

Titel der Arbeit:
Demand Management and Vehicle Routing
in Dynamic Ride-Sharing Systems

Schriftliche Promotionsleistung
zur Erlangung des akademischen Grades
Doctor rerum politicarum
vorgelegt und angenommen
an der Fakultät für Wirtschaftswissenschaft
der Otto-von-Guericke Universität Magdeburg

Verfasser: Jarmo Haferkamp

Geburtsdatum und -ort: 17.07.1991, Vorwerk

Arbeit eingereicht am: 01.09.2023

Gutachter der schriftlichen Promotionsleistung:

Prof. Dr. Marlin W. Ulmer

Prof. Dr. Jan Fabian Ehmke

Prof. Dr. Stefan Nickel

Datum der Disputation: 27.10.2023

List of associated publications

- Chapter 2: Haferkamp, J. & Ehmke, J. F. (2020). An Efficient Insertion Heuristic for On-Demand Ridesharing Services. In: Transportation Research Procedia. 22st EURO Working Group on Transportation Meeting, EWGT 2019, 18th-20th September 2019, Barcelona, Spain, pp. 107-114. <https://doi.org/10.1016/j.trpro.2020.03.074>.
- Chapter 3: Haferkamp J. & Ehmke J. F. (2022). Effectiveness of Demand and Fulfillment Control in Dynamic Fleet Management of Ride-Sharing Systems. In: Networks. 79(3):314-337. <https://doi.org/10.1002/net.22062>.
- Chapter 4: Haferkamp, J., Ulmer, M. W., & Ehmke, J. F. (2023). Heatmap-Based Decision Support for Repositioning in Ride-Sharing Systems. In: Transportation Science 0(0). <https://doi.org/10.1287/trsc.2023.1202>.
- Chapter 5: Haferkamp, J. (2023). Design of Multi-Optional Pickup Time Offers in Ride-Sharing Systems. *Under review in the EURO Journal on Transportation and Logistics*.

All papers have been reformatted to ensure a clear and consistent presentation.

Abstract

Ride-sharing systems contribute to the transition from private cars to a more sustainable utilization of urban traffic infrastructure. To use such systems, travelers indicate their origin and destination via a mobile application, receive a transportation offer, and, if they accept it, are transported to their destination, possibly sharing the vehicle. To ensure long-term acceptance of the service, ride-sharing operators aim to provide a high percentage of travelers with an acceptable transportation offer. To this end, they make use of demand management, i.e., the shaping of demand in its volumes and/or characteristics, and vehicle routing, i.e., routing of the vehicle fleet in order to fulfill transportation requests. In this regard, ride-sharing operators face the challenge that corresponding approaches should ensure both high system performance and fair service conditions, e.g., in terms of traveler fares and driver compensation.

The aim of this thesis is therefore to develop soft, i.e. non-monetary, approaches for demand management and vehicle routing in dynamic ride-sharing systems for the benefit of operators, travelers and drivers. To this end, we first focus on implementing the computational framework including state-of-the-art vehicle routing heuristics, as well as on building a comprehensive understanding of controlling demand and its fulfillment in ride-sharing systems. Secondly, we are developing new approaches that focus on demand management and vehicle routing to reduce request cancellations while involving travelers or drivers in the decision-making process. More precisely, we first design heatmaps to support repositioning decisions of idle drivers to balance supply and demand in decentralized ride-sharing systems. Secondly, we design multi-optional transportation offers to manage demand while enabling travelers to choose a convenient pickup time from a set of options.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Challenges in ride-sharing operations | 3 |
| 1.2 | Related literature on ride-sharing systems | 4 |
| 1.3 | Research questions and thesis outline | 5 |
| 2 | An efficient insertion heuristic for on-demand ridesharing services | 9 |
| 2.1 | Introduction | 10 |
| 2.2 | Problem description | 11 |
| 2.3 | Computational experiments | 14 |
| 2.4 | Conclusion | 20 |
| 3 | Effectiveness of demand and fulfillment control in dynamic fleet management of ride-sharing systems | 23 |
| 3.1 | Introduction | 24 |
| 3.2 | Demand and fulfillment control | 26 |
| 3.3 | Problem formulation | 33 |
| 3.4 | Evaluation framework | 36 |
| 3.5 | Computational experiments | 43 |
| 3.6 | Conclusion | 57 |
| 4 | Heatmap-based decision support for repositioning in ride-sharing systems | 61 |
| 4.1 | Introduction | 62 |
| 4.2 | Related literature | 64 |
| 4.3 | Problem statement and formulation | 70 |

Contents

| | | |
|----------|--|------------|
| 4.4 | Repositioning heatmaps | 76 |
| 4.5 | Experimental setup | 83 |
| 4.6 | Results | 87 |
| 4.7 | Final remarks | 100 |
| 5 | Design of multi-optional pickup time offers in ride-sharing systems | 103 |
| 5.1 | Introduction | 104 |
| 5.2 | Related literature | 106 |
| 5.3 | Problem description | 110 |
| 5.4 | Offer design | 114 |
| 5.5 | Experimental setup | 118 |
| 5.6 | Computational results | 123 |
| 5.7 | Final remarks | 130 |
| 6 | Conclusion | 131 |
| 6.1 | Summary | 131 |
| 6.2 | Outlook | 133 |
| | Bibliography | 135 |
| | Appendix | 149 |

Chapter 1

Introduction

Urban areas suffer from congested transportation infrastructure, causing daily traffic jams and parking shortages. To make more sustainable use of the limited urban space, authorities seek a transition away from private cars towards shared mobility solutions. Contributing to this transition are dynamic ride-sharing systems, which aim to provide (almost) door-to-door mobility on demand at affordable prices by consolidating transportation requests.

In this thesis, we understand ride-sharing systems to be services where travelers indicate their origin and destination via a mobile application to receive a transportation offer immediately. If they accept the offer, they are picked up accordingly and transported to their destination, possibly sharing the vehicle with other travelers for some part of the trip. In contrast, unacceptable offers lead to travelers canceling their request. Ride-sharing systems strive to minimize such cancellations, i.e., they try to satisfy a high percentage of travelers in order to ensure the long-term acceptance of the service.

To achieve this, ride-sharing operators make use of demand management and vehicle routing. Demand management refers to shaping the temporal and spatial demand volumes and/or characteristics. This can range from basic request acceptance mechanisms to sophisticated pricing approaches. Vehicle routing, in turn, refers to the routing of the vehicle fleet to fulfill new incoming and future expected transportation requests. This includes updating route plans to accommodate new requests, as well as proactive routing decisions, such as the repositioning of idle vehicles.

The appropriate implementation of demand management and vehicle routing depends on the business model of the ride-sharing system at hand. Decentralized ride-sharing platforms like *Uber*, *Lyft* or *Didi*, which outsource transportation to independent drivers, often apply surge pricing to balance demand and supply volumes. In terms of vehicle routing, they focus on matching travelers and drivers for direct transportation, while shared rides are optional for travelers to reduce the fare. In contrast, centralized systems, such as *MOIA*, which maintain a fleet of employed

drivers, may also use pricing to manage demand volumes. However, they also rely on advanced vehicle routing to consolidate transportation to ensure a high utilization of the costly fleet. Other ride-sharing systems, such as *Via* in Jersey City or *SSB Flex 2.0* in Stuttgart, are operated in cooperation with public transportation authorities, giving priority to the public interests. Demand management and vehicle routing thus have to ensure high system performance, e.g., in terms of low costs and burden on traffic infrastructure, as well as fair service conditions, e.g., in terms of traveler fares and driver compensation.

Considering the related literature, it can be found that dynamic vehicle routing has been researched for decades (see, e.g., Psaraftis et al. (2016) and Soeffker et al. (2022) for overviews). In contrast, integrated demand management and vehicle routing have only recently gained interest in the literature on dynamic optimization of transportation services (see Fleckenstein et al. (2023) for an overview). With respect to dynamic ride-sharing systems, the focus has been particularly on pricing, or surge pricing, to manage the highly volatile demand (and supply) (see, e.g., Bertsimas et al. (2019), Guda & Subramanian (2019), Nourinejad & Ramezani (2020), Chen et al. (2021), Hu et al. (2022), and Ma et al. (2022)). Even though pricing plays a major role in both research and practice of ride-sharing systems, it is extremely unpopular among both travelers and (independent) drivers due to the resulting uncertain and highly fluctuating fares and competition (see, e.g., Bertini & Koenigsberg (2021), Conger (2021), and Abrams (2022)).

In this thesis, we aim to develop softer, i.e., non-monetary means for demand management and vehicle routing in dynamic ride-sharing systems that involve travelers or drivers in the decision-making processes. More precisely, we first focus on implementing the computational framework, along with state-of-the-art vehicle routing heuristics. Second, we establish a comprehensive understanding of how demand and its fulfillment can be controlled considering the implications for system performance and travelers. Based on this groundwork, we develop two approaches that integrate demand management and vehicle routing to reduce request cancellations while involving travelers or drivers in the decision-making process. The first approach focuses primarily on advanced vehicle routing by designing heatmaps to guide idle drivers toward balancing supply and demand in decentralized ride-sharing systems. The second approach focuses more on advanced demand management by designing

transportation offers that ensure effective fulfillment and allow travelers to choose a convenient pickup time from a set of options.

In the following, we first discuss the challenges in dynamic ride-sharing operations. Then, in Section 1.2, we provide a brief overview of the related literature. Finally, in Section 1.3, we discuss the aim and outline of this thesis.

1.1 Challenges in ride-sharing operations

Ride-sharing systems have to manage a large fleet of vehicles in order to satisfy uncertain as well as temporal and spatial unbalanced transportation demands while meeting travelers' expectations. These expectations include on-demand, affordable, and reliable transportation with reasonable detours. To satisfy these expectations, ride-sharing operators must make dynamic decisions that include assigning requests to vehicles, providing transportation offers to travelers, and repositioning idle vehicles.

The most fundamental decision concerns the assignment of incoming transportation requests to incumbent vehicle routes, considering vehicle-sharing opportunities. The challenge is to quickly determine assignments within the large-scale fleet that satisfy traveler expectations and enable the effective fulfillment of the uncertain future demand. The assignment decision is thus primarily about dynamic vehicle routing, with demand management being a secondary concern.

Based on the potential assignment(s), a transportation offer must be made to the requesting traveler. Such an offer may include information on the fare, driver and expected time frame for pickup and drop-off. It is thus the basis on which travelers decide whether to complete or cancel the request. Accordingly, the challenge for operators is to balance between an offer that satisfies the travelers' (uncertain) expectations and a profitable offer that provides flexibility in vehicle routing. The offer decision is thus primarily a question of demand management, with vehicle routing considerations playing an important role.

In contrast, repositioning decisions are only indirectly related to the processing of transportation requests aiming to offset temporal and spatial imbalances in demand by means of idle vehicles. The challenge for operators is to find a (re)positioning that maximizes the coverage of the expected future demand. Moreover, the resulting vehicle routing should be comprehensible to the driver or, in decentralized ride-sharing

systems, support independent drivers in their repositioning decisions. Repositioning decisions are primarily considered as anticipatory vehicle routing, with demand management being involved by deciding which demand to cover. However, the corresponding demand management capabilities are limited by the availability of idle vehicles and thus by the temporal and spatial imbalance in demand.

1.2 Related literature on ride-sharing systems

In the following, we provide a brief overview of the related literature on demand management and vehicle routing in ride-sharing systems, focusing on the three operational decisions discussed before. For a more comprehensive classification of how demand and its fulfillment are controlled in the related literature, see Chapter 3. Moreover, for an overview of research considering demand management and vehicle routing in other applications, we refer to Fleckenstein et al. (2023). Most of the corresponding papers focus on time window management for next-day attended home deliveries (see Waßmuth et al. (2023) for a review). In contrast, comparatively few papers, such as Ulmer (2020) or Klein & Steinhardt (2023), consider dynamic vehicle routing as in ride-sharing systems, where fulfillment is performed simultaneously with the incoming new requests. However, ride-sharing also differs from same-day delivery applications due to the immediacy of transportation requests.

Focusing on the literature on ride-sharing systems, the majority of papers primarily concern dynamic vehicle routing in terms of assignment decisions. Early papers addressing related problems refer to the dynamic dial-a-ride problem as an extension of the well-known dynamic vehicle routing problem. For reviews on the static and dynamic dial-a-ride problem, see, e.g., Molenbruch et al. (2017) and Ho et al. (2018). For an overview of more recent assignment approaches for large-scale ride-sharing systems, see Wang & Yang (2019), who cover also repositioning decisions and offer decisions in terms of pricing.

The repositioning of idle vehicles to compensate for temporal and spatial imbalances in demand has also been explored widely (for a comprehensive overview, see Chapter 4). For example, queueing-based approaches (see, e.g., Zhang & Pavone (2016) and Sayarshad & Chow (2017)) and model predictive control approaches (see, e.g., Iglesias et al. (2018) and Pouls et al. (2022)) have been considered to make completely centralized repositioning decisions. Furthermore, agent-based reinforcement

learning has gained interest recently to obtain sophisticated repositioning policies (see, e.g., Holler et al. (2019) and Jiao et al. (2021)).

In contrast, demand management via comprehensive offer decisions has been neglected for a long time (for a comprehensive overview, see Chapter 5). Earlier papers considered only feasibility checks to ensure that accepted requests are fulfilled subject to waiting time and detour constraints (see, e.g., Berbeglia et al. (2011) and Alonso-Mora et al. (2017a)). More advanced offer decisions have been considered, particularly with respect to selective acceptance mechanisms and pricing approaches. Selective acceptance mechanisms aim to accept favorable transportation requests with regard to current opportunity costs or the expected future demand (see, e.g., Hosni et al. (2014) and Heitmann et al. (2023)). In contrast, pricing optimizes travelers' fares and/or drivers composition to attract drivers (and deter demand, see, e.g., Bimpikis et al. (2019), Guda & Subramanian (2019)), or to attract demand (and deter drivers, see, e.g., Hu et al. (2022)). These papers include the approaches mentioned above for repositioning: the study by Taylor (2018) is based on a queueing-based approach, Nourinejad & Ramezani (2020) develop a model predictive control approach, and Chen et al. (2021) rely on reinforcement learning.

1.3 Research questions and thesis outline

In this thesis, we now aim to contribute to the dynamic ride-sharing literature by developing non-monetary approaches that integrate demand management and vehicle routing to the benefit of operators, travelers, and drivers. In doing so, we aim to break away from the sole focus on centralized decision-making and offer alternatives to pricing, which can be considered the dominant approach to demand (and supply) management in the recent literature and practice of dynamic ride-sharing systems. To achieve these aims, we address three main research questions:

- RQ1: What constitutes a softer, i.e., non-monetary, approach to demand management and vehicle routing in dynamic ride-sharing systems?
- RQ2: What are the opportunities and implications of advancing demand management and/or vehicle routing for operators, travelers, and drivers?
- RQ3: How can travelers and drivers be involved in the decision-making process to their and the operators' benefit?

RQ1 is related to both gaining a comprehensive understanding of demand management and vehicle routing in dynamic ride-sharing systems and the development of new corresponding approaches. *RQ2* refers to the identification of opportunities for advanced demand management and/or vehicle routing to improve the performance of ride-sharing systems and the analysis of the (potentially adverse) implications for travelers and drivers. Finally, *RQ3* is about how to involve travelers and drivers in the decision-making overcoming the purely centralized perspective for the benefit of all stakeholders.

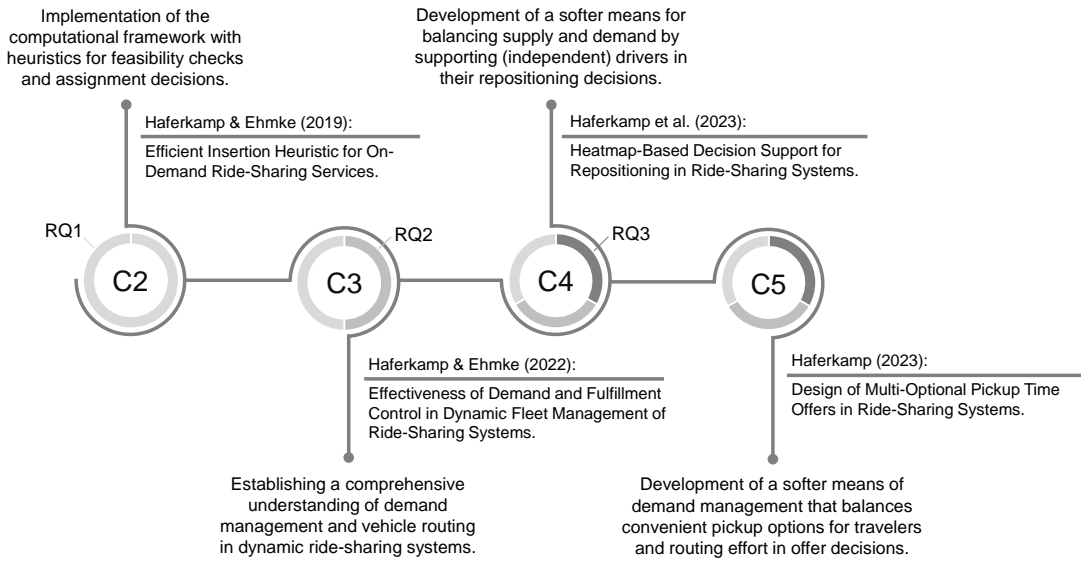


Figure 1.1: Thesis Overview

A comprehensive overview of the structure of the thesis is provided by Figure 1.1. The circles marked *C2* to *C5* symbolize the four main chapters. The color differences in turn show the increasing consideration of the introduced research questions, reflecting the chronological organization of the thesis. Furthermore, the figure indicates for each chapter the main contribution as well as the title and authors of the related papers. From the respective contributions, it can be noted that Chapters 2 and 3 are primarily devoted to laying the groundwork for the development of the solution approaches presented in Chapters 4 and 5.

More precisely, Chapter 2, titled “*An Efficient Insertion Heuristic for On-Demand Ridesharing Services*” and published in *Transportation Research Procedia*, establishes the computational framework by implementing efficient heuristics that integrate feasibility check and assignment of dynamically incoming transportation requests. With respect to *RQ1*, it thus implements a first non-monetary approach for demand management and vehicle routing in dynamic ride-sharing systems.

Chapter 3, entitled “*Effectiveness of Demand and Fulfillment Control in Dynamic Fleet Management of Ride-Sharing Systems*”, published in *Networks*, focuses on providing the fundamental understanding considering *RQ1* and *RQ2*. The contribution is twofold: (1) to classify how demand and its fulfillment are controlled in the related literature; (2) to explore the opportunities and implications of advanced demand and fulfillment control in a comprehensive computational study. The literature classification highlights the diversity of potential approaches, whereas the experimental results indicate that demand and fulfillment control affect the performance and service quality of ride-sharing systems quite differently.

Based on the previous two chapters, Chapter 4 titled “*Heatmap-Based Decision Support for Repositioning in Ride-Sharing Systems*” and published in *Transportation Science* addresses the three research questions by proposing anticipatory heatmaps to support idle drivers in their repositioning decisions. As such, it is one of the first non-monetary approaches for balancing demand and supply also in decentralized ride-sharing systems by highlighting driver-specific repositioning opportunities. However, as our heatmaps change the future driver distribution and hence demand coverage, we propose an adaptive learning algorithm that iteratively updates the heatmaps until the service cancellations cannot be reduced further. We show in comprehensive computational experiments the merits of such carefully designed heatmaps and investigate several scenarios with non-compliant drivers, analysing the potential for and against deviating from heatmap recommendations.

Finally, Chapter 5 titled “*Design of Multi-Optional Pickup Time Offers in Ride-Sharing Systems*” and currently under review in the *EURO Journal on Transportation and Logistics* addresses the three research questions by designing multi-optional pickup offers for an advanced non-monetary demand management. In this regard, the work is among the first to address travelers’ uncertain pickup time requirements by providing a set of options for the traveler to choose from. For the design of these

pickup time offers, we propose a parametric cost function approximation that balances acceptance probability and approximated routing effort. The offer design thus aims to provide travelers with an acceptable pickup option, while maintaining an effective utilization of the ride-sharing fleet. We demonstrate the effectiveness of this approach in a comprehensive computational study and provide managerial insights, particularly with respect to the value of information on traveler pickup time requirements.

Chapter 2

An efficient insertion heuristic for on-demand ridesharing services

Abstract In recent years, several ride-sharing operators have launched their services across the globe. For these services, mobility requests arrive dynamically and have to be realized with a limited number of vehicles. The problem of request acceptance and route planning can be modeled as Dynamic Dial-a-Ride Problem (DDRP). Due to the limited transportation capacity of the shared vehicles, an important objective of the new service operators is to maximize the number of accepted requests. Since not all requests can be fulfilled, it is necessary to inform passengers immediately about the acceptance or rejection of their request. One way to achieve this is via feasibility check of the DDRP, which in this case must be performed within a very short computing time. The aim of this contribution is to examine the trade-off between computing time and solution quality as well as the effects of rescheduling during the feasibility check under realistic conditions of a typical urban on-demand ride-sharing service. For this purpose, a Large Multiple-Neighborhood Search is proposed as an efficient approach to solve the DDRP. The analysis of different computing time limitation's as well as the performance evaluation of the developed heuristic is based on computational simulation.

Keywords ride-sharing Services, Dynamic Dial-a-Ride Problem, Large Multiple-Neighborhood Search

2.1 Introduction

In many countries, on-demand ride-sharing services have become an affordable and comfortable mobility alternative classified between public transport and taxi services. In major German cities, for example, an increasing number of mobility service operators has launched their services recently, creating a competitive market. In Berlin, services like BerlKönig, Clevershuttle, Allygator and MyTaxiMatch offer shared rides in highly populated parts of the inner city. The increased attractiveness of this business model can be attributed to improved operations due to digitization and the expected availability of autonomous driving vehicles in a couple of years, which is supposed to further increase the flexibility and profitability of on-demand ride-sharing services.

All recent on-demand ride-sharing services come with mobile applications, which allow passengers to submit requests on short notice. Therefore, some operators do not offer the possibility of pre-booking rides. Instead, the next possible pick-up time is determined and offered at the time of request submission. In this context, passengers expect to receive an immediate and reliable response to their request. For a response to be perceived by a user as immediate, the service operator must reply in less than one second (Nielsen, 1994). Hence, in order to send a reliable request acceptance confirmation, the service operator must check immediately whether a request can be fulfilled with their currently available transport capacity, taking into account the possibility of shared rides. These requirements pose significant challenges to the service operators.

Evaluation of request acceptance is known in the routing literature as “feasibility check” (Ehmke & Campbell, 2014) and can, in the considered case, be performed by solving the Dynamic Dial-a-Ride Problem (DDRP). During the feasibility check, at least one insertion position for a new request is determined within an existing solution which does not violate any constraints. If the search is successful, the request is inserted and therefore accepted. The new solution then serves as a route plan for fulfilling the accepted requests. Recent examples for literature providing overviews on solution approaches for the DDRP are Cordeau & Laporte (2007), Berbeglia et al. (2010) and Hosni et al. (2014).

In most studies, the underlying applications are the transportation of patients or elderly passengers and not on-demand ride-sharing services. A first contribution ad-

addressing this application is provided by Horn (2002). In this article, a framework for the operation of an on-demand mobility service is presented. The feasibility check is carried out with various minimum-cost insertion heuristics, which differ in the way they determine the set of insertion positions to be considered. Hosni et al. (2014) formulate a shared taxi problem and present solution approaches for the static and dynamic Dial-a-Ride Problem. For the solution of the dynamic variant, the cost-minimal insertion position of a request is determined by solving a mathematical program. In order to increase the efficiency of this approach, the cases in which no feasible solution can be found are identified in advance. Bischoff et al. (2017) examine the operation of an on-demand ride-sharing service in the city of Berlin, Germany, using the multi-agent transport simulation framework “MATSim”. A Minimum-Cost Insertion Heuristic (MCI) is integrated into the simulation framework. For an incoming request, this heuristic checks each possible insertion position and selects the one that is feasible and has the least additional travel time.

So far, the contributions dealing with ride-sharing services focus on the determination of a feasible and cost-effective insertion position. Extending this work, in our paper, we propose an advanced, state-of-the-art solution technique, which reschedules the already accepted requests very quickly upon potential integration of the new request. Furthermore, we analyze the performance of our technique under the demanding computational time constraints of an on-demand ride-sharing service operator. In particular, we introduce a Large Multiple-Neighborhood Search (LMNS) to carry out the feasibility check of the DDRP.

2.2 Problem description

We model the considered problem as a dynamic and stochastic optimization problem. Let $G = N, E$ be a complete, directed graph mapping the operating area of an on-demand ride-sharing service. The set of nodes N represents potential locations of origins and destinations of passengers, and the set of edges E are connections between these locations weighted according to the expected driving duration from origin to destination.

A fleet of V identical vehicles, each with a limited capacity c_v and a location l_v , is available for the fulfillment of mobility requests. For the operation of the service, an independent period over a limited number of hours is considered. In

the course of the period, mobility requests $r \in R$ are posed dynamically following a stochastic distribution while already accepted requests are being fulfilled successively. Each request is defined by an origin $i_r \in N$ and a destination $j_r \in N$ as well as a capacity requirement q_r , which corresponds to the number of passengers per mobility request, and a duration d_r for boarding or alighting. Furthermore, there is a discrete point in time, s_r , at which the request is submitted, as well as a time restriction given by a latest acceptable time of arrival a_r at destination j_r . It is assumed that passengers want to arrive at their destination as soon as possible. Their requests can therefore only be scheduled in such a way that the arrival time is sufficiently early. To determine the latest acceptable arrival time a_r , a maximal tolerance duration t_r for postponed destination arrivals is added to the earliest possible arrival time, given by s_r and the driving duration from i_r to j_r . The extent to which the tolerance duration t_r is required depends on how soon a vehicle can pick up the passenger, i.e., the waiting time as well as the actual ride duration. Increased ride durations may occur for passengers due to detours as well as boarding and alighting of other passengers if they share a part of their ride.

For each incoming request, it has to be checked whether its realization is feasible subject to time and capacity related constraints. This check is performed by attempting to find a valid route plan that contains the new received request as well as all previously accepted requests. It must be taken into account that request acceptance is carried out simultaneously with the ongoing fulfillment of requests. Therefore, only those parts of a current route plan can be changed which have not yet been executed, i.e. the parts where the scheduled arrival time is after the current time of request t_r . Moreover, the flexibility in rescheduling is constrained in such a way that the next stop remains unchanged during the journey. This ensures that drivers and passengers can be informed reliably about the next stop in order to avoid the perception of an arbitrary planning.

A new route plan is considered as feasible solution if for all $r \in R$, the origin node i_r and destination node j_r belong to the same route, and i_r is planned before j_r , as well as time and capacity constraints are satisfied. If this can be accomplished, the request is accepted and the route plan found is used for the fulfillment of the accepted requests. The objective is to reject as few passengers as possible, and therefore to maximize the number of mobility requests accepted.

2.2.1 Solution approach

We propose the following LMNS to solve the presented problem. Our LMNS is a variant of the Adaptive Large Neighborhood Search (ALNS), which was proposed by Ropke & Pisinger (2006) for a static Pickup and Delivery Problem. The general idea of the approach is to find a new feasible solution by inserting a new mobility request r_{new} into a given solution s_0 . The LMNS is therefore performed within the booking process whenever a request is received. The given solution s_0 is based on the last returned solution, but contains only those parts for which rescheduling is possible with regard to the time and operational constraints formulated above.

```

1 Simplified function  $LMNSI(s_0, r_{new})$ 
2    $s = s_0$ 
3   add  $r_{new}$  to requestbank
4   while (computation time limit not reached) do
5      $s^t =$  remove requests from  $s$  to requestbank
6      $s^t =$  insert requests from requestbank into  $s^t$ 
7     if (all requests are inserted) then
8       | return  $s^t$ 
9     else if ( $s^t$  is accepted) then
10    |  $s = s^t$ 
11  end
12  return  $s_0$ 

```

Figure 2.1: Pseudocode of the Large Multiple-Neighborhood Search

The use of a modified ALNS was inspired by the step of minimization of the number of vehicles used in Ropke & Pisinger (2006). Within this step, each route is removed from the current solution once in order to try to insert its requests into the remaining ones. Our LMNS differs from the original ALNS in several aspects. First, as shown in line 3 of the pseudocode, only the new request r_{new} is added to the requestbank before the iterative search begins.

Second, lines 4 and 7 to 9 show that the termination criteria have been adjusted. Since only the insertion of r_{new} has to be accomplished, the search is terminated as soon as a feasible solution has been found (line 7 to 9). The second termination criterion is a limit of the computation time, which ensures a maximum response time

within the booking process for each incoming request (line 4). When this criterion is fulfilled, the starting solution s_0 is returned and the request rejected, since no feasible solution could be found within the specified computation time.

