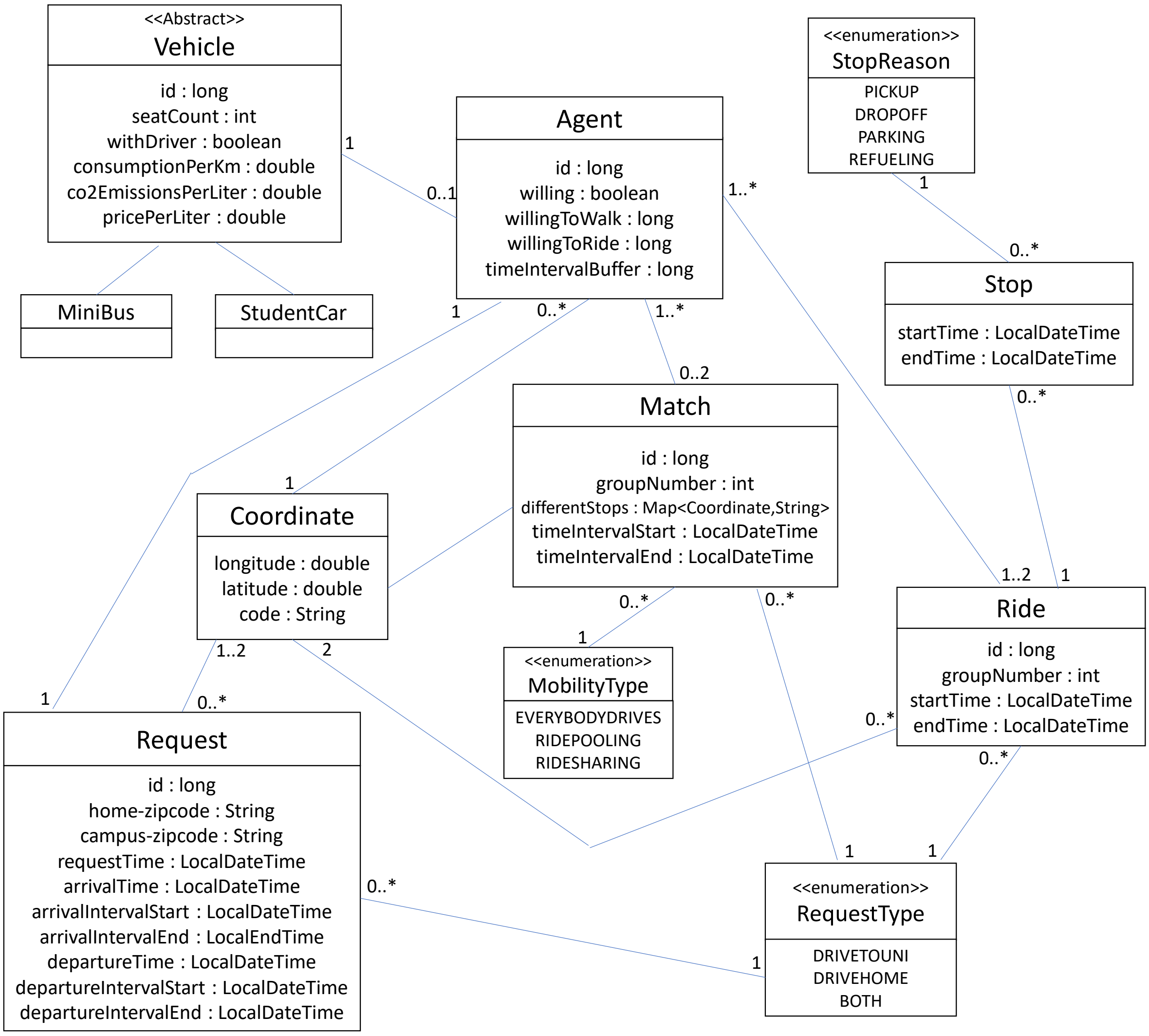


Figure 3.5: The framework’s classes and their relationships.

also saved for each request. The Requesttype is implemented as an enum which can either assume the values DRIVETOUNI, DRIVEHOME or BOTH.

The class Coordinate represents positions of, for example, homes or campus parking locations. Each Coordinate contains a longitude value and a latitude value.

The Vehicle class is abstract and can be extended by actual vehicle models, such as a specific car or car type like a minibus. Each vehicle entails an id, a seat count and specifications about its gas consumption per km, gas price and CO_2 emissions per liter, as well as whether a driver is present or the vehicle is driving autonomously. In this work the abstract Vehicle class is extended by the classes StudentVehicle and MiniBus.

StudentVehicle is supposed to represent a student owned private vehicle and MiniBus represents a ridepooling vehicle which is usually a minibus with a few seats more than a typical car.

Besides the input data multiple models are needed for portraying mobility elements that we implement as Java classes. These classes are a Ride with its Stops and Stopreasons, a MobilityType and a Match. The MobilityType represents the type of mobility model used; it is an Enum which, in this work, can assume the values EVERYBODY-DRIVES, RIDESHARING or RIDEPOOLING. A MobilityType attribute is used in the class Match in order to state by which method a specific Match came to be.

The Match class depicts a possible matching of students for a shared ride. This matching is not yet definitive but states that a joint ride for this group of students is feasible. A Match contains an id and information needed to form a finite matching, a Ride. This information includes the Agents of the Match, the driver (which can be null), the Match Vehicle, the Coordinates of the stops along the way, the RequestType and MobilityType, the start and end Coordinates, and possibly a group number which is needed for ridepooling. Furthermore, a match possesses multiple time-windows; one for the most crucial stop (either final arrival for a match to campus or first student departure for a ride home) and one time-window for each stop position.

Out of a Match a Ride can be derived which represents a finite grouping of students who ride in a vehicle together. This grouping can also consist of only one person who either wants to commute on their own or could not find other students to match with. Therefore, a Ride consists of similar attributes as a Match, i.e. an id, a list of agents, a driver, a vehicle, a group number and start and end positions. But, further specifications such as start and end times as well as actual Stops are added. The Stop class defines a stop executed during a ride. It contains a stop start and end time, a stop Coordinate, the students who the stop was for, and a Stopreason which is an enum with the possible values PICKUP, DROPOFF, PARKING and REFUELING.

3.4.2 Input Configuration

Multiple input parameters can be set in a config.json file in order to specify a simulation run. This way, we are able to evaluate different scenarios for each mobility method dynamically and fairly easily. Every simulation run consists of multiple separate simu-

```json
{
  "lower radius":2.5,"upper radius":25,"student car seat count":5,"time interval":15,"accepted walking distance":1200,
  "accepted driver time":"x + log1.4(x)","accepted ridepooling drive time":"x + log1.2(x)","stop time":1,
  "postcode mapping":{
    "97070":{"longitude":9.9326207,"latitude":49.7907874},
    "97080":{"longitude":9.958895,"latitude":49.8058},
    "97074A":{"longitude":9.951346,"latitude":49.786898},
    "97074B":{"longitude":9.96832,"latitude":49.782563}
  },
  "studentCarConsumptionPerKm":0.07,"studentCarCo2EmissionPerLiter":2400,"studentCarPricePerLiter":1.74,
  "busConsumptionPerKm":0.1,"busCo2EmissionPerLiter":2400,"busPricePerLiter":1.74,"request file":"agentsWithRequests.json",
  "postcode file":"Student_home_distribution_with_faculty_postcode.csv","postcodePairFile":"all_postcode_pairs.json",
  "countOfGroups":18,"radiusToExclude":1500,"bus count":250,"bus seat count":6,"busWithDriver":false,
  "centralCoordinate":{"longitude":9.9326207,"latitude":49.7907874}
}
```


Figure 3.6: An exemplary configuration json file.

lations for evaluating every mobility mode with the same config file whose path needs to be stated with the simulation start. This config file is then read in the start and entails all the needed information. An exemplary configuration is shown in Figure 3.6.

Generally, the configuration data can be split into the groups “file paths”, “general simulation specification”, and “model specification”. The categorization of the configuration parameters can be seen in Table 3.1.

The file parameters serve as support in the case that some data necessary for the simulation is missing as it can then be generated out of these files. The general specification parameters all specify important information for the execution of the simulation runs. For example, the radius parameters influence which agents are considered out of the base set with all 12780 generated agents. Lastly, the model specification parameters set attribute values in models such as Agent, StudentCar and MiniBus.

3.5 Approaches for the Mobility Concepts

In the following sections we present our approaches for the three mobility modes “everybodyDrives”, i.e. the baseline mode, ridesharing and ridepooling. This entails a description of the used algorithms, tools, input, output, and the structure and sequence of events of the simulation.

3.5.1 Simulation Sequence for all Mobility Modes

Since all mobility modes have the same goal of transporting agents to campus and back, they entail similar simulation sequences which are portrayed in Figure 3.7.

At first, before any of the mobility mode simulations can start, some data has to be prepared. The agents that are specified in the configuration file are read and the additional information contained in the input parameters such as “student car seat

File Path Parameters	request file, postcode file, postcodePairFile
General Specification Parameters	lower radius, upper radius, stop time, postcode mapping, countOfGroups, radiusToExclude, bus count, centralCoordinate
Model Specification Parameters	student car seat count, time interval, accepted walking distance, accepted driver time, accepted ridepooling drive time, studentCarConsumptionPerKm, studentCarCO_2EmissionPerLiter, studentCarPricePerLiter, busConsumptionPerKm, busCO_2EmissionPerLiter, busPricePerLiter, bus seat count, busWithDriver

Table 3.1: Parameter categorization.

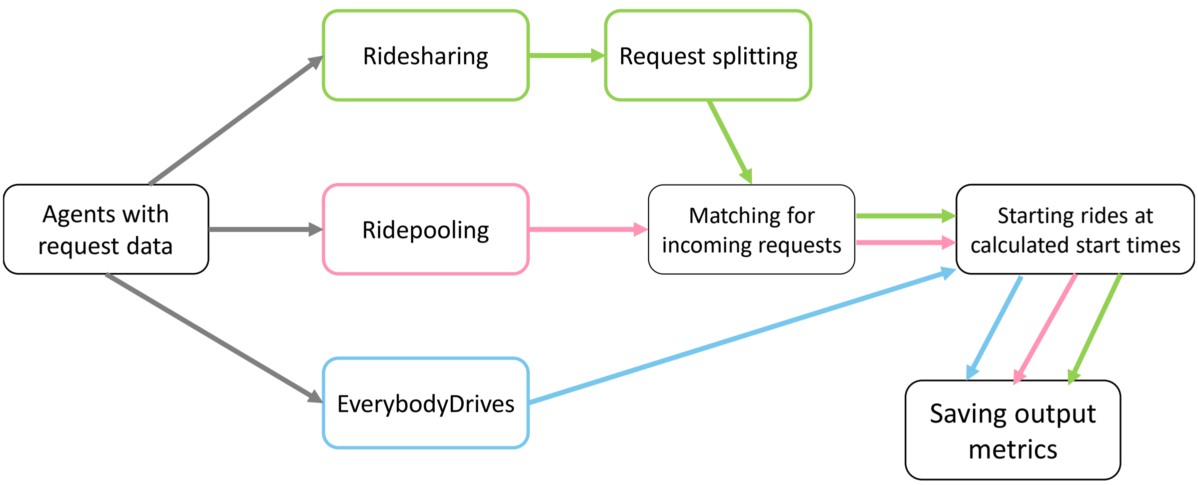


Figure 3.7: The rough simulation sequence for all mobility modes.

count" is applied to them. For the parameter "time interval" this specifically means that we determine the time-windows for each agent based on the time interval value (given in minutes).

We calculate the lower arrival time-window limit (inclusive) by subtracting the time interval minutes from the preferred arrival time set in the agent's request and determine the upper arrival time-window limit (inclusive) by adding the time interval minutes to the preferred arrival time.

For the lower departure time-window limit (inclusive) we adopt the preferred departure time set in the request (as we assume this to be the earliest possible departure) and for the upper departure time-window (inclusive) we add double the amount of the specified time interval minutes. This way, both the arrival and the departure time-windows have the same length.

In this work, we apply the same parameter values to all agents, i.e. all agents accept the same walking distance, for example. For a more realistic evaluation, the parameters could be set with probability distributions which could be investigated in future work.

These agents are then used as input for every mobility mode simulation. Since the everybodyDrives mode does not entail any matching of agents, the next step is to immediately determine when agents start their ride to or back from their university campus. On the other hand, ridesharing and ridepooling have to execute matching of agents for every request. Before this ridesharing contains an extra step in which we split the requests into ride-to-campus and home requests. While the ride starts are not strictly after all matching has been completed, the matching step starts earlier and is thus depicted as a earlier step in this visualization.

After all rides have been completed, every mobility mode simulation saves the resulting output metrics in several output csv files. These metrics are costs, CO_2 emissions, travelled kilometers and travelled minutes for each ride separately and for each agent summed up (i.e. the metrics for agent's ride to campus and back are added together).

For ridesharing and ridepooling we save additional metrics such as the number of occupied seats for each ride.

3.5.2 EverybodyDrives Approach

For the purpose of evaluating and comparing the effects of ridesharing and ridepooling as well as possibly more mobility modes down the line, a reference mobility mode is necessary. As estimating the metrics of the different possible ways of commute for several thousand students is difficult and time-consuming, we assume a baseline mobility mode of every student travelling by a privately owned car. Since we focus on students that live outside of Würzburg, this assumption comes a bit closer to the truth due to students choosing the car more the farther away they live [3]. We call this mode "everybodyDrives" in this work.

The basic idea for everybodyDrives is for every student to drive separately in their own car and for them to arrive and depart at the exact time they wish to.

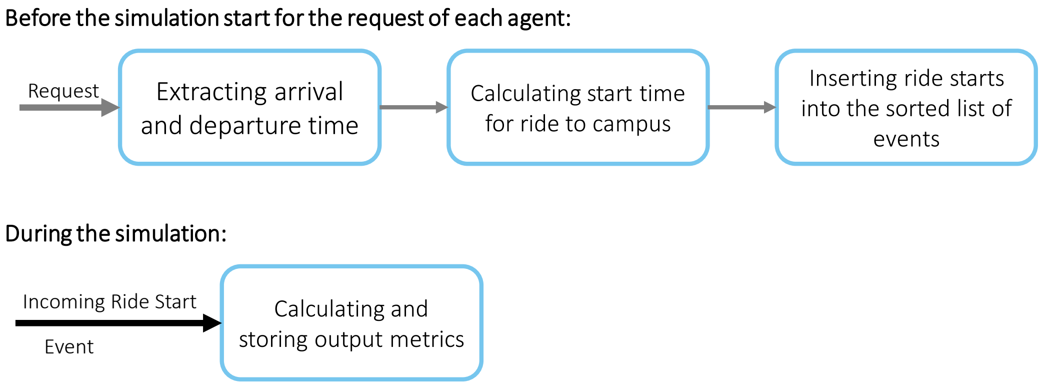


Figure 3.8: The everybodyDrives simulation sequence.

As an agent's transportation needs are, in this work, expressed in a request, we extract the necessary information from each agent's request. Since agents do not need to inform anybody about their transportation goals in the everybodyDrives approach, a request arrival time is not needed. With the specified preferred arrival time we calculate the start time for the ride to campus. We do this with the help of Graphhopper (see Subsection 3.2.3). The ride time of the path, given by Graphhopper, is simply subtracted from the arrival time. We are able to determine the exact drive start time up front since we do not consider any traffic and therefore, no ride time fluctuations exist. The specified preferred departure time from campus does not need to be adjusted and serves as the drive start for each agent's ride back home.

For everybodyDrives no further preparation of data is necessary.

Since we are working with an event-based simulation, the simulation simply starts with the student with the earliest drive start for their ride to campus, followed by the second earliest and so forth. As soon as some students start to drive home earlier than other agents driving to uni, the events consist of ride starts to uni interspersed with ride starts back home until, finally, mostly rides back home remain. For each event (i.e. ride start) the output metrics for this ride and agent are calculated, saved and also stored temporarily. This simulation sequence is further exemplified in Figure 3.8.

Finally, after all agents have driven to and back from their university campus, we examine for each agent if all of their constraints have been observed and if so, the output metrics for all agents and all rides are saved in result csv files. These results further serve as reference data for the other mobility approaches.

3.5.3 Ridesharing Approach

The idea for the ridesharing mobility concept is for students to share their rides to and from campus in their private cars without sacrificing the comforts of commuting by car such as short ride times and travelling whenever the driver wishes to.

3.5.3.1 Assumptions Made for the Approach

For avoiding inconvenient rides, we determine that the driving agent does not drive to other agents' houses for pickup or drop-off and instead expects the other passengers to walk to or from the driver's home position. This necessitates other agents to be willing to walk by foot. The maximum distance an agent is willing to walk can be set with the "accepted walking distance" input parameter in the configuration file.

Dropping off or picking up other students at different campuses is permitted in order to increase the possible groupings of students. However, in order to prevent large detour times, we limit the accepted time an agent is willing to ride in a car for ridesharing. This value is different for every student and is based on the time it takes the student to commute to campus on their own. We achieve this by inputting a function that specifies the accepted value for every student. We assume that a student with a short travel time alone is willing to enlarge this time more percentage wise than a student who already takes a long time to get to uni. As a result we choose the function "$x + \log_{1.4}(x)$" with x being the time for travelling alone for calculating the accepted time as a default. Nonetheless, this function can also be changed in the configuration file. This way we obtain a rather realistic approach for drivers accepting small detours for picking up fellow students along the way and refusing to drive to the other part of town for a pickup. Through the use of a logarithmic function we further set agents to accept a bigger change of ride time when living closer to campus and a smaller change when living far away. For example, an agent who travels for 3 minutes when alone would now accept travelling 6 minutes while an agent who originally travels for 40 minutes now accepts rides up to 50 minutes. If ridesharing rides without any detours are wanted, the function can be set to "x". Similarly, the function can be set to "$2x$" or the like if another linear relation such as doubling of accepted ridesharing drive time to drive time alone is preferred.