Third, for our problem, the set and selection of removal and insertion operators used in lines 5 and 6 has been adapted. For the removal of requests in line 5, the Random, Worst and Shaw operators are still applied, and for the subsequent insertion in line 6, only the Regret-2 insertion operator both with and without noise is carried out. The selection of a removal and insertion operator is made randomly at each iteration. It is not useful to apply an adaptive selection procedure, since the number of realizable iterations per function call is too limited due to the short computing times.

Fourth, also the acceptance criterion in line 9 has been modified. The objective function value of a solution is determined over the overall driving duration plus a penalty term β for each unintegrated request. Usually, simulated annealing is used as acceptance criterion, but its advantages, namely controlling the search process through decreasing temperature, does not apply in this case, as the number of iterations is too limited. Instead, a similar criterion is applied, where the temperature is recalculated within each iteration with the formula used in Ropke & Pisinger (2006) for the calculation of the start temperature. Therefore, solutions with an improved objective function value are accepted as well as deteriorated solutions with a certain probability depending on the degree. This degree is controlled by the parameter α , which specifies the factor of deterioration where the probability of acceptance equals 50%. For factors of deterioration below α , the probability of acceptance increases and decreases for those above α .

2.3 Computational experiments

Within the scope of this study, computational experiments have been carried out through simulating and optimizing an on-demand ride-sharing service operating in the road network of Berlin, Germany. The aim of the computational experiments is to obtain insights on the performance of the LMNS in the light of the challenging runtime restrictions of the problem. The results are then benchmarked against the MCI of Bischoff et al. (2017) to examine the extent to which rescheduling during the insertion leads to an improvement of the solution quality. The quality of a solution

is measured by the number of mobility requests accepted and the service quality, represented by the average waiting time and detour time of the passengers. Before the results are presented, we describe the parameters of the considered ride-sharing service and the design of the computational experiments.

2.3.1 Settings

In order to investigate a realistic scenario, a service similar to the recently established BerlKönig is assumed. BerlKönig is operated by the Berlin public transit authorities as a field test. Our assumed fleet therefore corresponds to BerlKönig's launch fleet, which consists of 50 vehicles with a seat capacity of $c_v = 6$ (Mercedes-Benz, 2019). It is further assumed that at the beginning of the planning period, the vehicles are idle and distributed randomly over the operating area. The area under consideration is also derived from the launch operating area of BerlKönig (BerlKönig, n.d.). In this area, 400 addresses (cf. Figure 2.2) were randomly selected as potential origins and destinations, and the real driving durations between these locations were determined using OpenStreetMap (OSM).



Figure 2.2: Potential Origin and Destination Locations (created using Leaflet | © OpenStreetMap)

Based on the selected 400 locations, 1600 requests are generated by randomly drawing a combination of origin i_r and destination j_r under the restriction that no combinations are duplicated and that the direct driving durations have a minimum duration of 5 minutes (i.e., it is therefore assumed that the use of an on-demand ride-sharing service is not reasonable if the driving duration is below 5 minutes). In addition, a capacity requirement q_r is determined for each request by drawing a random value from a Poisson distribution with $\lambda = 1$, ensuring that primarily individual passengers or couples request the service, and $q_r \leq c_v$ for all requests. Furthermore, a fixed value of 1 minute is assumed for the boarding and alighting duration d_r , and a value of 20 minutes as tolerance duration t_r reflecting possible postponement of destination arrivals. Additionally, a scenario with a less restrictive service constraint is considered, where t_r is set to 40 minutes. Overall, 100 instances are created, each covering a planning period of 3 hours, for which a constant surplus demand is assumed. For each instance, 800 out of the 1600 generated requests are randomly drawn as well as their time of request submission s_r .

The parameters of our LMNS have been adjusted on the basis of initial experiments with a computing time limit of 20 milliseconds. This involves tuning the parameters of the removal and insertion operators as well as the general search parameters. For the operators, the noise influence of Worst and Shaw removal as well as Regret-2 insertion have been adjusted; the noise of the Shaw-removal has been set to zero. For the other two operators, relatively high values have turned out to provide good solutions. In addition, the weighting of the similarity measurement factors has been adjusted for the Shaw removal operator. In particular, the planned pick-up and drop-off times have received a high weighting as a measure of similarity. As further search parameters, the following parameters were adjusted: α and β for the acceptance criterion, γ_1 and γ_2 , representing the minimum and maximum percentage of requests to be removed per iteration. The resulting values indicate that a higher degree of diversification is beneficial for the successful application of the developed insertion heuristic.

2.3.2 Results

The aim of the computational experiments is to obtain insights on the performance of the LMNS in the light of challenging computing time restrictions. For this purpose,

computing time limits between one millisecond and one second are examined. We will analyze the number of the accepted requests as an important metric for the service operator and try to understand further effects of the computing time limits on the performance of the LMNS. Then, we will examine the effect of the computing time limits and the associated changes on quality measures from the customer's point of view.

The obtained results are shown in Table 2.1, which gives the computing time limit in milliseconds, the average number of accepted requests, the average number of requests available for rescheduling, and the average number of iterations by rejection/acceptance. The column "requests for rescheduling" provides information about the average number of requests available for removal in one iteration of the LMNS. The columns "number of iterations by rejection" and the "number of iterations by acceptance" show how many iterations of the LMNS were performed on average before the search was terminated. For "iterations by rejection", the instances in which the computing time limit terminates the search are included, while for "iterations by acceptance", those in which a new feasible solution has been found.

The obtained results show that on average number of passenger requests between 705.7 and 708.3 can be accepted. The performance is best for a computing time limit of 40 milliseconds. While the number of accepted customers remains quite stable, the average number of requests available for rescheduling as well as the average number of iterations by rejection and acceptance grows continually with an increasing computing time limit.

While the increasing computing time does not create a performance gain with the number of accepted requests, the other three metrics show how the additional computing time effects the search. For instance, the increase of iterations by rejection and acceptance shows that not only more iterations can be performed, but also that requests can be accepted in later iterations which would have been rejected with lower computation time limits. For example, with a limit of 1 millisecond, it is not possible to accept requests that require more than one iteration, while with a limit of 100 milliseconds, an average of 2.5 iterations can be performed until a feasible solution is found. Note that the LMNS spends the same effort for each request so that the probability of accepting demanding requests increases also with the computing time limit. Furthermore, the increase in the average number of requests available

for rescheduling shows that with additional computing time, the number of requests increases which can be integrated simultaneously into the current planning horizon. That these do not lead to a higher number of accepted requests can be explained by the fact that, due to the high demand, there is a sufficient number of easy-to-integrate mobility requests at hand.

| Time limit in milliseconds | Accepted requests | Requests for rescheduling | Iterations by rejections | Iterations by acceptance |
|----------------------------|-------------------|---------------------------|--------------------------|--------------------------|
| 1 | 705.7 | 27.6 | 1.0 | 1.0 |
| 20 | 707.4 | 32.1 | 7.7 | 1.4 |
| 40 | 708.3 | 33.6 | 14.8 | 1.7 |
| 60 | 707.2 | 33.9 | 21.9 | 1.9 |
| 80 | 708.0 | 34.3 | 28.5 | 2.2 |
| 100 | 706.5 | 34.6 | 35.6 | 2.5 |
| ⋮ | | | | |
| 1000 | 706.8 | 36.8 | 300.7 | 12.3 |

Table 2.1: Average Statistics for Different Computing Time Limits

Table 2.2 contains the metrics related to the quality of service for the same computing time limits known from Table 2.1. The table contains the “average pooling rate”, the “average waiting time”, the “average detour” and the value of the overall “postponement of destination arrivals”. The pooling rate states the percentage of passengers which share a part of their ride with passengers of another request. The waiting time corresponds to the average difference between the time of request and the pick-up time, while the detour corresponds to the average difference between the time spent by the passenger in the vehicle and the duration of the direct journey. The postponement of destination arrivals results from the addition of the waiting time and detour.

Generally, we can see that the metrics seem to evolve consistently with higher computing time limits. The average number of passengers sharing a part of their ride as well as the average detour decrease while the average waiting time increase. These trends reflect the increasing chance of successful rescheduling. This leads, on the one hand, to an increased probability of accepting requests that are more difficult to consolidate, resulting in decreasing pooling rates and detours. On the other hand, it increases waiting times, since the number of simultaneously integrated requests

2.3 Computational experiments

| Time limit in milliseconds | Pooling rate | Waiting time in minutes | Detour in minutes | Postponement of destination arrivals |
|----------------------------|--------------|-------------------------|-------------------|--------------------------------------|
| 1 | 78.4% | 11.3 | 3.88 | 15.20 |
| 20 | 76.1% | 11.91 | 3.66 | 15.57 |
| 40 | 75.0% | 12.05 | 3.60 | 15.66 |
| 60 | 74.7% | 12.10 | 3.59 | 15.69 |
| 80 | 74.5% | 12.12 | 3.58 | 15.70 |
| 100 | 74.2% | 12.20 | 3.56 | 15.76 |
| ⋮ | | | | |
| 1000 | 72.7% | 12.49 | 3.51 | 15.91 |

Table 2.2: Average Impact of the Computing Time Limits on the Service Quality

increases, too. The values of the average overall postponement of destination arrivals show that the increasing waiting times cannot be offset by the decreasing detours so that the quality of service is marginally lower for higher computing times. Overall, in this scenario, our LMNS works well even for very small computing times, and larger computing times are neither beneficial for the number of accepted requests nor for service quality metrics.

We also want to evaluate the performance of our LMNS relative to the MIC heuristic. To this end, we use the results created at a computing time limit of 1 millisecond, which can be interpreted as the lower performance bound of the developed LMNS. The results shown in Table 2.3 distinguish between the first scenario with a maximum tolerance t_r for postponed destination arrivals of 20 minutes and the second less restrictive scenario with $t_r = 40$ minutes as well as between the MCI and LMNS approach. The metrics shown in the table correspond to those explained in context of Table 2.1 or Table 2.2.

| Scenario | Approach | Accepted requests | Pooling rate | Waiting time in minutes | Detour in minutes | Postponement of destination arrivals |
|--------------------|----------|-------------------|--------------|-------------------------|-------------------|--------------------------------------|
| $t_r = 20$ minutes | MCI | 685.0 | 69.3% | 13.05 | 3.14 | 16.19 |
| | LMNS | 705.7 | 78.4% | 11.31 | 3.88 | 15.20 |
| $t_r = 40$ minutes | MCI | 740.3 | 78.9% | 26.71 | 6.47 | 33.18 |
| | LMNS | 792.5 | 91.7% | 17.50 | 9.15 | 26.65 |

Table 2.3: Comparison of the Average Results of MCI and LMNS

First, we compare the results for the short maximal postponement duration of 20 minutes. In this context, LMNS is able to accept 20 requests (or 3%) more than MCI. Furthermore, the average pooling rate shows that LMNS achieves a 9.1% increase in the number of shared rides compared to the MCI. The results of MCI and LMNS also differ in terms of service quality metrics. The average waiting time is higher for the MCI result, and the average detour is longer in case of the LMNS result. However, the values of the average postponement of destination arrivals show that passengers reach their destination faster if the planning is done by the LMNS.

These results show that even with low flexibility due to a restrictive service level and an extremely limited computing time, the rescheduling of the developed LMNS enables the acceptance and successful consolidation of more mobility requests as well as the improvement of the service quality compared to MCI.

Finally, we examine results for a higher maximal postponement value of 40 minutes. As expected, the flexibility gained from the relaxation of the service restriction allows more mobility requests to be accepted on average. The average number of accepted requests increases by 12.3%/8.1% for LMNS and MCI, respectively. For LMNS, whereby 99.1% of all requests are accepted. The high flexibility also allows a more efficient consolidation of rides, so that the average pooling rate increases by 13.3%/9.6%. These improvements are at the cost of significantly longer average waiting times and detours. The results show, on average, an increase of the waiting time by 54.7%/104.7% and the detour by 135.8%/106.1%, which leads to an increase of the average postponement of destination arrivals by 75.3%/105.4% for LMNS and MCI, respectively.

The differences observed in the first scenario thus intensify in the second scenario for all metrics considered. The additional planning flexibility associated with relaxation of the service restriction thus increases the advantages of LMNS over MCI. Hence, a continuous rescheduling of the LMNS seems to be quite beneficial. In conclusion, if the design of the ride-sharing service allows for continuous rescheduling, LMNS can take advantage of that and provide better results than MCI.

2.4 Conclusion

Request acceptance is a core task for on-demand ride-sharing services operators. In this paper, we proposed an LMNS as a complex feasibility check and rescheduling

mechanism to solve the DDARP in the context of ride-sharing services. It could be demonstrated that the developed LMNS can be used for such services under the demanding requirements of tight computing time constraints. Furthermore, we could show that the developed LMNS is superior to a standard approach from the literature both in terms of the average number of requests accepted and in terms of the average quality of service.

In future work, it is our ambition to integrate the proposed LMNS into the simulation framework “MatSim” (Horni et al., 2016) to evaluate it under more realistic conditions. In addition, a comparison with other feasibility checks known from the literature is planned. Future research will also focus on profit-maximizing implementation of a ride-sharing service, especially with regards to service quality and reliability. In particular, the anticipation of future requests will be of interest, since this could lead to a higher acceptance rate through rejection of unfavorably located requests and smart relocation of idle vehicles.

Chapter 3

Effectiveness of demand and fulfillment control in dynamic fleet management of ride-sharing systems

Abstract

In recent years, innovative ride-sharing systems have gained significant attention. In such systems, dynamic fleet management covers demand and fulfillment control to determine which stochastically incoming requests are to be satisfied and how vehicle resources are utilized for their fulfillment, respectively. Demand and fulfillment control can be implemented ranging from straightforward myopic to more sophisticated anticipatory. In this paper, our aim is twofold: (1) we want to classify how policies implement demand and fulfillment control in the related literature on dynamic fleet management; (2) we want to explore the effectiveness of demand and fulfillment control under varying conditions in order to identify benefits and risks for ride-sharing systems. To this end, we define policies that differ in the optimization of demand and/or fulfillment control through the exploitation of either confirmed or complete information. Our experimental results demonstrate that demand and fulfillment control affect the performance and service quality of ride-sharing systems quite differently.

Keywords anticipation, dial-a-ride problem, dynamic vehicle routing, large neighborhood search ride-sharing, stochastic requests

3.1 Introduction

Worldwide increasing congestion in urban traffic networks and the associated air pollution have led to a growing interest in innovative shared mobility solutions. Among these are on-demand ride-sharing services like Uber Pool (n.d.), which promise to improve the efficiency of traditional taxi services by bundling travelers on the way from their origin to their destination. This increased level of efficiency allows for lower fares compared to individual taxi services and enables a more convenient travel experience compared to traditional local public transport through smaller transport cabins and direct trips.

Ride-sharing services have emerged in the light of advancing digitization, which allows travelers to submit requests on-demand. The resulting interaction of request acceptance and vehicle routing poses a great challenge on operators, as requests arrive stochastically and decisions have to be made dynamically. In request acceptance, trip requests submitted by travelers – often in expectation of instant confirmation – are processed. Here, it must be ensured that all accepted requests can be fulfilled with the given vehicle resources. Operators may also reject requests due to a lack of resources or in favor of potential future ones. At the same time, vehicle routing addresses the utilization of the fleet to fulfill accepted requests as well as those expected to be accepted in the future. Commonly, accepted requests must be fulfilled at short notice, which means that planning and execution are performed synchronously.

Dynamic fleet management, which comprises request acceptance and vehicle routing to control the demand to be satisfied and its fulfillment, is a key factor for a successful ride-sharing system. To handle the uncertainty caused by the stochastic nature of requests, *demand control* as well as *fulfillment control* can range from simple exploitation of confirmed information only to more sophisticated exploitation of information on the stochastic problem elements. Complex strategies for both demand and fulfillment control are reflected in the decision-making policies proposed in the literature. Given these policies, it remains unclear what extent of demand and fulfillment control is beneficial under which conditions and how it affects the performance of ride-sharing systems. Understanding the effectiveness of demand and fulfillment control is important, both for the systematic development of policies as well as for guiding operators in selecting a policy tailored to the intended service.

Our aim is to investigate the effectiveness of demand and fulfillment control in a ride-sharing system systematically. To this end, first, we define control strategies and classify the policies proposed in the literature accordingly. Secondly, we present policies that vary in the complexity of optimization in demand and/or fulfillment control. Finally, based on a comprehensive computational study, we analyze how these policies affect the performance metrics of a typical urban ride-sharing system as well as the quality of service perceived by travelers.

In our computational study, for the most part, we assume complete information to increase optimization possibilities instead of using truly anticipatory policies. Therefore, we are able to interpret results independently of anticipation capabilities and the quality of information on the stochastic demand. However, concerning the stochastic-dynamic problem, rather upper bounds of effectiveness are investigated than those obtainable by truly anticipatory policies. Nevertheless, it can be assumed that trends in performance differences will be reflected through similar patterns by policies that apply the same strategies towards demand and fulfillment control. Furthermore, to ensure the comparability of the implemented policies, all acceptance and routing decisions are obtained by solving variants of the static dial-a-ride problem (DARP) using a well-known large neighborhood search (LNS).

In summary, we contribute to a better understanding of demand and fulfillment control in dynamic fleet management. To this end, we give an overview of corresponding policies as proposed in the related literature. Moreover, we provide valuable insights into the overall effectiveness of demand and fulfillment control under various conditions of a ride-sharing system.

The paper is organized as follows. Section §2 defines demand and fulfillment control in dynamic fleet management, differentiates control strategies, and provides a classification of the related literature. In Section §3, the dynamic DARP (DDARP) faced by a ride-sharing system is presented and modeled as a Markov decision process. Section §4 covers the framework for investigating the policies as well as the presentation of the LNS. In Section §5, computational experiments are presented including study design and computational results. Finally, Section §6 provides a conclusion and outlines future research directions.

3.2 Demand and fulfillment control

The differentiation between demand and fulfillment control in dynamic fleet management is a key aspect of this paper. We define *demand control* as all dynamic decisions aimed at controlling which trip requests are to be satisfied. We define *fulfillment control* as all dynamic decisions aimed at controlling vehicle resources to fulfill the trip requests to be satisfied. Apart from this logical differentiation, there is a strong interdependence between demand and fulfillment control, since demand control determines the input for fulfillment control and fulfillment control strongly influences the availability of vehicle resources important for demand control. In the following, we detail this differentiation and then classify the policies proposed in the literature.

3.2.1 Control strategies

For both demand and fulfillment control, we begin with differentiating whether these are reflected in the decision-making process at all. This distinction is rooted in the few variants of the dynamic vehicle routing problem (DVRP) where either only demand or only fulfillment is dynamically controlled. Secondly, we distinguish how uncertainty is handled within the decision process. Uncertainty is a key challenge in dynamic fleet management caused by the stochastic nature of requests, implying that decisions are made based on incomplete information. We define *anticipatory* decision-making as the consideration of future stochasticity in order to maximize the expected cumulative reward. Anticipatory decision-making can be based on historical data, forecasts, or the distribution of the stochastic elements. In contrast, decision-making that maximizes immediate rewards only based on confirmed information is referred to as *myopic*. In the context of dynamic fleet management, a reward refers, for example, to an accepted request or its monetary compensation. Demand and fulfillment control can be carried out in a myopic or anticipatory manner. This differentiation leads to the following super-ordinate demand and fulfillment control strategies further detailed in the subsections:

- Demand control:
 - *None*: All requests received during the planning period are accepted.
 - *Myopic*: Requests received during the planning period are accepted if sufficient vehicle resources are available and this maximizes the immediate reward.
 - *Anticipatory*: Requests received during the planning period are accepted if sufficient vehicle resources are available and this maximizes the expected cumulative reward.
- Fulfillment control:
 - *None*: Requests accepted during the planning period are scheduled and fulfilled after its completion.
 - *Myopic*: Requests accepted during the planning period are scheduled and fulfilled synchronously within the period.
 - *Anticipatory*: Requests accepted during the planning period are scheduled and fulfilled synchronously within the period, taking into account expected future acceptances.

3.2.1.1 Demand control

For the scope of demand control, in this section, we detail the meaning of the strategies referred to as “none”, “myopic” and “anticipatory”. “None” means that all incoming requests will be accepted. To ensure this, corresponding DDARPs do not consider hard constraints on the quality of service in terms of maximum waiting time for travelers. Instead, service quality becomes part of the objective function in order to achieve a convenient service for all requesting travelers. In practice, however, the demand is usually still controlled indirectly, either on a strategic level through determining a suitable service area, or through inconvenient service offers with long waiting times.

In contrast, myopic and anticipatory demand control accept only a subset of the incoming requests under the objective of maximizing the number of accepted requests or the revenue. This is usually accompanied by strict constraints on the quality of service so that the amount of feasible demand is limited. A key feature of these demand control strategies are customer acceptance mechanisms through which dynamic acceptance decisions are made for each incoming request. The basis for these

mechanisms is the so-called *feasibility check*, which ensures that a feasible route plan can be found at any state of the decision process. In the myopic case, demand control is often limited to feasibility checks, as any additional accepted request increases the immediate reward (see, e.g., Coslovich et al. (2006)). Enhanced myopic demand control additionally proactively rejects requests if they seem unfavorable at the time of request, e.g., in terms of incremental transportation costs (see, e.g., Xiang et al. (2008)).

Anticipatory demand control can be implemented in the following ways. The first is through sophisticated customer acceptance mechanisms, which extend the feasibility check to an anticipatory acceptance decision. This ensures that only requests that seem favorable with respect to the expected cumulative reward will be accepted. Whether a request is favorable or not is either reflected in its expected vehicle resource consumption (e.g., Ulmer et al. (2018) for a related DVRP) or quantified by a revenue management approach (e.g., Yang & Strauss (2017) for a related DVRP). The second way is through the combination of feasibility checks and proactive allocation of vehicle resources by anticipatory routing decisions. Such decisions may concern the relocation of idle vehicles (e.g., Horn (2002)) or the incorporation of dummy requests to reflect future expected ones within a scenario-based approach (e.g., Ichoua et al. (2006)). As a result, the feasibility check is only successful for favorable requests to which vehicle resources have been allocated. However, the effectiveness of such demand control is strongly dependent on the extent of anticipatory routing decisions and the strictness of service quality constraints.

3.2.1.2 Fulfillment control

The classification of fulfillment control into the presented strategies follows the implementation of the routing decisions. Therefore, in case of no control (“none”), no dynamic routing decisions are made. Such policies can be found in the context of reservation systems, where the fulfillment is carried out after a booking process has been completed. Such reservation systems have been investigated in the context of time slot management for attended home deliveries (e.g., Campbell & Savelsbergh (2005), Ehmke & Campbell (2014)).

When routing decisions are made dynamically as required in on-demand systems, fulfillment control is often based on myopic re-optimization of vehicle route plans

in the event of a newly accepted request. This re-optimization can be achieved by applying (meta)-heuristics initially developed to solve a static vehicle routing problem (e.g., in Attanasio et al. (2004)), or through newly developed routing algorithms (e.g., in Alonso-Mora et al. (2017a)).

Anticipatory fulfillment control enhances myopic strategies by considering expected future request acceptances in routing decisions. However, the specific approach can differ greatly in terms of complexity and comprehensiveness. For example, waiting strategies determine at which locations vehicles should wait in order to efficiently accommodate future requests (e.g., Branke et al. (2005)). In relocation strategies, vehicles are relocated within the service area for the same purpose (e.g., Horn (2002)). The first two examples therefore do not interfere directly with the key task in fulfillment control, namely route planning. In contrast, (multiple)-scenario approaches often plan anticipatory routes with the help of dummy requests (e.g., Ichoua et al. (2006)). Furthermore, for example, approaches of approximate dynamic programming (ADP) provide comprehensive anticipatory fulfillment control for rather small problem instances (e.g., Yu & Shen (2020)).

3.2.2 Classification of the related literature

In this section, we provide an overview of the related research on dynamic fleet management. In particular, we classify the proposed policies according to the introduced strategies for demand and fulfillment control (see Table 3.1). To provide a broad overview, we complement the primarily covered papers dealing with DDARP with related ones dealing with customer acceptance mechanisms or DVRP. For a comprehensive literature review on the DARP, we refer to Molenbruch et al. (2017) and Ho et al. (2018). For the DVRP, we refer to Psaraftis et al. (2016) and Ritzinger et al. (2016).

The first studies on dynamic fleet management of a ride-sharing system were conducted by Dial (1995) and Madsen et al. (1995) in 1995. Dial (1995) decomposes the problem into a set of travelling salesman problems, while Madsen et al. (1995) suggest an insertion heuristic in order to solve the DDARP. These first contributions focus on myopic fulfillment control without considering demand control. More confirmed policies following this control structure can be found in Ma et al. (2013) and Riley et al. (2019). Ma et al. (2013) compute solutions for large ride-sharing systems

by performing a grid-based service area decomposition. Riley et al. (2019) propose a column generation based policy to satisfy all incoming requests under the objective of minimizing waiting times.

| | | Demand control | | |
|---------------------|--------------|---|--|--|
| | | None | Myopic | Anticipatory |
| Fulfillment control | None | | Campbell & Savelsbergh (2006) Ehmke & Campbell (2014) Cwioro et al. (2019) | Campbell & Savelsbergh (2005) Yang et al. (2016) Yang & Strauss (2017) Mackert (2019) |
| | Myopic | Dial (1995) Madsen et al. (1995) Ma et al. (2013) Riley et al. (2019) | Attanasio et al. (2004) Coslovich et al. (2006) Xiang et al. (2008) Beaudry et al. (2010) Berbeglia et al. (2011) Berbeglia et al. (2012) Hosni et al. (2014) Alonso-Mora et al. (2017a) Lowalekar & Jaillet (2019) | Ulmer et al. (2018) Ulmer et al. (2019) |
| | Anticipatory | Mitrović-Minić & Laporte (2004) Schilde et al. (2011) Hyytiä et al. (2012) Riley et al. (2020) | Branke et al. (2005) Thomas (2007) | Horn (2002) Ichoua et al. (2006) Alonso-Mora et al. (2017b) Shah et al. (2020) Yu & Shen (2020) Pouls et al. (2020) |

Papers dealing with demand and fulfillment control in a dynamic dial-a-ride problem are written in bold.

Table 3.1: Literature Classification

The policy introduced in Riley et al. (2019) was later extended in Riley et al. (2020) to an anticipatory fulfillment control through a periodic relocation of vehicles in idle mode by means of demand forecasts. More comprehensive anticipatory fulfillment is proposed by Schilde et al. (2011) and Hyytiä et al. (2012), again without considering demand control. Schilde et al. (2011) adapt a multiple-scenario approach, originally introduced by Bent & van Hentenryck (2004) for the DVRP, using a variable neighborhood search. Hyytiä et al. (2012) propose a theoretical approach combining Markov decision processes and M/M/1 queues to develop an anticipatory policy focusing on fulfillment control for the single-vehicle case.

Anticipatory fulfillment control via waiting time strategies are primarily investigated within the scope of the general DVRP. For example, Mitrović-Minić & Laporte (2004) analyze waiting time strategies under the objective of minimizing travel times, while Branke et al. (2005) and Thomas (2007) maximize the number of accepted requests. Both papers establish demand control through a myopic feasibility check by means of an insertion heuristic. Here, the waiting time strategies themselves are not considered as anticipatory demand control, since stops along the route at which waiting is feasible results from the myopic feasibility checks.

The counterpart to policies that focus only on fulfillment control are customer acceptance mechanisms that focus only on demand control. These mechanisms are primarily examined concerning the problem of managing delivery slots in attended home deliveries. Myopic customer acceptance mechanisms are presented for example in Campbell & Savelsbergh (2006), Ehmke & Campbell (2014), and Cwioro et al. (2019). Campbell & Savelsbergh (2006) examine financial incentives to encourage customers to choose a delivery time slot that is favorable in terms of myopically planned delivery routes. Ehmke & Campbell (2014) compare simple static and dynamic customer acceptance mechanisms under consideration of stochastic travel times. Cwioro et al. (2019) propose an adaptive large neighborhood search (ALNS) for the feasibility check to maximize the number of time slots that can be offered. Beyond that, several anticipatory demand control approaches have been considered. Campbell & Savelsbergh (2005) adapt the multiple-scenario approach introduced by Bent & van Hentenryck (2004) as a customer acceptance mechanism applying an insertion heuristic. Yang & Strauss (2017) propose a pricing approach based on approximate dynamic programming using a sophisticated customer choice model developed in Yang et al. (2016). More confirmedly, Mackert (2019) approximates opportunity costs using a mixed-integer linear program. Furthermore, the pricing of individual and shared rides for a ride-sharing system is examined in Qiu et al. (2018), assuming complete information. Apart from slotting problems, ADP-based customer acceptance mechanisms have also been investigated by Ulmer et al. (2018) and extended by Ulmer et al. (2019), taking into account fulfillment control. In these two examples, demand control assesses incoming requests with respect to their long-term vehicle resource demand, while fulfillment is myopically controlled using an insertion heuristic.

The papers reviewed above demonstrate that the implementation of demand and fulfillment control within and between DVRPs varies greatly. In the following, we list the variants most relevant for our study, which considers demand and fulfillment control in the scope of a DDARP. Many of those papers propose purely myopic policies. In these cases, well-known solution methods are used to perform a quick feasibility check in demand control as well as to re-optimize route plans in fulfillment control. Such a policy is proposed by Attanasio et al. (2004) applying a parallel tabu search (TS) for both tasks, by Coslovich et al. (2006) through a two-stage insertion heuris-

tic, by Beaudry et al. (2010) through an insertion heuristic for the feasibility check and a TS for re-optimization, and by Berbeglia et al. (2011) proposing constraint programming for the feasibility check, which in Berbeglia et al. (2012) is extended to a combination of TS and constraint programming.

Some papers propose improved myopic demand control in a DDARP context. The idea is that only cost-effective requests are accepted. This was first discussed for such a problem by Horn (2002), yet dismissed due to the potential unfairness towards requests with certain characteristics. Potential discrimination of requests is therefore one aspect that will be examined in our computational study. Xiang et al. (2008) and Hosni et al. (2014) proposed a policy that proactively rejects requests that seem cost-ineffective myopically. To this end, both papers check feasibility and whether the incremental costs exceed a threshold value. Xiang et al. (2008) implement this with an insertion heuristic. Hosni et al. (2014) introduce a model-based approach that integrates each incoming request into the incumbent route plan at minimal incremental costs. Furthermore, Alonso-Mora et al. (2017a) and Lowalekar & Jaillet (2019) improve demand control by postponing acceptance decisions until a batch of requests has been received, even if this does not allow for instant trip confirmations. Here, all dynamic decisions are made within a two-stage process. In the first step, a set of potential routes is created, before in the second step an assignment problem is solved to select the routes to be realized under the objective of maximizing the number of accepted requests. While Alonso-Mora et al. (2017a) determine all feasible combinations of unfulfilled requests in the first step, Lowalekar & Jaillet (2019) only consider promising ones with the help of a zone path construction approach.

For the DDARP, several policies have been proposed that implement anticipatory demand control and fulfillment control via routing decisions. A first policy presented by Horn (2002) involves the relocation of idle vehicles. More confirmedly, Pouls et al. (2020) proposed a policy focusing primarily on anticipatory relocation. In contrast, Ichoua et al. (2006), Alonso-Mora et al. (2017b), , Shah et al. (2020) and Yu & Shen (2020) focus on the anticipatory planning of vehicle routes. Ichoua et al. (2006) adapt a multi-scenario approach using TS. Alonso-Mora et al. (2017b) extends Alonso-Mora et al. (2017a) by incorporating expected future requests via dummy requests. Another extension of Alonso-Mora et al. (2017a) towards anticipation is proposed by Shah et al. (2020), who use ADP for the selection of routes within the allocation

problem. ADP is also used by Yu & Shen (2020) to solve the DDARP in connection with a decomposition of the problem.

This literature review summarized the different strategies for demand and fulfillment control and how they are implemented through proposed policies. In contrast to the presented literature, with our work, we provide a comparative meta-analysis of demand and fulfillment control to investigate their effectiveness for ride-sharing systems. Moreover, we show under varying system conditions when and how fleet management benefits from which degree of control. With all this, we want to contribute to a better understanding of dynamic fleet management in ride-sharing systems and encourage the systematic development and selection of policies concerning their intended effectiveness.

3.3 Problem formulation

In this section, we define the components of the DDARP under consideration. Then, we model the stochastic and dynamic problem as a Markov decision process, enabling demand and fulfillment control in a ride-sharing system through dynamic acceptance and routing decisions.

3.3.1 Problem components

Let \mathcal{L} be a set of locations in the service area of a ride-sharing system. For each location $l \in \mathcal{L}$, it is assumed that a (deterministic) service time p_l for travelers getting on and off a vehicle is known, as well as for all pairs of locations $(i, j) \in \mathcal{L}$, a (deterministic) travel time of $c_{i,j}$ is defined. The considered ride-sharing system faces a demand represented by trip requests $r \in \mathcal{R}$. Each request is characterized by its receiving time t_r , its origin $o_r \in \mathcal{L}$, its destination $d_r \in \mathcal{L}$, as well as its fulfillment time window $[b_r, e_r]$, which defines the earliest pick-up time b_r and latest drop-off time e_r . We assume that the earliest pick-up time b_r corresponds to the receiving time of the request t_r . This means that travelers must be ready for departure at the time when they pose their request, which excludes pre-bookings. The latest drop-off time e_r is defined by addition of earliest pickup time b_r , direct travel time c_{o_r, d_r} , and a parameter α , which defines the maximum arrival delay tolerated by travelers. Arrival delays arise from waiting time to be picked up as well as detours caused through the

bundling of requests. Detours include both additional travel time to reach the origin or destination of other travelers and the service time required by them for getting on or off the vehicle. To satisfy the demand, a fleet of identical vehicles \mathcal{V} is available. We assume that the capacity of a vehicle is not constraining, i.e., passenger seats are never fully occupied due to tight time windows for the request fulfillment.

3.3.2 Markov decision process

The considered decision process consists of a series of decision epochs $k \in \mathcal{K}$, covering a temporally limited planning period of a DDARP. At the beginning of the planning period, the service is in an initial state s_0 . For this state, we assume that the vehicles $v \in \mathcal{V}$ are waiting in idle mode at an initial location $l_v \in \mathcal{L}$. Furthermore, it is assumed that in the initial state s_0 no trips are waiting for fulfillment. Accordingly, a degree of dynamics as defined by Lund et al. (1996) as the ratio of the demand stochastically received to the total demand, of 100% is assumed. Each decision epoch $k \in \mathcal{K}$ is triggered by a stochastically incoming request $r_k \in \mathcal{R}$ leading to a pre-decision state s_k . The pre-decision state reflects all decision-relevant characteristics such as the activities of the vehicles and pending requests. Formally, the pre-decision state s_k is defined by the time t_r at which the service operator has received the new request r_k . Furthermore, it contains the state of the resources described through the tuple $(l_k^v, \mathcal{O}_k^v | \forall v \in V)$, where $l_k^v \in \mathcal{L}$ specifies the current vehicle locations and $\mathcal{O}_k^v \subset \mathcal{R}$ for each vehicle the set of accepted requests whose travelers are currently being transported. Finally, it represents the demand described through the tuple (r_k, \mathcal{U}_k) , where r_k refers to the new request and $\mathcal{U}_k \subset \mathcal{R}$ to the set of accepted requests whose travelers still have to be picked up. These three parts result in the state definition $s_k = (t_r, (l_k^v, \mathcal{O}_k^v | \forall v \in V), (r_k, \mathcal{U}_k))$.

Based on the pre-decision state s_k , an action $A^\pi(s_k)$ is derived from a policy $\pi \in \Pi$. A policy $\pi \in \Pi$ thus defines for each pre-decision state s_k all decisions to be taken and can thus be considered as a solution approach to the stochastic-dynamic problem. Here, an action consists of two hierarchically dependent decisions. The first decision is whether to accept or reject the new request r_k . This acceptance decision is represented by the binary decision variable $x_k \in \{0, 1\}$, where $x_k = 1$ represents acceptance and $x_k = 0$ represents the rejection of a request. The second decision is the selection of a feasible route plan, defining the utilization of all vehicles $v \in \mathcal{V}$ until

the next decision epoch. A route plan is considered feasible if all accepted requests have been assigned to a vehicle subject to the following constraints:

- i) For all pending accepted requests $r \in \mathcal{U}_k$ and the new request r_k , in case of $x_k = 1$, the pick-up at origin o_r is planned before the drop-off at destination d_r for the same vehicle $v \in \mathcal{V}$.
- ii) For all currently executed requests $r \in \mathcal{O}_k^v$, the drop-off at destination d_r is planned for the same vehicle $v \in \mathcal{V}$.
- iii) For all origins, the planned pick-up z_o is later or at the same time as the corresponding earliest pick-up time b_r .
- iv) For all destinations, the planned drop-off z_d is earlier or at the same time as the corresponding latest drop-off time e_r .

Let $y_k \in \mathcal{F}_x$ be the routing decision variable, with \mathcal{F}_x as a finite set of all feasible route plans under consideration of decision x_k . Such a set could for example be determined heuristically by adapting a solution method developed for a static vehicle routing problem. The acceptance decision x_k requires that the set of all route plans \mathcal{F}_x must not be empty. The execution of action $A^\pi(s_k)$ leads to a deterministic transition from the pre-decision state s_k to a post-decision state $s_k^a = (y_k)$. This state consists of the selected feasible route plan y_k , which serves for the routing of the vehicles until the next decision epoch $k+1$. This is triggered by a stochastic transition W_{k+1} , which reflects that the operator has received the next request $r_{k+1} \in \mathcal{R}$.

Let B_k be the partial reward function for one decision epoch $k \in K$ and let the value of B_k be equal to the acceptance decision x_k , so that the cumulative reward $v^\pi(s_0)$ corresponds to the number of accepted requests. The objective is to find an optimal policy $\pi^* \in \Pi$ that maximizes the expected cumulative reward $v^\pi(s_0) = \max^\pi \mathbb{E}\{\sum_{k=0}^K B_k(s_k, A^\pi(s_k), W_{k+1}) | s_0\}$ over all decision epochs $k \in \mathcal{K}$. Having formally introduced the stochastic-dynamic problem under consideration, in the next section, we present policies that exploit only confirmed or complete information to solve the problem with varying degrees of optimization in demand and/or fulfillment control.

3.4 Evaluation framework

In this section, we describe our evaluation framework for investigating the effectiveness of demand and fulfillment control within dynamic fleet management of a ride-sharing system. Using the control strategies discussed in Chapter 2, we detail the implemented policies in Section 4.1 and discuss an established LNS that we use to realize them in Section 4.2.

3.4.1 Implementation of policies

From the control strategies defined in Section 2.1, we derive possible combinations of demand and fulfillment control to define policies for dynamic fleet management. These policies differ in optimization capabilities in demand and/or fulfillment control through the exploitation of confirmed information only or complete information. Table 3.2 summarizes the policies with regard to the related control strategy.

| | | Demand control | |
|---------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | | Inspired by myopic strategies | Inspired by anticipatory strategies |
| Fulfillment control | Inspired by myopic strategies | <i>Basic Control</i> | <i>Advanced Demand Control</i> |
| | Inspired by anticipatory strategies | <i>Advanced Fulfillment Control</i> | <i>Advanced Control</i> |

Table 3.2: Investigated Policies

Basic Control refers to policies employing purely myopic demand and fulfillment control. They are implemented through a feasibility check for demand control and re-optimization of routes for fulfillment control. In particular, the feasibility check for the acceptance decision x_k is made by an insertion heuristic, checking whether an incoming request $r_k \in \mathcal{R}$ can be inserted into the incumbent route plan y_{k-1} , where y_0 refers to the initial empty route plan. The route plan obtained is then re-optimized in the scope of the routing decision y_k . For this purpose, a static DARP is solved considering all accepted, not yet fulfilled requests under the objective of minimizing total travel time. Note that for both acceptance and routing decisions, already fulfilled requests as well as locations approached by a vehicle will not be rescheduled. This means that vehicles are not tracked along the path between two locations $i, j \in$

\mathcal{L} , which reduces the rescheduling opportunities but also the computational effort related to locating vehicles. Moreover, it allows drivers and travelers to be reliably informed about the next stop, avoiding frequent diversions of vehicles.

From an operator perspective, *Basic Control* could be advantageous in case of highly uncertain conditions, where the inclusion of additional information in optimization does not pay off. Moreover, *Basic Control* does not require any sophisticated technological and computational resources. The key argument against *Basic Control* is the high risk of insufficiently informed decision-making both for demand and fulfillment control.

Advanced Demand Control aims at improving the performance of a ride-sharing system through the acceptance of favorable requests in terms of vehicle resource occupancy (e.g., requests which can be bundled more easily). It is inspired by policies that implement anticipatory demand control through sophisticated customer acceptance mechanisms while fulfillment is controlled myopically. Therefore, complete information is exploited to enable enhanced acceptance decisions x_k , while routing decisions y_k are made based only on confirmed information.

In particular, the acceptance decision x_k is made for each incoming request $r_k \in \mathcal{R}$ in a two-step procedure. First, a feasibility check is carried out by an insertion heuristic (as in *Basic Control*). If the feasibility check has been successful, the favorability of the request is investigated in the second step. To identify favorable requests, a static team orienteering problem (TOP) with equal scores for each considered request is solved. The TOP is a well-known variant of the static vehicle routing problem, in which only the most cost-efficient locations are visited. The objective is to find the optimal set of visited locations which maximizes the operator's benefit (Chao et al., 1996). As input for the TOP serves all requests of the incumbent route plan y_{k-1} , the current request r_k as well as all future requests. All requests have equal scores, representing that the route plan found maximizes the number of integrated requests. All requests of the incumbent route plan y_{k-1} must be covered to identify favorable requests among current and future requests. In the end, a request r_k is accepted if it is contained in the best route plan found. After the acceptance decision has been made, a new route plan y_k is determined by solving a static DARP without taking future requests into account, following re-optimization based fulfillment control in *Basic Control*.

In summary, with *Advanced Demand Control*, a ride-sharing system operates more efficiently through the controlled selection of the requests to be satisfied. However, since we assume basic fulfillment control, vehicle resources may be utilized in an unfavorable way making beneficial demand control much more challenging. Furthermore, there are risks associated with a selective demand control such as incomprehensible rejections as well as rejections perceived as proactive, leading to dissatisfied travelers. Moreover, the continuous rejection of certain requests identified as unfavorable may prevent such trips from being requested, regardless of whether their assessment might change over time.

Advanced Fulfillment Control improves request fulfillment by considering all future request acceptances in conjunction with basic demand control. It is inspired by policies that implement anticipatory approaches for fulfillment control only.

In particular, complete information is exploited to obtain a favorable route plan in advance with respect to a feasibility check based demand control. To this end, request acceptance decisions are simulated for each future request $r \in \mathcal{R}$ in order of appearance through an insertion heuristic. However, the fulfillment of accepted requests is *not* simulated, so that the incumbent route plan can be changed flexibly throughout these checks. Once all decisions on request acceptance have been simulated, a route plan is created from the obtained acceptances as a blueprint for fulfillment control.

Summarized, *Advanced Fulfillment Control* enables an optimized fulfillment of the accepted requests without changing the concept of demand control. The acceptance of a request therefore depends primarily on the time a request is posed and not, like in case of *Advanced Demand Control*, on its characteristics. However, *Advanced Fulfillment Control* may even reinforce the drawbacks of such an basic demand control by enabling the acceptance of more demanding requests through improved vehicle routing.

Finally, *Advanced Control* exploits complete information on future demand for both demand and fulfillment control allowing all dynamic decisions to be made in advance. It is inspired by policies in which demand and fulfillment are controlled through anticipatory approaches. The exploitation of complete information is done by solving a static TOP with the same score for each request $r \in \mathcal{R}$. This results in a route plan that maximizes the number of integrated requests so that the requests

to be accepted and the routes to be taken can be selected accordingly. It therefore naturally outperforms the other three policies.

3.4.2 Large neighborhood search

In the following, we describe how the different policies are implemented based on an LNS. We apply the same heuristic to all occurring DARPs to ensure the comparability of policies within computer experiments. The developed LNS is based on the ALNS proposed by Ropke & Pisinger (2006). It was chosen because it has been applied over years to a variety of complex vehicle routing problems and has achieved consistently good results in relatively short run times.

3.4.2.1 Overview

The basic idea of an LNS is to destroy and repair solutions iteratively (Pisinger & Ropke, 2010). For the problem at hand, a solution w is represented by a route plan n_w and a set of unplanned requests $m_w \in \mathcal{R}$, whose fulfillment is not yet considered in route plan n_w . A route plan n_w consists of a plan for each vehicle $v \in \mathcal{V}$, which specifies the sequence of the locations $l \in \mathcal{L}$ to be visited as well as their planned arrival times z_l^v . The LNS aims to maximize the number of request fulfillments $|n_w|$ and/or to minimize the required total travel time $c(n_w)$. Algorithm 1 presents the pseudocode of our LNS implementation.

The search is initialized with a solution w_0 as input, which is saved as incumbent solution w and best known solution w_{best} (line 2 and 3). Next, the iterative search for a superior solution is performed until a termination criterion is met. As a termination criterion, the maximum number of iterations β is defined as well as further criteria depending on the respective purpose of the search. Each iteration of the LNS begins with the creation of a new solution (lines 5 to 7). For this purpose, the incumbent solution w is saved as the basis of the new solution w_{new} . Afterwards, w_{new} is destroyed through an operator that moves between γ_1 and γ_2 percent of the requests from the route plan $n_{w_{new}}$ to the set of unplanned requests $m_{w_{new}}$. If in the dynamic environment the origin o_r has been visited already, the corresponding destination d_r is no longer removable. The exact number of requests to be removed is determined in each iteration by a random value q_1 with $\{q_1 \in \mathbb{N} \mid (\gamma_1 \times |n_{w_{new}}|) \leq q_1 \leq (\gamma_2 \times |n_{w_{new}}|)\}$. In the next step, a repair operator inserts as many requests from the set

of unplanned requests $m_{w_{new}}$ into the route plan $n_{w_{new}}$ as feasible. For both destroy and repair operators, in contrast to a classical ALNS, the particular operator is selected randomly for each iteration. This is a consequence of the implementation of the LNS in a dynamic environment, where multiple searches are performed over a few iterations so that automatic adaptation of the operator selection during the search is neither feasible nor advantageous.

```

1 Function  $LNS(w_0)$ 
2    $w = w_0$ 
3    $w_{best} = w_0$ 
4   while termination criterion is not met do
5      $w_{new} = w$ 
6     remove requests from  $n_{w_{new}}$  to  $m_{w_{new}}$ 
7     insert requests from  $m_{w_{new}}$  into  $n_{w_{new}}$ 
8     if ( $w_{new}$  is accepted) then
9        $w = w_{new}$ 
10      if ( $w_{new}$  is an improvement to  $w_{best}$ )
11        then
12           $w_{best} = w_{new}$ 
13        end
14      end
15    return  $w_{best}$ 

```

Algorithm 1: Large Neighborhood Search

Removal operators correspond to those used in Ropke & Pisinger (2006). We summarize them as follows:

Random-removal: This operator randomly selects the requests to be removed and thus provides a maximum diversification in terms of the set of selected requests.

Worst-removal: The aim of this operator is to remove requests that are not placed well. For this purpose, all requests of a route plan are sorted in descending order in a list according to the travel time that could be saved if the request was removed. In order to avoid the repeated removal of similar sets of requests, “noise” is applied when selecting a request for removal. Following Ropke & Pisinger (2006), we use the formula $q_2^{\delta_1} \times |list|$ to determine the list position of the next request to be removed. In this formula, q_2 stands for a random value with $\{q_2 \in \mathbb{Q} | 0 \leq q_2 \leq 1\}$ and δ_1 for the parameter that controls the degree of noise.

Shaw-removal: Originally introduced by Shaw (1998), this operator removes similar requests, since they can be shuffled around more easily so that improved route plans can be found more likely. In particular, first, a request is randomly selected. All other requests are then sorted in ascending order according to their similarity to the selected request and removed corresponding to the sorting. The similarity between two requests r_1 and r_2 is calculated by the distances between origins $c_{a_{r_1}, a_{r_2}}$ and destinations $c_{d_{r_1}, d_{r_2}}$ as well as between their planned arrival times $\Delta(z_{a_{r_1}}, z_{a_{r_2}}) + \Delta(z_{d_{r_1}}, z_{d_{r_2}})$. Before the geographical and temporal values are added up, they are min-max normalized.

For the subsequent insertion of the removed requests, there is a wide range of operators. We discuss only those operators that turned out promising in previous tests, one with and one without noise:

Regret-2-insertion: The regret-insertion heuristic was first proposed by Potvin & Rousseau (1993) for the vehicle routing problem with time windows. The idea is to insert requests where the regret would be largest if the best found insertion option was no longer feasible. An insertion option comprises a position in a route for the origin and the destination of a request. For the regret-2 variant, the regret is calculated from the difference between the most and the second most cost-effective feasible insertion option. The costs correspond to the additional travel time which would result from the request being inserted. In case that only one feasible insertion option can be found, the difference to the maximum integer value is calculated instead. For each selection of the next request to be inserted in the route plan, the regret value of each unplanned request $r \in m_{w_{new}}$ is calculated and sorted in descending order. For the operator without noise, the request with the highest regret value is inserted into the most cost-effective feasible position. For the operator with noise, the selection of the next request to be inserted is made in the same way as described for the worst-removal operator. The degree of noise is controlled in this case by the parameter δ_2 .

A new generated solution w_{new} is accepted if the number of planned requests $|n_{w_{new}}|$ remains equal or increases relative to the incumbent solution w (see line 8). Since mostly fully utilized services are investigated, which often show limited routing flexibility, this acceptance criterion has the advantage of allowing a maximum diversification with respect to the overall travel time and prevents deterioration of the

number of planned requests. After accepting and saving s_{new} as incumbent solution w , it is checked whether this solution is superior to the best known solution w_{best} (line 10 to 12). This is the case if the number of planned requests $|n_{w_{new}}|$ is higher or remains equal with a shorter total travel time $c(n_{w_{new}})$. After evaluating the new solution w_{new} , the next iteration is performed until the search is terminated, and the best known solution w_{best} is returned (line 15).

3.4.2.2 Execution

In the following, we briefly describe how the LNS is applied to execute the before outlined acceptance and routing decisions of the four policies under consideration.

Basic Control: In each decision epoch $k \in \mathcal{K}$, the LNS is first executed to perform the feasibility check based acceptance decision x_k . To this end, the input set of unplanned requests m_{w_0} is represented by request r_k , while input route plan n_{w_0} corresponds to an empty plan in case of $k = 0$ and route plan y_{k-1} otherwise, yet updated with respect to the time of request t_k . Through the update, input route plan n_{w_0} covers only stops at locations whose planned arrival time z_l^v plus service time p_l is greater or equal to the time of request t_k . Furthermore, the first location of each plan contained in n_{w_0} represents the current respectively next location of a vehicle and will not be rescheduled. Based on this input, the LNS searches for a solution w_{new} that covers all requests in the route plan $n_{w_{new}}$. The search is terminated when either such a solution has been found ($x_k = 1$) or a maximum of β iterations has been performed ($x_k = 0$). Note that in case of an unsuccessful feasibility check, the returned solution is discarded, while the updated routing decision y_{k-1} serves as routing decision y_k . In case of a successful feasibility check, the routing decision y_k is determined by re-optimization of the found solution n under the objective of minimizing the total travel time $c(n_w)$ in β iterations.

Advanced Demand Control: This policy requires in each decision epoch $k \in \mathcal{K}$ a feasibility check for the acceptance decision x_k as well as a re-optimization for the routing decision y_k . It generally corresponds to the procedure of *Basic Control*. However, after each successful feasibility check follows the additional favorability check of the acceptance decision x_k . To solve the corresponding TOP, the initial solution w_0 consists of the same route plan as in the feasibility check. The set of unplanned requests m_{w_0} contains, besides the new request r_k , all trips to be requested

in the following decision epochs $k + 1, k + 2, \dots, k + n$. Based on this input, the LNS is executed to maximize the number of planned requests $|n_w|$. For the acceptance of a new solution w_{new} as best solution w_{best} , an additional criterion is applied, which evaluates if all requests contained in the initial route plan n_{w_0} are as well contained in $n_{w_{new}}$. The search terminates after either finding a solution w_{new} that contains all requests considered in the search or after β iterations have been performed. Once the search has been terminated, it is examined whether the candidate request r_k is contained in the returned route plan $n_{w_{best}}$, which represents that the request has passed the favorability check.

Advanced Fulfillment Control: In this case, the LNS is primarily executed as a feasibility check to obtain the future request acceptances. The input of this feasibility check differs from that presented for *Basic Control* by omitting time-related updates of the incumbent route plan so that it remains flexible throughout all checks. After completion of the last feasibility check, the LNS is executed to determine the route plan thought of as a blueprint for the decision process. To this end, the solution returned by the last successful feasibility check is optimized under the objective of minimizing the total travel time $c(n_w)$ in β iterations.

Advanced Control: Here, the LNS is executed to solve a TOP again to obtain a route plan thought of as a blueprint for the decision process. The initial solution w_0 consists of an empty route plan n_{w_0} , and the set of unplanned requests m_{w_0} includes all requests $r \in R$. The solution w is then optimized in β iterations with respect to the number of planned requests $|n_w|$ and the total travel time $c(n_w)$.

3.5 Computational experiments

In this section, we analyze the impact of the presented policies on the effectiveness of demand and fulfillment control with respect to the performance of the ride-sharing system. We introduce our instances and present the results of the computational study. The description of the parameter tuning of the LNS is given in the appendix.

From the computational results, we first analyze the performance regarding the achieved solution quality expressed as acceptance rate, defined by the number of accepted requests divided by the number of received requests. Secondly, further metrics that describe the operational performance of the ride-sharing system are discussed. This provides insights into the nature of such systems and contributes to

a better understanding of the context-related effectiveness of demand and fulfillment control. Thirdly, we investigate the effect on the service quality perceived by travelers through a detailed trip-specific evaluation. Last, we analyze how acceptance rates change when information becomes incomplete.

3.5.1 Experimental design

Our case study is based on taxi trip data collected in the urban area of New York City, USA. This data set is provided by the City of New York and contains a total of 165,114,361 million trips fulfilled by the Yellow Cab taxi fleet in the year 2014 (NYC Taxi and Limousine Commission, n.d.). Each record contains the start and end time of the trip, the distance traveled as well as the origin and destination locations in terms of geographical coordinates. Figure 3.1 shows the temporal distributions of the trips. In order to simplify the data handling and to ensure consistent trip patterns, we only include weekday trips from January 2014 that operate in the evening peak (as indicated in Figure 3.1 between 17:30 and 20:30) in the area of Manhattan. Furthermore, only trips with a distance greater than zero are considered.

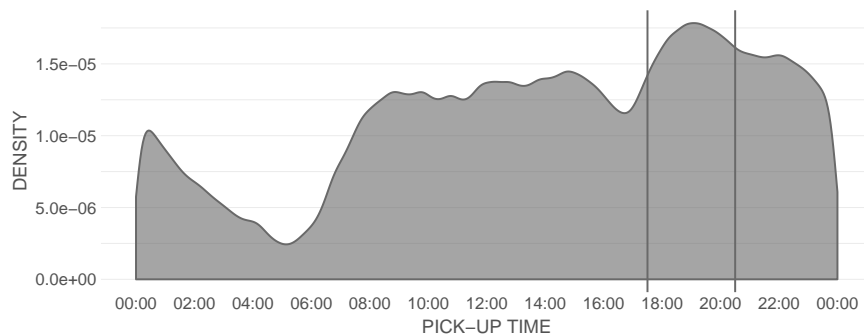


Figure 3.1: Pick-up Time Distribution

Given the taxi trip data, we derive the characteristics of our ride-sharing system as follows. First, potential initial vehicle locations are determined. For this purpose, 40 locations were randomly sampled from the set of locations where a trip ends at 17:30. Second, potential trip requests including origins and destinations are defined. To this end, of all included trips, 180 were randomly sampled. Thus, we assume one incoming request per minute on average. A constant set of trip requests is used in all experiments to enable trip-specific evaluations. The selected locations are visualized

in Figure 3.2, indicating that there is a centrally located area in Manhattan with a higher demand density. Next, free-flow travel times between all locations were computed using the GraphHopper routing engine (GraphHopper, n.d.). Free flow travel times are multiplied by factor ϵ to provide a simple approximation to the real travel times during peak hours.