Furthermore, the aspect of drivers and passengers only travelling at convenient times is covered by using time-windows as hard constraints and limiting the travel-time as this results in observing the wished for arrival and departure time of students and not starting their ride to campus much too early or arriving back home too late for their liking.

Additionally, our approach allows for agents, who have left their car at home, to not find a match for travelling from the university campus back home. Thus, if no matching ride home is offered by someone else, the affected agents are left stranded at their campus and are denoted as "lost" in this work. Naturally, this would not be such a problem in the real-world as public transport exists even though it might be very bothersome to use for students living outside the university town. Furthermore, the hard constraint time-windows could be disregarded in order to wait longer for another, though more inconvenient, match that might also never emerge. However, as we do not consider public transport in this work and strive to observe the hard constraints, these options are not examined further.

We instead offer a hypothetical carsharing service to these students with the optimistic characteristic of the carsharing vehicle having the same rate of CO_2 consumption and being allowed to be left at the student's home position after use. This way, no ride back has to be considered for the utilized carsharing vehicle. We, however, set this option to be costly (30 cents for every travelled kilometer) and are thus able to take into account the negative effect of not finding a match home.

3.5.3.2 Matching Strategy

Having ensured that ridesharing is not inconvenient for participants (with the exception of agents categorized as lost), we develop an approach for ridesharing for campus mobility.

Since we modelled each student to own their own car, they all could be the driver of a grouping. Therefore, the strategy is for them to first check whether another person with a home position in the accepted walking radius to them already offers a ride to uni at a fitting time. In this case the student then leaves their car at home and opts to become a passenger of the already offered ride. If multiple such fitting ride offers exist, the student chooses the ride with the smallest walking distance. Should no fitting ride offers exist, the student would offer to drive themselves in their own car and other students could then join this new ride offer.

The more detailed matching process looks as follows for every considered request: First, we examine already existing matches for the considered request, from which matches with the false RequestType or that are already full (i.e. all seats are occupied) are eliminated.

These remaining matches are sorted based on the similarity to the request. The considered aspects are the distance between the driver's home and the request-agent's home and the similarity of the decisive time-windows of the match and of the request. Matches with a small distance between the driver and the request-agent as well as a similar time-window to the request's time-window are preferred and therefore at the front of the list of possible matches. Of this list the first 20 matches are examined.

For each of these 20 matches (or less if there are no 20 possible matches), we check if the assignment of the request-agent to the match is actually feasible under the given constraints. If so, we add the match to a list of eligible matches. This check is carried out by investigating multiple conditions:

- The driver's home position is inside the specified accepted walking distance to the request-agent
- There is an overlap between the match's time-window and the request's time-window
- All time-windows are observed and
- All agents accept the resulting ride times

The time-window overlap is evaluated differently depending on whether the driver's campus is at the same location as the request-agent's campus. Should the locations be the same, a true overlap has to exist. If the request-agent's campus location is different, the overlap is calculated by shifting the request-time-window either forward for rides to uni or backwards for rides home by the time it takes to drive from the request-campus to the driver's campus. This is due to the driver's campus location always either being the last or first stop of a shared ride and the other stops always being executed before or afterward, respectively. For example, a match's driver wants to arrive between 12:00 and 12:30 at uni, another student of this match wants to arrive between 11:25 and 11:55 at a different campus. Since the drive from the second student's campus to the driver's campus takes five minutes, the second student can be dropped off at 11:55 with the driver arriving at 12:00 at their campus.

The check for the time-windows and ride times can be evaluated after solving the VRP of this matching which we accomplish with the help of jsprit and Graphhopper (see Subsections 3.2.3 and 3.2.4).

Should all conditions be true, the match is categorized as feasible for the examined request and added to the list of eligible matches.

Out of these eligible matches we currently pick the one with the smallest walking distance. In the future this could also be changed to pick the match with the least overall time entailed. The request-agent is then added to the chosen match and a new start-time, fitted to the new match solution, is calculated. Additionally, if the request-agent has the same campus as the driver, we adjust the match's time-window to the overlap of the previous match time-window and the request time-window. Should the campus be different, we update the match's stops. If the stops already contain the request-agent's campus due to another passenger, the stop time-window is updated with the overlap of the request's time-window and the other passenger's time-window.

3.5.3.3 Preparation of Data

The sequence of the ridesharing simulation is depicted in Figure 3.9 and first starts with a preparation of data which is also depicted at the top of the Figure.

Since we want the students to request their rides to uni and back home separately and the input agents are configured to have only one request, which entails both preferred arrival and departure times and one request arrival time, we first split this request into two requests. One request for the arrival at campus and one for the departure from campus. The source request's arrival time is given to the arrival-at-uni request and a new request arrival time is created for the departure-from-uni request. For this, a randomizer with a seed of "1234" generates a number between 30 and 120 minutes using an equal probability distribution.

This way, students request a ride to campus even though they are not guaranteed a ride back. We chose this approach as it presents a difference to the ridepooling approach

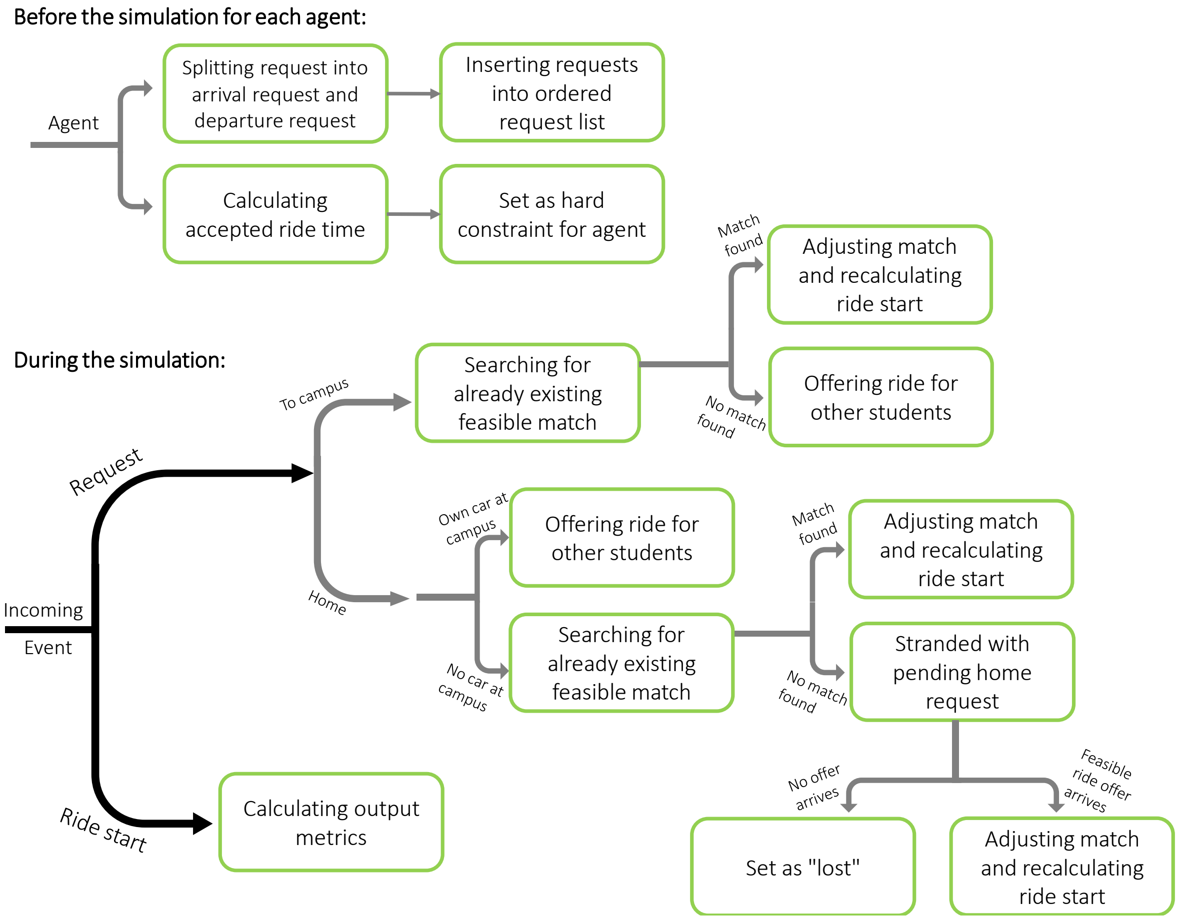


Figure 3.9: The ridesharing simulation sequence.

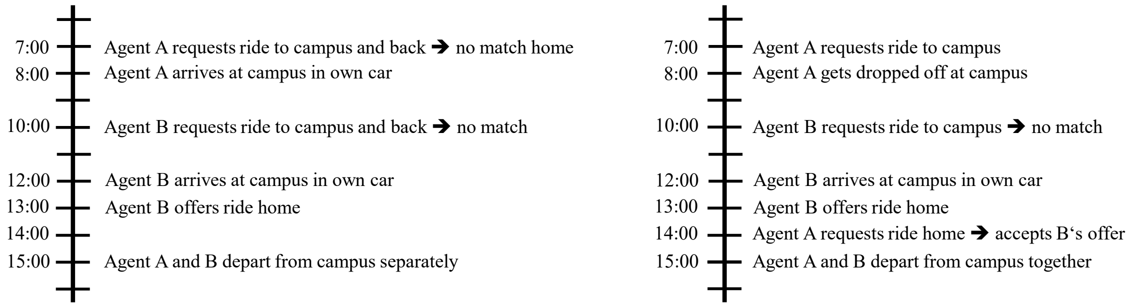


Figure 3.10: The sequence when agent A ignores a ride offer to campus from agent B due to no ensured back ride (left) and the sequence for accepting the ride offer (right).

and also due to the fact that some possible matchings for the ride home only appear once some students of this future match have already left their home. For example, a student wants to arrive at campus at 8:00 and wants to leave at 15:00. Thus, this student requests a ride to campus at 7:00 which results in a match. However, no matches are yet available for the ride home at 15:00. If the student would now decide to ignore the match for the ride to campus, as no ride back is guaranteed, they would miss out on an incoming offer for a ride back home from a different student who does not want to arrive at uni until 12:00 and also leaves at 15:00. This would result in no lost students but also much less ridesharing matches. Hence, students who are using our ridesharing approach have to be willing to trust that an opportunity for a ride home will emerge later on and be tolerant of the possibility of being lost and having to use the more costly alternative carsharing option. The presented example is further exemplified in Figure 3.10 in which two possible timelines of the same example scenario are presented. On the left, the example's timeline without request splitting is depicted which results in more cars on the road. On the right, said example is presented with the splitting of requests.

After splitting the requests into arrival and departure requests, all resulting requests are saved and chronologically ordered regarding their request arrival time since requests can only be considered in the simulation as soon as they arrive.

Then, we calculate for every student how long they are willing to ride in a car based on the input parameter function ("$x + \log_{1.4}(x)$" for our use) and their travel time alone and set this value as another hard constraint.

After these preparations the ridesharing simulation can finally start.

3.5.3.4 Simulation Sequence

As our approach is implemented as an event-based simulation, we jump from incoming event to the next incoming event. This event can either be an incoming request for a ride (to uni or back home) or the start of a ride. Each step of the event processing is depicted in Figure 3.9.

As soon as a request arrives, our algorithm for finding a match starts (see Subsection 3.5.3.2).

In the case of no eligible matches existing for the request, two scenarios have to be considered. If the request is for a drive to uni, the request-agent offers to drive themselves using their own car. Then, the next incoming requests can consider to join this new offer until the ride has to start. In the second scenario the request is for a ride home. Since the request-agent was looking to join a ride home and does not have a car at campus (as it was left at home), they are, as of now, stranded at their university location. However, there is still the possibility of another agent who has a car at campus offering a ridesharing opportunity for a ride home that fits the stranded agent. Thus, the stranded agent is not yet officially added to the "lost" group of agents; instead, their request is added to a list of pending requests and is only officially categorized as lost when their time-window for departure is in the past.

As mentioned, a student who drove their car to campus, naturally will also drive back without considering other already existing matches. So, if a request from student with a car at campus for a ride home arrives, this student automatically becomes the driver of a new ridesharing match that other agents can join. With the emergence of this new match, the pending request list is considered. Should a pending request of a student who is so far stranded at campus exist, we examine whether this student can be added to the new match. If so, they are no longer stranded.

Finally, the other kind of event that can occur in the ridesharing simulation is the start of a ride. Each match update also updates the start time for this match. Then as soon as it is a match's turn (based on the start time), we convert the match into an actual ride that has a fixed start time, end time and information abouts stops for passengers that travel to a different campus. Then, the output metrics (travelled kilometers and minutes as well as emissions and costs for each passenger) are calculated for this ride. The travelled kilometers and minutes are calculated as the kilometers/minutes between the pickup and drop-off for each agent. The emissions and costs of the ride are split between the passengers.

3.5.4 Ridepooling Approach

Similarly to the ridesharing approach, our approach for ridepooling strives to offer a comfortable alternative way of commuting to and from campus.

3.5.4.1 Assumptions Made for the Approach

Yet again, time-windows and maximum accepted driving times have to be observed in order to ensure the desired convenient nature. However, as ridepooling does not entail participants driving in private cars, no agent will be preferred for the start location of a ride. On the contrary, we decided that each agent would be picked up and dropped off directly at their home location, further contributing to the convenience of a ridepooling ride. For this reason we set the students to accept slightly longer detours than for ridesharing, which is why we specify "$x + \log_{1.2}(x)$" as the default input function for calculating the accepted ridepooling ride time for each agent with x being the time needed for driving alone.

The general idea of our ridepooling approach is for students to request a ride to and a ride back from university in one single request that arrives before the ride to uni. If both rides are guaranteed, the students are being picked up at their home location by ridepooling vehicles and dropped off at their campus with some detours along the way, the same happens for the other direction. Should there not be two guaranteed rides, the affected student uses their own car.

The new input general specification parameters that are needed for ridepooling are the count of vehicles used, a number of groups and a central Coordinate for categorizing incoming requests as well as a radius for excluding requests from categorization. Additionally, the new input model specification parameters consist of the ridepooling vehicle gas consumption per kilometer (in liters) as well as the price and CO_2 emissions per liter. Then, there is also the seat count for the ridepooling vehicle and a specification for whether the bus is driven by a person or drives autonomously.

We further assume that multiple, locally distributed depots with fixed capacities exist for housing the ridepooling vehicles from which the vehicles depart from and return to. Furthermore, we set the service times for the vehicle times from 4:00 in the morning until 1:00 of the next day. The last ride of the vehicle has to be executed before the service time is over. With the utilization of multiple depots we reduce the empty distance a ridepooling vehicles has to cover at the start or the end of the day as long as it can return to the closest depot. We further set the depot that is most centrally located to have a higher capacity than the other depots, namely 40%.

3.5.4.2 Matching Strategy

The first step is for our algorithm to find a match for the ride to campus. For this, we first need to assign the request to a group number in order to limit the examination of existing matches since primarily matches with the same group number are considered. A match is assigned the group number of the first request that lead to this match. This assignment is based on the input central coordinate which represents a chosen central point for calculating the angle between the request-agent's home and the y-axis of an imaginary coordinate system with the central point as the origin. Based on the input parameter count of groups we define which angle-interval belongs to which group. For example, if a count of 12 groups is input, we determine that each group contains a span of $360/12 = 30$ degrees. If a request-agent's home location leads to an angle of

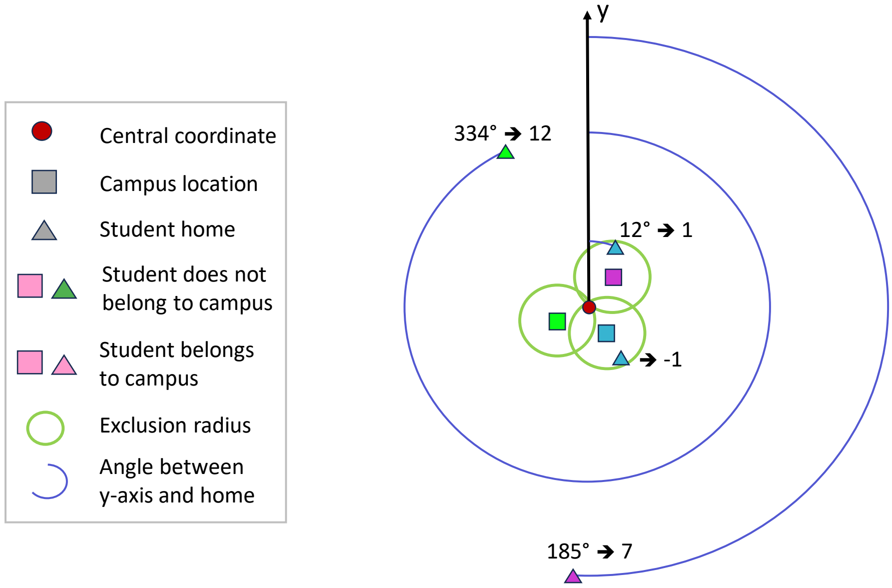


Figure 3.11: An example for assigning groups to agent requests based on home locations.

12 degrees, the request would belong to group 1. A request with a home with an angle of 185 degrees will belong to group 7, 334 degrees equals group 12 and so on.

An exception to this assignment are requests with home positions located inside the exclusion radius around the request-agent's campus that was specified in the input parameters. Such requests are assigned the group number -1 and do not exclude any matches based on their group number. We chose such an exception because including agents who live somewhat close to their university campus should in many cases be feasible without high detour times. A representation of this categorization of several home-positions can be seen in Figure 3.11. Additionally, Figure 3.12 displays an exemplary splitting of agent locations into 18 groups where agents in proximity to campus locations are excluded from this grouping and are colored black.