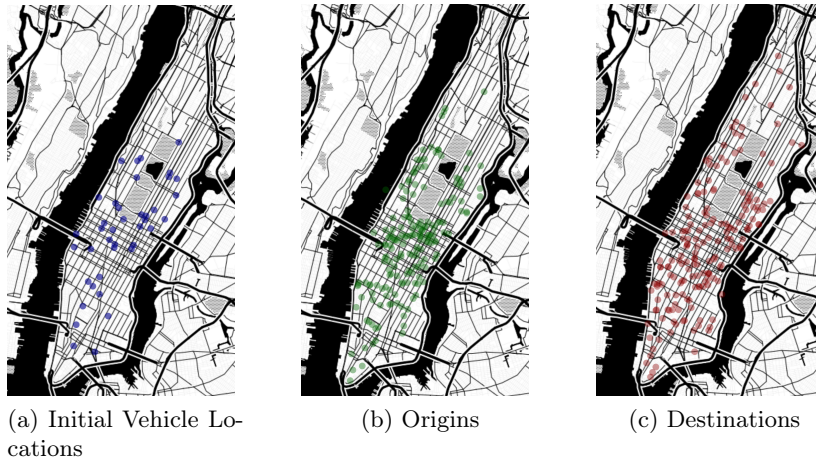


Figure 3.2: Location Distributions (created using Leaflet | © OpenStreetMap)

In summary, we create 110 problem instances as follows: 10 instances are used for the parameter tuning of the LNS, and 100 for our computational study. These instances differ in the receiving times of each request as well as in the initial vehicle locations. Moreover, a baseline scenario is defined for all instances as follows: a fleet of 10 vehicles, a planning period from 17:30 to 20:30 (180 minutes), a travel time factor $\epsilon = 3$, and a maximum arrival delay for each request of 15 minutes. We vary the baseline scenario as follows. First, we vary the fleet size to analyze varying resource-demand ratios. Second, we analyze the impact of temporally varying demand density. To this end, the length of the planning period is varied, and receiving times of requests are adjusted to the corresponding time frame under investigation, whereby 19:00 always marks the middle of the planning period. Third, we analyze geographically varying demand densities by adjusting the travel time factor. Fourth, we examine the impact of the fulfillment time window by varying the allowed maximum arrival delay.

For each analysis, four variations of the base value are considered, representing a decrease of 40% and 80% as well as an increase of 40% and 80% of its parameters (see Table 3.3). With these parameter intervals, we can cover a wide range of possible objective function values and at the same time create insights into where and when the effectiveness of the considered policies is changing. However, to keep the computational effort manageable, only one parameter is varied at a time, while all others keep the value highlighted in Table 3.3.

| Sensitivity analyses | Varying characteristic | | | Values | | |
|------------------------------------|------------------------|--------|---------|----------------|---------|---------|
| <i>Resource Demand Ratio</i> | Fleet size | 2 | 6 | 10 | 14 | 18 |
| <i>Temporal Demand Density</i> | Planning period | 36 min | 108 min | 180 min | 252 min | 324 min |
| <i>Geographical Demand Density</i> | Factor on travel time | 0.6 | 1.8 | 3 | 4.2 | 5.4 |
| <i>Fulfillment Time Window</i> | Maximum arrival delay | 3 min | 9 min | 15 min | 21 min | 27 min |

Table 3.3: Values for the Sensitivity Analyses

In the next section, we will discuss the results of all four sensitivity analyses concerning their impact on acceptance rates. In the subsequent sections, we focus on *Resource Demand Ratio*, while detailed results for the other sensitivity analyses can be found in the appendix, as the results are structurally similar.

3.5.2 Analysis of acceptance rates

We begin by analyzing the effectiveness of the presented policies with respect to acceptance rates. We particularly analyze the value of more advanced optimization in demand and fulfillment control. Overall results are presented in Figure 3.3, which shows the acceptance rate on the Y-axis and the fleet size on the X-axis. The acceptance rates are calculated based on the 100 instances solved 5 times with the varying fleet sizes for each of the four policies. The differently shaped points represent the numeric results and the trend is highlighted by connecting lines. Additionally, the standard deviations of the average acceptance rates are illustrated by a lighter color range around the lines.

Generally, with increasing fleet size, achievable acceptance rates increase as well. As expected, *Basic Control* leads to the smallest acceptance rates, while *Advanced Control* creates the best acceptance rates with an increase about 10–20% compared to *Basic Control*. Interestingly, for smaller fleet sizes, *Advanced Demand Control* yields better results, while for larger fleet sizes, *Advanced Fulfillment Control* can

create significantly higher acceptance rates. The standard deviations increase with increasing fleet sizes. They are negligible for *Advanced Control*. It can be concluded that the highest potential lies in the advanced control of both demand and fulfillment, regardless of the resource demand ratio. However, the contribution to this potential shifts from demand to fulfillment control with an increasing acceptance rate.

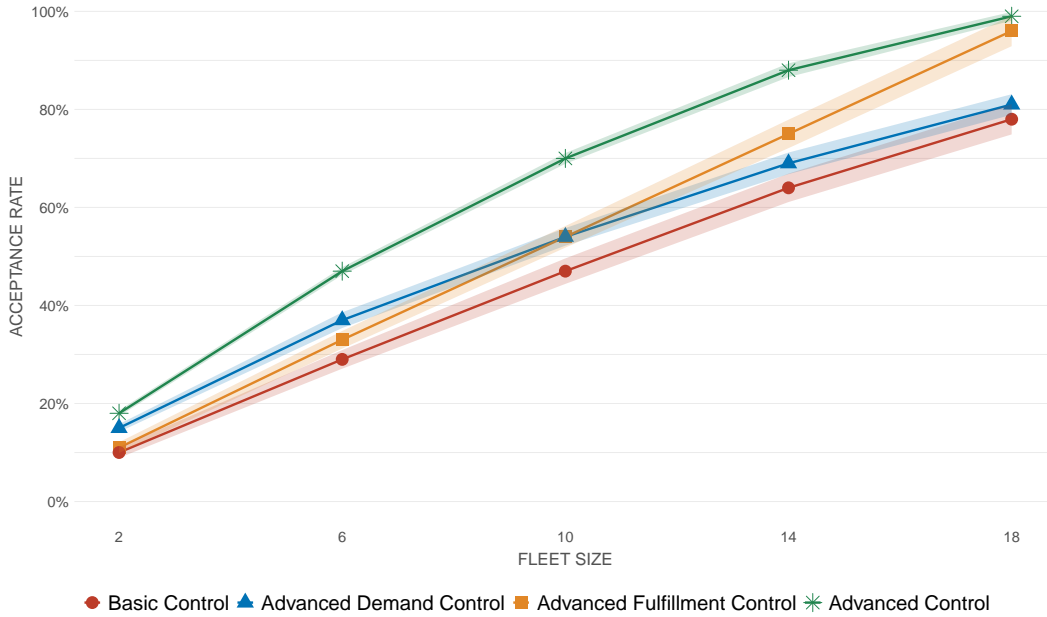


Figure 3.3: *Resource Demand Ratio*: Average Acceptance Rates with their Standard Deviation

We now analyze the results of the further sensitivity analyses (see Figure 3.4). We begin with (a) *Temporal Demand Density*, where we manipulate the demand through temporal variation of the planning period. Generally, results are similar to those obtained for the *Resource Demand Ratio* analysis. For the same fleet size, a relatively larger planning period allows accommodating more requests, with a high benefit of advanced fulfillment control for a large temporal spread of requests and a high benefit of advanced demand control for a small temporal spread of requests. For (b) *Geographical Demand Density*, instead of the time of the planning period, the travel time factor ϵ is used to vary the geographical density of the service area. As expected, when the relative travel times become larger and the area of operation becomes more

“stretched” out, the acceptance rates decrease. The acceptance rate of *Advanced Control* is about 20% higher than for *Basic Control*. Advanced control at either demand or fulfillment can improve this by about 5% only. Here, a high geographical density diminishes the benefits of advanced demand control and increases those of advanced fulfillment control. However, when the geographical density decreases, unfavorable requests from remote regions may automatically be infeasible to fulfill.

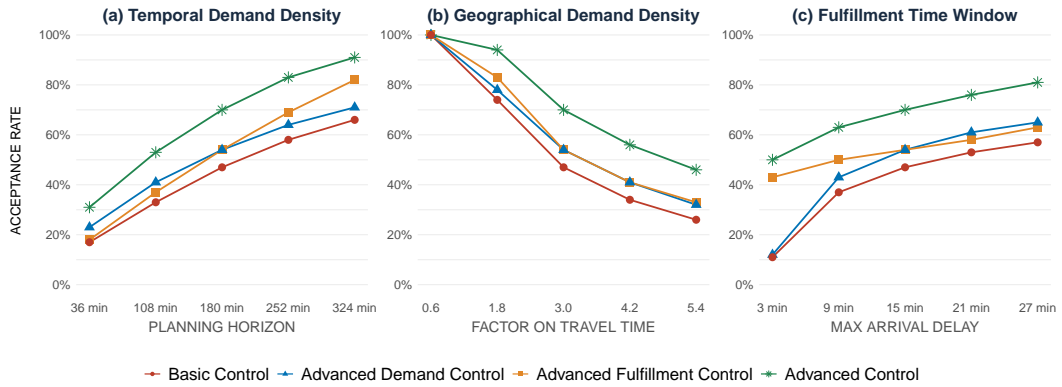


Figure 3.4: Average Acceptance Rates per Sensitivity Analyses

Finally, we analyze for (c) *Fulfillment Time Window* how the variation of the maximum delay, consisting of waiting time and detour, influences the effectiveness of demand and fulfillment control. As expected, acceptance rates increase for all policies with an increased maximum delay. However, the gap between *Basic Control* and *Advanced Control* is very large for small maximum delays. In contrast, *Advanced Fulfillment Control* yields quite stable results for all maximum arrival delays. The benefit from enhanced demand control is higher when the maximum delay is higher. In contrast, the benefit from enhanced fulfillment control is higher when the maximum delay value is lower.

The above findings demonstrate that the effectiveness of demand and fulfillment control depends highly on the system characteristics. However, particularly advanced control of demand and fulfillment shows great potential to increase the acceptance rate of a ride-sharing system. Furthermore, it becomes clear that the potential for demand and fulfillment control differs in response to the characteristics of the system under consideration. The value of advanced demand control is particularly high when

(1) insufficient resources (due to small fleet size or dense temporal demand) require a significant proportion of requests to be rejected, and (2) when a sufficiently large and heterogeneous pool of potentially acceptable demand (due to moderate geographic demand density and sufficiently wide fulfillment time windows) enables the selection of more favorable requests. For advanced fulfillment control, the analysis of different fulfillment time windows demonstrates its importance when offering immediate pick-up times, while the others highlight the dependency on a sufficiently high acceptance rate. Hence, with only a few accepted requests, the trips to be fulfilled are so unfavorable that an increase in performance through advanced fulfillment control alone is barely achievable. Overall, the results imply that the potential of policies focusing on an advanced demand *or* fulfillment control only vary greatly depending on the nature of the ride-sharing system.

3.5.3 Analysis of operational performance

The aim of this subsection is to gain further insights into how demand and fulfillment control impact further performance metrics of a ride-sharing system. The following metrics are considered:

- The average travel time per fulfilled request, defined as the total travel time divided by the total number of fulfilled requests.
- The pooling rate, which measures the percentage of travelers who shared a part of their ride with at least one other traveler.
- The percentage share of each vehicle mode, defined by the total time all vehicles have spent in the mode, is divided by the total time spent by the entire fleet.

The considered modes are:

1. *Shared Travel Time*: Time a vehicle transports more than one traveler,
2. *Single Travel Time*: Time a vehicle transports exactly one traveler,
3. *Unoccupied Travel Time*: Time a vehicle drives without a traveler, i.e., empty trips,
4. *Service Time*: Time required for travelers for getting on or off a vehicle,
5. *Waiting Time*: Time a vehicle waits at a location for a traveler or the next request assignment.

The first metric examined is the average travel time per fulfilled request in minutes, plotted in Figure 3.5 against the varying fleet size. *Basic Control* creates constantly high average travel times per request even with increasing fleet size. Again, *Advanced Control* represents the counterpart, with travel time savings of 3 to 10 minutes on average, highlighting the potential of a combined advanced demand and fulfillment control for ride-sharing systems. *Advanced Demand Control* works almost as well as *Advanced Control*; only for the largest fleet size, *Advanced Fulfillment Control* becomes more efficient. Hence, the reduction of the average travel time per fulfillment is mainly rooted in demand control. Furthermore, a positive correlation can be observed between lower average travel times and the previously identified high acceptance rates, so that the reduction does not seem to be related to an overly restrictive demand control.

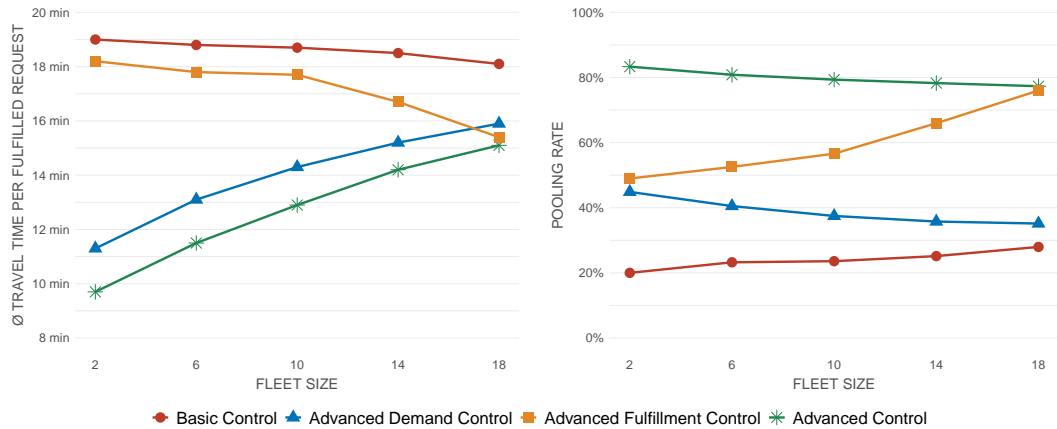


Figure 3.5: *Resource Demand Ratio*: Average Travel Time per Fulfilled Request and Pooling Rate

The second metric of interest is the pooling rate shown in Figure 3.5. *Basic Control* and *Advanced Control* define lower and upper bounds with a gap of 60%. Here, fulfillment control is the key for a good pooling rate as shown by the results of *Advanced Fulfillment Control*; with increasing fleet size, it almost becomes as effective as *Advanced Control*. However, if the fleet size is small, there is a similarly high potential for improving the pooling rate through demand control.

So far, we have seen that the effectiveness of demand and fulfillment control can vary quite a bit. Advanced demand control tends to achieve a reduced average travel

time per fulfillment by accepting a set of favorable requests, while advanced fulfillment control tends to offer higher pooling rates through more successful bundling of travelers. Finally, we examine the proportion of all modes a vehicle can have for the four policies (see Figure 3.6).

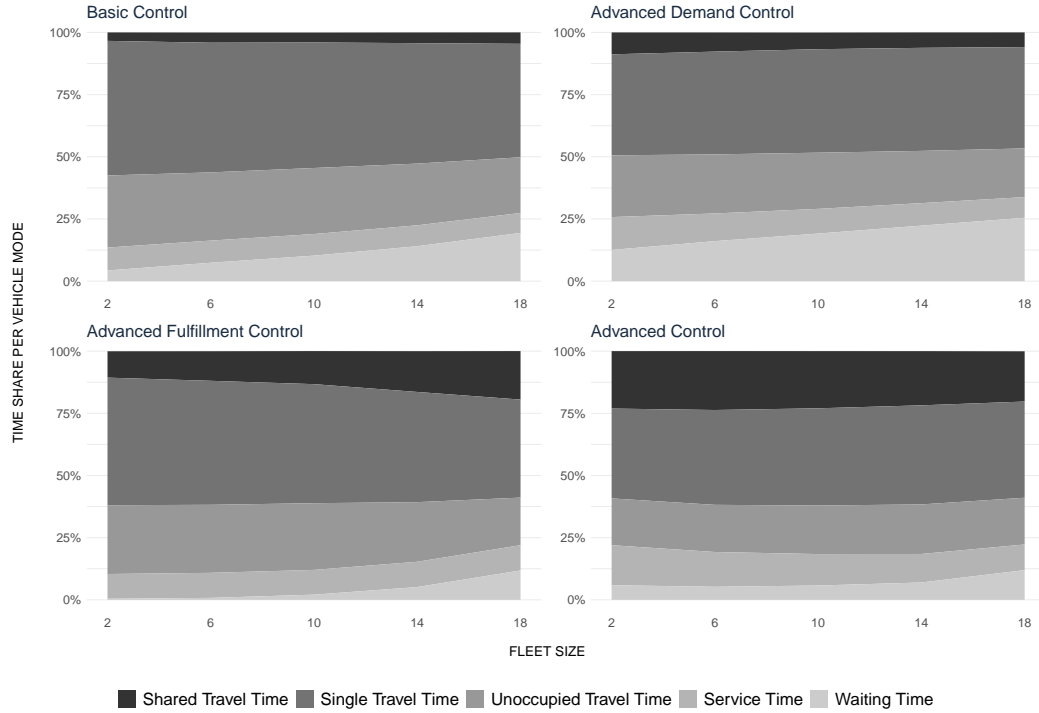


Figure 3.6: *Resource Demand Ratio: Average Time Share per Vehicle Mode*

For all policies and fleet sizes, a rather stable proportion of *Unoccupied Travel Time*, as well as the relatively large share of *single travel time*, is clearly visible. Interesting differences can be observed with respect to the *Shared Travel Time* and *Waiting Time*. For *Shared Travel Time*, again, advanced fulfillment control seems to be the key. Interestingly, even at *Advanced Control*, only about 25% of the total fleet time is used for the simultaneous transport of more than one traveler. However, this is a significantly increased proportion compared to *Basic Control*. Major differences are also apparent for the *Waiting Time*. Especially for *Advanced Fulfillment Control* and *Advanced Control*, lower waiting times can be observed. The lower waiting times in case of an advanced fulfillment control may root in a proactive approach towards

future requests. In contrast, the share of the waiting times is highest for *Advanced Demand Control*. Here, the higher waiting times arise as an advanced demand control may have vehicles wait for favorable request instead of accepting unfavorable ones. Overall, these results show different strategies regarding the handling of waiting time, whereby a smart combination of both strategies appears to be most promising

3.5.4 Analysis of service quality

Finally, we examine the impact of demand and fulfillment control on the quality of service experienced by travelers. Service quality metrics are derived for each of the trips and summarized per policy. The first step is to investigate whether different service quality levels can be observed and if the trip-specific quality of service varies. We analyze the following metrics:

- The acceptance probability per trip, defined by the number of times the trip is requested divided by the number of times the request is accepted.
- The average waiting time per trip, based on the difference between the time of the request and the time the corresponding traveler is picked-up.
- The average detour duration per trip, defined as the average difference between the direct travel time of the trip and the actual time between executed pick-up and drop-off.

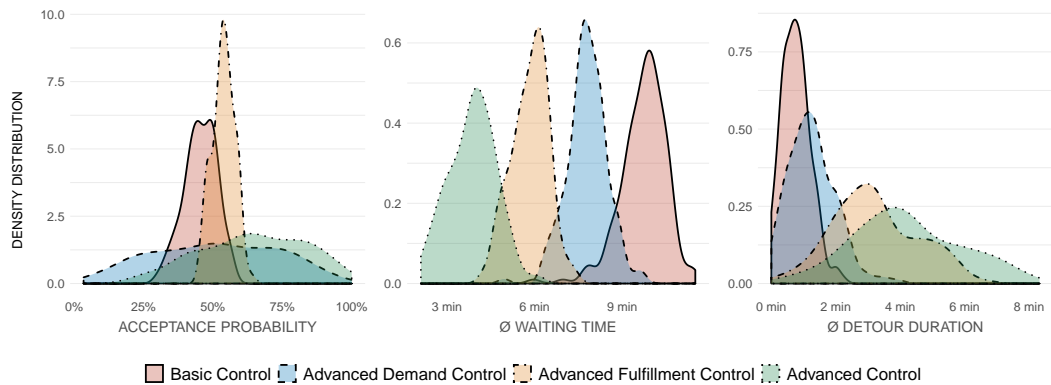


Figure 3.7: Quality of Service per Trip

The results are shown in Figure 3.7 by means of density plots. With regard to acceptance probability, there are clear differences in the distributions. For *Basic Control* and *Advanced Fulfillment Control*, the diversification is relatively low, with a high density at about 50%. Distributions for *Advanced Demand Control* and *Advanced Control* are very flat. This indicates that the probability of being accepted is quite dissimilar among the trips regardless of the circumstances of their request, indicating that the acceptance probability depends on trip inherent characteristics. Interestingly, these characteristics seem to have a relatively minor influence on whether it is feasible to accept a trip.

As seen for the analysis of acceptance probability, the average detour duration per trip also follows different distributions. What is particularly surprising is the shape of the distributions, which shows, especially for *Advanced Fulfillment Control* and *Advanced Control*, that the average detour duration varies depending on the trip. The opposite order of the distribution peaks, compared to those of the average waiting time, results from the limitation through the maximum delay parameter. The shorter waiting times achieved by an advanced demand and fulfillment control are thus partly offset by longer detours.

In the following, trip characteristics are further investigated to find correlations between acceptance probability and detour duration. To this end, we consider the location of the origin and destination as well as the distance between them. For a DVRP, Soeffker et al. (2017) have already shown that anticipatory acceptance discriminates the peripheral regions of the operating area, i.e., the locations there have a lower probability of acceptance. For *Advanced Demand Control*, Figure 3.8 illustrates this correlation separately for origin and destination of all trips, using a color scale as well as sizes that reflect the acceptance probability. The small, dark dots indicate trips with a very low acceptance probability and large, bright ones with a very high acceptance probability. A preference for the regional center and the discrimination of upper and lower periphery is evident, illustrating, for demand control via enhanced acceptance decisions, the positive correlation between the acceptance probability of a trip and the geographical centrality of its origin and destination. In contrast, the analyses of average detour duration for *Advanced Demand Control* and *Advanced Fulfillment Control* did not reveal any recognizable discrimination patterns.

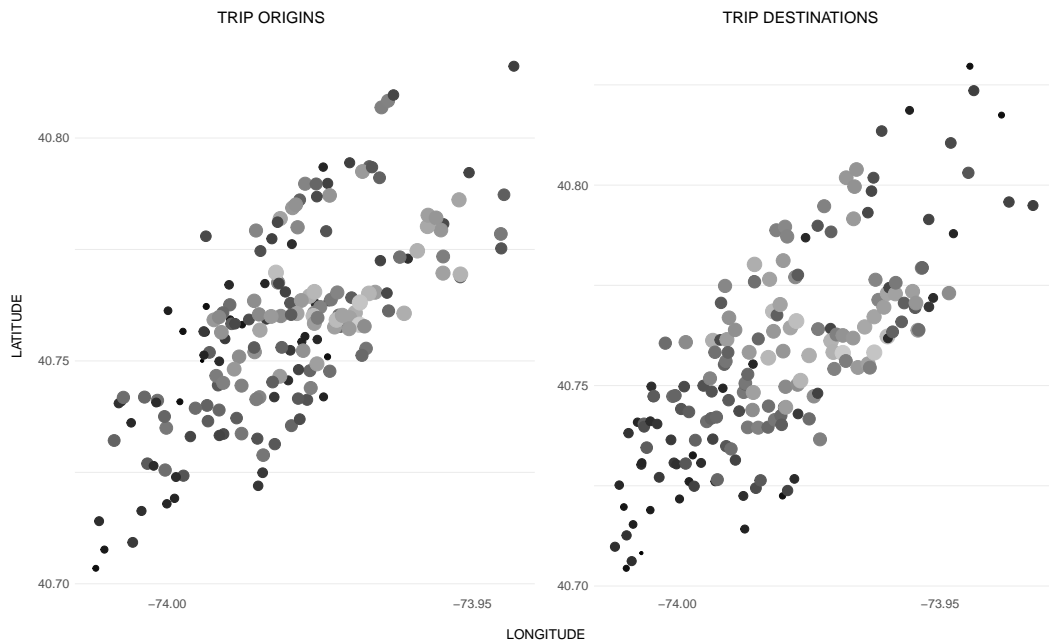


Figure 3.8: Acceptance Probability per Trip Depending on Origin and Destination

As a further characteristic, we examine the trip distance in the light of acceptance probability and detour duration. Results are shown in Figure 3.9. It becomes evident that there is a distinct negative correlation in the case of *Advanced Demand Control*. Implicitly, the advanced demand control utilizes the trip distance as a further criterion to assess requests. For the average detour per trip, a positive correlation with trip distance is noticeable for both cases. This correlation, however, is much more pronounced for *Advanced Fulfillment Control*. Hence, advanced fulfillment control penalizes long-distance trips, yet in a way that limits the usability of the ride-sharing system for such trip requests not as strict as an advanced demand control does.

In summary, demand and fulfillment control have a very different impact on the service quality of ride-sharing systems as experienced by travelers. For advanced demand control, the quality depends significantly on the nature of the requested trip. A ride-sharing system applying such a policy would be very suited for short trips in the center of the service area. However, as their requests would be rejected frequently, travelers requesting trips with unfavorable characteristics are likely to

switch to other mobility services. In contrast, for advanced fulfillment control, the service would be much more balanced in terms of the acceptance probability. Yet, the increasing average detour in proportion to the distance traveled could diminish the perceived quality of service, even if this would be perceived as fair by the traveler. Finally, it should be noted that a policy exploiting the optimization potential in demand and fulfillment control would not only incorporate the performance benefits as shown in the previous sections but also the varying quality of service depending on the characteristics of the trip.

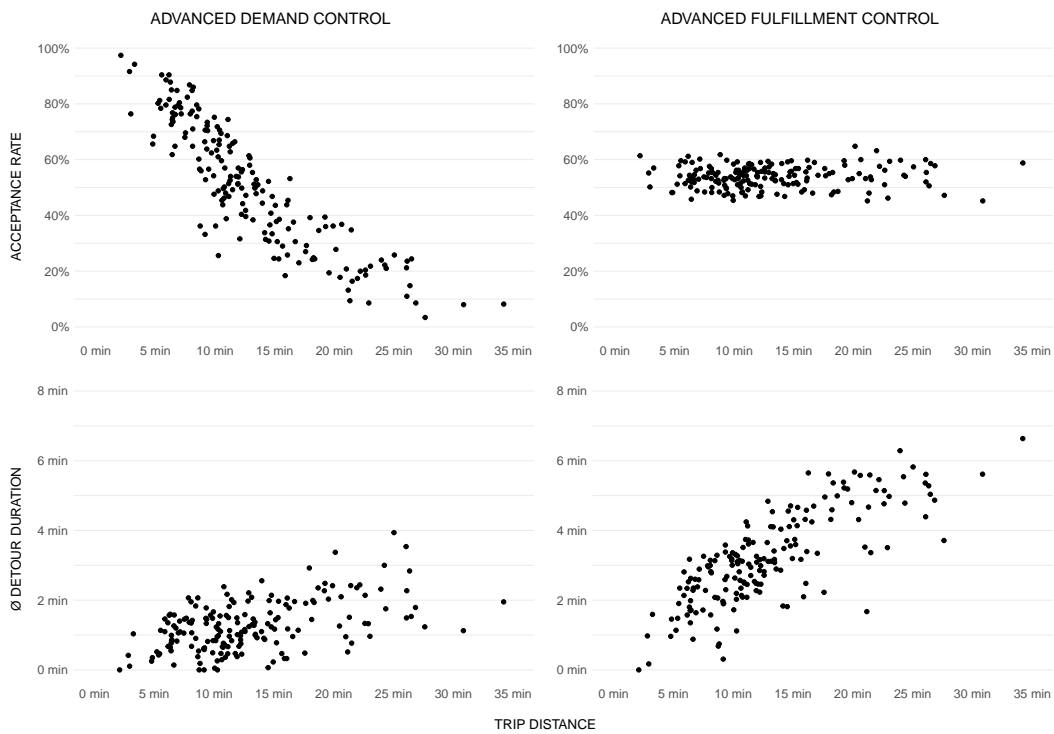


Figure 3.9: Acceptance Probability and Detour Duration Depending on the Trip Distances

3.5.5 Analysis of incomplete information

In the following, based on the *Resource Demand Ratio* sensitivity analysis, we investigate to which extent the above results change when information becomes incomplete. For this purpose, the average acceptance rates obtained under complete information

are compared to those obtained for a perfect information horizon limited to the next 10 minutes. This analysis provides insights into the value of information defined by Mitrović-Minić et al. (2004) as the “performance gap between solving an instance with incomplete and complete information”.

Limiting the information horizon requires some minor adjustments to the three policies considering advanced demand and/or fulfillment control. With respect to *Advanced Demand Control*, the set of requests considered in the TOP-based favorability check is reduced from all future requests to those that will be received in the next 10 minutes. *Advanced Fulfillment Control* is adapted so that for each incoming request, feasibility checks are performed to determine which requests will be accepted in the next 10 minutes, to be able to re-optimize the route plan accordingly. In case of *Advanced Control*, a TOP is solved for each incoming request, taking into account the already accepted requests as well as all requests that will be received in the next 10 minutes to decide upon the acceptance as well as the new incumbent route plan. The results are visualized in Figure 3.10, which shows the acceptance rates for the four considered policies as a solid line for the unlimited information horizon and as dashed line for the information horizon limited to 10 minutes. The gap between the two lines of each policy is further highlighted by the respective color.

For *Basic Control*, per definition, no difference is visible, since it does not take into account any information about future demand. As expected, decreased acceptance rates can be observed in case of the limited information horizon among the other three policies. Consequently, policies that are most affected by information incompleteness are those that exploit the information most extensively. However, the structural differences between the policies implementing advanced demand and/or fulfillment control remain similar to those obtained under complete information.

These results indicate, on the one hand, that the value of information in demand and fulfillment control is directly reflected by the acceptance rates and its proportional deterioration. On the other hand, it can be observed that the previously presented findings are less pronounced in the considered case of incomplete information, yet structurally still valid. However, this investigation represents only one of various possible variations from complete over incomplete to erroneous information. It would therefore be interesting to investigate in future work under which horizon

and/or quality of information the structural consistency persists and when and how it may alter.

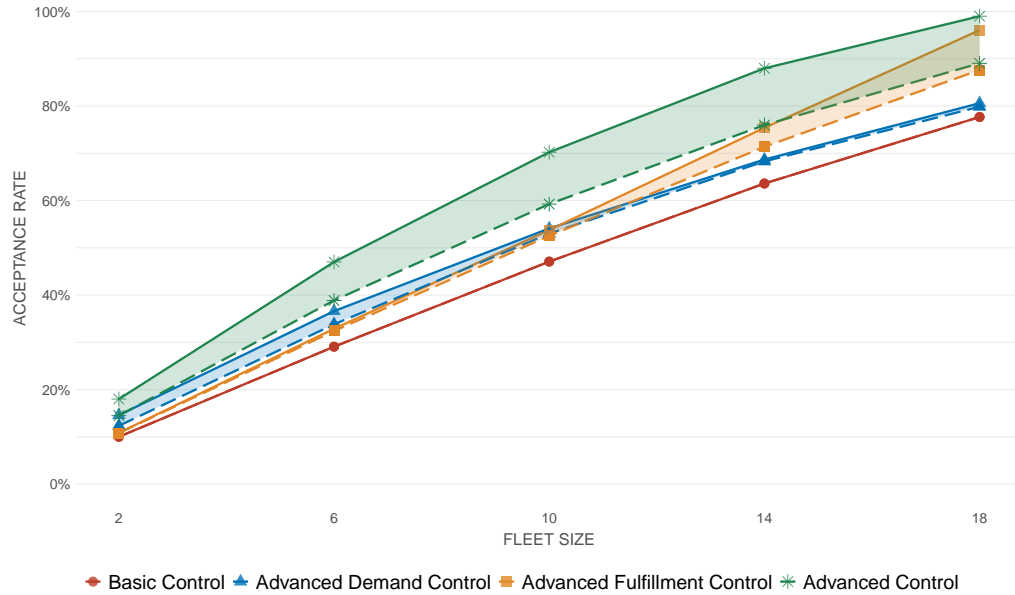


Figure 3.10: Average Acceptance Rates for Unlimited and 10 minute Information Horizon

3.6 Conclusion

In our paper, we investigated the effectiveness of demand and fulfillment control in dynamic fleet management of ride-sharing systems. To this end, we first differentiated strategies for demand and fulfillment control and classified the related literature accordingly. Second, we defined four policies, which differ in the complexity of optimization and the amount of information exploited by demand and/or fulfillment control. The impact of these policies on dynamic fleet management was investigated in a comprehensive computational study, highlighting the operator's perspective as well as the consequences for travelers. Overall, our results demonstrated great potential for combined advanced demand and fulfillment control in dynamic fleet management. Potential benefits range from increased acceptance and pooling rates to decreased

travel and idle times. However, acceptance probability and detour duration depend considerably on the nature of the requested trip.

A particular contribution of our paper is the differentiation of dynamic fleet management according to the effectiveness of demand and fulfillment control. This created insights about whether optimization potential can be attributed to either demand or fulfillment control or a reasonable combination of them. This is important since advanced demand and fulfillment control differ in their requirements as well as in the effect on the performance of the ride-sharing system. Advanced demand control is especially beneficial if there is a sufficient surplus of demand, i.e., when there is a decent subset of favorable requests that can be selected from a larger pool of feasible requests. Furthermore, advanced demand control can increase the acceptance rate primarily through a significant decrease of average travel time per fulfilled request. The acceptance probability is highly correlated with the nature of the requested trip, leading to an acceptance of short trips that are centrally located in the service area. The potential of fulfillment control is primarily associated with the acceptance rate and the promised fulfillment time window. Taking acceptance of future requests into account, advanced fulfillment control enables a rather stable performance despite increasingly narrow fulfillment time windows. However, advanced fulfillment control can only be beneficial if demand control has only a minor impact or is of advanced nature, too. In particular, performance improvement through advanced fulfillment control can be traced back to a much more successful bundling of requests. The consequence for travelers is that the detour duration increases proportionally to the distance of the trip.

Our paper offers operators of ride-sharing systems an orientation on how to implement demand and fulfillment control. For instance, advanced demand control could be more suitable for large systems or systems with a few regular travelers, where the satisfaction of individual travelers is negligible. Furthermore, it could be implemented in order to efficiently manage a temporary demand surplus on special occasions. Advanced fulfillment control would be particularly suitable for systems with stable demand allowing precise anticipation of future acceptances. Moreover, under the assumption that a selective demand control avoids, the demand target should be fully operable.

Moreover, we contribute to research on dynamic fleet management by providing a more differentiated view of how policies control demand and fulfillment in a ride-sharing system. We believe that this can be the basis for a better understanding of the varying effectiveness of existing policies as well as the development of new ones.

In the future, a more detailed overview of anticipatory decision-making in dynamic fleet management could provide a better understanding of what types of anticipation are reasonable for ride-sharing systems. Furthermore, for our study, we performed the evaluation mostly assuming complete information while the implications of incomplete information were only briefly examined. An intuitive next step would be to perform demand and fulfillment control under different degrees of incomplete or imperfect information to investigate the link between information quality and the exploitation of the identified potentials. Moreover, state-of-the-art policies for demand and/or fulfillment control could be evaluated to compare their effectiveness with the results obtained. This would include the development of sophisticated customer acceptance mechanisms for anticipatory demand control in ride-sharing systems.

Chapter 4

Heatmap-based decision support for repositioning in ride-sharing systems

Abstract In ride-sharing systems, platform providers aim to distribute the drivers in the city to meet current and potential future demand and to avoid service cancellations. Ensuring such distribution is particularly challenging in the case of a crowdsourced fleet, as drivers are not centrally controlled but are free to decide where to reposition when idle. Thus, providers look for alternative ways to ensure a vehicle distribution that benefits users, drivers, and the provider.

We propose an intuitive mean to improve idle ride-sharing vehicles' repositioning: repositioning heatmaps. These heatmaps highlight driver-specific earning opportunities approximated based on the expected future demand, current and expected future fleet distribution, and the location of the specific driver. Based on the heatmaps, drivers make decentralized yet better-informed repositioning decisions. As our heatmap policy changes the driver distribution in the future, we propose an adaptive learning algorithm for designing our heatmaps in large-scale ride-sharing systems. We simulate the system and generate heatmaps based on the previously learned policy in every iteration. We then update the policy based on the simulation's outcome and use it in the next iteration. We test our heatmap design in a comprehensive case study on New York ride-sharing data. We show that carefully designed heatmaps reduce service cancellations therefore revenue loss for platform and drivers significantly while leading to a better service level for the users and to a fairer treatment of drivers.

Keywords mobility-on-demand, vehicle repositioning, crowdsourced transportation, heatmap, stochastic dynamic decision making, adaptive learning

4.1 Introduction

The trend for ride-sharing services like UberXShare and MOIA is unbroken. In this paper, we understand ride-sharing services as systems where users spontaneously submit transportation requests online, are picked up a short time afterward, and are driven to their destination, while possibly sharing parts of their ride with other users. In some cases, no driver from the ride-sharing fleet may be available in the vicinity of a user, creating unacceptable high waiting times and, consequently, a poor level of service. Service cancellations due to insufficient levels of service lead to loss of revenue and user dissatisfaction. Thus, service providers aim for a good distribution of the drivers in the city to meet current and potential future demand.

The tools for ensuring such a distribution depend on the type of service provider. Services like MOIA own a fleet of employed drivers and, therefore, can make repositioning decisions centrally. Since the future demand is uncertain, efficient repositioning decisions are a challenging task on their own, even if they can be made centrally (Pouls et al., 2020). Other providers such as UberXShare crowdsource transportation for their ride-sharing services to private individuals who are paid on a per-job basis. In that case, drivers are not directly controlled by the service provider but are free to decide where to reposition in the city when unoccupied. This is an additional challenge, as decentralized repositioning likely inconveniences users and drivers.

For example, many drivers may prefer waiting in the city center, where new requests are more likely to occur, which may lead to an overflow of resources, while there is a driver shortage at other locations. Furthermore, new and inexperienced drivers may be lost in the system without guidance and receive only limited earnings (Cook et al., 2021). For the platform, all this means poor service availability for some users on the one hand and frustrated drivers on the other hand. Another challenge with crowdsourced drivers is that they might be reluctant to follow directions by the platform if they do not understand them (Möhlmann et al., 2021). Thus, service providers look for alternative and intuitive ways to ensure a good distribution of crowdsourced drivers that benefits users, drivers, and the provider.

In this paper, we propose an intuitive mean to improve the repositioning of unoccupied vehicles in ride-sharing systems: Repositioning Heatmaps (RH). In case a driver becomes unoccupied, these RH highlight the driver-specific earning opportunities, approximated based on their travel time to areas with a shortage of drivers to

satisfy the expected demand in the near future. In the RH, repositioning locations with high expected opportunities are shown in green shades, while repositioning locations with low expectations are shown in red shades. Both shades can have different intensities, e.g., dependent on the relative opportunity volumes. Drivers then decide on repositioning in a decentral and independent manner. Ideally, our RH provide non-monetary incentives that guide drivers to increase service availability, reduce cancellations, and improve drivers' earning opportunities.

Creating RH is challenging for several reasons. As with centralized approaches, future user demand is uncertain. Therefore, the earning opportunities are unknown. Furthermore, showing RH changes the repositioning decisions of drivers in the system, which in turn may lead again to too many or an insufficient number of drivers in certain areas. We propose an adaptive learning algorithm for designing our RH to address this issue. In every iteration, we simulate the system and generate RH based on previously learned opportunities. We then update the earning opportunities based on the simulation's outcome and use the updated opportunities in the next iteration. Eventually, the expected opportunities and the heatmap design policy converge.

We test our heatmap design in a comprehensive case study on New York ride-sharing data with 200 drivers and around 6400 expected users per planning period. We show that carefully designed RH reduce service cancellations, and, therefore, reduce revenue loss for platform and drivers significantly. Furthermore, providing heatmaps to drivers increases the average earnings per driver and reduces the volatility in earnings among the drivers. Even though analyzing these results is beyond the scope of this paper, such "fair" earnings might be an important factor for the long-term commitment of a driver to a platform. We also show that RH lead to a better and more balanced distribution of service availability in the city, another important factor for long-term user retention. We demonstrate that our strategy is relatively robust to non-compliant drivers deviating from our recommendations. Finally, we show that providing RH to new and inexperienced drivers can give them an easy entry into the system and a more leveled playing-field.

Our paper makes the following contributions: First, we are among the first to introduce RH for nudging crowdsourced drivers. We show that RH are a powerful, intuitive tool for managing complex and dynamic systems, which is likely applicable for other applications such as restaurant meal delivery. Second, we investigate and

define a new problem and formalize the corresponding sequential decision process. Third, we present a large-scale, real-time decision policy based on adaptive learning with very limited calculation efforts. Fourth, we show that our policy is superior to a variety of benchmark policies and can even compete with a centralized repositioning heuristic from the literature. And fifth, we provide a comprehensive computational study including managerial insights on central and decentral management of crowd-sourced ride-sharing systems and the experience of the involved stakeholders.

We begin with an overview of related literature in Section 4.2. The problem is stated and formalized in Section 4.3. How we develop RH by adaptive learning is presented in Section 4.4. The experimental setup is discussed in Section 5.5 and computational experiments are reported in Section 4.6. We conclude with final remarks in Section 4.7. We also present a comprehensive appendix with additional experiments and details.

4.2 Related literature

Our work addresses the management of crowdsourced drivers for ride-sharing services. In the following, we briefly discuss the related work from crowdsourced transportation and then embed our work in the literature on ride-sharing optimization. Last, we mention related work from the area of shared mobility, where dynamic repositioning also plays an important role.

4.2.1 Crowdsourced transportation

Crowdsourced transportation service providers outsource jobs to private individuals, inducing cost advantages for the service provider on the one hand and flexible working hours and uncertain earning opportunities for the drivers on the other hand. Uncertainty in crowdsourced transportation plays a major role for service providers as well, as it is not clear when, where, and how many drivers will be available. Moreover, the freedom of choice of the drivers leads to further planning uncertainty, since they decide on the acceptance of a job assignment and the execution of a repositioning recommendation. For the improvement of such systems, it is important to understand the supply market and thus the individual behavior of crowdsourced drivers. Relevant literature ranges from a detailed description of driver characteris-

tics (see, e.g., Ashkrof et al. (2020)) across the analysis of differences between driver segments (see, e.g., Cook et al. (2021)) to empirical market analyses (see, e.g., Rai et al. (2021)). Several studies focus on the interdependence of driver behavior and earning potential. For instance, Henao & Marshall (2019) examine how behavior affects earning potential, while Castillo et al. (2022) study how tipping affects the behavior of crowdsourced drivers. Optimizing transportation systems in the face of uncertain driver behavior is a major challenge that has only recently come into the focus of research. For an overview, we refer to Savelsbergh & Ulmer (2022). The rather limited work in this area focuses on the uncertain number of drivers being in the system (see, e.g., Dayarian & Savelsbergh (2020) or Ulmer & Savelsbergh (2020)), or on hedging against drivers rejecting offered requests (see, e.g., Gdowska et al. (2018) or Ausseil et al. (2022)).

Uncertainty in repositioning has not been explored much so far. One of the few works that address repositioning under these conditions is by Alnaggar (2021). Similar to our work, Alnaggar (2021) proposes a heatmap to guide drivers, in their case, towards earning opportunities for crowdsourced last-mile deliveries. They use a short-term demand forecast to derive global heatmaps with up to three different levels for systems with up to nine repositioning locations. We differ from their work as follows. First, the problems vary as in our work, we consider the problem of ride-sharing with consolidation potential and tighter time commitments. Heatmaps are provided to an individual driver in real-time while in Alnaggar (2021), heatmaps are updated every 15 minutes and are globally set. Second, the methodology is different: Alnaggar (2021) suggests a stochastic lookahead method that samples future demand and creates the heatmaps accordingly. The lookahead rather focuses on the short-term demand as the lookahead horizon is limited, and future values are discounted. The lookahead model is static, i.e., it does not capture that in the future new information is revealed and dynamic decisions are made. Our method captures both as we apply a learning algorithm where values are updated based on future demand realizations and heatmap decisions. Our method also considers long-term effects which is very valuable as we show in our experiments. Last, our work differs in scale with hundreds of repositioning locations and drivers and thousands of customers.

4.2.2 Ride-sharing

In the following, we present related work from ride-sharing, which suffers from an unbalanced driver distribution due to spatial and temporal imbalanced demand (Jiao et al., 2021). Besides the assignment of transport requests, imbalanced demand is one of the most considered challenges for the efficient operation of ride-sharing services (Wang & Yang, 2019). In the following, we examine the extent to which the literature for large-scale repositioning of unoccupied drivers in ride-sharing services meets the requirements of a crowdsourced fleet (see Table 4.1).

| | | Supports drivers in decision making | Indicates multiple repositioning options | No discrimination against inconvenient drivers/users |
|--|-----------------------------|--|---|---|
| Pavone et al. (2012) Zhang & Pavone (2016) Sayarshad & Chow (2017) Braverman et al. (2019) | Queueing-based | | | ✓ |
| Zhang et al. (2016) Iglesias et al. (2018) Wallar et al. (2018) Pouls et al. (2020) Lei et al. (2020) Li et al. (2021) Pouls et al. (2022) | Model predictive control | | | ✓ |
| Wen et al. (2017) Holler et al. (2019) Jiao et al. (2021) Liu et al. (2021) Xi et al. (2021) Yu & Hu (2021) Zhu et al. (2021) | Reinforcement learning | ✓ | | ✓ |
| Taylor (2018) Bimpikis et al. (2019) Guda & Subramanian (2019) Nourinejad & Ramezani (2020) Chen et al. (2021) Hu et al. (2022) | Surge pricing | ✓ | ✓ | |
| RH (this work) | Heatmaps | ✓ | ✓ | ✓ |

Table 4.1: Literature Classification

With regard to the unique requirements of crowdsourced fleets, four criteria have been identified as particularly relevant: (1) Pursuing a system-wide balance between demand and supply in the best interest of provider revenue and driver earning opportunities; (2) The balance is achieved by supporting drivers in their decision about when and where to reposition; (3) The recommendations include comprehensive information to enable well-founded decision making (no “take it or leave it”); (4) Non-compliance is not penalized to ensure long-term driver satisfaction and therefore

retention. The related approaches can be divided into five main streams: queueing-based, model predictive control, reinforcement learning, data-based driver guidance, and surge pricing. Next, we will briefly present the related work per category and highlight the relationship to our work.

We first consider *queueing-based approaches* (QBA) and *model predictive control* (MPC), as they are similar in their applicability for crowdsourced fleets. QBA for determining optimal repositioning policies have been widely researched (e.g., Pavone et al. (2012), Zhang & Pavone (2016), Sayarshad & Chow (2017), and Braverman et al. (2019)). Sayarshad & Chow (2017) prove that these are also applicable for services of real-world size. Braverman et al. (2019) show that, in addition to classical repositioning, the optimal routing for unoccupied vehicles in search of a next passenger can be determined. MPC typically builds on demand forecasts combined with mathematical programming to solve the repositioning problem online periodically. The first contribution in this direction comes from Zhang et al. (2016). Others like Iglesias et al. (2018), Wallar et al. (2018), Pouls et al. (2020), and Pouls et al. (2022) focus on large-scale or shared rides. Lei et al. (2020) propose MPC to train a neural network offline that allows for a quick repositioning policy prediction online. Li et al. (2021) also propose a neural network but use it to improve the demand prediction within an MPC approach.

Even though QBA and MPC are based on very different concepts of ride-sharing systems, related contributions have in common that they focus on the centralized control of the entire fleet. This makes transferring to the guidance of individual drivers in a crowdsourced ride-sharing system difficult. In contrast, with our RH, we provide a decision-support tool at the driver’s request that helps making better-informed decisions on an individual basis. For our experimental setup, we rely on the work by Pouls et al. (2020) as a benchmark policy and show that the right heatmap strategy can compete with a centralized MPC-based policy.

A different approach for repositioning in large-scale ride-sharing services that gained attention in recent years is *agent-based reinforcement learning* (e.g., Wen et al. (2017), Holler et al. (2019), and Jiao et al. (2021)). In agent-based approaches, decentralized decision-making for individual drivers is considered instead of centralized decisions for the entire fleet. While Wen et al. (2017) focus on decentralized learning of a repositioning policy, Holler et al. (2019) compare the benefits of both

centralized and decentralized learning. Of particular interest for our work are the results of Jiao et al. (2021). The results show that policies learned via reinforcement learning can be transferred to the real world and are advantageous over the intuitive decisions of drivers. It is further shown that driver collaboration is particularly relevant for large fleets. Moreover, it is argued that driver repositioning may deviate from central recommendations, but this has not been considered further in the experiments. This conflict of interest has been addressed by Zhu et al. (2021) by means of RL, introducing monetary subsidies to nudge drivers to perform the recommended repositioning. Finally, recent contributions by Liu et al. (2021), Xi et al. (2021), and Yu & Hu (2021) propose technical innovations for reinforcement learning of repositioning policies.

As Jiao et al. (2021) have shown, reinforcement learning can be a good foundation for learning recommendations for the repositioning of drivers. However, due to the nature of reinforcement learning, the recommendations as described in the above papers are limited to the driving direction to be taken or the neighboring area to be approached. Furthermore, these are black-box approaches. Drivers therefore only have the choice of complying with recommendations or making unsupported repositioning decisions. Our RH differ from this, despite their learning capability, by providing drivers with an assessment of repositioning locations distributed across the service area while highlighting the best individual option. Moreover, the information used to generate the heatmap as well as the resulting indications can be communicated in a comprehensible way.

Finally, another way to reposition a crowdsourced fleet while focusing on platforms' revenue is *surge pricing*. Here, user fees and driver wages are optimized to attract drivers (and deter demand, see, e.g., Bimpikis et al. (2019), Guda & Subramanian (2019)), or to attract demand (and deter drivers, see Hu et al. (2022)). These papers include the approaches discussed above, e.g., the study by Taylor (2018) is based on a QBA, Nourinejad & Ramezani (2020) develop an MPC approach, and Chen et al. (2021) rely on RL. In this context, also heatmaps have already been applied to inform drivers about the current distribution of prices (Guda & Subramanian, 2019). Unlike our idea of heatmaps for non-monetary driver guidance, surge pricing relies on a strong monetary incentive to get drivers to behave as the platform requires. Surge pricing is usually used to serve peak demand and does not necessarily increase

service availability in every part of the city (Bimpikis et al., 2019). Moreover, the value of drivers entering the system or moving towards certain areas is paid for by the customers. Thus, surge pricing may discourage customers in areas with already limited demand from ordering service. Another challenge of surge pricing is that volatile and therefore uncertain prices are often not well received by users and drivers (see, e.g., Dholakia (2015), Goncharova (2017), and Conger (2021)). As a result, first cities have begun to ban surge pricing (Spielman, 2021). Our heatmap strategy does not shift the cost to the users (or drivers), but can rather be a helpful tool to guide drivers towards earning opportunities, and, as we show in our experiments, provide better service availability system-wide. With our RH, we, therefore, aim to provide a less controversial mean supporting ride-sharing service providers in managing a crowdsourced fleet without inducing disadvantages for particular drivers or user groups.

4.2.3 Shared mobility

Dynamic repositioning is also well-known for shared mobility services such as bike sharing, car sharing, or scooter services (see, e.g., Luo et al. (2022), Martin et al. (2021), and Greening & Erera (2021)). In such systems, rental and return of vehicles vary over time and in space. Thus, stations (or demand areas in station-free systems) run out of vehicles while others have more than sufficient vehicles. Measures are necessary to reposition vehicles in anticipation of future demand. There are two main differences to our problem. First, in such systems, repositioning is usually done manually, i.e., a workforce travels through the city to pick up or drop off shared vehicles. Second, decision-making in such systems can be done centrally by the service provider. In our problem, drivers decide for themselves if and how they follow our repositioning recommendation. While the work on shared mobility is vast, the work likely closest to ours is presented by Brinkmann et al. (2019), where dynamic decisions are made about repositioning bikes to minimize expected future cancellations. Similar to Pouls et al. (2020), Brinkmann et al. (2019) propose an MPC approach where decisions are informed by samples of future demand, carefully balancing cancellations and travel time. They show that with the right strategy, improvements of more than 10% can be achieved relative to conventional strategies ignoring future demand.

4.3 Problem statement and formulation

In the following, we give a problem narrative to define the system and its dynamics. We then provide an example and model the problem as a sequential decision process.

4.3.1 Problem narrative

We take the perspective of a ride-sharing service provider that connects transportation requests with self-employed drivers following the goal of minimizing the overall number of daily service cancellations. Over the course of the service horizon, users request instant transportation from an origin to a destination within the city. To fulfill a transportation request, the service provider can assign a nearby driver that is either currently idling, or assign an already busy driver who is currently repositioning or transporting another user. Thus, pooling is possible with respect to a maximum vehicle capacity. If no driver is nearby, e.g., the user cannot be picked up within 10 minutes, the user cancels the service request. If the transportation request can be fulfilled, the user pays a fee. This fee is split between driver and service provider. For simplicity, we assume that the fleet of self-employed drivers work the entire service horizon¹, accept all transportation requests assigned to them, and follow the routing suggested by the service provider. However, the drivers are free in their decision where to reposition after finishing a job, which could be an area with high expected demand (as described by Ermagun & Stathopoulos (2018), for example), or an area in the driver’s neighborhood (as described by Rai et al. (2021), for example).

We assume decentralized decision-making about the repositioning of the drivers. The drivers’ repositioning behavior impacts the fleet distribution in the city and therefore future revenues respectively earning opportunities for service providers and drivers. Having not enough drivers in one area of the city leads to service cancellations and revenue loss for the service provider. Having too many drivers in another area leads to fewer earning opportunities for the drivers. Therefore, both service providers and drivers are interested in the effective distribution of the fleet. However, research shows that self-employed drivers are reluctant to follow service providers’ directions straightforwardly, especially if their reasoning is not immediately clear (Möhlmann et al., 2021). Hence, in our problem, we propose an alternative tool

¹We note that our proposed methodology can also handle dynamic fleet sizes as analyzed in Appendix B.5.

that is both intuitive and leaves drivers with the final repositioning decision: driver guidance through RH.

Driver-specific RH indicate earning opportunities for the drivers in the city. Earning opportunities depend on the repositioning duration from the driver’s current to the repositioning location, the expected demand in the area, as well as how well the area is already covered by the fleet. Promising repositioning locations imply significant opportunities due to short travel times, high expected demand, and/or few drivers in the surrounding area. These repositioning locations are indicated in shades of green with different intensities; other, rather unfavorable repositioning locations are colored in different shades of red. Whenever drivers are without an assignment to fulfill and consider repositioning, they consult their heatmap to make a well-informed repositioning decision. This decision may be fully compliant with the platform’s recommendation or may deviate based on the driver’s personal preferences, e.g., with respect to travel distance or potential competition.

The service provider aims for a heatmap strategy that nudges drivers towards lucrative areas while still maintaining a flexible and effective distribution of the entire fleet to avoid future service cancellations. Such a strategy creates RH every time a driver becomes idle, based on the current state information and the expected demand. Besides the design of RH, the provider also faces the operational challenge of assigning and routing new requests. Since our work aims on investigating effective heatmap design, we assume that for assignment and routing, the service provider follows an externally given strategy (details will be provided in Section 5.5.1). However, we create our RH in a way that it can capture different routing and assignment strategies, and, in Appendix B.4, we show its effectiveness for a variety of alternative strategies.

4.3.2 Example

In the following, we give an example to illustrate the dynamics of the system under consideration (see Figure 4.1). We note that in the example and throughout the paper, we replace the green and red shadings of the heatmap with different shades of grey to allow readability in black and white. The darker the shade, the “greener” the heatmap.

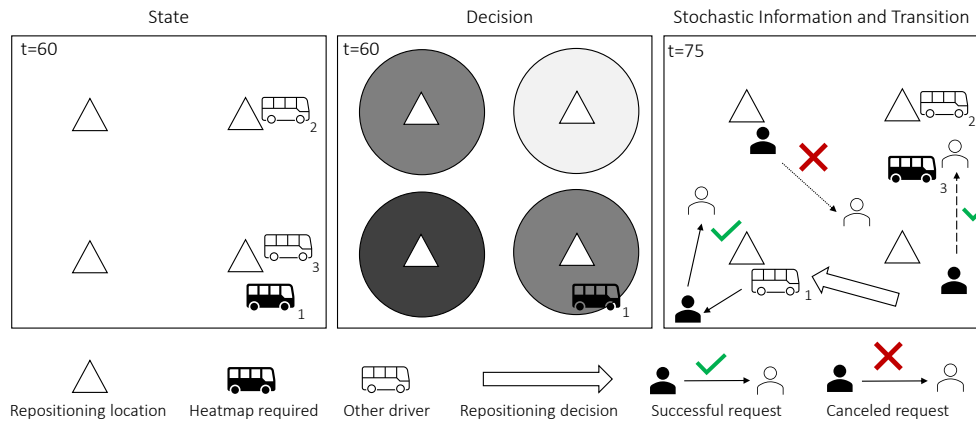


Figure 4.1: Example for a State, Decision, the Realization of Stochastic Information, and Transition to the Next State

The system can be described by its state (on the left), potential decisions (in the center), and a realization of stochastic information including the resulting transition to a new state (on the right). The example is at time $t = 60$ of the service horizon. The city consists of four regions, each with one repositioning location. For the purpose of the example, we assume that the expected demand is equal in all regions. The locations of the three drivers are given. Two of them, drivers two and three, depicted in light, already have done their repositioning and are now waiting for new requests to be assigned. Driver one, depicted in dark, just finished a trip and now needs to make a repositioning decision via a heatmap provided by the service provider.