All existing matches of the same and adjacent group categories are then further filtered for the same Requesttype, not being fully occupied already and, similarly to ridesharing, for an overlap of time-windows. Since we did not want for it to be possible that one ride visits the same stop multiple times, we have to ensure that agents with the same campus location have overlapping time-windows for either arrival times for rides to uni or departure times for rides back home. Should a match, that is examined by our filter, not yet contain a stop at the request-agent's campus, we instead check for a possible shifted overlap of time-windows between other stops. This means that we perform the same overlap check as for ridesharing but for all stops of the match. For each stop of the examined match we check whether a time-window overlap between the stop time-window and the request's time-window exists where we once shift the request

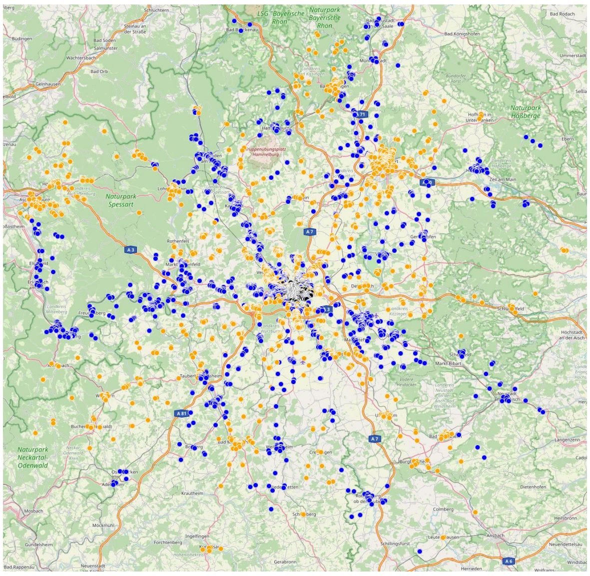


Figure 3.12: An example for assigning groups to agent requests based on home locations.

time-window forward by the time it takes to drive from the request's campus to the stop campus and once backward by the time it takes to drive from the stop campus to the request's campus. This way we try to ensure that a ridepooling solution should exist where no idle times waiting for time-windows exist during a ride. Figure 3.13 exemplifies this time shift consideration by showing an example in which an overlap exists for the time-windows A [12:30 - 13:00] and B [11:55 – 12:25] since the drive from B to A takes 5 minutes.

The filtered matches are then sorted by three standards. Most importantly, matches with the same group number as the request get priority and will be the first in the

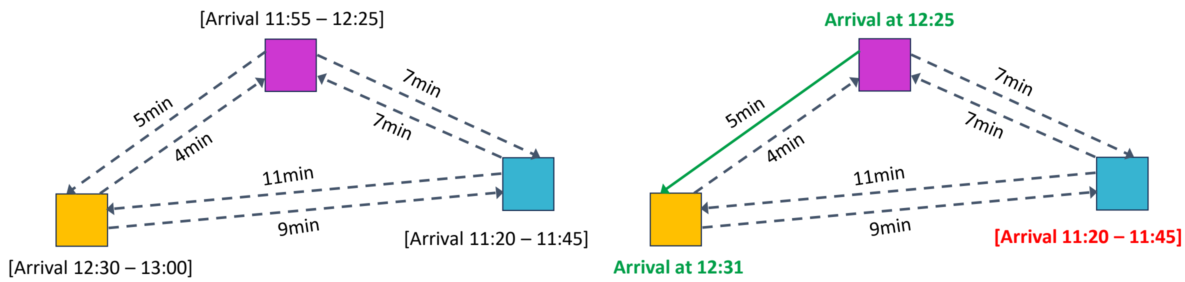


Figure 3.13: A depiction of two stops being possible in a ride even though their time-windows are not overlapping.

list. Then, we prefer matches that already stop at the request's campus. Lastly, these gradations are ordered by the closest distance of another student's home of the match to the request-agent's home. Afterwards, we take the first five matches of this sorted list and check with the help of jsprit and Graphhopper whether fitting in the request is actually feasible for each of these matches. It is feasible if the following conditions are fulfilled:

- all time constraints of all agents of the match and of the request-agent are observed
- the resulting ride would fit between the vehicle's rides before and after
- the vehicle's next ride still has to be feasible with the new ride's resulting vehicle position and end time
- no waiting times occur during the resulting ride
- the ride time is accepted by all agents of the match

Out of the feasible matches of the five examined ones we chose the match with the lowest total ride time and use this for a comparison with the best free vehicle at the time.

We determine which vehicles might be free by excluding vehicles that are already executing a ridepooling ride during the request's time-window. Furthermore, we calculate the drive time between the request-agent's home and campus and check for every not yet eliminated vehicle if it could fit this drive time into its schedule in a way that the request's time-window is observed; i.e. can the vehicle fit the drive from the agent's home to campus so that it arrives during the time-window or, in the case of a ride home, can the vehicle fit the drive from the agent's campus to their home so that we pick up the agent during the time-window?

The vehicles that pass this condition are then sorted based on each vehicle's distance between its last parking location and the request-agent's pick-up location. The vehicle with the smallest distance is favoured. The first five matches of this sorted list are, if feasible, put into a list of eligible matches of which the match with the shortest ride time is picked as best match. Should no match of the first five be feasible, we start to examine the next matches of the ordered list of filtered possible matches. The first match to be feasible according to the aforementioned conditions is picked as the best match for the request.

We then go through the ordered list of possibly free vehicles and check if the ride would actually be feasible since the vehicle still has to drive to the pick-up location which might not be possible time-wise and the resulting ride might affect the vehicle's next ride due to an adjusted vehicle position.

We then compare the first feasible free-vehicle match with the best match that was found for already existing groupings and choose the free vehicle if its distance to the pickup location is smaller than the distance between our new pickup location and the closest other pick-up location of the group match. In the case of no match being feasible for the incoming request (for example, when no other matches exist yet), we also look at ridepooling vehicles that might be free at the right time for the request.

This match finding process is done for both the request-agent's ride to and back from campus. Should there be a problem with either ride, due to all already present matches not fitting or not being feasible and no free vehicles existing, the agent will prefer to not take part in ridepooling and drive their own car.

3.5.4.3 Preparation of Data

In order to start a ridepooling simulation run with the aforementioned parameters, some data has to be prepared first yet again. This is demonstrated at the top of Figure 3.14 which depicts also the whole simulation sequence.

Firstly, the specified count of vehicles is generated with their starting position and corresponding position time being their stay at their assigned depot and the earliest possible service start. In contrast to ridesharing we do not need to split the requests into arrival and departure requests as only one request is needed to plan the ride to and the ride back from campus. Then, similarly to ridesharing, the accepted ridepooling ride times are calculated and saved for each agent based on the input function and the drive time alone. Finally, all requests are ordered by their request arrival time for the start of the event-based simulation and thus, the simulation can start.

3.5.4.4 Simulation Sequence

The ridepooling simulation sequence is depicted in Figure 3.14. At first, the occurring events of the simulation will be request arrivals for both a ride to campus and a ride back home. Each incoming request is put into our ridepooling algorithm for finding and guaranteeing a match for both rides.

If matches were found for both rides of an agent, we add said agent to both matches and recalculate the matches' start times and stop time-windows. Should the agent's campus now be the last stop for the ride to or the first stop for the ride back from uni, we also recalculate the main time-window of the affected match. The new start-time of the matches lead to a reordering of events in the simulation as this changes the actual ride starts.

As soon as a ride start event occurs, we drop the accompanying match from the active match list and calculate the actual stop times that happen during the ride as well the resulting output metrics for each agent, i.e. the travelled kilometers and minutes, as well as the emissions and costs of the ride. For each agent we compute the kilometers and minutes from their pickup to their drop-off. For the splitting of the total emissions and costs of the ride, we consider the bee-line distance between each agent's home to their campus. We then calculate the emissions for this agent as their percentual share of the

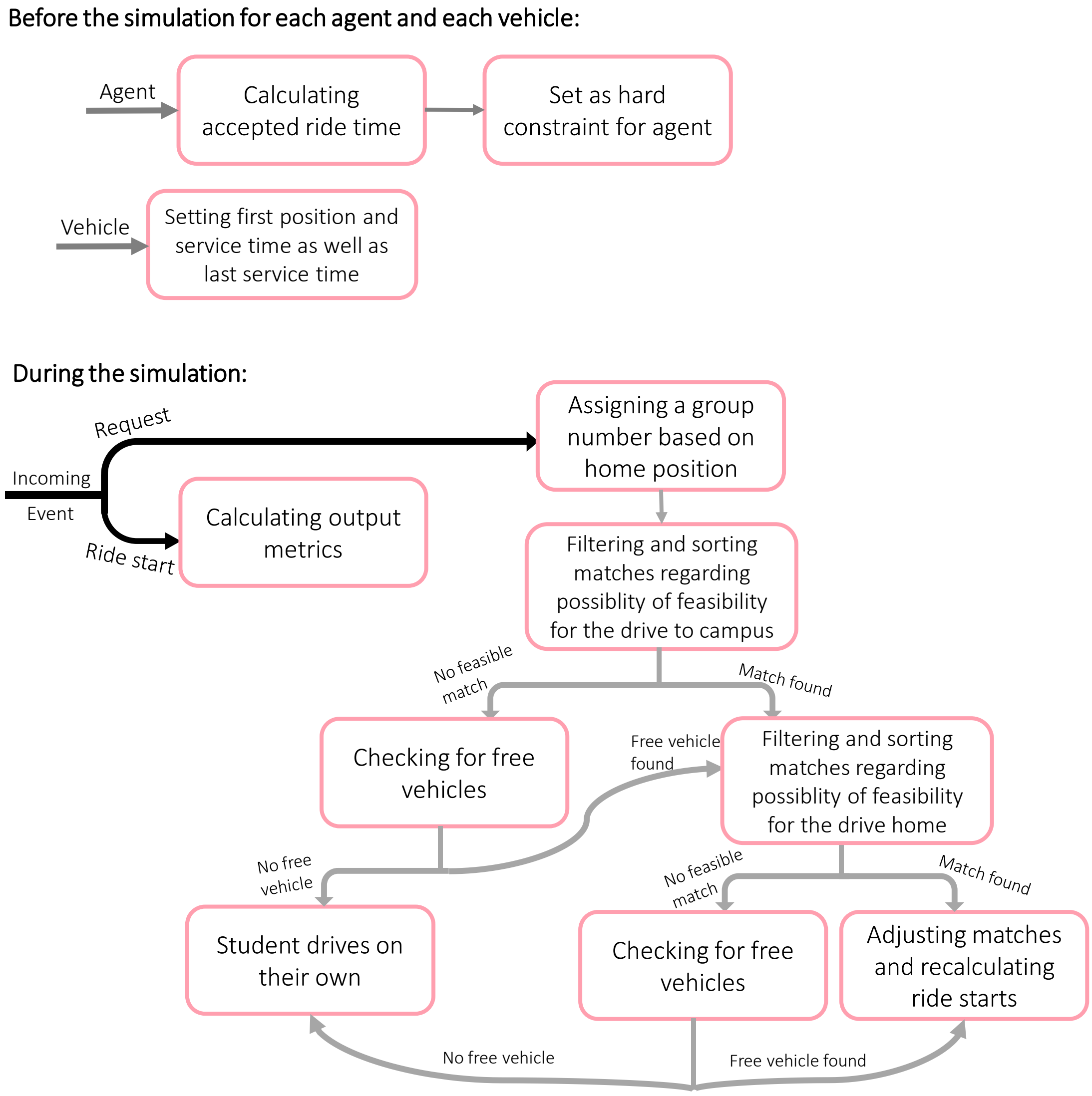


Figure 3.14: The ridepooling simulation sequence.

sum of all agents' distances multiplied with the total emissions which also include the vehicle's empty drive to the first pickup location. The same is done for the costs.

After all events have been handled in the simulation, we ensure that all agents' constraints are fulfilled in their rides.

3.6 Calculation of the Output Metrics

We calculate the results for the output metrics as follows:
Travelled kilometers per agent: Adding up the kilometers from the agent's pick-up location to the agent's drop-off location.
Travelled minutes per agent: Adding up the driven minutes from the agent's pick-up location to the agent's drop-off location and adding the stop times for every stop along the path (not for the pick up or drop-off).

Ridesharing costs for one ride per agent: $\frac{RC}{n}$

Ridesharing emissions for one ride per agent: $\frac{RE}{n}$

with RC being the total ride (gas) costs, RE being the total ride (CO_2) emissions and n the number of agents of the ride.

Ridepooling costs for one ride per agent: $\frac{CD_A}{\sum_{i=1}^{n} CD_i} \cdot RC$

Ridepooling emissions for one ride per agent: $\frac{CD_A}{\sum_{i=1}^{n} CD_i} \cdot RE$

with CD_i being the beeline distance of agent i's home to campus and A being the considered agent.

3.7 Comparison of the Mobility Mode Approaches

In order to make sensible statements about the usefulness and sustainability of ridesharing and ridepooling, we have to ensure the comparability between our implemented approaches that we presented above.

The most important aspect of this is to use the same input data for each approach. This particularly concerns the agents who must have the same home and campus locations as well as request data. Additionally, we use the same general and model configuration parameters for comparable everybodyDrives, ridesharing and ridepooling scenarios.

Furthermore, we developed our ridesharing and ridepooling approaches with comparability in mind and thus use the same tools for VRP-solving, distance and drive time calculation and created few but significant differences in addition to the basic difference between ridesharing and ridepooling.

Aspect with similarities	Explanation
Campus stops	The campus stops are defined the same
Agents	The same agents and transport demands are considered for all mobility modes
Time-windows	The same time-window constraints exist for all mobility modes
Considered costs	Only gas costs are considered for the cost calculation
Stop times	The same stop standing times are considered for stops during rides

Table 3.2: The similarities between our ridesharing and ridepooling approaches.

The Tables 3.2, 3.3 and 3.4 present an extensive overview of the similarities and differences between our ridesharing and ridepooling approaches as well as aspects where a difference between both modes depends on the parameter input.

3.8 Examined Scenario

In this work we examine if the application of ridesharing and ridepooling would achieve a success regarding the reduction of CO_2 emissions and costs for students if utilized for the campus mobility in and around Würzburg. Therefore, we choose a realistic scenario for evaluating the different approaches under the same circumstances. For the actual simulation runs we now calculate the actual distances between the students' campus and home positions and only consider students located in a specified radius. Since expecting minibuses to cover huge distances would result in wasting time driving around without any passengers and not picking up other students, we set the outer radius to 25km for a realistic scenario. The lower radius is set to 2.0 for ensuring that students are not too close to their destination campus which would make using ridesharing or ridepooling nonsensical.

We set the maximum accepted walking distance to 1200 meters for each agent as this is the distance that several German federal ministries ascribed to be the maximum sensible walkable distance [33]. In order to configure a flexible approach of students being fine with arriving some minutes earlier or later than their favoured arrival time and leaving campus even more minutes later than their favoured departure time, we set the time interval to 15 minutes. This way, if a student wishes to arrive at 10:00, they would

Aspect with differences	Ridesharing	Ridepooling
Requests	Requests are split into to-campus and home requests	One request per agent arrives which contains to-campus and home ride
Accepting of to-campus matches	Matches to campus are accepted despite matches home not being ensured	To-campus matches are only accepted if home-matches are immediately found and binding
CO_2 and cost splitting	CO_2 and costs are split evenly for a ride's agents	CO_2 and costs are split based on distance to each agent's destination
Metric calculations for rides	Ride metrics are calculated by considering the path from the first agent stop to the last agent stop	Ride metrics are calculated by considering the ride start from the vehicle's last standing position to the last agent stop
Home stops	Stops are not directly at agents' home locations	Stops are directly at agents' home locations
Rides home	Agents without a car at campus may not get a home match and be stranded at campus (lost)	Agents are guaranteed a home ride (either through ridepooling or private ride home
Empty rides	Rides can not be empty	Rides are often empty while driving to the first pickup positions

Table 3.3: The differences between our ridesharing and ridepooling approaches.

Parameter aspects	Explanation
Vehicle seat count	The seat counts of the ridepooling and student vehicle are configured with the input and can thus be the same or different
Gas consumption per km	The gas consumption per km of the ridepooling and student vehicles are configured with the input and can thus be the same or different
CO_2 emission per liter	The CO_2 emission per liter of the ridepooling and student vehicles are configured with the input and can thus be the same or different
Gas price per liter	The gas price per liter of the ridepooling and student vehicles are configured with the input and can thus be the same or different
Accepted ride times	The accepted ride times can be configured separately for ridesharing and ridepooling by inputting functions. Since the function input values is the same for both cases, the accepted ride times can be the same or different for ridesharing and ridepooling.

Table 3.4: Similarities and differences between our ridesharing and ridepooling approaches.

still be fine with 9:45 or 10:15. Similarly, if they wish to depart at 12:00, we assume that this is the earliest possible time for departure which is why a later departure up to 12:30 would still be acceptable. As already mentioned, the function for setting the accepted ride-time for each agent is "$x + \log_{1.4}(x)$" for ridesharing and "$x + \log_{1.2}(x)$" for ridepooling with x being the time for travelling alone. For both ridesharing and ridepooling the stop time during a ride is set to one minute to account for aspects like payment, choosing a place to sit and stowing luggage.

Furthermore, we use five campus locations, as five distinct locations are found in the study and choose their locations as shown in Figure 3.15 which also depicts the chosen central coordinate that needs to be set for ridepooling. Additionally, Figure 3.16 depicts the agents located in our specified radius (a distance between 2.0 and 25.0 to their campus) who are colored according to the campus they belong to. We also set the additional ridepooling input parameters count of groups to 18, radius for exclusion to 1500 meters and count of buses to 250. We chose 250 buses since we calculated the maximum amount of agents that have overlapping arrival or departure intervals and thus similar transportation times. This maximum amount is 1364 for our scenario and as we aim to fulfill most agents' travel demands we chose a number of buses that should cover most of the demand when multiple agents travel together. For example, if each bus would transport 5 students of the 1364 simultaneously, we could cover 1250 of the 1364 at the same time.