A potential heatmap recommendation is shown in the center. The heatmap combines several ideas. As the expected demand is evenly distributed and the two areas on the right are already covered by a driver, a reasonable decision would be to color these locations in lighter grey (i.e., red) to indicate fewer earning opportunities. The areas on the left are both uncovered currently, thus, the heatmap may highlight them in darker grey (i.e., green), indicating more earning opportunities for these two locations. Finally, the vehicle's distance to the regions comes into play, leading to darker shading for the bottom locations.

The most right part of Figure 4.1 shows the realization of stochastic information and the transition to the next state at time $t = 75$. This part is relatively complex for this problem. The stochastic information is twofold:

1. First, the repositioning decision of driver one has been revealed, indicated by the large white arrow. Here, driver one has been compliant and decided to reposition to the bottom left.
2. Second, new demand occurs (and is assigned to drivers if possible) between the current and the new decision epoch, indicated by dark and light customer icons connected by a thin arrow. In the example, three new requests occurred between $t = 60$ and $t = 75$. Two of them could be assigned. The one in the bottom right was served by driver three, the other in the bottom left was served by driver one. The request in the center of the city could not be assigned and was canceled. As the platform aims on minimizing cancellations, cost of one realize.

The new decision epoch is initialized when the next driver, namely driver three in this example, requires a repositioning heatmap. As the demand is uncertain, it is also uncertain when the next decision epoch occurs ($t = 75$ in the example). The transition leads to a new distribution of the fleet, based on the repositioning decision of driver one, and the assignments of users to drivers and the corresponding trips.

4.3.3 Sequential decision process

The problem at hand is a stochastic dynamic problem where decisions are made repeatedly and under incomplete information. Furthermore, decisions made now impact the future states of the system. For example, using a specific heatmap strategy changes the distribution of drivers in the future. To capture these dependencies in the mathematical model, we rely on the modeling framework by Powell (2022) on sequential decision processes. This model connects decision states with decisions, revelation of new information, and a transition function. In the following, we present the sequential decision process for the problem, i.e., the definition of decision epochs, states, decisions, stochastic information, and transitions. First, we introduce some preliminary notations.

Preliminaries

We assume operations during a time horizon $T = [0, t^{\max}]$, where time units are discretized by rounding down to minutes. Operations take place in a service area (N, E, \mathcal{T}) with N being the set of locations in the city, E the set of edges between the locations, and \mathcal{T} the constant travel times on the edges. We define a set of potential repositioning locations $R \subset N$. We further assume a fleet of m drivers working all day with initial idling positions $\rho_0 = (\rho_{01}, \dots, \rho_{0m}) \in R^m$ and homogeneous capacity and service duration per stop. There is no termination location, but all operations end (latest) at time t^{\max} .

Decision epochs

Decision epochs occur whenever a driver has finished service and checks the app for RH. Thus, the time of the next decision epoch and the overall number of decision epochs is uncertain. We denote decision epochs as $k = 1, \dots, K$ with K being a random variable.

States

A state S_k at decision point k contains the following information:

- The current point of time t_k .
- The location information of the driver currently requesting a heatmap: n_k .
- The status of the other drivers: Given the large scale of the underlying problem in drivers and customers, we refrain from modeling the individual routes and stops for every driver (and user) in the system. Instead, we focus on the o_k currently unoccupied drivers either idling at or assumed traveling to a repositioning location.

We model the corresponding information using vectors $\rho_k = (r_{1k}, \dots, \rho_{o_k k})$ and $\chi_k = (r_{1k}, \dots, \chi_{o_k k})$ to represent the current location and the last recommended repositioning. Notably, in case of $\rho_{jk} \in R$, the driver j is already at a repositioning location, whereas otherwise it is assumed that the driver is heading towards the last recommended one $\chi_{jk} \in R$.

In essence, a state can be summarized as $S_k = (t_k, n_k, \rho_k, \chi_k)$. The initial state is at time zero with no driver requesting a heatmap and all drivers idling at their initial locations, $S_0 = (0, -, \rho_0, \chi_0)$. There is no decision made in the initial state.

Decisions

A decision x_k is the heatmap of repositioning opportunities shown to the driver. A heatmap decision is a vector $x_k = (x_{1k}, \dots, x_{|R|k})$ of values $x_{rk} \in \mathbb{R}_+$ for each repositioning location $r \in R$. Higher values of x_{rk} indicate higher opportunities. There are no direct costs associated with a decision.

Stochastic information and transition

The stochastic information $w_{k+1} = (r_{k+1}^w, D_{k+1})$ is twofold and reflects repositioning and the occurrence and treatment of new demand:

First, it contains a new repositioning location r_{k+1}^w for the driver requesting a heatmap in S_k based on heatmap decision x_k . In our experiments, we assume that the probability of location $r \in R$ being selected is based on the value x_{rk} (higher probability with higher value; the full details will be provided in Section 4.5.2). The exact location selected becomes known with the driver's arrival.

Second, demand D_{k+1} realized and is served via the platform's assignment and routing procedure until another driver requests a heatmap. The demand is realized until a point of time t_{k+1} where the assignment and routing procedure induces the next free driver requesting a heatmap and therefore the next decision epoch $k + 1$ at time t_{k+1} . It also leads to updated sets ρ_{k+1} and χ_{k+1} in case drivers arrive at a location or are assigned to a request. Furthermore, cancellations D_{k+1}^c between t_k and t_{k+1} due to insufficient driver availability realize with information about time t^p and the nearest repositioning location $l^p \in R$ for a cancellation p . The cancellations define the cost $C(S_k, x_k, D_{k+1}) = |D_{k+1}^c|$. In case $t_{k+1} = t^{\max}$, the process terminates.

Solution and objective function

The solution for the problem is a policy π assigning a heatmap decision $\mathcal{X}^\pi(S_k)$ to every state S_k . An optimal policy π^* minimizes the expected costs (cancellations) when starting in state S_0 and applying policy π^* throughout the process:

$$\pi^* = \arg \min_{\pi \in \Pi} \mathbb{E} \left[\sum_{k=0}^K C(S_k, \mathcal{X}^\pi(S_k)) | S_0 \right]. \quad (4.1)$$

4.4 Repositioning heatmaps

In the following, we present the methodology behind creating RH. We first give a general motivation for and overview of our method and then present the details.

4.4.1 Motivation and overview

When designing effective RH, several factors come into play. RH should enable many future earning opportunities for drivers, but it is not self-evident how to capture them as they are impacted by driver location, future demand, and the behavior of other drivers. Ideally, the travel from the driver’s current location should be kept at a minimum to avoid ineffective empty miles. At the same time, drivers should be guided to an area where significant demand volumes can be expected. These two goals can be competing and hence need to be carefully balanced as, e.g., Larsen et al. (2004) have shown for a single vehicle dynamic service routing problem. In our problem, an additional challenge emerges: the other drivers and their repositioning decisions. When providing an RH to a driver, the overall success also depends on the current fleet distribution as well as the future RH provided to other drivers. For instance, in the worst case, they might all accumulate at the same location. Thus, instead of only balancing travel time and expected demand, an RH should rather consider the current fleet distribution as well as the future “net demand”, i.e., the difference between expected future supply and demand, as other vehicles may enter the respective region in the future. The latter is equivalent to expected future cancellations and therefore particularly challenging to evaluate as it depends on the realized demand, the provided RH, and the driver decisions. We therefore approximate the net demand iteratively as described later in this section.

To capture these considerations algorithmically, we base the future earning opportunities resp. the utility of a repositioning location r for the driver requesting a heatmap in k on three factors:

- The *net demand* that drivers can expect when repositioning to r in the “near” future. Higher net demand will increase earning opportunities. However, net demand in the far-away future may be less relevant and may lead to unnecessary waiting for the driver as well as missed earning opportunities elsewhere.
- The number of *unoccupied drivers* that are currently idling at or that are assumed to be on their way to location r . A larger number will decrease earning opportunities for the drivers. At the same time, sending another driver will likely lead to driver shortage and future cancellations in other areas of the city.
- The *repositioning duration* for the driver between n_k and r . This prioritizes quickly reachable locations to minimize inefficient empty miles.

Integrating these intuitive factors in one holistic RH is already difficult as we discuss in the following section (we will later show that each part is essential for an effective RH design). However, we face an additional challenge, namely, that the net demand (and consequently, the future earning opportunities) is also affected by the policy applied since the policy impacts not only the repositioning location of the driver in the current state but also the distribution of all drivers in later states.

To this end, we propose an adaptive learning procedure that does not change the design process of the heatmap itself but carefully adapts its most important component, the expected net demand. The learning procedure starts with initial expected net demand values, iteratively applies the corresponding policy, and adapts the expected net demand based on the observed values. Therefore, the resulting RH maintain their intuition and integrate the stochastic dynamic developments at the same time. The details of the learning process are presented later in this section, after introducing the initial holistic RH design.

4.4.2 Repositioning heatmap design

The intention of our heatmap design is to link state and learned information in a smart way to provide effective and intuitive repositioning recommendations.

To achieve this, RH are designed as a combination of three underlying heatmaps: one reflecting information on the expected net demand, one reflecting the distribution of unoccupied drivers, and one reflecting the repositioning duration for the driver under consideration. Before formalizing this process, we illustrate how a heatmap is created (see Figure 4.2):

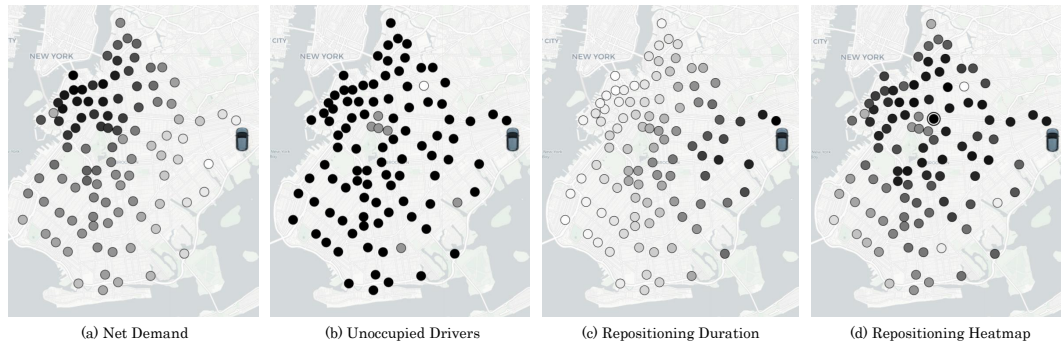


Figure 4.2: Exemplary Repositioning Heatmap (created using Leaflet | © OpenStreetMap contributors © CARTO)

- 4.2(a) The first heatmap component shows the distribution of the expected net demand given the current point of time. Consequently, it indicates where otherwise unmet demand in the near future can be covered through proactive repositioning. In the example, in the northwest, high net demand and therefore earning opportunities are highlighted in darker colors, while in the eastern regions where the requesting driver is located, brighter colors indicate only few opportunities for the near future.
- 4.2(b) The second heatmap component is state-dependent. It displays the current distribution of the unoccupied drivers among the repositioning locations. The information is partially generated by the idle drivers and partially by the drivers that were provided with a heatmap recently. Therefore, it indicates where to expect competing drivers and how many. This is crucial since drivers often become unoccupied in the same area at about the same time (i.e., they are interchangeable from the system’s perspective) and yet should be guided to different repositioning locations to meet the expected net demand. For such drivers, the platform assumes they are traveling to the most fitting repositioning

tioning location suggested by the RH. As drivers do not confirm the location they travel to and may not comply with the heatmap recommendations, the assumed values might differ from the real values. In the example heatmap component, vacant repositioning locations are colored in black, and the most frequented ones are in white.

- 4.2(c) The third heatmap component indicates the travel duration between the driver’s current location and the potential repositioning locations. Incorporating repositioning duration into the heatmap design allocates expected net demand geographically and personalizes opportunities. Accordingly, the repositioning locations are colored in the example from black to white with increasing travel duration.
- 4.2(d) Based on these three components visualized in (a) to (c), the final RH shown in Figure 4.2(d) is created. As can be noticed, a well-balanced recommendation results from the three one-dimensional heatmap components, considering a trade-off between expected net demand and travel time while avoiding direct competitors.

After understanding how the RH are constructed, we formalize the generation in the following. Let us first consider again the net demand shown in Figure 4.2(a). The information displayed here is based on a learned matrix $c = (c_{rt})_{r \in R, t \in T}$ of values $c_{rt} \in \mathbb{R}_+$ for each repositioning location $r \in R$ and each time $t \in T$. The adaptive learning process applied to obtain reliable estimates of the expected net demand c forms the core of our approach and is discussed in detail in the next section. While Figure 4.2(a) is based on learned information, the Figures 4.2(b) and (c) are based on state information S_k . Figure 4.2(b) is determined by the frequencies b_{rk} with which each repositioning location $r \in R$ occurs. This is derived from the vector of current locations of unoccupied drivers ρ_k , i.e., the vector of the last recommendations χ_k for all drivers j with $\rho_{jk} \notin R$. Figure 4.2(c) results from the set θ_k comprising the travel duration $\mathcal{T}(n_k, r)$ from the current location n_k to the repositioning locations $r \in R$.

To determine the final RH, shown in Figure 4.2(d), the three input vectors c_k , b_k , and θ_k are min-max normalized to ensure comparable scales. RH are then derived

according to Equation 4.2, where the signs emphasize that high values for \bar{c}_k indicate high opportunities and for $\bar{b}_k, \bar{\theta}_k$ low ones:

$$x_k = \bar{c}_k - \bar{b}_k - \bar{\theta}_k . \quad (4.2)$$

The values \bar{b}_k and $\bar{\theta}_k$ are derived directly from state S_k . Thus, the resulting policy depends on the parameterization of c . In the following, we will describe how c is approximated.

4.4.3 Adaptive learning process

In the presented heatmap design, the expected net demand c plays a crucial role since this reflects the complex interplay of future demand, its routing and assignment, and future driver repositioning. The challenge in approximating these is that using the expected net demand in the heatmap design results in shifts in driver distribution that may affect subsequent states and eventually lead to new net demand in different areas of the city. We therefore propose a learning process that uses and adapts the approximated net demand such that, as we show in our experiments, net demand is stepwise reduced (thus, service availability is increased) until the values converge to a final matrix c . In the following, we first give an overview of the learning process based on Figure 4.3 before explaining the algorithmic details. Figure 4.3 shows the data involved as well the four main steps of the learning process. The data are:

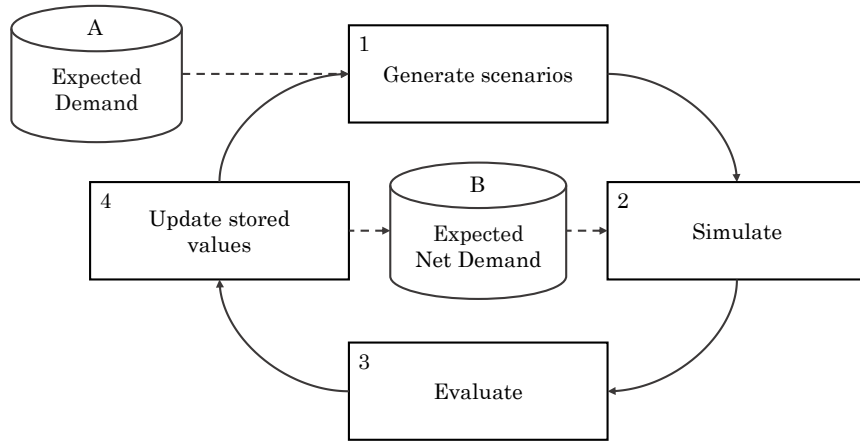


Figure 4.3: Adaptive Learning Process

A: As input the expected demand \mathbb{D} . We assume that in practice, the expected demand can be derived from forecasts or historical booking data. However, for our experiments, we use a pool of trip requests to sample expected demand scenarios.

B: The estimated expected net demand values learned from the previous iteration. The values are updated after each iteration and used as the expected net demand matrix c in the design of the policy applied in the next iteration.

Using this data, the following four process steps are executed iteratively:

- 1: Generation of multiple scenarios based on the expected demand \mathbb{D} . Each demand scenario depicts the demand for the entire service horizon. For this purpose, the demand is sampled (from A) on the basis of varying random seeds. This implies that the spatial and temporal distribution of demand remains fairly constant, but the actual trip requests vary.
- 2: Simulation of the scenarios. In the simulations, RH are determined in each occurring decision epoch k based on the state information S_k and the expected net demand c (stored in B).
- 3: The occurrence of cancellations in time and space is evaluated for all scenarios.
- 4: The obtained cancellations are then used to update the stored expected net demand (in B).

In the following, we discuss the details of Algorithm 1. This algorithm processes observed cancellations, whose temporal and spatial contribution to the expected net demand must be carefully defined. To this end, we first motivate a temporal discounting function $F(t)$ as input as well as a spatial discounting matrix \bar{g} :

- **Time:** With respect to $F(t)$, preliminary experiments indicated that the expected net demand should be estimated uniformly based on the cancellations observed for the next 60 minutes. Shorter time horizons do not provide sufficient lead times to counteract impending cancellations, while longer ones do not indicate repositionings currently required.

- **Space:** Cancelled demand could have been satisfied not only from the closest but from several repositioning locations. To integrate this, we define a *spatial discount matrix* \bar{g} . This matrix provides the converted min-max normalized travel times between all repositioning locations $r \in R$. Thus, neglecting the temporal aspect, cancellations are factored with a value of 1 for the nearest repositioning location and with a value of 0 for the most distant one. The continuously decreasing spatial impact of cancellations favors the balancing of expected net demand. Moreover, the indirect consideration of travel times also increases the value of conveniently repositioning locations to a greater extent.

```

input : Expected demand  $\mathbb{D}$ , Repositioning Locations  $R$ 
         Temporal-Discount-Function  $F(t)$ ,
         Spatial-Discount-Matrix  $\bar{g}$ 
output: Policy  $\pi_c$ 
1 Function ALP( $\mathbb{D}, R, F, G$ )
2    $c(R, t^{\max}) \leftarrow 0$ ; // Initialize net demand matrix
3    $i \leftarrow 0$ ; // Initialize learning iterations
4   /* Perform  $i^{\max}$  learning iterations: */
5   while ( $i < i^{\max}$ ) do
6      $\pi_c \leftarrow c$ ; // Create policy from net demand matrix
7      $P_i \leftarrow \emptyset$ ; // Initialize empty set of cancellations
8      $j \leftarrow 0$ ; // Initialize simulation iterations
9     /* Simulate  $j^{\max}$  scenarios: */
10    while ( $j < j^{\max}$ ) do
11       $D_j \leftarrow \text{scenario}(\mathbb{D})$ ; // Generation of a scenario
12       $O_{D_j, \pi_c} \leftarrow \text{simulate}(D_j, \pi_c)$ ; // Simulation of the
13      scenario
14       $P_i \leftarrow P_i \cup \text{cancellations}(O_{D_j, \pi_c})$ ; // Collecting
15      the cancellations from the results
16       $j \leftarrow j + 1$ 
17    end
18    /* Update entries in net demand matrix: */
19    for ( $r \in R$ ) do
20      for ( $t \in T$ ) do
21         $c_{rt} \leftarrow$ 
            $\frac{i}{i+1} \cdot c_{rt} + \frac{1}{i+1} \cdot \text{getUpdateValues}(r, t, P_i, F, G)$ 
           // Update of an entry in the net demand
           matrix
22      end
23    end
24     $i \leftarrow i + 1$ 
25  end
26  return  $\pi_c$ 

```

Algorithm 1: Adaptive Learning Process

In addition to the two discount functions, the input of the algorithm consists of the expected demand \mathbb{D} for sampling scenarios, and the set of repositioning locations

R . The output is a policy π_c that defines a repositioning heatmap for each state S_k based on the expected net demand matrix c .

The algorithm is initialized in the first three lines. As line 2 indicates, no net demand matrix is given initially. Repositioning is thus restricted to the nearest repositioning location in the initial iteration. In line 4 the actual learning process starts running over i^{max} iterations. Each iteration starts by deriving a heatmap generating policy π_c based on the current expected net demand c (line 5). In addition, an empty set P_i is defined to collect cancellations to be processed later (line 6). Lines 8 to 13 cover the steps scenario generation, simulation and evaluation. How often these steps are executed depends on the number of scenarios j^{max} considered for each learning iteration. In line 9, a scenario D_j is generated from the expected demand \mathbb{D} by sampling of trip requests (function $scenario(\mathbb{D})$). The simulation of the scenario D_j is performed next, applying the incumbent policy π_c (function $simulate(D_j, \pi_c)$). From line 11, the cancellations are transferred from the simulation result O_{D_j, π_c} to the set P_i (function $cancellations(O_{D_j, \pi_c})$).

The cancellations collected in P_i are subsequently used in lines 14 to 18 to update the expected net demand c . For this purpose, it is iterated for each repositioning location $r \in R$ over each time $t \in T$. The update of the matrix entry c_{rt} is executed in line 16. We update via a gradient descent of decreasing step size to ensure convergence of the process. More specific, we rely on the cumulative average over all previous iterations. Thus, the value c_{rt} is updated as shown in line 16 of the algorithm with the function $getUpdateValues(r, t, P_i, F, G)$, returning the updated values based on the temporally and spatially discounted cancellations in P_i . After the expected net demand c has been updated, the next iteration of the learning process begins. For our experiments, we set the number of iterations performed $i^{max} = 10$, as well as a number of simulated scenarios per iteration $j^{max} = 20$, as preliminary tests indicated that this amount is more than sufficient. As a result, the final policy is our RH.

4.5 Experimental setup

In this section, we present the main setup for the computational evaluation of the proposed RH (sensitivity analyses are provided in Appendix B.4). We present the design of the test instances and the modeling of the drivers' decentralized repositioning.

tioning decision-making. We also introduce the repositioning policies that serve as performance benchmarks.

4.5.1 Instances

In the following, we describe how we design the service area from real-world data, and how we create and assign the demand of the ride-sharing system at hand.

Service area

For the computational evaluation, we investigate ride-sharing systems operating in the urban area of New York City. More precisely, we analyze the performance of two systems that operate separately in the boroughs of Manhattan and Brooklyn. The boroughs differ in size, shape, and demand distribution. Manhattan is comparatively smaller, characterized by its island shape, and shows a more temporally and spatially unbalanced demand. In contrast, Brooklyn is much broader and relatively circular, with a demand concentration north of the center, although demand appears to be more evenly distributed. In our simulations, the areas are represented by 3000 unique locations sampled from taxi trip data from January 2014 (NYC Taxi and Limousine Commission, n.d.). Travel times are based on OpenStreetMap free-flow travel times which are multiplied by two to obtain a simple approximation of rush-hour traffic congestion. For the operation of the ride-sharing system, for each service area, 100 of the 3000 possible locations are defined as taxi rank-like repositioning locations. To achieve a fairly even distribution of these locations, the selection is made by means of a k -medians clustering algorithm using latitude and longitude values. The ride-sharing fleet in our main experiment is assumed to consist of 200 drivers operating homogeneous vehicles with four passenger seats for the entire service horizon. In practice, crowdsourced systems may experience drivers entering and exiting the fleet. Therefore, in Appendix B.5, we further evaluate the effectiveness of our heatmaps in case of dynamically varying fleet sizes.

Demand creation

The planning period under consideration covers an 8-hour afternoon shift from 14:00 to 22:00. To exclude warm-up and cool-down phases of the simulation, the first and

last 30 minutes are not taken into account for the evaluation. At the beginning of the planning period, the drivers are randomly distributed among the repositioning locations ready to fulfill incoming transportation requests. A number of 6400 incoming transportation requests is assumed per simulation run. Precomputational experiments revealed that a total of 6400 requests would allow for a reasonable relationship between fleet size and transportation requests, so that cancellation rates range from a minimum of 1.4% to a maximum of 30%. Each transportation request includes the transportation of one user. For each simulation run, transportation requests are sampled from a pool of 100,000 trips per service area performed by Uber or Lyft in September 2019 (NYC Taxi and Limousine Commission, n.d.).

We obtain a realistic spatial and temporal distribution following the real demand as follows. Since information on request times is not provided in the data set and information on pickup and drop-off locations is only available at zone level, we interpret the pickup times as request times and randomly select the pick-up and drop-off locations of the sampled trips for each simulation run from the subset of locations belonging to the respective zones. Each performed simulation run thus differs in the initial locations of the drivers, the request times, and the pick-up and drop-off locations of the trips. For the requesting users, it is assumed that a maximum waiting time of 10 minutes and a maximum travel time of 1.5 times the direct trip’s travel time is acceptable.

Demand assignment and routing

Our heatmap-policy is not bound to a specific routing and assignment strategy and, as we show in Appendix B.4, performs well for a set of different strategies. For our main experiments, we use the most effective assignment and routing strategy tested. For the assignment of demand, we balance efficiency and flexibility based on the insights from Ulmer et al. (2021). Efficiency refers to the consolidation of rides by minimizing the additional travel time, i.e., making use of sharing opportunities. Flexibility is maintained for the vehicle resources by minimizing pickup times. This helps meeting demand as quickly as possible, possibly from drivers who would otherwise be idle. By taking both aspects into account, a more balanced assignment is achieved. The strategy is implemented as follows. When a transportation request comes in, the assignment is performed instantly. To this end, the routes of

all vehicles are checked and all feasible assignments are determined subject to time and capacity-related constraints. If no feasible assignment exists, the transportation request is cancelled. Otherwise, it is confirmed, and the feasible assignment with the minimum sum of additional travel time and pickup time is selected. In case of a tie, the driver who has been waiting longer for an assignment is prioritized. We assume that drivers always accept both the centralized assignments and corresponding routing. With respect to vehicle routing, it is assumed that drivers either serve requests via the centrally planned routes or take the shortest path to a self-selected repositioning location. However, drivers are able to divert from their next stop to serve a new assignment, i.e., drivers do check the app while driving. To this end, in the experiments, out of the 3000 locations considered per service area, those located on a traveled shortest path are considered as possible deviation points.

4.5.2 Modeling driver decisions

In the following, we describe how a driver j in epoch k selects a repositioning location given a heatmap value x_k . We assume that this selection is made as soon as a driver becomes unoccupied, and that it is carried out using the shortest path. The driver decisions are modeled via an additive utility model as presented in Train (2009). The utility of a location r for the driver j in epoch k is the weighted sum of the heatmap value x_{rk} and an additional value u_{rjk} reflecting driver-specific preferences and experiences, e.g., with respect to repositioning duration and earning opportunities:

$$r_{k+1}^w = \arg \max_{r \in R} \{ \alpha_j \times x_{rk} + \beta_j \times u_{rjk} \}. \quad (4.3)$$

The weights α_j, β_j determine the overall compliance of the driver with the provided RH and are unknown to the platform. These weights differ for every driver, but they are assumed to be constant over time. We assume three value combinations for α_j and β_j , resulting in three types of drivers: (1) *Fully compliant* drivers with $\alpha_j, \beta_j = \{1, 0\}$, relying solely on the heatmap recommendations; (2) *Partial non-compliant* drivers with $\alpha_j, \beta_j = \{0.75, 0.25\}$ who enrich the RH with their own experiences or preferences; and (3) *Non-compliant* drivers with $\alpha_j, \beta_j = \{0, 1\}$ who are not willing to use the RH and thus decide on the basis of their own experience only.

4.5.3 Benchmarks

In our computational experiments, we will compare RH with the two benchmarks policies: “Nearest Repositioning” (NR), and an MPC-based method. These two policies were selected as they feature different degrees of repositioning idle drivers. NR specifies that drivers always repositions to the nearest repositioning location as soon as they become idle, which minimizes repositioning efforts. For MPC, we implement a mixed-integer programming model proposed by Pouls et al. (2020). This model is periodically solved for centralized repositioning of all idle drivers in order to maximize the coverage of the predicted demand at the minimal number of repositionings and minimum travel times. Details of how we adapted this approach can be found in Appendix B.1.

4.6 Results

We will present the results in two parts. First, we investigate the effectiveness of RH under “perfect” operating conditions, evaluating the impact of its application on the different actors of a ride-sharing system. For this purpose, we will assume compliant drivers who follow the repositioning recommendations provided by our RH. The evaluation will be made in comparison to the two benchmark policies NR and MPC. Second, we analyze repositioning under different compliance levels. We first examine the case that all drivers use RH but deviate from their recommendations. We compare the performance of cases where only a part of drivers uses the heatmap, i.e., the case of “experienced” versus “inexperienced” drivers, where the first relying solely on their own experience of earning opportunities. Then, we assume that drivers do not have any knowledge and analyze how RH-based guidance changes their earning opportunities. All results represent the average over 100 simulation runs, with instances generated based on different random seeds.

4.6.1 Evaluation under compliance

In the following, we will provide an overall analysis of the results for compliant driver behavior. Based on this, we will demonstrate the learning process of the heatmaps and show their impact on service availability and driver fairness.