For modeling the student vehicles we used the model VW Golf 2.0 TDI (year of manufacture 2010)[5] with 5 seats total and a consumption of 0.048 liters gas per km, 2625 grams of CO_2 emissions per liter gas and a price of 1.719 euros per liter gas. For the minibuses we used the model Citroen Grand C4 Picasso (year of manufacture 2018)[6] with 6 passenger seats and a consumption of 0.038 liters gas per km, about 2578 grams of CO_2 emissions per liter gas and a price of 1.719 euros per liter gas.

[5]https://www.adac.de/rund-ums-fahrzeug/autokatalog/marken-modelle/vw/golf/vi/270771/#technische-daten

[6]https://www.adac.de/rund-ums-fahrzeug/autokatalog/marken-modelle/citroen/c4-picasso/2generation-facelift/264035/#technische-daten

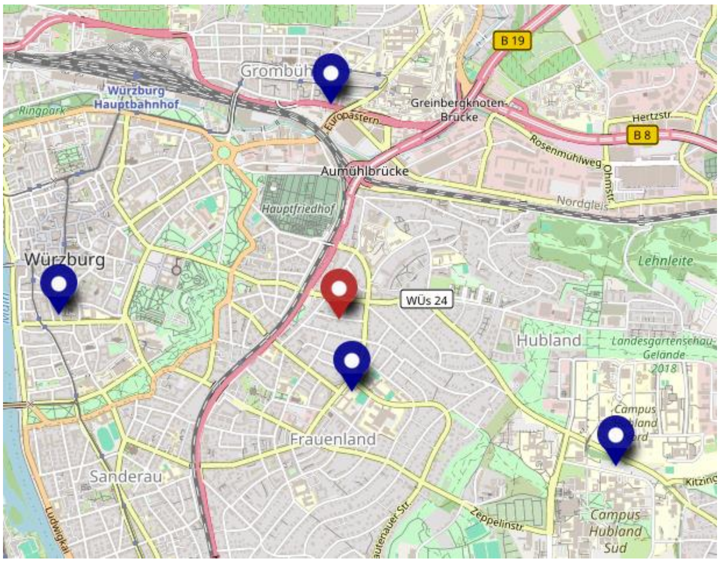


Figure 3.15: The used campus locations in blue and the central coordinate for ridepooling in red.

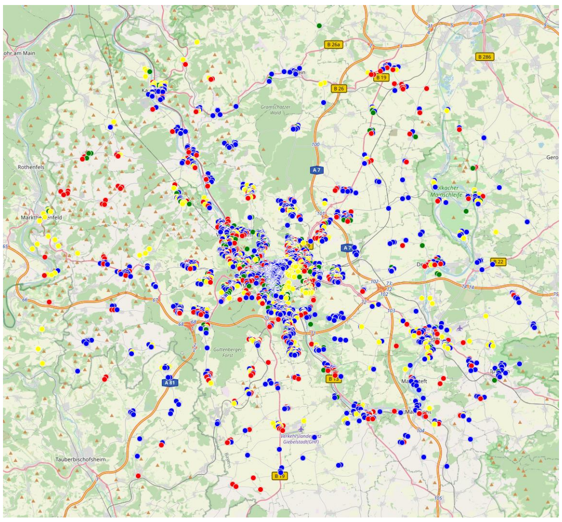


Figure 3.16: Agents considered in our scenario.

4 Evaluation

In this chapter we present the evaluations for the aforementioned realistic scenario for the mobility modes everybodyDrives, ridesharing and ridepooling and discuss and compare these results. This entails the results for the output metrics that are travelled kilometers and minutes as well as cost and CO_2 emissions. These output metrics are split into output for the rides to campus, the rides home and the sum of both (i.e. per day). Additionally, we demonstrate the possible reasons for the results with the help of more output values such as the average number of occupied seats per ride or the average amount of stops. We further address the characteristics of students who are categorized as lost during ridesharing and validate the consistent nature of our results.

4.1 Effects of Ridesharing and Ridepooling

In order to examine the output metrics' results, we analyze them for the travels to campus and back separately as well as their sum for both rides. We do this both for all agents separately and for the daily total sum of the metrics.

4.1.1 Daily Travelled Kilometers and Minutes of the Agents

Figure 4.1 shows the distribution of the travelled kilometers per day for all students as violin plots. The baseline mobility mode (everybodyDives) is colored in blue, ridepooling in pink and ridesharing in green. Additionally, the quartiles and the mean value are displayed in each violin plot. We can see that both ridepooling and ridesharing do not have a big effect on the distances covered by each agent with a mean of about 18km a day for everybodyDrives to about 19km for ridesharing and 23km for ridepooling. Furthermore, all quartiles are slightly higher for ridepooling than for ridesharing which both have higher quartiles than everybodyDrives. Thus, we conclude that ridepooling leads to slightly more kilometers travelled per agent than ridesharing. This is due to the detours that are added for picking up or dropping off other students. Since we set agents to be more accepting of longer ride times for ridepooling than for ridesharing, ridesharing leads to less or shorter detours as we can conclude by the lower mean for travelled kilometers.

The travelled minutes per day for all students, shown in Figure 4.2, behave similarly to the travelled kilometers since more kilometers typically mean more minutes travelled.

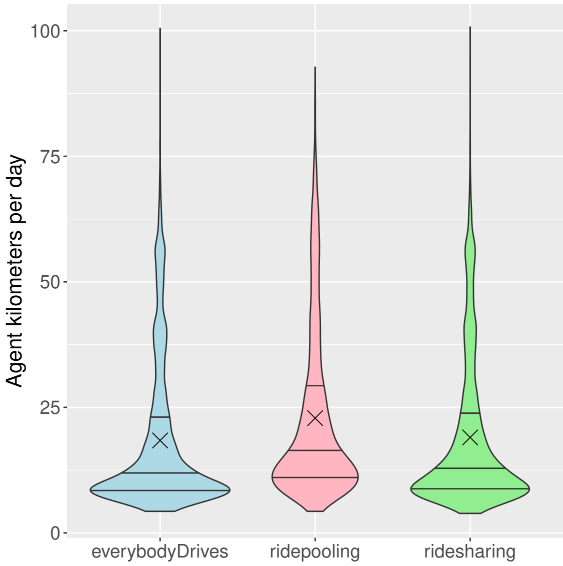


Figure 4.1: The distribution of the daily travelled kilometers of the agents.

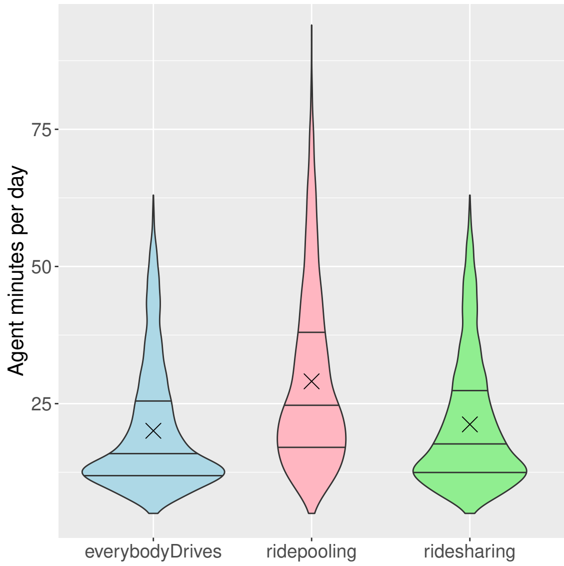
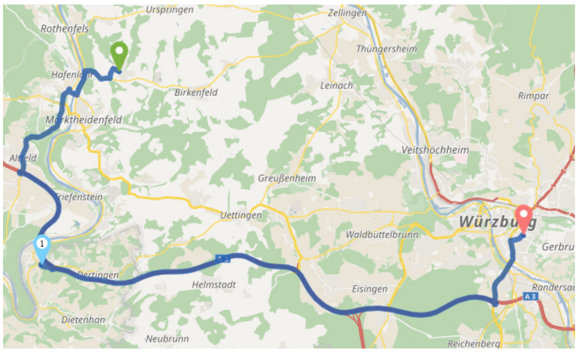


Figure 4.2: The distribution of the daily travelled minutes of the agents.

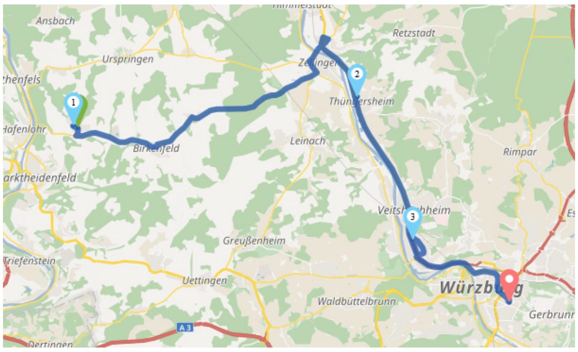
However, the effects of ridepooling look to be more drastic for the travelled minutes which might be because of more detours being accepted in ridepooling which leads to more stops experienced by passengers. Each stop adds more time to the ride (in our case 1 minute) whereas the travelled kilometers are unaffected.

Furthermore, we can see that there are some agents who live far away and travel less kilometers when ridepooling is used than in the baseline scenario while also not having shorter travel times. This is unexpected since ridepooling adds detours to agents' rides by picking up more passengers. When looking into the case for an agent that travels nearly 100km a day when driving alone, we discover that this agent chooses to use the freeway when travelling alone, as it is the faster option, and is forced to drive through more rural areas when using ridepooling as other agents are picked up in these rural areas during the ride. Figure 4.3 displays this agent's case in which the agent drives the quicker but longer route when possible and rides through the slower but shorter path when grouped with other people. Therefore, this is yet another effect of ridepooling for reducing travelled kilometers and thus also emissions and costs.

Similarly, there are cases for ridesharing where an agent might have a lower travel time than in the baseline scenario. This is due to such an agent being a passenger and walking to (or from) the driver's house which is located closer to the passenger's campus. As we do not consider the kilometers the passengers travel by foot since they are limited and do not produce any emissions or costs, the output metric travelled kilometers is reduced for this agent. Figure 4.4 demonstrates this effect for a specific case in our results.

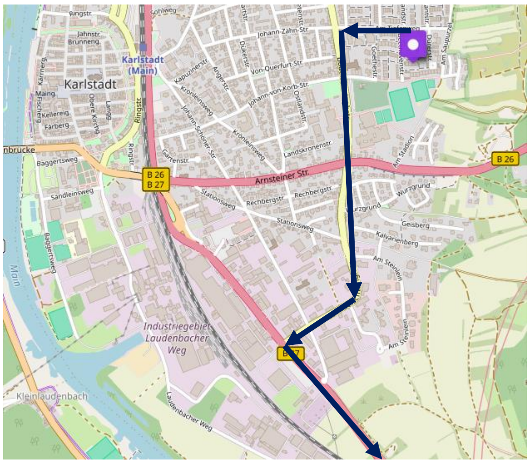


(a) Path of agent when travelling alone by private car.

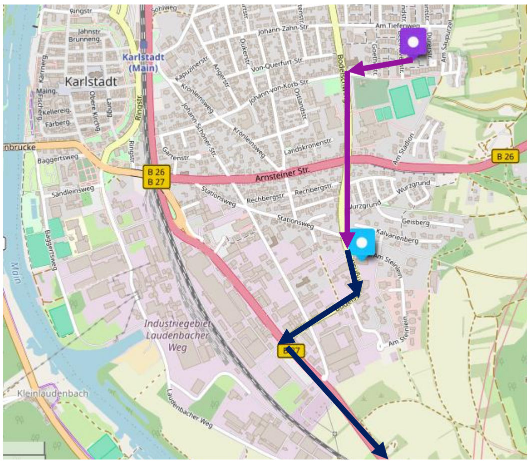


(b) Path of agent when using ridepooling.

Figure 4.3: An agent preferring a path via freeway in the baseline scenario (left) and being forced to drive through more rural areas when ridepooling is used (right).



(a) Path of agent when travelling alone by private car.



(b) Path of agent when using ridesharing and being a passenger.

Figure 4.4: An agent (violet) walks to the driver (blue) of a ridesharing match and then travels less kilometers by car.

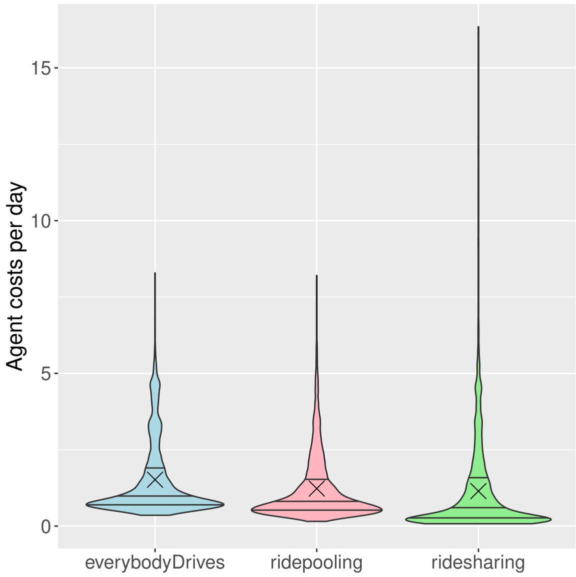


Figure 4.5: The daily gas costs (€) for the agents.

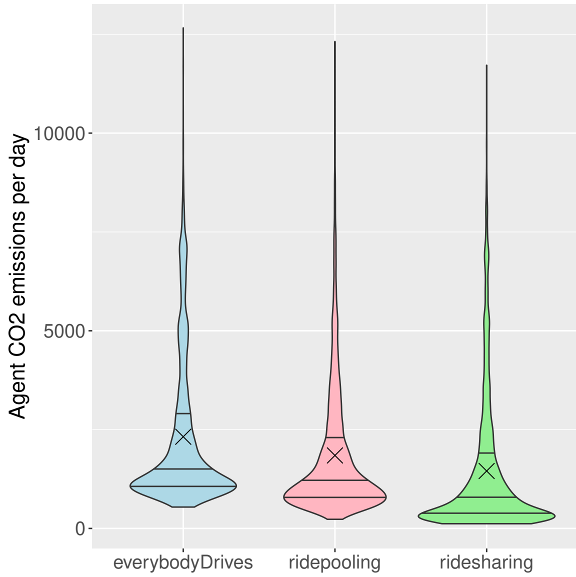


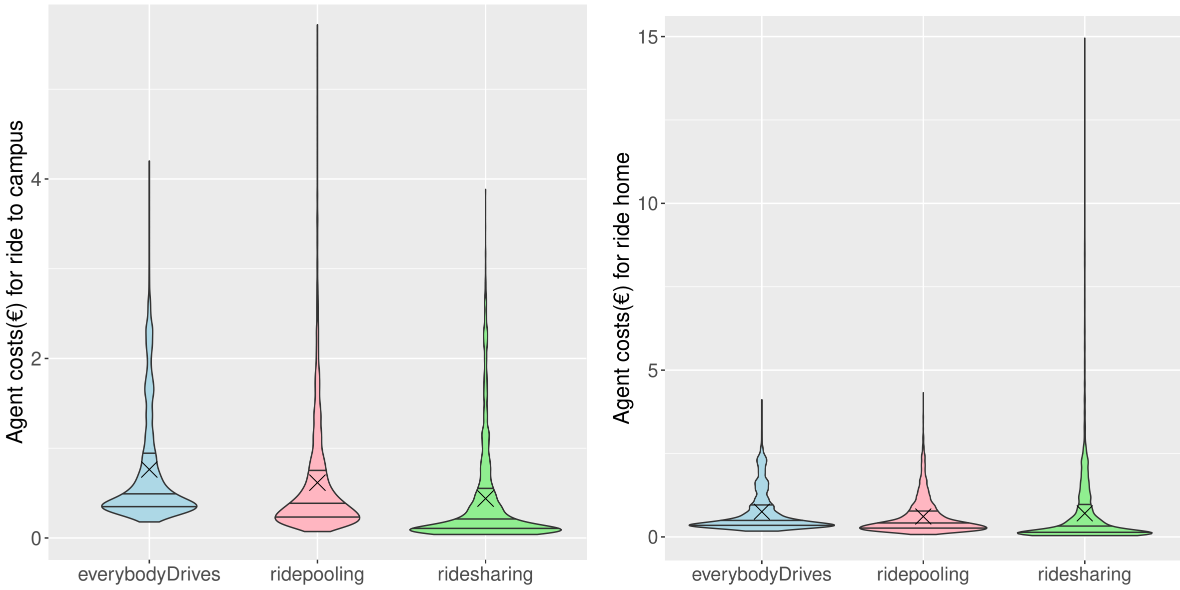
Figure 4.6: The daily CO_2 emissions (g) for the agents.

4.1.2 Daily Costs and Emissions for the Agents

When examining the cost per day distribution for all agents, shown in Figure 4.5, we notice that ridesharing results in similar cost savings per agent as ridepooling and that, generally, both modes result in less costs for the agents in our evaluated scenario.

However, both mobility modes leads to some extreme outliers where an agent has to pay a lot more than when travelling by themselves. This becomes especially evident when looking at the costs for a single ride per agent, displayed in Figure 4.7. The ridesharing cost outliers for the home rides can be attributed to the lost agents. As for ridepooling, we can see that especially the one-way rides are affected by the outliers since we only encounter either an agent's ride to campus or the ride back being affected in our ridepooling simulation run. For example, an affected agent pays a lot more for the ride to campus and then pays a smaller amount for the ride home. This way, the cost sum for the agent is not influenced as drastically as the costs of the affected one-way ride.

We ascribe these cost outliers to two factors. Firstly, the agents of a ride have to account for the empty ride to the first pick-up position (in the likely case of the ride-pooling vehicle not already being located at this position). Should an agent happen to be the sole passenger of a ride, they would consequently have to pay the gas costs for more kilometers than when travelling alone. If we executed ridepooling with vehicles that had the same emissions per kilometer as the student vehicles, this would result in our approach needing at least two agents per ridepooling ride to be profitable.



(a) Costs for the ride to campus per agent.

(b) Costs for the ride home per agent.

Figure 4.7: The costs for one-way rides where huge outliers are visible.

Secondly, even if multiple people share a ridepooling ride, the result may still not be profitable. This is due to our method of splitting the costs between the agents of a ride since agents who live farther away from their campus have to pay a bigger portion of the cost sum. Consequently, an agent who lives much farther away than the other passengers of a shared ride has to pay most of the resulting costs. We deem this fair as this agent also contributed the most to said costs. Nonetheless, in the worst case this can lead to the explained costly outliers even with more sustainable ridepooling vehicles.

Since our method for calculating the CO_2 emissions for each agent of a ride is the same as for the costs, we encounter the same results as can be seen in Figure 4.6. While the average CO_2 reduction per agent is significant for ridepooling, there are outliers to which we assign huge CO_2 increases due to them either using a ridepooling ride alone or living much farther away than the other passengers.

This problem could be addressed in future work by either developing a new cost splitting method or by optimizing matches for each agent's resulting costs and CO_2 share which could be done by pooling people with similar distances to campus together.

To summarize, in our evaluated scenario ridesharing leads to higher reductions of CO_2 emissions and costs for the students than ridepooling while also showing better (i.e. lower) results for the Quality of Service indicator of travelled minutes.

4.1.3 Total Differences between the Mobility Modes

Table 4.1 depicts the total output metrics for the modes. For ridesharing we present both the results that include the alternative home rides and results without. We can see that

Metric	EverybodyDrives	Ridesharing	Ridepooling
Total travelled kilometers	120103.4	75249.89 incl. alt. rides, 69419.88 excl. alt. rides	123115.9
Total travelled minutes	130833	78651 incl. alt. rides, 72265 excl. alt. rides	149273
Total CO_2 emissions (g)	15133034	7405582 incl. alt. rides, 6671001 excl. alt. rides	12324333
Total gas costs (€)	9909.976	6117.555 incl. alt. rides, 4368.553 excl. alt. rides	8201.169
Total empty covered distances (km)	0	0	45792.57
Total to-campus detour distances (km)	0	1658.512	14232.87

Table 4.1: The output metrics for the mobility modes in sum.

while ridepooling increases the driven kilometers and minutes, the emissions and costs get reduced. Table 4.2 displays additional output metrics for the modes like the average seat count which shows that ridepooling leads to larger groupings than ridesharing.

We further investigate the metric results for the ridepooling minibuses. Figures 4.8 and 4.9 depict the daily travelled kilometers per minibus and the daily travelled minutes per minibus, respectively. We can see that there is a wide range of vehicle activity with some vehicles travelling only 150km a day and some up to 770km. Similarly, the active ride times range from 230 up to 930 minutes (i.e. 3.8 to 15.5 hours) per minibus.

Figures 4.10 and 4.11 show very similar distributions for the daily gas costs and CO_2 emissions as for the travelled kilometers and minutes. This is due to the costs and emissions linearly depending on the travelled kilometers.

The aforementioned range of travelled kilometers and minutes as well as costs and emissions for the vehicles is a result of some ridepooling vehicles servicing more rides than others and some vehicles being more idle and having longer standing times as depicted in Figures 4.12 and 4.13.

This is due to our approach not balancing vehicle utilization. Furthermore, we prefer vehicles with the smallest distance to a new stop location. This leads to vehicles with remote positions (e.g., due to taking students home to their remote home locations) not being assigned to further matches. This could be solved in future work by either

Metric	EverybodyDrives	Ridesharing	Ridepooling
Number of agents	6524	6524	6524
Number of lost students	0	653	0
Number of driving students	6524	2856	466
Number of rides	13048	5712	4574
Number of rides with > 1 students	0	2898	3166
Average seat count	1	≈2.17	≈2.85
Average seat count to campus	1	≈2.28	≈2.81
Average seat count home	1	≈2.06	≈2.9

Table 4.2: The additional output metrics for the mobility modes.

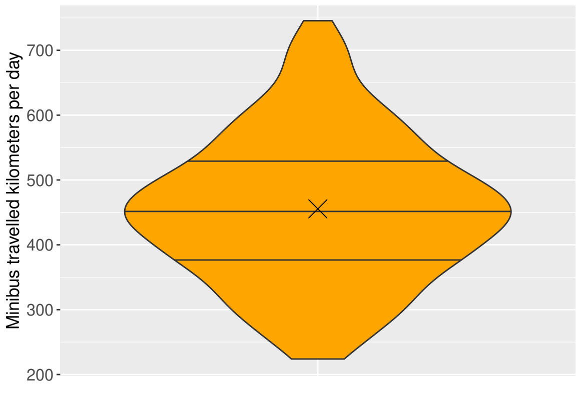


Figure 4.8: The daily total travelled kilometers for the ridepooling vehicles.

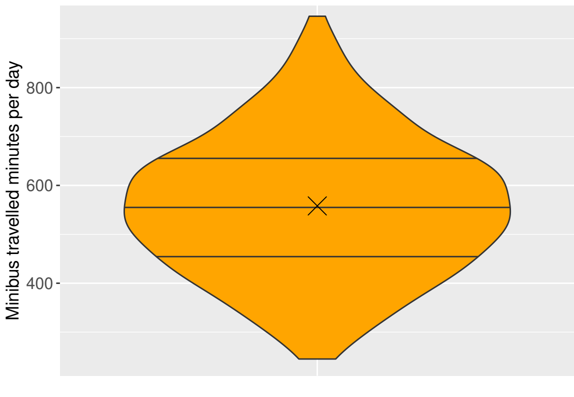


Figure 4.9: The daily total travelled minutes for the ridepooling vehicles.

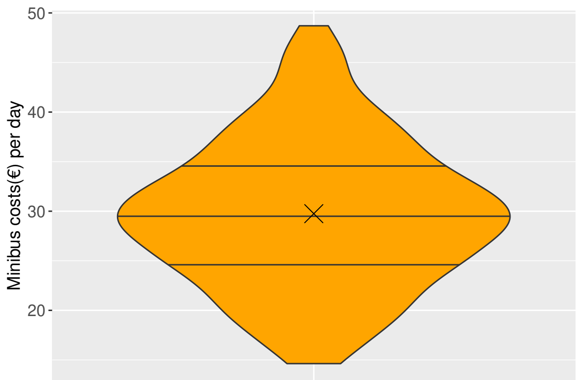


Figure 4.10: The daily total gas costs (€) for the ridepooling vehicles.

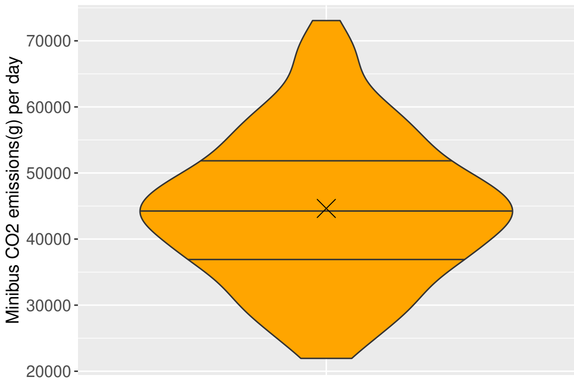


Figure 4.11: The daily total CO_2 emissions (g) for the ridepooling vehicles.

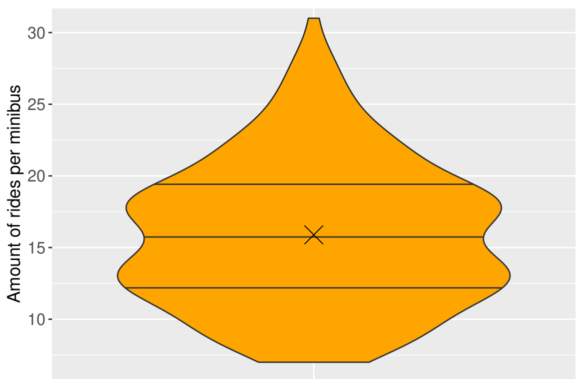


Figure 4.12: The amount of rides per ridepooling vehicle in a day.

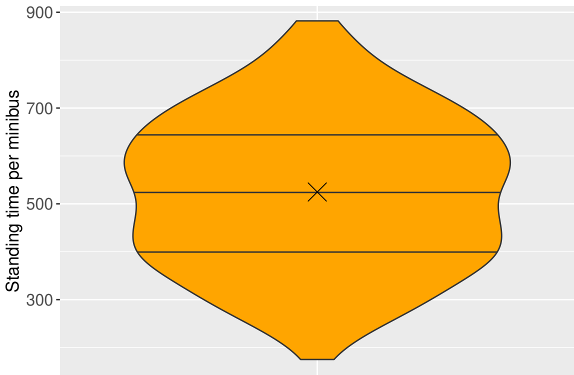


Figure 4.13: Total idle standing time (minutes) per ridepooling vehicle in a day.

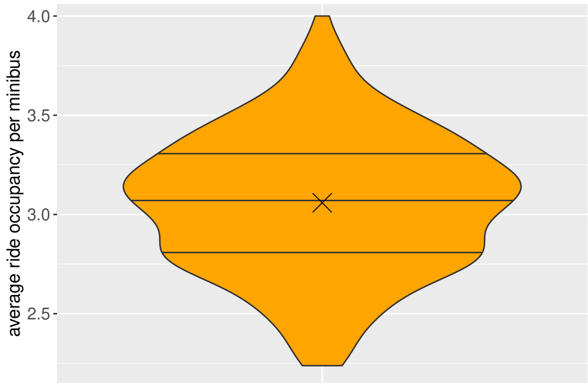


Figure 4.14: The daily average seat occupancy for the ridepooling vehicles.

favouring vehicles with longer idle times or repositioning remotely located vehicles and summoning them back to the depot, for example.

Another indicator for our vehicle utilization not being balanced is the average ride occupancy per ridepooling vehicle. As we can see in Figure 4.14, most vehicles have an average ride occupancy between 2.8 and 3.3 passengers. However, some vehicles only transport 2 passengers on average while other vehicles have an average occupancy of 4. This is again due to our approach not considering distributing passengers between vehicles evenly and instead aiming for shorter travel times for the students.

4.2 The Influence of Vehicle Choice

As we mentioned above, the effect of ridepooling depends on the difference of sustainability between the student vehicle and the ridepooling vehicle. This is due to our method of calculating the costs and emissions as said relationships are solely linearly dependent on the travelled kilometers.

Since ridepooling entails vehicles covering distances unoccupied, we can not just sum up the travelled kilometers with passengers. Instead, we have to sum up the total travelled ridepooling vehicle kilometers and further add the distances travelled by agents who drove in their own car. The formula for calculating the total kilometers per day is thus $D_{RV} \cdot y + D_{SV} \cdot x$ for ridepooling with x as the student vehicle emissions per km and y as the ridepooling vehicle emissions per km as well as D_{RV} as travelled ridepooling vehicle distance and D_{SV} as travelled student vehicle distance. The formula for calculating the total kilometers per day for everybodyDrives is $D_{ED} \cdot x$ with D_{ED} as total distance travelled in everybodyDrives.

4.2.1 Break Even Point

With this we can calculate how much more sustainable the ridepooling vehicles have to be in order to produce the same or less emissions or gas costs. We equate the ridepooling and everybodyDrives total kilometer formulas with a less-than-or-equal relation and can this way determine the maximum emissions per kilometer for the ridepooling vehicles.

$$
\begin{aligned}
D_{RV} \cdot y + D_{SV} \cdot x &\leq D_{ED}x \\
D_{RV} \cdot y &\leq D_{ED} \cdot x - D_{SV} \cdot x \\
D_{RV} \cdot y &\leq (D_{ED} - D_{SV}) \cdot x \\
y &\leq \frac{(D_{ED} - D_{SV}) \cdot x}{D_{RV}}
\end{aligned}
$$

This means that the emissions per kilometer for the ridepooling vehicle need to be smaller or equal to the difference between the total student car emissions in both modes divided by the total travelled kilometers of the ridepooling vehicles in order to not create more emissions.

Now we can apply this to our evaluated scenario. As previously explained in Subsection 4.1.3, we have a total of 113866.7 kilometers travelled by the ridepooling vehicles (D_{RV}) and 9249.155 kilometers travelled by agents in their car (D_{SV}). The total covered distance for the everybodyDrives mode, all travelled by student car, is 120103.4 kilometers (D_{ED}).

Inserting this into our equation we obtain

$$
y \leq \frac{(120103.4 - 9249.155) \cdot x}{113866.7}
$$

$$
y \leq 0.973544 \cdot x
$$

as a result. Consequently, the ridepooling vehicle needs to have a maximum of 97.3544% of the CO_2 emissions per kilometers of the student vehicle.

In our specific case we set the student vehicle to have a consumption of 0.048 liters per kilometer and emissions of 2625 grams of CO_2 per liter. This equates to 126 grams of CO_2 per kilometer which is our x. Inserting this into the equation

$$
\begin{aligned}
y &\leq 0.973544 \cdot 126 \\
y &\leq 122.666544
\end{aligned}
$$

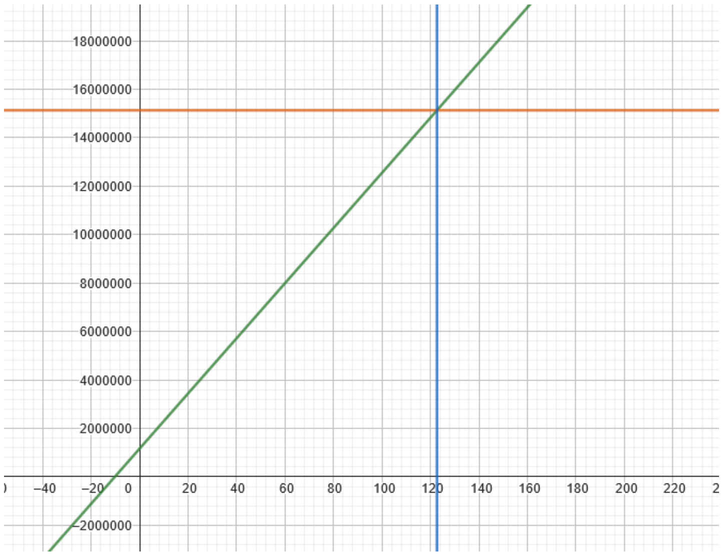


Figure 4.15: The function $f(x) = 113866.7x + 9249.155 \cdot 126$ (green) breaks even at $x = 122.666544$ (blue) where it reaches the total amount of CO_2 emissions in everybodyDrives, namely 15133028.4 grams of CO_2 (orange).

shows us that ridepooling will not produce more CO_2 than everybodyDrives in our scenario as long as the ridepooling vehicles have a CO_2 per kilometers rate of 122.666544 at most.

This point of breaking even is further demonstrated in Figure 4.15 which displays the function $f(x) = 113866.7x + 9249.155 \cdot 126$.

4.2.2 Possible Utilization of Electric Vehicles For Ridepooling

In order to make the ridepooling mode even more sustainable, electric vehicles could be used. Therefore, we briefly examine how much CO_2 could be saved when using electric minibuses for transporting students. The German Federal Ministry for the Environment described that electric vehicles also produce CO_2 emissions due to the production and provision of electricity [34] that is not yet fully sustainable as long as ecological electricity is not used. Using the German energy mix, the ministry states that a typical electric vehicle leads to about 70 grams of CO_2 per driven km (only considering energy provision and not production or maintenance).

With this we can calculate the total CO_2 emissions for ridepooling with electric vehicles (while student vehicles still use gas) for our scenario.

$$113866.7 \cdot 70 + 9249.155 \cdot 126 = 9136062.53$$

These resulting 9136062.53 grams of CO_2 are only about 60.37% of the 15133034 grams for the everybodyDrives mode and thus a considerable reduction of emissions. If

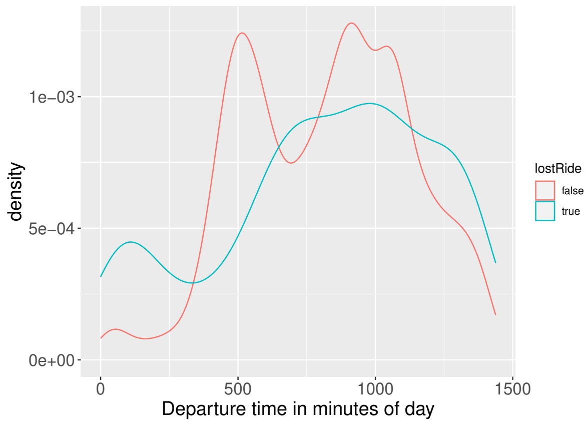


Figure 4.16: The kernel densities for the departure times (displayed in minutes of day) for the agents who end up lost and the agents that do not.

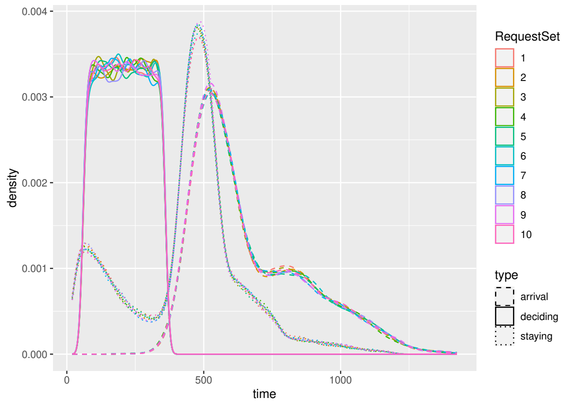
we were to use ecological energy for the ridepooling vehicles, the reduction would be even greater.