System performance

For the setting of decision-making with compliant drivers, we first analyze the impact of heatmap-based repositioning on system performance, focusing mainly on minimizing cancellations. The idea is to demonstrate the performance of RHs compared to the benchmark policies and to investigate the impact of the information considered to create RHs.

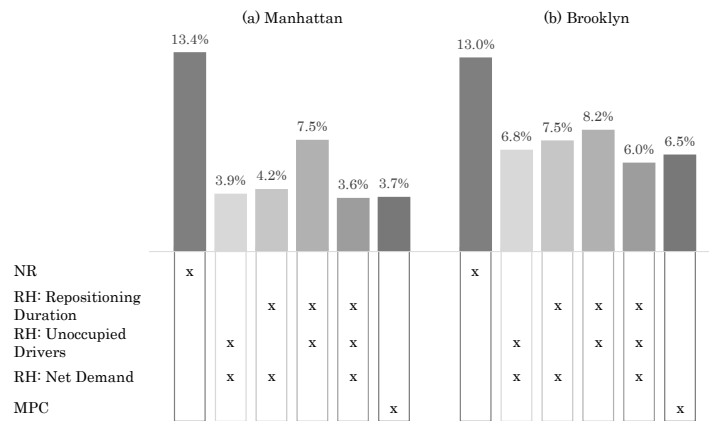


Figure 4.4: Average Cancellation Rates for Manhattan and Brooklyn

Cancellation rates are shown in Figure 4.4 for the Manhattan and Brooklyn service areas. The crosses below indicate the corresponding repositioning approach. We compare the proposed RH to the benchmark policies NR and MPC as well as to RH variants in which some components have been omitted. Additional results for alternative setups and further performance metrics can be found in Appendix B.4. Our RHs yield the lowest cancellation rates for both service areas of 3.6% and 6.0%, respectively. NR is performing worst, with about 13.4% of transportation requests being canceled in Manhattan and 13.0% in Brooklyn. The second best results are obtained with MPC, with slightly higher cancellation rate of 3.7% in Manhattan and 6.5% in Brooklyn. For the other RH variants, neglecting the net demand information proves especially disadvantageous.

Insight: RH are a subtle, but powerful tool for the effective repositioning of drivers, even when compared to an MPC approach. Successful heatmap strategies should consider both the current setup (fleet distribution, driver location) as well as future demand and fleet movement (represented through net demand).

Learning process

A critical feature of our RH is the adaptive learning process. Our experiments show that the importance of learning differs for different instance settings. For some, learning is very important, for others, the first approximation already provides very effective decisions. To illustrate the learning and analyze when it is particularly valuable, we show the extreme cases of our experiments. The corresponding learning curves for Manhattan and Brooklyn are shown in Figure 4.5, with the reduction of the cancellation rate relative to the first iteration plotted on the y -axis, and the process iteration plotted on the x -axis. The light gray curve reflects the learning curve of the setup with the least observed potential to reduce the cancellation rate through learning, while the dark gray curve refers to the setup with the maximum observed potential (We discuss the dependency between setup and learning later in this section). Iteration 1 corresponds to the cancellation rate after the initial expected net demand has been included in the RH.

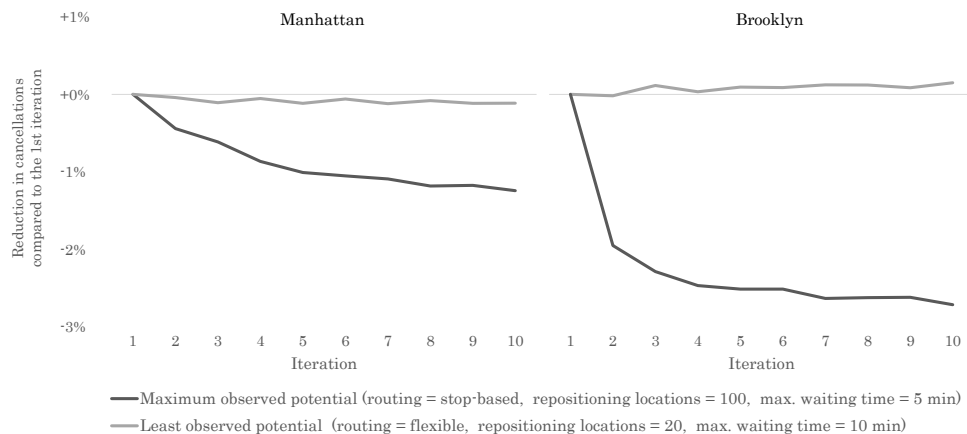


Figure 4.5: Exemplary Learning Curves for Different Instance Settings

We see convergence in all learning curves. Convergence is usually achieved after 10 iterations for all tested settings. We also see that the learning potential differs with respect to region and setting. In the case of Manhattan and the smallest learning value, cancellation rates remain nearly on the same level as in the first iteration. The high-learning potential example shows an additional reduction of up to 1.5% compared to the initial cancellation rate. For Brooklyn, the cancellation rates in the low potential example also remain on the same level. In the high-potential learning case, however, the iterative process helps to decrease the relative cancellations by additional 3% compared to the cancellation rate after the first iteration.

The presented learning curves demonstrate that the iterative learning process improves the overall performance of the heatmaps. The difference between the improvement levels shows that the overall setup determines whether the initial net demand values are already effective. In particular, the learning potential appears to be high when the destination selection is more complex (more distributed demand in Brooklyn, large number of repositioning locations), the commitment of the decision is high (stop-based routing without deviation vs. flexible routing), and anticipation is more important since service requests are more urgent (low maximum waiting time).

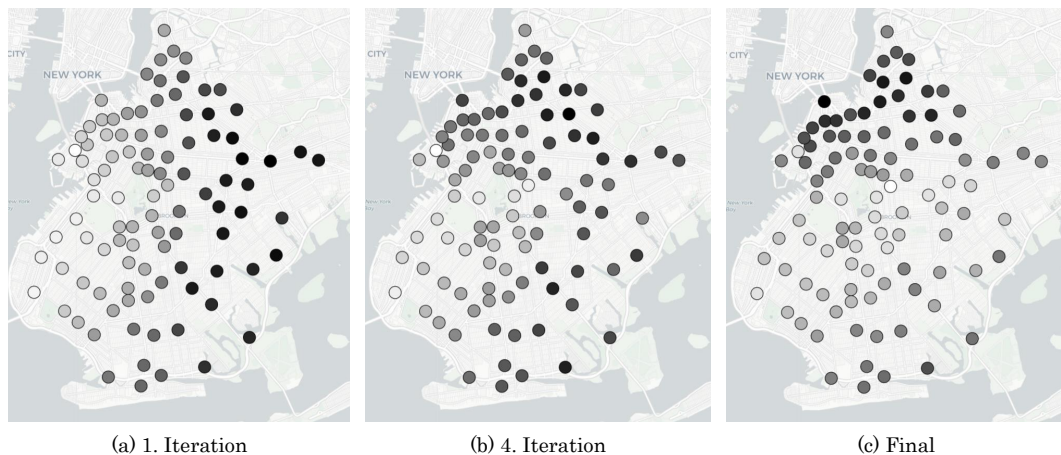


Figure 4.6: Net Demand Heatmaps for Exemplary Learning Process (created using Leaflet | © OpenStreetMap contributors © CARTO, darker shades reflect higher values)

In addition to the general benefits of adaptive learning, we want to illustrate the adjustments of the expected net demand for an exemplary simulation run. For this

purpose, we compare for a fixed point in time the distribution of expected net demand after one and four iterations as well as after completion of the learning process. Figure 4.6 shows the corresponding net demand heatmaps for 2:30pm, with the repositioning locations indicated as circles colored following the normalized expected net demand from black (high) over grey (medium) to white (low). The first heatmap (4.6a) shows a clear distribution characterized by decreasing expected net demand from the east (white) to the west (black) of the service area. After four iterations of the adaptive learning process, this distribution appears to be diminishing, with expected net demand increasing, particularly for repositioning locations in the north and decreasing for those in the east (4.6b). For the final heatmap, this trend is concluded with service opportunities expected primarily in the north of the service area (4.6c).

Insight: With our adaptive learning strategy, the distribution of expected net demand systematically adjusts over the course of the learning process, leading to better repositioning recommendations and reduced cancellations. Thus, considering the dynamic interactions between demand development, RH-based guidance, and driver decision-making is crucial for successful development and communication of RH strategies.

Service availability

In evaluating the learning process, it became visible that the distribution of expected net demand can be learned systematically. This poses the question of whether the associated avoidance of cancellations leads to an improved and more balanced service availability throughout the service area. This would be critical for user retention, as insufficient service availability or systematic discrimination against certain parts of the service areas could induce user dissatisfaction and churn.

To examine regional service availability, we analyze the cancellation rates per repositioning location compared to those of NR. For this purpose, the requests are assigned to the repositioning location that is nearest to the pickup location in terms of travel time. Figure 4.7 shows the corresponding cancellation rates per location by circles of different shades of grey and their demand volumes using their size. Here, the scale of cancellation rates ranges from a minimum of 0% (white) to a maximum

of 50% (black). A small white circle, for instance, indicates a low cancellation rate for a low demand volume in the vicinity of the repositioning location, while a large black circle indicates a high cancellation rate for a high demand volume.

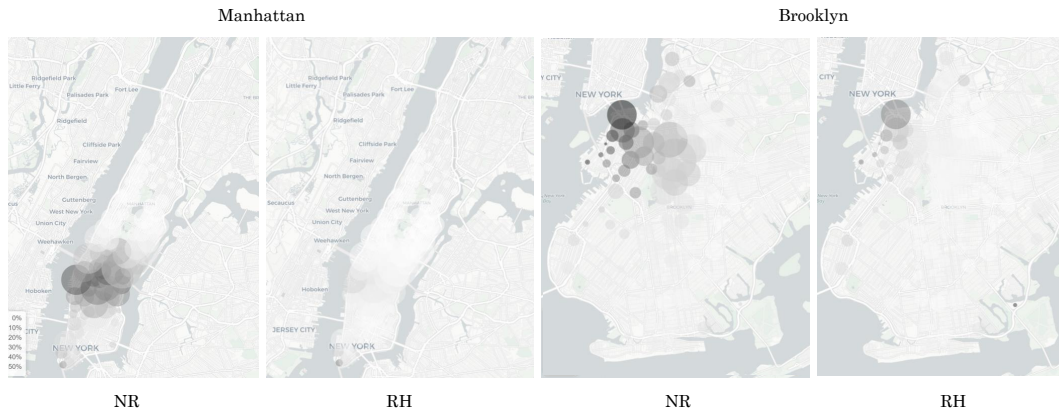


Figure 4.7: Distribution of Cancellation Rates by Color and Demand by Size (created using Leaflet | © OpenStreetMap contributors © CARTO)

For Manhattan, the demand center in the middle of the island is clearly visible, as well as a relatively large region with very low demand in the north. Here, for NR, the grey circles in the lower part of the demand center are noticeable, indicating increased cancellation rates. For RH, we do not see big differences, indicating relatively balanced (and low) cancellation rates. For Brooklyn, the demand center appears slightly north of the geographic center. Here, for NR, the light to dark grey circles north of the demand center are particularly prominent, indicating comparatively high cancellation rates in this region. Again, in the case of RH, all circles are colored brightly and thus indicate relatively low cancellation rates, with one small exception in the southeast, where the size of the circle indicates a very scarce demand.

From the results of both service areas, it can be concluded that RH greatly contribute to increased and more balanced service availability compared to NR. It is apparent that RH particularly help to decrease the otherwise high cancellation rates in high-demand regions. Moreover, these improvements are only slightly detrimental to regions with very scarce demand located at the outermost corners of the service area.

Insight: RH lead to a more balanced and higher service availability not only in demand hotspots but across the entire service area. Consequently, providers can serve more users with a constant pool of drivers and ensure a more balanced and higher service availability. These enhancements are likely to increase user satisfaction and, therefore, their commitment to the ride-sharing system.

Driver fairness

Having shown that platform providers would benefit from the application of RH, it remains to be investigated whether the same is true for drivers. To evaluate driver satisfaction, we focus on the total ride time per driver, which is a proxy for a driver's earnings. The corresponding boxplots for Manhattan and Brooklyn are shown in Figure 4.8 for RH and are compared to those of NR and MPC. Each of these boxplots represents the total time passengers are transported by a driver, i.e., the time the driver is paid for by the passengers, for 200 daily drivers \times 100 simulations = 20,000 drivers.

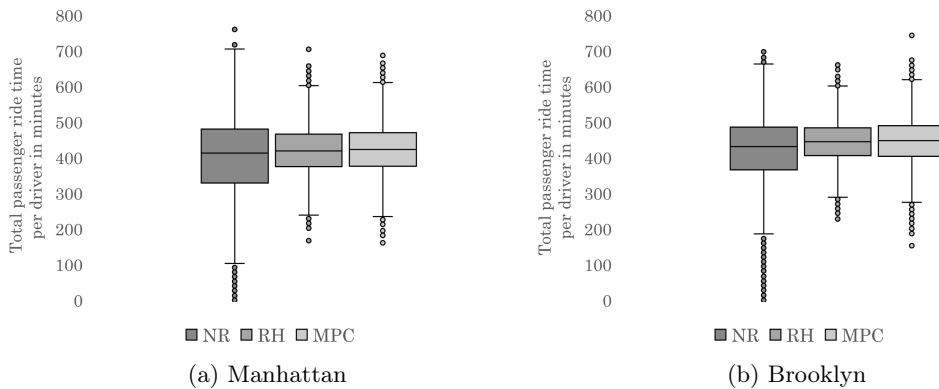


Figure 4.8: Total Passenger Ride Time per Driver

For Manhattan and Brooklyn, the median of the total ride time is about 400 for all approaches, with slightly higher values for RH and MPC, as they lead to fewer cancellations. More distinct differences can be observed among the interquartile ranges. This range is clearly most comprehensive for NR with a total time of about 325 to 475 and decreases to about 375 to 450 for MPC and RH. The same ranking can

be observed with respect to the length of the whiskers. For NR, outliers indicate that some drivers cannot earn much. In contrast, the outliers in RH and MPC are less pronounced and occur both positively and negatively. The decreasing magnitudes of interquartile ranges illustrate that the distribution of earnings among drivers is more balanced for RH than NR and, to a lesser extent, for MPC. This involves far fewer drivers receiving significantly below-average or above-average earnings.

Insight: RH contribute to a more balanced and fairer distribution of earnings among drivers. Thus, it is likely that drivers can be retained more easily. However, even though the average and minimum earnings increase, an even distribution of earnings can negatively affect a driver’s personal income. Thus, the extent to which the equitable distribution contributes to the acceptance of the proposed heatmaps depends on prior earnings as well as the driver’s personal mindset toward equal opportunities.

4.6.2 Evaluation under non-compliance

Next, we examine what happens when drivers do not always follow the recommendations indicated by their heatmaps or do not consider them at all. To this end, we investigate different levels and variants of non-compliance. We first conduct experiments where drivers are only partially non-compliant to see how the performance of RH depends on strict driver compliance. Second, we analyze what happens when compliant drivers compete with inexperienced and experienced non-compliant drivers to further explore the value of our RH.

Partially non-compliant drivers

In the first set of experiments dealing with partial non-compliance, we focus on the most prominent factors to model the driver specific values u_{rjk} , namely repositioning duration and supply/demand information (Urata et al., 2021). We treat the two factors individually, and for each factor, we assume that there are two groups of drivers evaluating repositioning options differently. For the travel duration, some drivers are called *time savers* trying to avoid much repositioning. Others are called *time investors* traveling more to reach potentially fruitful demand areas. For the

supply/demand factor, we again consider two groups, separated by how drivers value supply and demand. Here, some drivers try to avoid areas with many competitors (*supply-driven*) while others focus on regions with high demand having competition only as an afterthought (*demand-driven*). For more details, we refer to Appendix B.2.

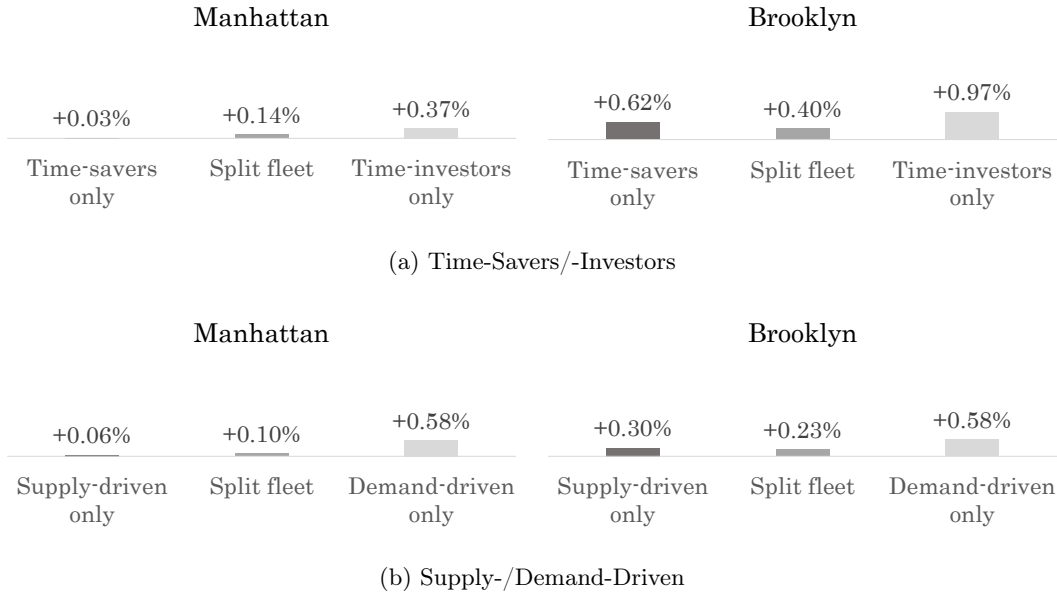


Figure 4.9: Cancellation Rates Relative to the Full Compliance Case

Figure 4.9 visualizes the cancellation rates relative to the case of full compliance for experiments in which all drivers are homogeneously partially non-compliant, as well as for two split-fleet experiments. For the split fleets, each driver is randomly assigned to the group of (1) heatmap-compliant ones, (2) time savers, or (3) time investors. For the second experiment, we have drivers who are either (1) heatmap-compliant, (2) supply-driven, or (3) demand-driven.

With time savers and time investors, for Manhattan, there is no significant impact on the overall system performance: cancellation rates increase only slightly (between +0.03% and +0.58%). They are slightly higher for Brooklyn, with a max increase of 0.97%. With supply-driven and demand-driven drivers, the picture is quite similar (only slightly increasing cancellation rates). In essence, even when drivers partially follow their own agenda, the overall performance of our heatmap-policy remains relatively stable, but the best performance is achieved when all drivers comply.

Let us now have a look at how compliant and non-compliant driver behavior impacts the different groups' earnings. In Figure 4.10, we see the differences in the average total passenger ride time again relative to the case where all drivers are compliant. For time savers versus time investors, time investors collect assignments when driving to and at opportunity hotspots, increasing their earnings by +13.3% and +5.4% for Manhattan and Brooklyn, respectively. Compliant drivers suffer from this considerably (-5.7% and -2.3%), and time savers are off even worse (-9.2% and -5.1%).

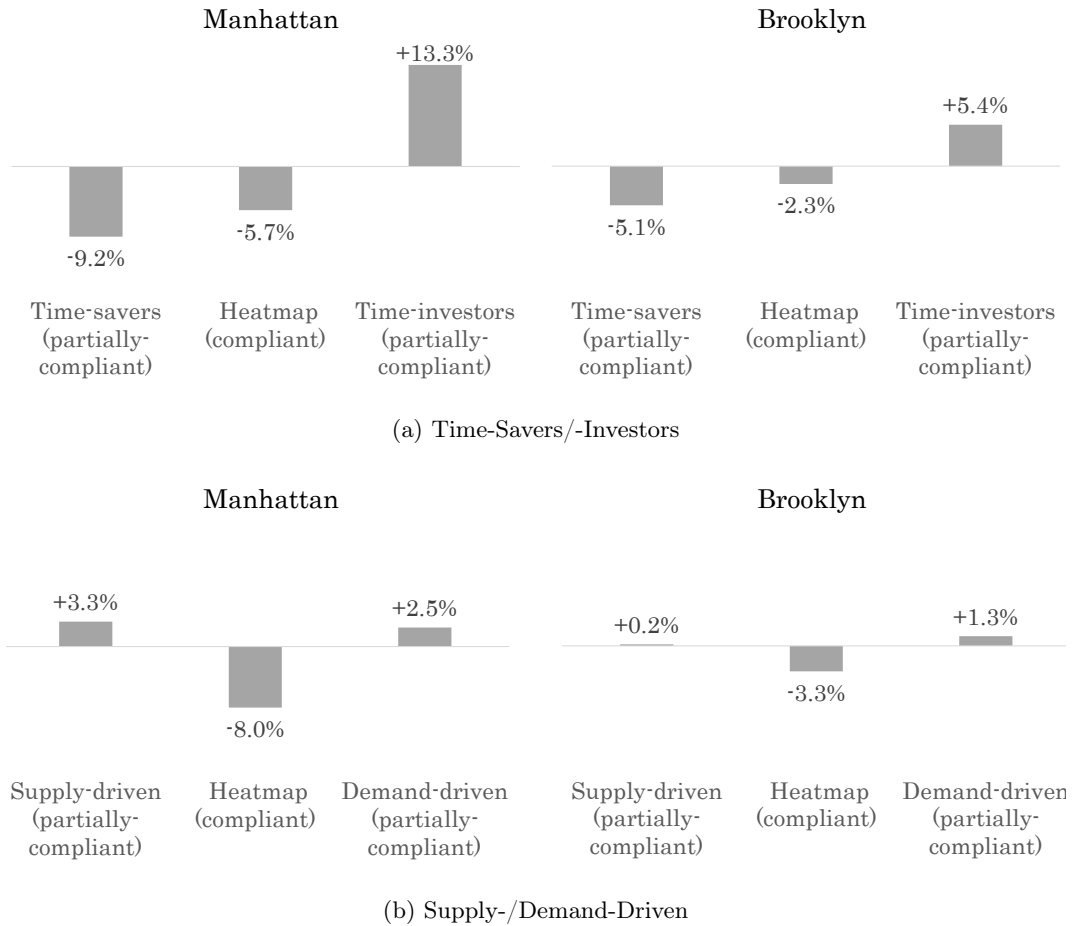


Figure 4.10: Split Fleets: Earnings for Compliant Drivers Competing with Partial Non-Compliant Drivers

For supply-driven versus demand-driven drivers, again, the general development for Manhattan and Brooklyn is similar, but variations are larger for Manhattan. Both supply-driven and demand-driven drivers can increase their earnings (+3.3% and +0.2% for supply-driven, +2.5% and +1.3% for demand-driven). Meanwhile, compliant drivers suffer considerably (−8.0% and −3.3%). However, if we further analyze the empty travel time, in Manhattan, for instance, earnings increase by around 30% and 25% for supply-driven and demand-driven drivers, respectively. The increased earnings are thus indirectly attributable to a very extensive repositioning, either to escape the competitors, or to reach demand hotspots.

Insight: Our results demonstrate that a ride-sharing system can achieve comparable results even if drivers are not completely compliant with RH. However, to some extent, drivers can “game the system” through more extensive repositioning leading to less earnings for compliant drivers. Such potentially undesirable behavior could be alleviated by providers through a compliance encouraging assignment strategy.

(In)Experienced drivers

We now let compliant drivers compete against *inexperienced* and *experienced* non-compliant drivers. For the inexperienced drivers, it is assumed that they do not bother much with repositioning and therefore always select a repositioning location in their vicinity (like partially-compliant *time savers* but without a heatmap). In the case of experienced drivers, by contrast, we assume that they have already successfully devised their own repositioning strategy. This is modeled by combining the information from *time investors* and *demand-driven drivers*, since this looked very promising in the previous experiments. For more technical details, we refer to Appendix B.2.

We first analyze the performance of a system with only inexperienced or experienced drivers (and no heatmaps). In Figure 4.11, we compare the corresponding cancellation rates relative to the fully-compliant case. In the case of a split fleet, cancellations increase for Manhattan and Brooklyn moderately by 1.0% and 1.5%, respectively. With a fleet of experienced drivers, cancellation rates in Manhattan increase comparably by +0.9%, while in Brooklyn this is more pronounced with +2.0%.

So even in the ideal case that all drivers are experienced, following the heatmaps lead to a better system performance. With a fleet of inexperienced drivers, cancellation rates increase significantly (+4.9% for Manhattan, +4.1% for Brooklyn). Thus, heatmaps are particularly valuable in case of many new drivers or when entering a new service area.

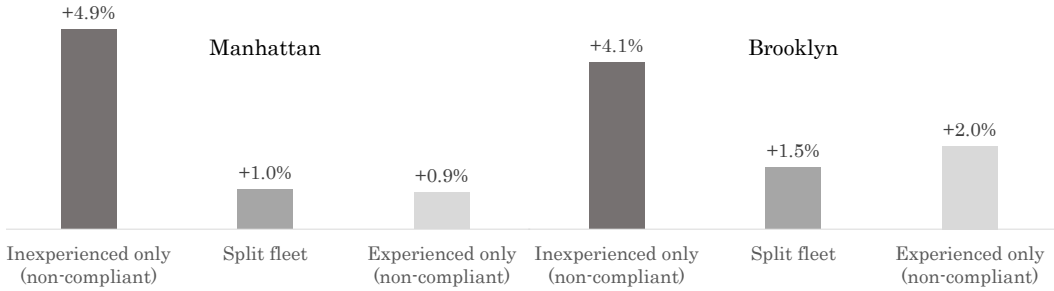


Figure 4.11: Cancellation Rates Relative to the Fully-Compliance Case

Now, what happens when compliant drivers have to compete with inexperienced and experienced drivers? We show the results in the upper part of Figure 4.12. We see that in case of a split fleet, the results differ for Manhattan and Brooklyn. In Brooklyn, being compliant is very valuable for inexperienced drivers, but experienced drivers still earn more. In Manhattan, compliant drivers are even worse off than inexperienced non-compliant drivers. Given the demand structure in Manhattan, experienced drivers seek and inexperienced drivers often end up in demand hotspots. Thus, the compliant drivers have to fulfill the demand in less attractive areas and essentially suffer from the other drivers being non-compliant.

In that case, the provider needs to ensure that compliant behavior is rewarded. To control the benefits for compliant and non-compliant drivers, the provider can modify routing and assignment strategies (see Appendix B.4 for an in-depth analysis). Here, we slightly modify the provider-controlled assignment strategy, now prioritizing compliant drivers in tie-breaker cases and in cases where compliant drivers may require slightly longer travel (< 10%). This does not change the overall system performance significantly, but the experience for the non-compliant and compliant drivers as shown in the bottom part of Figure 4.12. We see that in case of Manhattan, compliance pays off, at least for inexperienced drivers, and in Brooklyn, compliant drivers earn even more than experienced ones.

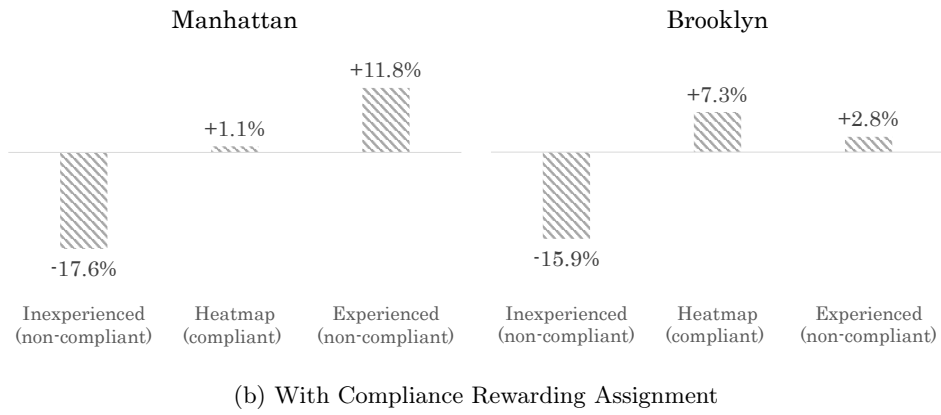
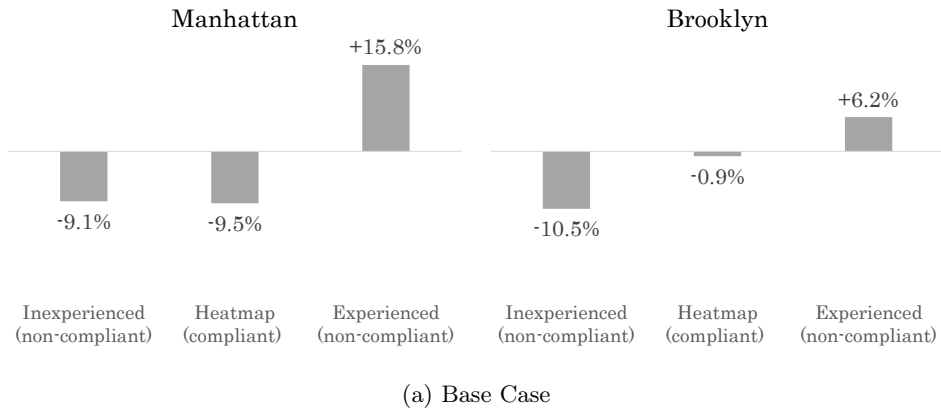


Figure 4.12: Split Fleets: Earnings for Compliant Drivers Competing with Inexperienced and Experienced Non-compliant Drivers

Insight: Our results demonstrate that RH *can* enable inexperienced drivers to close the earning gaps to experienced drivers and that in some cases even experienced drivers are better off using RH – if the platform ensures assignment priority for compliant drivers. This may alleviate the severe practical issue of earning gaps and driver churn of inexperienced drivers as discussed by Cook et al. (2021). Still, as a profit-maximizing driver’s behavior depends crucially on the behavior of the other drivers, there is a substantial potential for “gaming” the system, as also observed by Wang et al. (2023).