4.3 Lost Students

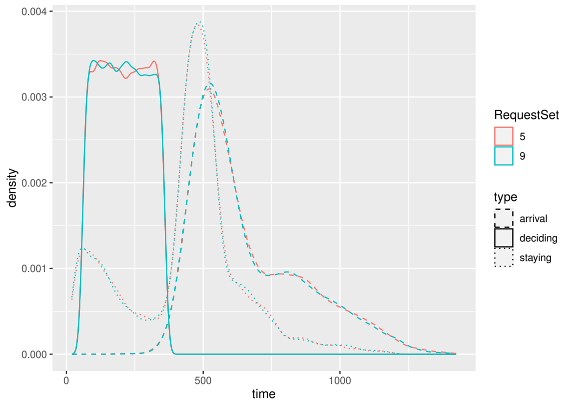
While our ridesharing approach shows the best results for CO_2 emissions and costs on average, some students have to face the consequences of being lost and then paying more than thrice for their ride home. This makes the approach as a whole more unattractive. Thus, we examine if these lost students of our scenario have certain characteristics in common in order to exclude such students from ridesharing in the future and possibly reduce the percentage of lost students.

When we compare the departure times of agents categorized as lost to the other agents, we can see that lost agents are more likely to want to leave early in the morning or late in the evening as shown in Figure 4.16. Since there will not be many other agents wanting to drive home at these times, the likelyhood of being lost will be higher. Therefore, in a real-world application of our ridesharing approach we would recommend agents who plan on leaving either early or late not to use ridesharing.

Nevertheless, we can see that most lost agents want to depart at similar times than other agents. Consequently, the departure times can not be the only reason leading ot agents being lost. This could be examined more closely in future work in order to assess if agents that end up lost can be accurately predicted before a simulation run.



(a) Kernel densities for the request times grouped by ten different request sets.



(b) Kernel densities for the request times displayed for the two request sets with the most statistical differences.

Figure 4.17: Kernel densities for the request times.

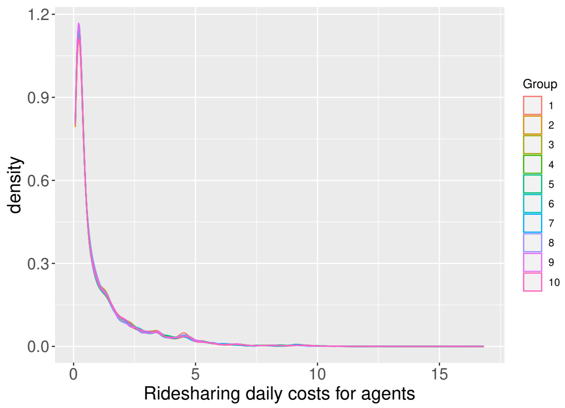
4.4 Consistency of Results

In order to validate that our results for the mode choices are of a consistent nature and do not vary greatly for different agent specifications, we also evaluate other input configurations. Firstly, we examine whether the demand data of agents is crucial for the results of the mobility modes. Then, we create different agent home distributions to check whether the effects of our mobility modes depend on location.

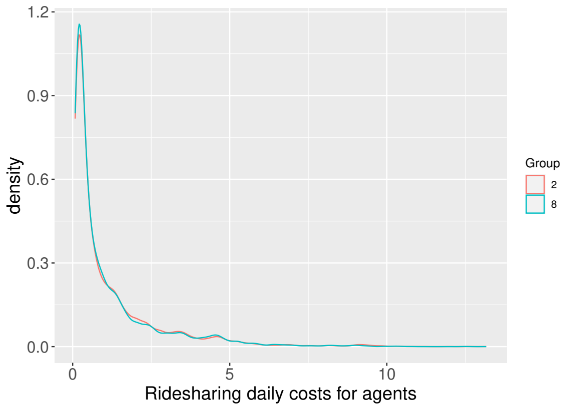
4.4.1 Evaluation for Two Work Weeks

For the comparison between multiple days of campus commute we create nine more possible request sets in which each request entails an arrival, departure and request arrival time. We do this as described in Section 3.3, i.e. we use the same probability distribution as before that is based on the MID survey and where the departure time depends on the arrival time. We then create nine agent-request sets by yet again assigning each agent one request randomly for all nine generated request sets. Therefore, each agent-request set contains the same agents with the same home location; Only the time aspect of their demand is different.

Together with our first agent-request set that we used for evaluating our scenario, we now have ten such sets that could represent two weeks of commuting to campus and back, i.e . two times Monday to Friday. This could in turn represent one week in two semesters each. Figure 4.17 demonstrates that the time distributions of all ten days are very similar.



(a) Ridesharing kernel densities for the daily costs for the agents grouped by ten different request sets.



(b) Ridesharing kernel densities for the daily costs for the agents displayed for the two request sets with the most statistical differences.

Figure 4.18: Ridesharing kernel densities for the daily costs for the agents.

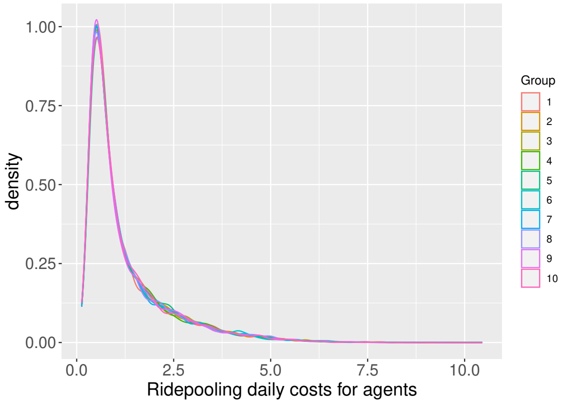
Still, as the requests containing this time information are assigned randomly to the agents, different demand data could still lead to different results since groupings of agents for travelling together are time-dependent.

However, our evaluations of all ten days show that the differences for both ridesharing and ridepooling effects are negligible between the ten days. In Figure 4.18 we can see the kernel density of the values of the daily costs for agents for the mobility mode ridesharing. There we can observe in the left Subfigure that each day's distribution of daily costs look very similar. On the right we demonstrate that even the two days with the biggest differences in their cost distributions (according to t-tests) seem to have nearly identical cost distributions.

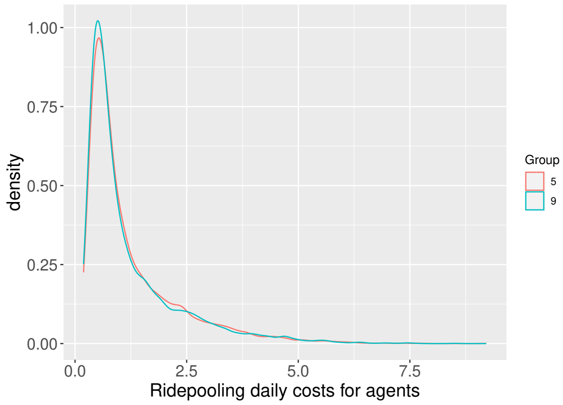
Similarly, ridepooling shows no significant differences for the daily costs for agents between the ten days. Figure 4.19 visualizes the kernel densities of all daily cost values for all days on the left and the two densities for the two days with the biggest difference where we can yet again see that these densities look nearly identical.

Another crucial metric is the number of agents per ride as this metric is indicative for the success of the mobility modes. Looking at Figures 4.20 and 4.21, which display how many rides have x occupied seats for ridesharing and ridepooling respectively, we can conclude that this metric also behaves similarly for the ten days.

When examining all of the above-mentioned and multiple more metrics through an anova, we verify that the ten days, i.e. ten different agent-request sets, show no statistically significant differences. Thus, we confirm that both our ridesharing and our

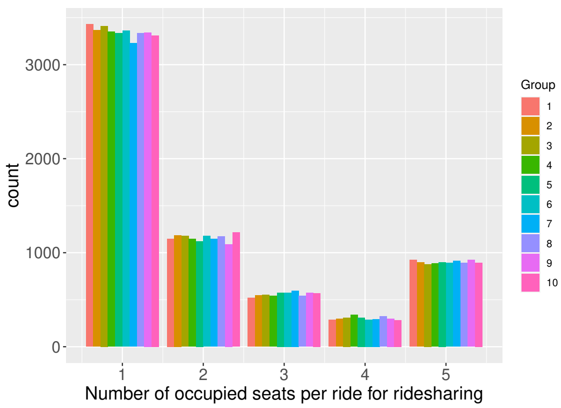


(a) Ridepooling kernel densities for the daily costs for the agents grouped by ten different request sets.



(b) Ridepooling kernel densities for the daily costs for the agents displayed for the two request sets with the most statistical differences.

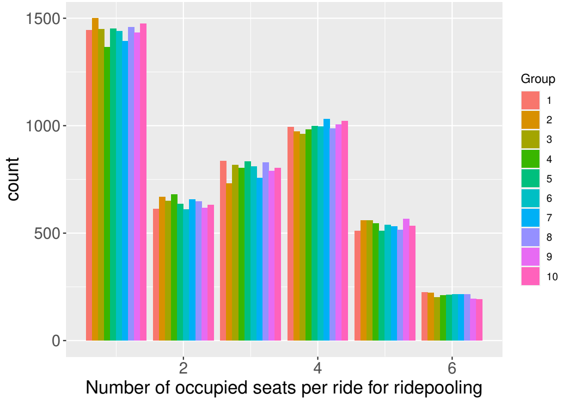
Figure 4.19: Ridepooling kernel densities for the daily costs for the agents.



(a) Number of occupied seats per ride of a day for ridesharing grouped by all request sets.

(b) Number of occupied seats per ride of a day for ridesharing for the two request sets with the biggest statistical differences.

Figure 4.20: Number of occupied seats per ride of a day for ridesharing.



(a) Number of occupied seats per ride of a day for ridepooling grouped by all request sets.

(b) Number of occupied seats per ride of a day for ridepooling for the two request sets with the biggest statistical differences.

Figure 4.21: Number of occupied seats per ride of a day for ridepooling.

ridepooling approach produce consistent results for demand with different time information.

4.4.2 Different Agent Home Locations

After verifying that our approach is stable regarding different mobility demand timewise, we now also want to examine whether different origin destination pairs lead to different effects of the mobility modes. This way we want to ensure that our observation of CO_2 and cost reductions are not a result of chance.

For this we utilize the tool OMOD once more and generate four additional sets of agent locations that reflect the zipcode home location distribution of the university students in Würzburg. These five different home location sets are shown in Figure 4.22 where the first one is the set we used for the evaluation of our scenario. Since all home location sets are based on the same zipcode, they look roughly the same. However, inside each zipcode the agents' home locations are placed noticeably different.

While we obtain different results for the all four main output metrics, we do not find this surprising since different home locations naturally lead to different travelled kilometers. However, the crucial question that we need to answer is whether ridesharing and ridepooling have the same effects on the output metrics, i.e. if all home-location sets experience the same degree of reduction of CO_2 emissions and costs.

To evaluate this, we calculate for all agents how the costs and emissions developed from everybodyDrives to both ridesharing and ridepooling and do this for all five home-location sets. This way, we can assess if a significant difference exists between the five home distributions regarding the effects of ridesharing and ridepooling. Using an anova,

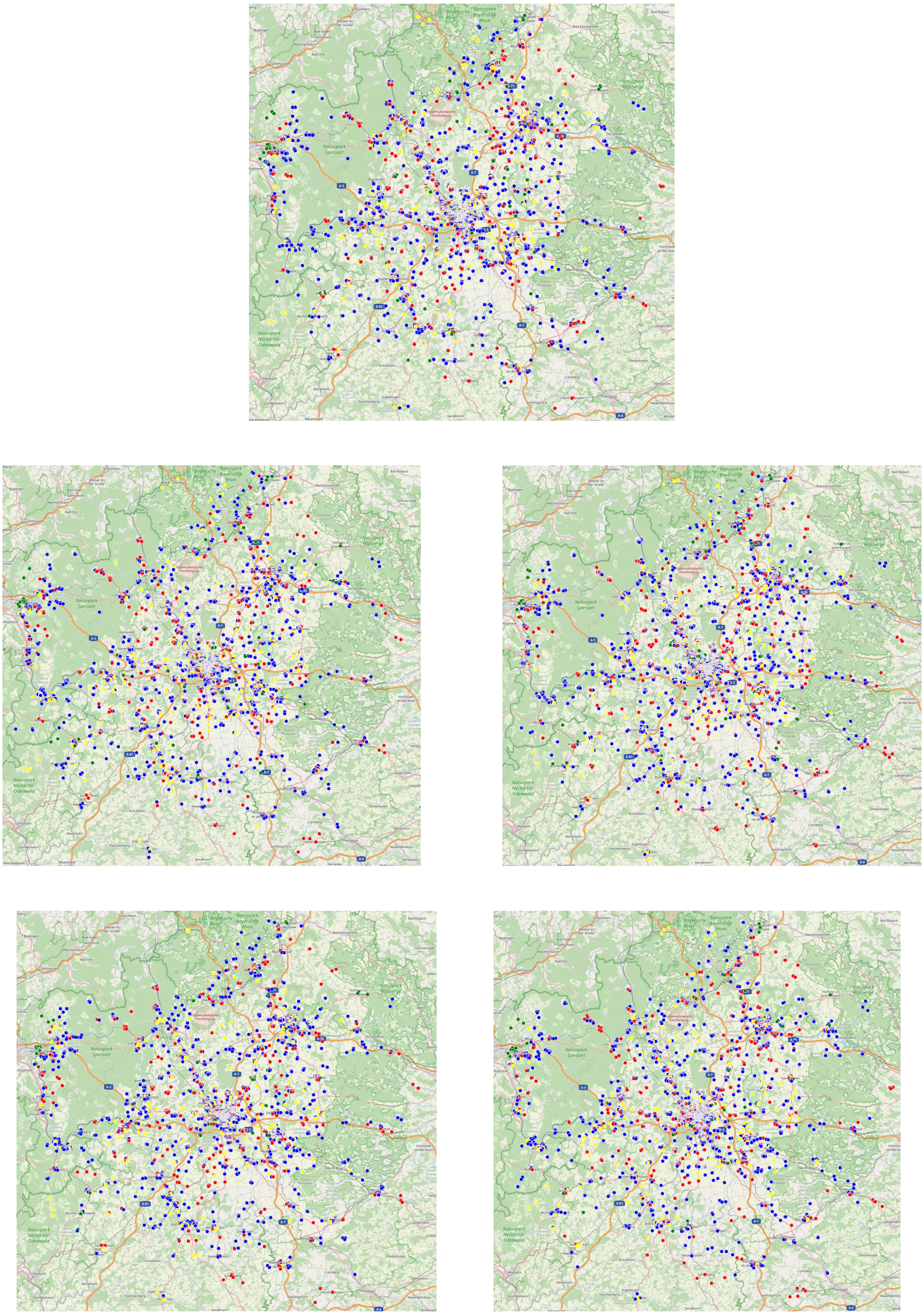
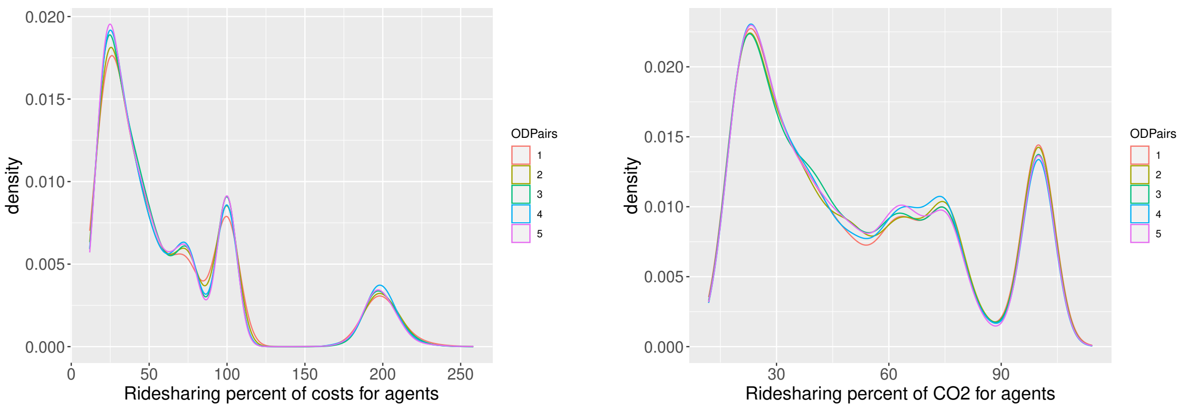


Figure 4.22: The five different home distriutions for the agents with our utilized one at the top.

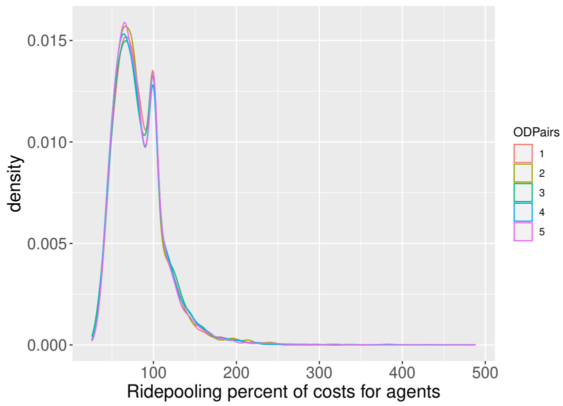


(a) The kernel densities of the ridesharing effect on agent costs in percent.

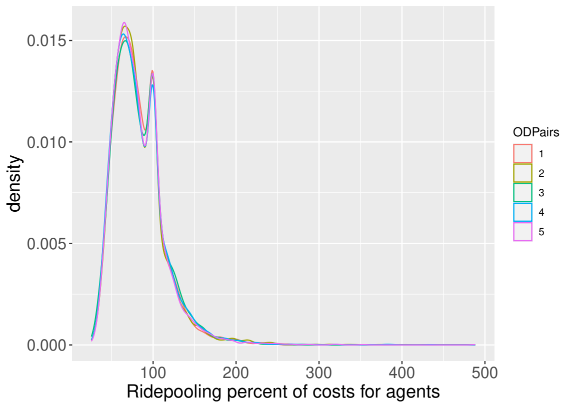
(b) The kernel densities of the ridesharing effect on agent CO_2 emissions in percent.

Figure 4.23: The effect of ridesharing compared for all five home disributions.

we learn that there is indeed no statistically significant difference for the reductions of CO_2 and costs per agent between the home distributions which is further demonstrated in Figures 4.23, 4.24 and 4.25.



(a) The kernel densities of the ridepooling effect on agent costs in percent.



(b) The kernel densities of the ridepooling effect on agent CO_2 emissions in percent.

Figure 4.24: The effect of ridepooling compared for all five home disributions.

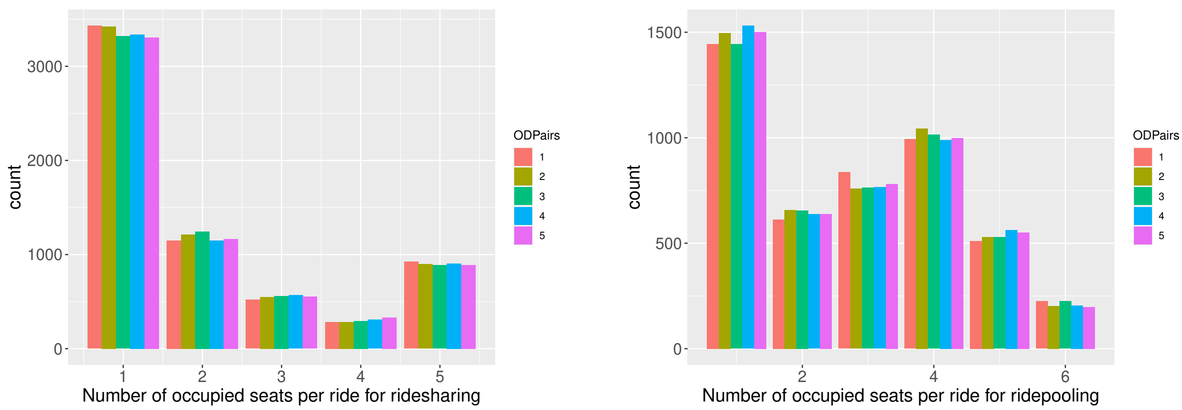


Figure 4.25: The distribution of ride occupancy for ridesharing (left) and ridepoling (right) compared for all five home distributions.

5 Sensitivity Analysis

This chapter entails the analysis of the influence of input parameters on the mobility mode results. We investigate the influence of both mode specific parameters and general parameters.

5.1 General Influence Factors

In this chapter we examine input parameters that influence multiple mobility modes such as the distance based radius in which agents are considered for our simulations.

5.1.1 Radius

Naturally, the outer radius, that is input as a cut-off point for which agents should be considered based on their home distance to their campus, significantly affects all output metrics for all mobility modes. This is because of the population of agents being changed for every non-negligible change of radius. We exemplify this in four subfigures of Figure 5.1 that show the effect of changing the input outer radius on four different output metrics.

On the top left of the Figure we can see that the increase of the radius also leads to an increase of the average daily travelled minutes per agent in a nearly linear relation for all mobility modes. This is due to every increase of radius adding more agents who live farther away and thus travel a longer amount of time. The ridepooling characteristic of having higher travelled minutes per agent is a result of more stops per ride which each add a stop time.

The top right subfigure displays the effect of the outer radius on the average gas costs back home per agent. Since the ride home is a lot costlier for agents categorized as lost in ridesharing, the ridesharing costs are higher than the ridepooling home ride costs. Furthermore, we can see that the ridesharing costs converge to the everybodyDrives costs for higher radiuses. This might be a result of both the number of lost agents increasing for higher radiuses, as demonstrated in the bottom right subfigure, and the decreasing average seat count for higher radiuses. We presume this decreasing seat occupancy to be a consequence of the more remote agents not having as many other agents located in their vicinity and also leading to ride times too long to fit into a ridepooling vehicle's schedule; thus, leading them to drive their own car.

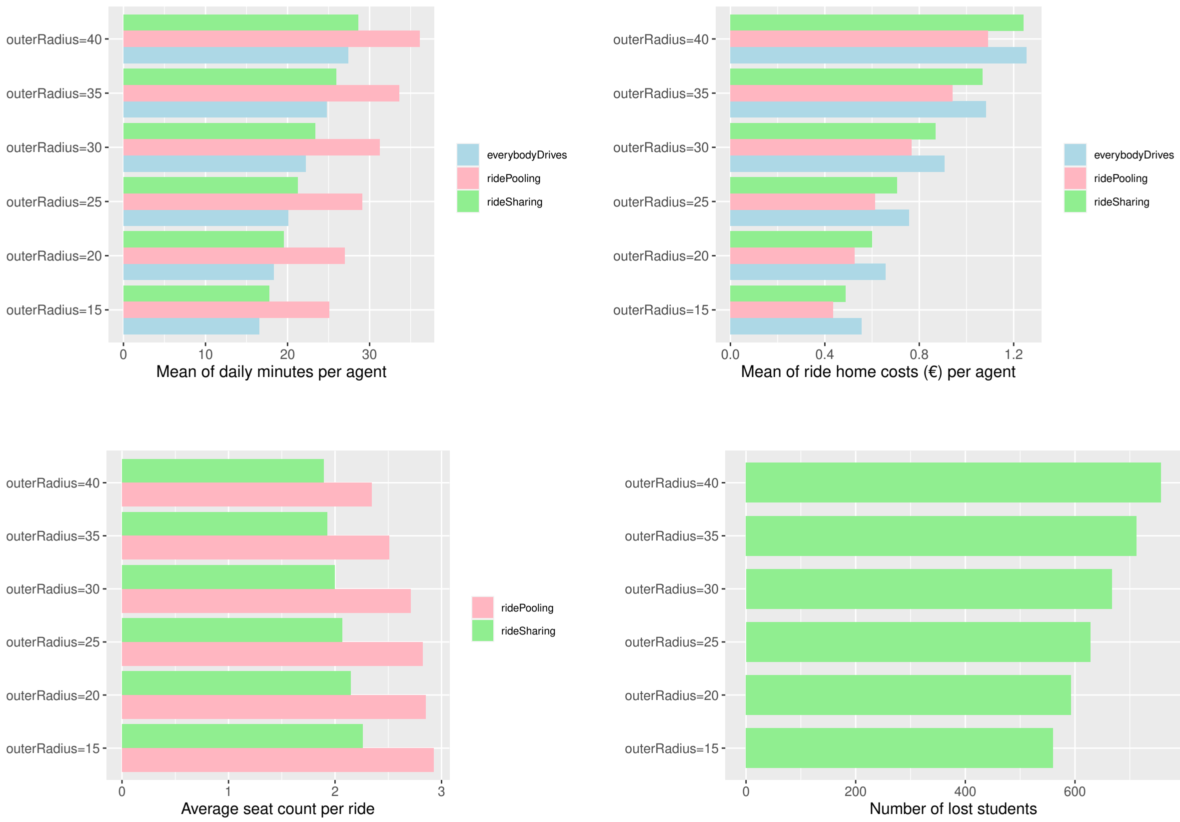


Figure 5.1: The absolute values for different input values for the radius parameter. The bottom right figure is only applicable to ridesharing.

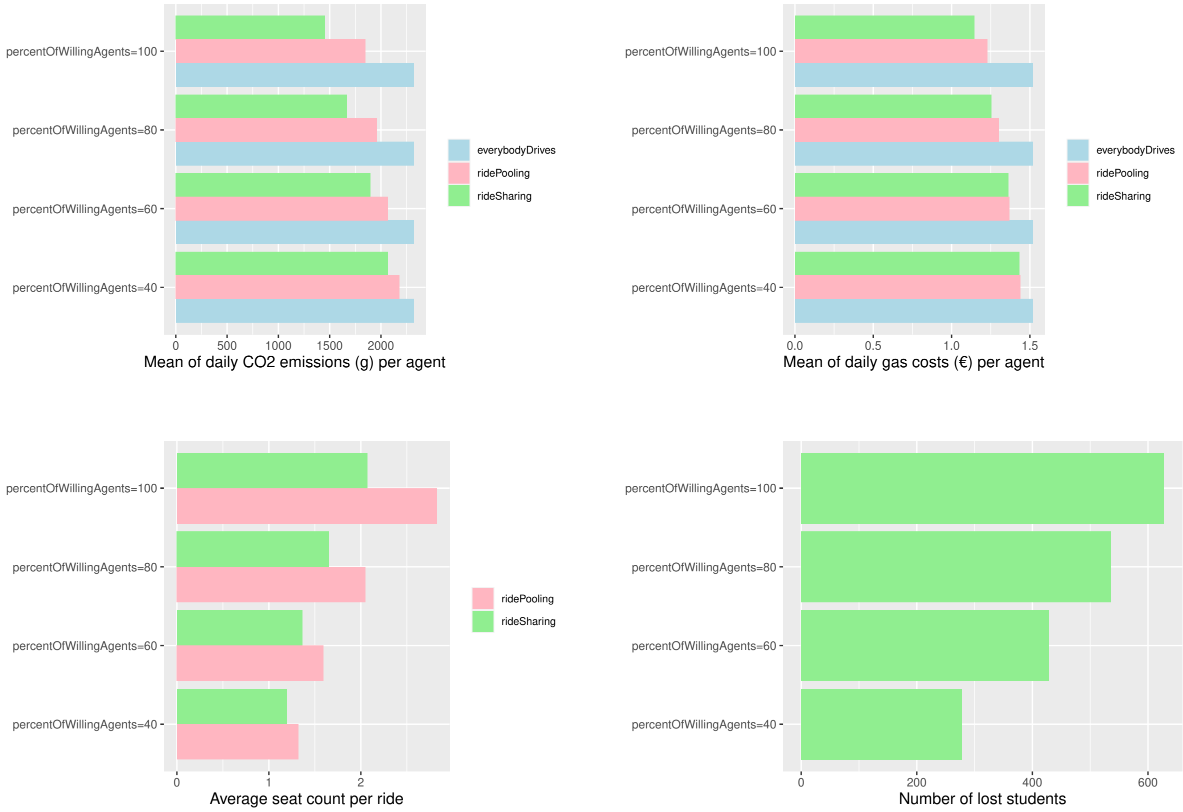


Figure 5.2: The absolute values for different percentages of willing agents. The bottom right figure is only applicable to ridesharing.

Consequently, the choice of outer radius is of great importance when designing a real-world application of ridesharing and ridepooling.

5.1.2 Willingness of Agents

Since the assumption that all students outside of Würzburg would be willing to use alternative mobility modes is rather unrealistic, evaluating the different modes for sub-portions of agents is of great interest.

Figure 5.2 displays the effects of different percentages of agents being willing to use alternative mobility modes (i.e. ridesharing or ridepooling in our case) on output metrics. Since the everybodyDrives mode is not affected by this input parameter, its output metrics are not affected as well. On the top left we can observe how the average daily CO_2 emissions per agent is influenced and can see that, as expected, the emissions are greatly dependent on how many of the agents are inclined to use alternatives for both ridesharing and ridepooling with ridesharing being slightly more dependent.

The average daily gas costs for the agent, displayed on the top right of Figure 5.2, behave similarly as the emissions (as both depend on the travelled kilometers) with the exception of ridesharing costs also containing the higher home ride costs of lost students. Therefore, the values for ridesharing and ridepooling are closer together for the costs.

Both the costs and emissions can be explained when we observe the two subfigures on the bottom of Figure 5.2. Naturally, with fewer agents using mobility modes for sharing rides, the average seat occupancy of the rides per day decreases greatly. Since more people use their own car, the number of lost agents in ridesharing also decreases.

In summary, both ridesharing and ridepooling only yield satisfying results for the reduction of CO_2 emissions and gas costs when enough agents are willing to utilize these modes. In our scenario at least 80% willingness is needed. We can further see that ridesharing leads to greater reductions in all cases.

5.1.3 Time Constraints

Next, we examine the effect of both relaxing and restricting the time-windows for arrival and departure. Figure 5.3 shows how the adjustment of these time constraints influence four exemplary output metrics. As we can see in the top two Subfigures, ridepooling is less affected by relaxing the time-windows than ridesharing which might be because of ridesharing being more limited regarding the amount of possible matches (due to the walking radius) which is where less time constraints open more matching possibilities in the walking radius. On the other hand, ridepooling does not seem to benefit greatly from more match options.

The direct comparison of costs for the ride to campus and costs for the ride home yet again demonstrate the negative effect lost agents have on the average costs for the home ride. However, both ride types (to campus and home) are similarly affected for their respective mobility mode with the exception of the home rides also not experiencing a notable improvement. This might be due to the amount lost students also not significantly decreasing between the values 20 and 25.

The behavior of the number of rides in response to different time-intervals indicates why ridepooling is more robust in regard to time constraints. In the lower right of Figure 5.3 we can see that the amount of daily rides only slightly decreases, as well.

5.2 Specific Influences on Ridesharing

Some of our input parameters only influence ridesharing, namely the number of seats of the student vehicle and the distance an agent is willing to walk to or from the driver's home of a match.

In the top of Figure 5.4 we can see the effect of changing the student car seat count on the average seat occupancy and the number of lost students. Interestingly, while the average ride occupancy is indeed higher for more available seats in the student car, this negatively affects the amount of lost students. The reason for this is that more agents

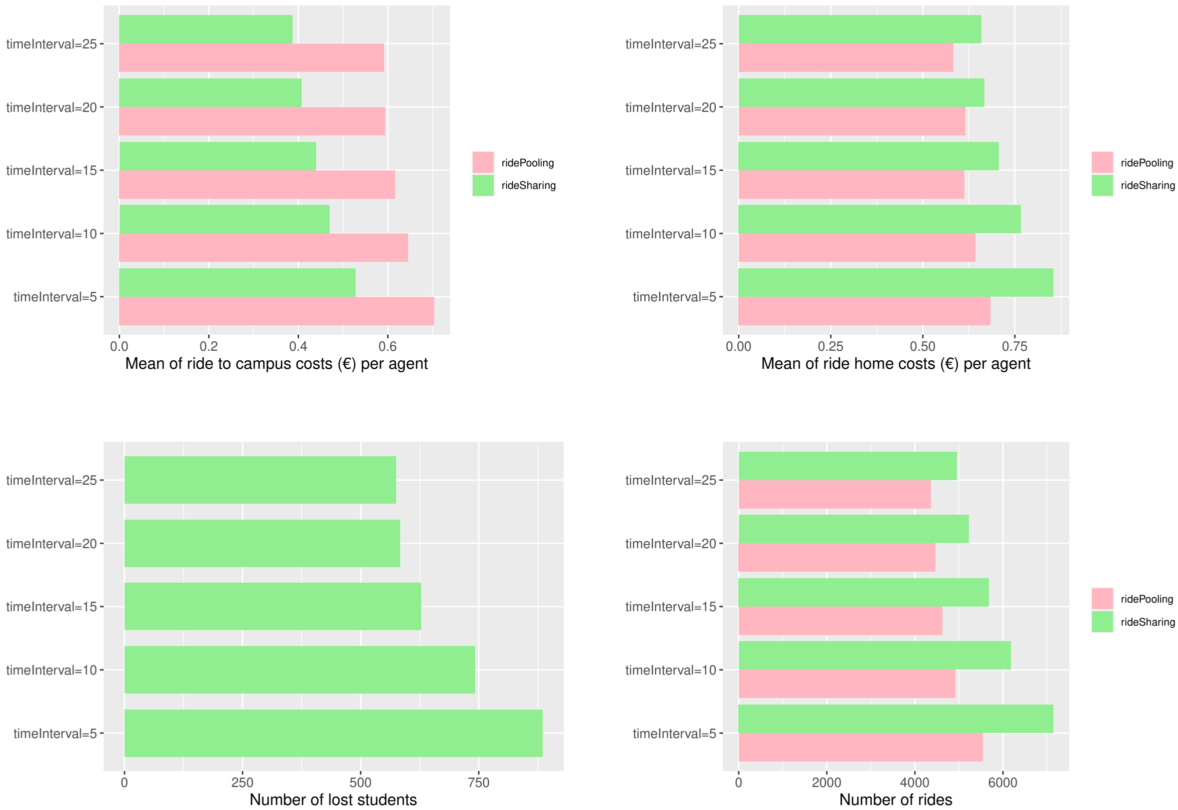


Figure 5.3: The absolute values for different input parameter values. The bottom left figure is only applicable to ridesharing.

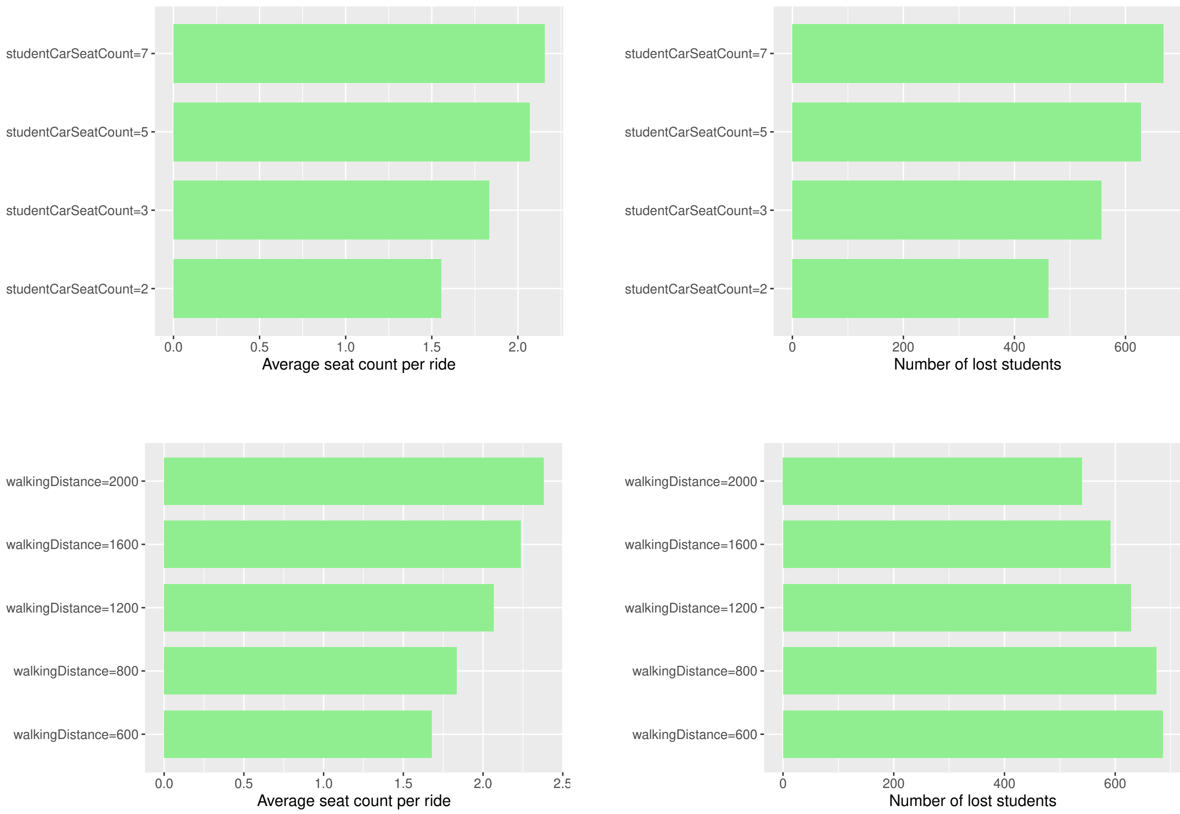


Figure 5.4: The absolute values for different input parameter values for ridesharing.

travelling as passengers (i.e. a higher average seat occupancy) leads to more agents depending on other students taking them along for the ride home.

On the other hand, a larger accepted walking distance, too, leads to a higher average seat occupancy but also reduces the number of lost students. This leads us to believe that the accepted walking distance is a crucial factor for the success of our ridesharing approach. However, expecting students to walk higher distance would greatly impact the quality of service of ridesharing.

5.3 Specific Influences on Ridepooling

As for ridesharing, some input parameters only affect ridepooling which is why we present these separately in this section. The parameters are the fleet size (i.e. amount of available ridepooling vehicles) and the amount of seats in the ridepooling vehicle.

As visible at the top of Figure 5.5, a smaller fleet of ridepooling vehicles (i.e. minibuses) leads to a significantly larger amount of students driving in their own car since the vehicle fleet was at capacity and no matches were found. However, while the average seat occupancy is reduced by a smaller fleet, the resulting changes are not as drastic as for the number of driving agents.

We assume this to be because of many of the driving students for a small fleet, are the sole passenger of a ridepooling ride for a larger fleet size. Consequently, the resulting CO_2 emissions of our evaluated scenario are not as affected by changing the fleet size. This leads to the conclusion that the fleet size can be reduced (e.g, for cost saving purposes) without having a great negative impact on emissions. However, this only applies to our scenario of the ridepooling vehicle not being significantly more sustainable than the student vehicle. Should the ridepooling vehicle have a really low emission rate and be electric, for example, the CO_2 emissions would be reduced immensely. Therefore, a reduction of the fleet with more driving students as a consequence would thus lead to higher emissions.

The influence of the ridepooling vehicle seat count is demonstrated at the bottom of Figure 5.5. Here we can see that changing the number of available seats does greatly affect the average seat occupancy for the rides of a day and thus also influences the CO_2 emissions with vehicles with more seats result in less emissions. Therefore, using a ridepooling vehicle with a sufficient amount of seats seems promising for the sustainability of ridepooling. Nonetheless, it should be noted that vehicles with more seats are typically bigger with greater emissions as a consequence which should be taken into account when a ridepooling vehicle is chosen.

Figure 5.5: The absolute values for different input parameter values for ridesharing.

6 Conclusion and Outlook

In this work we examined alternative mobility modes for the campus mobility of the University of Würzburg. For this we developed a framework with which we are able to model and evaluate different mobility modes under the same circumstances and hard constraints in an event-based simulation.

For this, we created agents that realistically represent the university students in and around Würzburg regarding their home locations and transportation demand. We then implemented the mobility modes ridesharing and ridepooling and evaluated them for a realistic scenario with a maximum service radius of 25km around Würzburg. Afterwards, we compared their results to each other and to a baseline mode in which every student drives a private car.

We judged the modes' performance based on sustainability and quality of service output metrics such as the amount of CO_2 emissions and travelled minutes per day. Additionally, we performed a sensitivity analysis and assessed which input parameters and constraints have the biggest effect on the modes' results in our evaluated scenario.

6.1 Discussion

The results show that, for our assessed scenario in which all vehicles run on gas, ridesharing produces the least amount of CO_2 emissions as the total amount of travelled kilometers is considerably reduced. On the other hand, since ridepooling vehicles have to execute empty rides in order to to pick-up students and drive back to the depot, a reduction of kilometers could not be accomplished. Consequently, the sustainability of ridepooling solely depends on the chosen ridepooling vehicles. As we utilized vehicles with slightly less emissions per kilometer in our scenario, we do indeed accomplish reductions for both CO_2 emissions and gas costs, and that at the same rate. In contrast, Ridesharing has a higher rate of CO_2 reduction than costs reduction. This is due to about 10% of agents not finding a feasible match home during ridesharing and thus being forced to use a more expensive alternative for getting home. Nevertheless, the average gas costs are also reduced for ridesharing.

When another quality of service metric, namely travelled minutes, is examined, we find that ridepooling leads to worse results since the average amount of travelled minutes is significantly higher due to more executed pick-up stops.

In summary, since our approaches for both ridesharing and ridepooling lead to CO_2 and cost reductions, we conclude that they are indeed feasible and successfully so under hard quality of service constraints. We further conclude that, while ridesharing performs better for our scenario (in regard to all average emissions, costs, travelled kilometers and minutes), the results highly depend on circumstances such as how far an agent is willing to walk by foot, how many seats each vehicle has or, especially, the emissions per km of the used vehicles.

Therefore, there is no clear answer for the question of which mobility mode is better for campus mobility as a whole. We would advise universities or institutions that seek to improve the sustainability of their campus mobility to do a thorough survey for the travel demand and requirements of their students to gouge which circumstances are present and which quality of service metrics are crucial.

For example, while our ridepooling approach leads to more minutes travelled in a vehicle, students are being picked up directly at their homes. Our ridesharing approach, on the other hand, expects students to walk to the driving student, up to a certain distance. It is possible that questioned students highly prefer not to walk at all and would rather sit in a vehicle for longer. Furthermore, this might also differ between universities.

We additional deem it highly important that universities that encourage the use of alternative mobility modes, ensure that all students have alternative options in case of not finding matches for their home ride.

As for our evaluated scenario, based on our results we would advise policy makers to consider employing ridepooling with electric vehicles that, preferably, use ecologically sustainable energy sources as this leads to the least amount of emissions and ensures that agents can get a ride home.

However, since we evaluated a simplified scenario without traffic influences and no unexpected incidents (such as students being late for pickup) more studies are necessary for making definite claims about the performance of the examined alternative mobility modes.

6.2 Future Work

In order to definitively judge which mobility modes are best suited for campus mobility, further research is necessary, especially regarding the aspect of realism. For example, we did not yet implement or explore the impact of traffic on the feasibility and results of the modes. In turn, the effect of each mode and traffic and congestion is also of importance for decision making.

Furthermore, as we set each agent to have the same quality of service constraints and configured all vehicles of a type to be the same, it would be interesting to investigate the effect of differing agent demands and vehicle attributes. For both the agents and the vehicles these parameters could be set based on probability distributions.

When calculating the resulting costs of each mode, we presently do not consider acquisition costs, service fees or the wear on the cars. For more realism in future work, these costs should also be taken into account. The same applies to examining the effect of necessary refueling stops or employing drivers for ridepooling and thus having to observe shifts.

This way, the sustainability of alternative campus mobility modes can be evaluted and promoted for more realistic scenarios.

Figures

Tables

Bibliography

[1] "Klimafreundlich zur arbeit - konzept zur reform des pendelns," https://www.sueddeutsche.de/wirtschaft/verkehr-klimafreundlich-zur-arbeit-konzept-zur-reform-des-pendelns-dpa.urn-newsml-dpa-com-20090101-220405-99-797989, accessed: 2024-01-15.

[2] "Verkehrsmittel für den weg zur hochschule," https://www.che.de/wp-content/uploads/upload/Im_Blickpunkt_Verkehrsmittel_fuer_den_Weg_zur_Hochschule.pdf, accessed: 2024-01-17.

[3] "Verkehrsmittel im Überblick: Wie kommst du zur uni?" https://www.jobmensa.de/lifestyle/verkehrsmittel-im-berblick-wie-kommst-du-zur-uni, accessed: 2024-01-15.

[4] "Sustainable mobility drives the newest employee perk," https://www.greenbiz.com/article/sustainable-mobility-drives-newest-employee-perk, accessed: 2024-01-24.

[5] "Mobility-as-a-service: Mit neuer campusmobilität bequem und nachhaltig zum nächsten termin," https://www.scs.fraunhofer.de/de/referenzen/digitale-innovation-und-nachhaltigkeit-am-flughafencampus-muenchen/mobility-as-a-service-konzept.html, accessed: 2024-01-24.

[6] F. Li, X. Long, S. Du, J. Zhang, Z. Liu, M. Li, F. Li, Z. Gui, and H. Yu, "Analyzing campus mobility patterns of college students by using gps trajectory data and graph-based approach," in *2015 23rd International Conference on Geoinformatics*. IEEE, 2015, pp. 1–5.

[7] I. V. McLoughlin, I. K. Narendra, L. H. Koh, Q. H. Nguyen, B. Seshadri, W. Zeng, C. Yao *et al.*, "Campus mobility for the future: the electric bicycle," *Journal of Transportation Technologies*, vol. 2, no. 01, p. 1, 2012.

[8] M. Lukesch, *Sharing Economy in der Logistik: Ein theoriebasiertes Konzept für Online-Mitfahrdienste*. Springer-Verlag, 2019.

[9] P. Vansteenwegen, L. Melis, D. Aktaş, B. D. G. Montenegro, F. S. Vieira, and K. Sörensen, "A survey on demand-responsive public bus systems," *Transportation Research Part C: Emerging Technologies*, vol. 137, p. 103573, 2022.

[10] “Historische ridesharing-dienste,” https://www.spiegel.de/fotostrecke/moias-vorgaenger-rufbusse-aus-der-analogen-welt-fotostrecke-168921.html, accessed: 2024-01-19.

[11] “Moia,” https://www.moia.io/, accessed: 2024-01-20.

[12] L. Melis and K. Sörensen, “The static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment,” *International Transactions in Operational Research*, vol. 29, no. 3, pp. 1417–1453, 2022.

[13] R. Gomes, J. P. de Sousa, and T. G. Dias, “A grasp-based approach for demand responsive transportation,” *International Journal of Transportation*, vol. 2, no. 1, pp. 21–32, 2014.

[14] S. Liyanage and H. Dia, “An agent-based simulation approach for evaluating the performance of on-demand bus services,” *Sustainability*, vol. 12, no. 10, p. 4117, 2020.

[15] J. Narayan, O. Cats, N. van Oort, and S. Hoogendoorn, “Performance assessment of fixed and flexible public transport in a multi agent simulation framework,” *Transportation Research Procedia*, vol. 27, pp. 109–116, 2017.

[16] S. Hörl, “Agent-based simulation of autonomous taxi services with dynamic demand responses,” *Procedia Computer Science*, vol. 109, pp. 899–904, 2017.

[17] K. Liu, J. Zhang, and Q. Yang, “Bus pooling: A large-scale bus ridesharing service,” *IEEE Access*, vol. 7, pp. 74 248–74 262, 2019.

[18] F. Zwick, N. Kuehnel, and S. Hörl, “Shifts in perspective: Operational aspects in (non-) autonomous ride-pooling simulations,” *Transportation Research Part A: Policy and Practice*, vol. 165, pp. 300–320, 2022.

[19] J. Si, F. He, X. Lin, and X. Tang, “Vehicle dispatching and routing of on-demand intercity ride-pooling services: A multi-agent hierarchical reinforcement learning approach,” *arXiv preprint arXiv:2307.06742*, 2023.

[20] K. Meneses-Cime, B. Aksun Guvenc, and L. Guvenc, “Optimization of on-demand shared autonomous vehicle deployments utilizing reinforcement learning,” *Sensors*, vol. 22, no. 21, p. 8317, 2022.

[21] “Blablacar,” https://www.blablacar.de/, accessed: 2024-01-15.

[22] S. Banerjee and R. Johari, “Ride sharing,” *Sharing Economy: Making Supply Meet Demand*, pp. 73–97, 2019.

[23] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transportation Research Part B: Methodological*, vol. 57, pp. 28–46, 2013.

[24] A. Tafreshian and N. Masoud, "Using subsidies to stabilize peer-to-peer ridesharing markets with role assignment," *Transportation Research Part C: Emerging Technologies*, vol. 120, p. 102770, 2020.

[25] Y. Wang, S. Winter, and M. Tomko, "Collaborative activity-based ridesharing," *Journal of Transport Geography*, vol. 72, pp. 131–138, 2018.

[26] B. Wang, H. Liang, S. Hörl, and F. Ciari, "Dynamic ride sharing implementation and analysis in matsim," *hEART 2017: List of Abstracts*, 2017.

[27] J. Long, W. Tan, W. Szeto, and Y. Li, "Ride-sharing with travel time uncertainty," *Transportation Research Part B: Methodological*, vol. 118, pp. 143–171, 2018.

[28] F. Bistaffa, C. Blum, J. Cerquides, A. Farinelli, and J. A. Rodríguez-Aguilar, "A computational approach to quantify the benefits of ridesharing for policy makers and travellers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 119–130, 2019.

[29] M. Haliem, G. Mani, V. Aggarwal, and B. Bhargava, "A distributed model-free ridesharing algorithm with pricing using deep reinforcement learning," in *Proceedings of the 4th ACM Computer Science in Cars Symposium*, 2020, pp. 1–10.

[30] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis.* Mcgraw-hill New York, 2007, vol. 3.

[31] L. Strobel and M. Pruckner, "Omod: An open-source tool for creating disaggregated mobility demand based on openstreetmap," *Computers, Environment and Urban Systems*, vol. 106, p. 102029, 2023.

[32] "Berufspendler," https://www.destatis.de/DE/Themen/Arbeit/Arbeitsmarkt/Erwerbstaetigkeit/Tabellen/pendler1.html, accessed: 2023-11-08.

[33] "Erreichbarkeit des Öffentlichen verkehrs (haltestellen)," https://www.deutschlandatlas.bund.de/DE/Karten/Wie-wir-uns-bewegen/103-Erreichbarkeit-Nahverkehr-Haltestellen.html#_pxztgqa17, accessed: 2023-11-20.

[34] "Klima und erneuerbare energien: Ist elektromobilität wirklich klimafreundlich?" https://www.bmuv.de/themen/verkehr/elektromobilitaet/klima-und-energie, accessed: 2024-02-20.

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Versicherung zur selbständigen Leistungserbringung

Titel der Masterarbeit:

Modeling and Evaluation of Sustainable Mobility Concepts for the Campus Mobility

Thema bereitgestellt von (Titel, Vorname, Nachname, Lehrstuhl):

Prof. Dr.-Ing. Marco Pruckner, Lehrstuhl für Informatik III
Prof. Dr. Marie Schmidt, Lehrstuhl für Informatik I

Eingereicht durch (Vorname, Nachname, Matrikel):

Helena Fehler, 2210435

Ich versichere, dass ich die vorstehende schriftliche Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die benutzte Literatur sowie sonstige Hilfsquellen sind vollständig angegeben. Wörtlich oder dem Sinne nach dem Schrifttum oder dem Internet entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht.

Weitere Personen waren an der geistigen Leistung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich nicht die Hilfe eines Ghostwriters oder einer Ghostwriting-Agentur in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar Geld oder geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Arbeit stehen.

☐ Mit dem Prüfungsleiter bzw. der Prüfungsleiterin wurde abgestimmt, dass für die Erstellung der vorgelegten schriftlichen Arbeit Chatbots (insbesondere ChatGPT) bzw. allgemein solche Programme, die anstelle meiner Person die Aufgabenstellung der Prüfung bzw. Teile derselben bearbeiten könnten, entsprechend den Vorgaben der Prüfungsleiterin bzw. des Prüfungsleiters eingesetzt wurden. Die mittels Chatbots erstellten Passagen sind als solche gekennzeichnet.

Der Durchführung einer elektronischen Plagiatsprüfung stimme ich hiermit zu. Die eingereichte elektronische Fassung der Arbeit ist vollständig. Mir ist bewusst, dass nachträgliche Ergänzungen ausgeschlossen sind.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Ich bin mir bewusst, dass eine unwahre Erklärung zur Versicherung der selbstständigen Leistungserbringung rechtliche Folgen haben kann.

Würzburg, 26.02.2024, H. Fehler

Ort, Datum, Unterschrift