4.7 Final remarks

We have illustrated how RH can lead to improved operations for service providers, drivers, and users. There are several avenues for future research. First, our experiments have shown that carefully designed heatmaps reduce service cancellations even in cases that drivers are less compliant with the provider’s recommendation. Future research may focus on explicitly identifying such non-compliant behavior by analyzing the drivers’ previous decisions. This analysis could then be used to adapt the heatmap design accordingly, e.g., by providing “time savers” reluctant to leave their neighborhood with recommendations nearby and use “time investors” to cover areas further away. However, as we have seen in our experiments, heterogeneous driver behavior already leads to imbalances in their earnings even when they are treated equally by the service provider. The imbalances may increase in case the provider further differentiates driver preferences. This leads to the question of fairness as very picky drivers may get very lucrative jobs while others do the heavy lifting. Furthermore, such differentiation may increase the number of drivers gaming the system once they realize that their behavior influences their recommendations. Future research may therefore focus on a fair and balanced repositioning given the heterogeneous driver preferences and ways to disengage drivers from gaming the system. As we observed, one way providers can control the value of compliance is through their routing and assignment strategies. Future work may thus focus on balancing operational efficiency with rewarding compliant behavior.

Another interesting aspect of our experiments is that the impact of driver compliance differs for services areas with different spatio-temporal characteristics. While in the rather small area of Manhattan, even less compliant drivers can achieve a good demand coverage, in the larger area of Brooklyn, non-compliance results in an increase in cancellations. Thus, for larger and more “complex” service areas such as Brooklyn, future research may focus on a better balance when drivers are not fully compliant. Potential options could be different driver compensation in different areas of the city. Alternatively, the provider might complement the crowdsourced fleet with dedicated drivers. Finding the right balance in the fleet and using the potentially more expensive dedicated drivers effectively may be an interesting challenge for future research.

Future research may combine our heatmap strategy with surge pricing. Our RH has shown to be a powerful tool to steer drivers in the right direction to the benefits of both drivers and users using non-monetary guidance. Thus, the heatmap policy does not change the customer demand and fleet size. Still, there might be situations where even with perfect heatmap-based decisions, the current demand or the fleet size are too small. In that case, future work may consider adding monetary incentives by combining our policy with surge pricing strategies. A potential starting point might be the learned net demand information to measure future demand and demand coverage and adapt the prices accordingly. In that case, an integrated learning of heatmap and pricing strategy might be required.

Finally, in our experiments, we have shown that heatmaps lead to more and more fairly distributed earnings amongst the drivers and better service availability for the users. Future research may provide empirical evidence for these improvements by examining driver behavior in response to heatmap information. It could further analyze the long-term impact in comparison with the status quo, e.g., with respect to user retention, business growth, and the drivers' trust in and adoption of the heatmap-based guidance.

Chapter 5

Design of multi-optional pickup time offers in ride-sharing systems

Abstract Ride-sharing systems strive to provide affordable on-demand mobility in urban areas by effectively consolidating incoming transportation requests. To ensure that transportation offers meet travelers' individual time requirements and constraints, service operators offer multiple pickup times from which travelers can choose. Designing such pickup time offers is challenging due to the uncertainty of both the requirements of the requesting traveler and the efficient fulfillment of future demand. We propose a parametric cost function approximation to balance between maximizing the probability that a traveler will choose an offered pickup time and minimizing the expected vehicle routing effort. We demonstrate the effectiveness of the proposed approach in a comprehensive computational study and provide managerial insights, particularly with respect to the value of information on traveler pickup time requirements.

Keywords demand management, ride-sharing, stochastic dynamic decision making, vehicle routing

5.1 Introduction

To support the transition to a more sustainable use of today's congested urban traffic infrastructure, the shift from private cars to shared mobility is publicly subsidized. In cooperation with local transportation authorities, ride-sharing operators such as MOIA or VIA are contributing to this change by operating a fleet of vehicles to offer shared on-demand transportation at affordable prices. Their operational aim is to maximize the level of service, i.e., to satisfy a high percentage of travelers to ensure long-term acceptance and financial viability.

To use such ride-sharing systems, travelers request transportation by entering their preferred pickup and drop-off locations in a mobile application. In turn, they receive a transportation offer which specifies the assigned vehicle, expected pickup time, and fare. Depending on the offer, the traveler either completes or cancels the request. A decisive factor for the offer acceptance is whether the pickup time meets the traveler's requirements (Wang et al., 2020). However, these requirements can vary widely, depending on the circumstances of the request. For example, let us imagine a traveler requesting transportation home on a rainy night. This traveler would certainly prefer to be picked up immediately, but in the absence of alternatives may also accept later pickup times. In contrast, let us think of a traveler who is planning the last leg of their home journey shortly before arriving at a central train station. Such a traveler will need some time to get to the pickup location and can resort to alternative means of transportation if the offered pickup time is infeasible. Lastly, we can think of a group approaching the end of a restaurant visit. In this scenario, for one, they are probably not in a hurry to leave, and for another, may prefer to have similar pickup times. To accommodate such individual pickup time requirements, often unknown to the ride-sharing system (and perhaps even to the traveler), we are among the first to explore the benefits of offering multiple pickup time options. Such offers enable travelers to choose from a set of options the pickup time that suits them best.

Designing such multi-optional pickup time offers is a challenge, as they need to satisfy the requirements of travelers and ensure the effective utilization of the ride-sharing fleet. This is particularly difficult because both traveler requirements and the effectiveness of fulfillment operations are uncertain. While the requirements depend on the current request, the effectiveness depends on future ones. However,

cleverly designed offers enable ride-sharing systems to maximize the service level by striking a trade-off between covering a wide range of requirements and offering the pickup that is deemed most efficient. For example, during low-demand periods with sufficient vehicle resources, the aim could be to cover potential pickup times evenly, whereas, in high-demand periods, the focus might be on offering pickup times that enable the consolidation of transportation requests. The offer design thus provides a soft means of demand management that integrates dynamic vehicle routing to evaluate fulfillment opportunities. By incorporating demand management, i.e., shaping demand in terms of its volume or characteristics, and vehicle routing, i.e., finding efficient routes to fulfill a given transportation demand, this work is in line with recent research on dynamic optimization of transportation services (see, e.g., Fleckenstein et al. (2023)).

For the design of multi-optional pickup time offers, we propose a parametric Cost Function Approximation (CFA) that balances the acceptance probability and the associated routing effort. The two design criteria thus reconcile the satisfaction of the current request and the preservation of vehicle resources in favor of future ones. Regarding the acceptance probability of an offer, we assume that probabilistic information is available based on historical booking data, for example. Vehicle routing effort, in turn, is approximated by aggregating a rating of the associated pickup time options. Here, each pickup time option corresponds to a feasible assignment of the requested transportation to a vehicle route. The rating (following the findings of Ulmer et al. (2021)) indicates whether the assignment contributes to consolidating transports and maintaining fleet flexibility. Finally, parameterization enables balancing the acceptance probability and approximate routing effort to strike the trade-off that minimizes the rate of canceled requests.

In summary, our paper makes the following contributions: we are among the first to consider diverse and uncertain pickup time requirements in ride-sharing systems and, in turn, explore the benefits of multi-optional pickup time offers. To this end, we first introduce the new dynamic and stochastic problem and formalize the corresponding sequential decision process. Second, we propose a solution approach that adapts the concept of a parametric CFA recently introduced by Powell & Ghadimi (2022). Third, we provide a comprehensive computational study that demonstrates

the effectiveness of the approach particularly in comparison to benchmarks exploring different levels of information on traveler pickup time requirements.

The paper begins with a discussion of the related literature in Section 2. A comprehensive problem description follows in Section 3. Section 4 presents how pickup time offers are designed via the CFA. The experimental setup and computational results are discussed in Sections 5 and 6. We conclude with final remarks in Section 7.

5.2 Related literature

In the following, we give an overview of the related literature, focusing, in particular, on the implementation of demand management and vehicle routing in research on ride-sharing systems. For an overview of research considering both aspects in other applications, we refer to Fleckenstein et al. (2023). Most of the corresponding papers focus on time window management for next-day attended home deliveries (see Waßmuth et al. (2023) for a respective review). In contrast, comparatively few papers, such as Ulmer (2020) or Klein & Steinhardt (2023), consider dynamic vehicle routing as in ride-sharing systems, where fulfillment is performed simultaneously with the incoming of new requests. However, ride-sharing also differs from such same-day delivery applications due to the immediacy of transportation requests, which require flexible availability of vehicle resources throughout the service area.

Focusing on the literature addressing ride-sharing systems, it should be noted that part of the research refers to the term dynamic dial-a-ride problem as an extension of the well-known dynamic vehicle routing problem. For reviews on the static and dynamic dial-a-ride problem see, for example, Molenbruch et al. (2017) and Ho et al. (2018). General reviews concerning the dynamic vehicle routing problem are provided by, for example, Psaraftis et al. (2016) and Soeffker et al. (2022). Soeffker et al. (2022) thereby provides an interesting comparison of the CFA adapted in this work with other method classes for sequential decision processes. Another related research stream focuses on shared autonomous vehicles; in this regard, we refer to Narayanan et al. (2020) for an overview. Furthermore, managing ride-sharing systems is not limited to processing transportation requests. We thus refer to Wang & Yang (2019) for a comprehensive review of related optimization problems. This review covers ride-hailing as well, i.e., matching requests and (independent) drivers for direct transportation. In the following classification, we focus on work in which a

ride-sharing system operates a fleet of vehicles aiming to consolidate transportation requests.

The classification is based on five criteria that are decisive for demand management in our work: (1) it reflects the possibility of cancellations due to the unavailability of vehicle resources; (2) it proactively manages demand; (3) it considers uncertain individual traveler decisions; (4) it provides transportation options for the traveler to choose from; (5) it considers uncertain pickup time requirements. The relevant research can be divided into five categories: no demand management, feasibility checks, acceptance mechanisms, pricing, and mode choice. In the following, we will provide a brief overview of the related literature organized by these categories and emphasize the relationship to our work.

| Paper | Approach | Request cancellations | Proactive demand management | Traveler's choice | Transportation offer sets | Uncertain pickup time requirements |
|--|----------------------|-----------------------|-----------------------------|-------------------|---------------------------|------------------------------------|
| Schilde et al. (2011) Hyytiä et al. (2012) Ma et al. (2013) Riley et al. (2019) | No demand management | | | | | |
| Horn (2002) Attanasio et al. (2004) Coslovich et al. (2006) Berbeglia et al. (2011) Alonso-Mora et al. (2017a) Simonetto et al. (2019) | Feasibility-check | ✓ | | | | |
| Xiang et al. (2008) Hosni et al. (2014) Alonso-Mora et al. (2017b) Lowalekar & Jaillet (2019) Shah et al. (2020) Heitmann et al. (2023) | Acceptance mechanism | ✓ | ✓ | | | |
| Sayarshad & Chow (2015) Bimpikis et al. (2019) Ma et al. (2022) | Differential pricing | ✓ | ✓ | ✓ | | |
| Jacob & Roet-Green (2021) Jiao & Ramezani (2022) Sharif Azadeh et al. (2022) | Mode choice | ✓ | ✓ | ✓ | ✓ | |
| This work | Pickup time choice | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 5.1: Literature Classification

First, we consider the *no demand management* category. The corresponding papers assume that all requests must be fulfilled promptly to ensure travelers' satisfaction. Their objective is therefore to minimize waiting times through improved vehicle routing. For example, Schilde et al. (2011) and Hyytiä et al. (2012) focus on increasing effectiveness by anticipating future transportation requests, while Ma et al. (2013) and Riley et al. (2019) aim at increasing efficiency to manage large-scale ride-sharing

systems. However, this problem setup neglects the traveler's ability to cancel requests when pickup offers are inconvenient, which is particularly important because realized waiting times can vary greatly depending on the requested transportation and the current system load.

In contrast, papers belonging to the other categories consider that travelers have the option to decline an unfavorable transportation offer. However, for the sake of simplicity, most assume that requests are only canceled if the pickup time exceeds a predefined waiting time threshold. The assumption is based on the image of travelers waiting at their pickup location and thus being able to be picked up anytime. This overlooks the advantage of digitized systems that allow travelers to make requests in advance from any convenient location, resulting in individually varying lead time requirements. Moreover, the requirements regarding the pickup time depend not only on the traveler but on the overall circumstances. For example, during periods of high demand, Wang & Bei (2022) observed that higher average waiting times correlate with lower average cancellation rates, as alternative transportation options such as cabs are at capacity as well.

With respect to the given waiting time threshold, the approaches in the second category only examine the *feasibility* of a new request without considering anticipatory demand management. However, they have to be further differentiated according to when requests are processed. Horn (2002), Attanasio et al. (2004), Coslovich et al. (2006), and Berbeglia et al. (2011) assign new incoming requests immediately to a vehicle, following a strictly myopic first-come, first-served principle. In contrast, Alonso-Mora et al. (2017a) and Simonetto et al. (2019) perform assignments for batches of requests to alleviate the disadvantage of myopic decisions. This advantage is offset by the disadvantage that travelers may receive their transportation offer delayed, which can contribute to their dissatisfaction. Consequently, recent publications related to crowdsourced systems focus on optimal response delays (for example, Yang et al. (2020), Ke et al. (2022), or Wang et al. (2022)). In contrast, we adopt the idea of immediate assignments, assuming that travelers are less willing to wait for an offer in centralized systems where the availability of vehicles is not subject to uncertainty.

The third category covers approaches in which, in addition to the feasibility, the favorability of new incoming requests is evaluated. These so-called selective *accep-*

tance mechanisms focus either on the current opportunity cost (as, for example, in Xiang et al. (2008), Hosni et al. (2014), and Lowalekar & Jaillet (2019)) or anticipate future demand (as, for example, in Alonso-Mora et al. (2017b), Shah et al. (2020), and Heitmann et al. (2023)). The disadvantages of such selective acceptance mechanisms are systematic discrimination against certain requests, as shown in Haferkamp & Ehmke (2022), and the permanent abandonment of the service by rejected travelers, as discussed in Geržinič et al. (2023). We, therefore, propose an approach towards proactive demand management that refrains from the rejection of transportation requests.

An alternative approach to managing demand in ride-sharing systems proactively is *differential pricing*, as proposed, for example, in Bimpikis et al. (2019) and Ma et al. (2022). They aim to maximize revenue given finite vehicle resources, while likewise assuming that all travelers require a short-term pickup. Their advantage is that all requesting travelers receive a transportation offer but on the condition that a sufficient amount of travelers have to cancel due to a high fare. In contrast to our work, differential pricing thus aims primarily at the control of the demand volume and less on shaping the characteristics of the given demand. Moreover, the implementation of differential pricing is very unpopular among travelers and may also discourage the use of ride-sharing systems (see, for example, Bertini & Koenigsberg (2021) or Abrams (2022)).

The papers of the last category assume shared rides to be optional following the example of services like Uber or Didi. Accordingly, Jacob & Roet-Green (2021), Jiao & Ramezani (2022) and Sharif Azadeh et al. (2022) focus on offering direct and/or shared transportation options mostly in conjunction with differentiated pricing. Thus, similar to our approach, travelers are offered multiple transportation options to choose from. However, their demand management focuses on how rather than when requests are fulfilled.

Additionally, it should be noted that Sayarshad & Oliver Gao (2018) and Liu et al. (2019) also consider travelers' mode choices. While Sayarshad & Oliver Gao (2018) focus on competing ride-sharing operators, Liu et al. (2019) study alternative means of transportation. Again, for both papers, minimizing waiting time is assumed to be the deciding factor for travelers' choices.

In summary, to the best of our knowledge, there is no work that has considered demand management under uncertain pickup time requirements by offering multiple pickup options to provide an effective and convenient ride-sharing system.

5.3 Problem description

In the following, we present the dynamic and stochastic problem under consideration. We first outline the problem in a narrative. The underlying sequential decision process is then illustrated with an example and modeled following the framework of Powell (2022). With respect to the involved vehicle routing problem, for clarity, we restrict ourselves to a descriptive presentation inspired by the modeling of Cordeau & Laporte (2007).

5.3.1 Problem narrative

We envision a ride-sharing system that operates a fleet of homogeneous vehicles to provide on-demand mobility in an urban area. In the course of a day, travelers request transportation via a mobile application, specifying pickup and drop-off locations. Considering the request as well as the incumbent vehicle routes, the service operator immediately offers a set of pickup time options associated with a feasible assignment. For determining the options, a maximum lead time, i.e., a maximum time interval between request and pickup, is assumed in order to comply with the on-demand nature of the system. Moreover, the number of options is limited to ensure that they are clearly distinctive and convenient to display in a mobile application. Based on the offer, the traveler either selects the most suitable pickup time or cancels the request (for details on traveler's choice modeling, see Section 5.5.2).

Once a traveler has chosen an offered option, the associated pickup time and assignment are considered binding to enable reliable planning for both system operators and travelers. This limitation reflects findings from Geržinič et al. (2023) that travelers rate unexpected pickup delays three times more negatively than an agreed-upon deviation from their preference. However, minor deviations are considered tolerable to allow for additional pickups or drop-offs along a planned vehicle route. Furthermore, on the way from the pickup to the drop-off location, detours in favor of shared rides are considered tolerable as a function of the direct travel time.

Accordingly, when choosing a pickup time option, travelers can only be informed about the current planned and latest feasible arrival time.

The objective of the ride-sharing system is to minimize the number of canceled requests by designing convenient pickup time offers for currently requesting travelers while ensuring their effective fulfillment in favor of future ones.

5.3.2 Illustrating example

In the following, we give an example to further illustrate the request processing (see Figure 5.1). The corresponding sequential decision process can be described by the system state (depicted on the left), potential decisions (in the center), and a realization of stochastic information including the resulting transition to a new state (on the right). The example state is at time $t = 90$. The decision is triggered by a traveler requesting transportation.

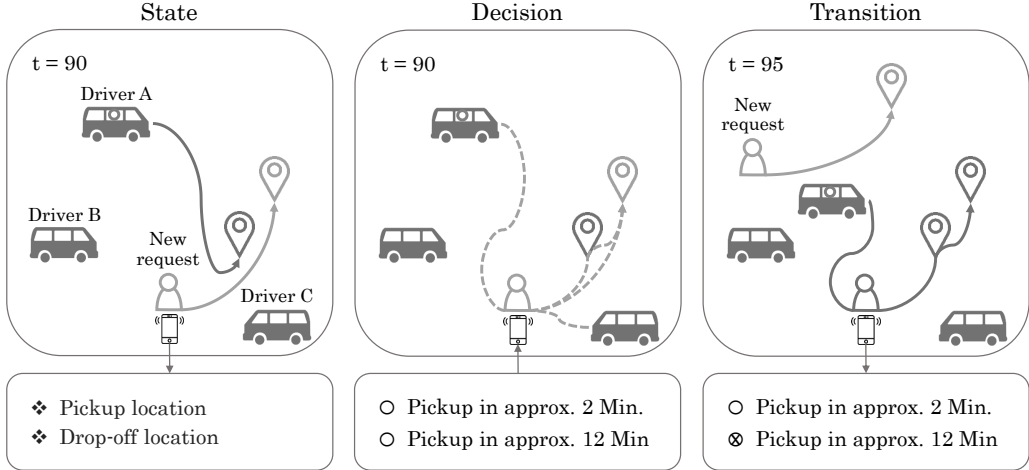


Figure 5.1: Example for a State, Decision, Stochastic Information, and Transition

In this example, three ride-sharing vehicles are available in the service area. While vehicle *A* is currently transporting a traveler to their drop-off location, vehicles *B* and *C* are waiting idle. The decision of the ride-sharing operator consists in determining and selecting feasible assignments, based on which pickup times are offered to the traveler. The service operator in this example decides to select an assignment to the vehicles *A* and *C*, which results in offering the traveler to be picked up in approx. 2

or 12 minutes. On the one hand, this decision ensures a high acceptance probability due to the heterogeneous pickup time options. On the other hand, it aims at a resource-efficient assignment, since the assignment to vehicle A allows to consolidate transports, while the assignment to vehicle C minimizes the duration in which the request restricts the flexibility of a vehicle. The transition phase following the offer is twofold. First, it includes the traveler's decision about the offer. In the example, the traveler chooses to be picked up in about 12 minutes, which leads to an update of vehicle A 's route plan. Second, it involves another transportation request being received by the ride-sharing system, which triggers the next decision epoch.

5.3.3 Sequential decision process

In the following, the underlying sequential decision process is formally modeled by introducing preliminary notations, decision epochs, states, decisions, stochastic information, and transitions.

Preliminaries

Let a ride-sharing system offer transportation within a service area during a time horizon $T = [0, t^{\max}]$ discretized in minutes. The area is defined by a set of locations N , a set of edges E between the locations, and constant travel times \mathcal{T} on the edges. The ride-sharing system operates a fleet of vehicles V with homogeneous capacities c_v . At the beginning of the time horizon, all vehicles are idle at an initial location $M_0 = (m_{01}, \dots, m_{0v}) \in N$.

Decision epochs

A decision epoch k occurs whenever a new transportation request r_k is received from a traveler.

States

The state information S_k for a decision epoch k include:

- The current point in time t_k with $0 < t_k < t^{\max}$.
- The requested pickup location p_k and drop-off location $d_k \in N$.

- The next locations where vehicles can be re-routed M_k .
- The number of travelers currently transported by the vehicles $W_k = (w_{k1}, \dots, w_{kv})$.
- The incumbent route plans $H_k = (H_{k1}, \dots, H_{kv})$, consisting of stop lists $H_{kv} = (h_{k1}, \dots, h_{k|H_{vk}|})$. Each stop $h_k \in H_{kv}$ is comprised of location, service time window, number of travelers to be picked up or dropped off, and service time. The service time window results from the agreed pickup time and, for pickups, from a maximum delay constant ω resp., for drop-offs, from the direct travel time multiplied by a maximum transportation duration factor ϵ .

A state can be summarized as $S_k = (t_k, p_k, d_k, M_k, W_k, H_k)$. The initial state is at time $t_k = 0$ with no requesting traveler and all vehicles idling at their initial location, $S_0 = (O, -, -, M_0, W_0, H_0)$.

Decisions

A decision concerns the pickup times to be offered and the assignments to be used. To this end, we assume that each assignment a_k is associated with exactly one pickup time $t_{p_k}^{a_k}$. We further assume that the number of pickup time options, i.e., the maximum offer size, is restricted by a constant parameter ζ . A decision can thus be defined as a vector X_k of maximal ζ feasible assignments a_k . An assignment a_k is considered feasible if:

- a) The pickup stop $h_{p_k}^{a_k}$ is planned before the drop-off stop $h_{d_k}^{a_k}$ for the same vehicle route H_{vk} .
- b) The interval between request time t_k and pickup time $t_{p_k}^{a_k}$ is less than or equal to a maximum lead time ψ and the interval between pickup time $t_{p_k}^{a_k}$ and drop-off time $t_{d_k}^{a_k}$ is less than or equal to the maximum transportation duration $\mathcal{T}(p_k, d_k) \times \epsilon$.
- c) The rescheduled arrival time $t_{n_k}^{a_k}$ is within the service time window for each stop $h_k \in H_{vk}$.
- d) The vehicle capacity c_v will not be exceeded.

Stochastic information and transition

The stochastic information contains two pieces of information. First, the traveler's choice y_k regarding transportation offer X_k , resulting in the update of the corresponding vehicle route H_{kv} or a canceled request $c(S_k, X_k)$. Second, the occurrence of a new transportation request r_{k+1} leading to the next decision epoch with updated state information S_{k+1} .

Solution and objective function

The solution for the problem is a policy π making an offer decision $X^\pi(S_k)$ to every state S_k . An optimal policy π^* minimizes the expected request cancellations $c(S_k, X_k)$ when starting in state S_0 and applying policy π^* throughout the process:

$$\pi^* = \arg \min_{\pi \in \Pi} \mathbb{E} \left[\sum_{k=1}^K c(S_k, X^\pi(S_k)) | S_0 \right]. \quad (5.1)$$

5.4 Offer design

Whenever a new request is received, a transportation offer has to be made immediately. The aim of designing such offers is to maximize the probability that a pickup option will be chosen while minimizing the routing effort in favor of future demand. To achieve this, we present a four-step process summarized in Figure 5.2 that includes the identification of eligible assignments and the offer design by means of a parametric CFA.

The identification of eligible assignments is performed in the first two steps, which are accordingly dedicated to vehicle routing. Steps three and four then comprise the actual offer design by means of CFA and are thus more closely related to demand management. The process steps are detailed in the following.

5.4.1 Identification of assignments

The identification of assignments follows the concept of a well-known insertion heuristic. This heuristic evaluates all feasible assignments for a transportation request to select the best-rated one. Our process follows this approach, with the difference that we determine among the feasible assignments the best rated for each eligible pickup

time discretized in minutes. This provides a selection of assignment candidates as input for the subsequent offer design.

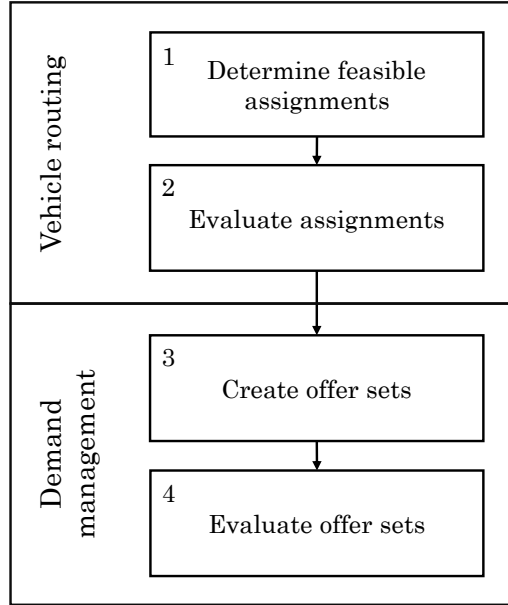


Figure 5.2: Overview of the Offer Design Process

The key to an effective selection is the evaluation of an assignment. We, therefore, first define the corresponding evaluation function and then discuss in detail the identification of assignment candidates according to Algorithm 1. The design of the evaluation function $\mathcal{E}(a_k)$ follows the insights of Ulmer et al. (2021), according to which they should balance efficiency and flexibility. Efficiency here refers to minimizing driving time by consolidating transports, whereas flexibility refers to minimizing the duration for which a vehicle is restricted by an assignment. Both aspects are reflected in the evaluation function $\mathcal{E}(a_k)$ by means of a metric: efficiency by the additional caused travel time of an assignment δ_{a_k} , and flexibility by the fulfillment duration γ_{a_k} , defined as the interval between request time t_k and planned drop-off time $t_{d_k}^{a_k}$. Finally, to determine the overall rating of an assignment $\mathcal{E}(a_k)$, we compute the parameterized sum of both metrics:

$$\mathcal{E}(a_k) = \alpha \times \gamma_{a_k} + \beta \times \delta_{a_k}. \quad (5.2)$$

In addition to the evaluation function $\mathcal{E}(a)$, the inputs to the assignment identification Algorithm 1 are the fleet of vehicles V and the current state S_K . The output is a list of assignments A_k as candidates for the succeeding offer design.

```

input : Vehicle  $V$ , State  $S_K$ ,
        Assignment-evaluation-function  $\mathcal{E}(a_k)$ 
output: List of assignments  $A_k$ 
1 Function getListOfAssignments( $V, S_K, \mathcal{E}$ )
2    $A_k \leftarrow \text{Assignment}[\psi]$ 
3   for ( $v \in V$ ) do
4     for ( $i, j \in H^{kv} | i \leq j$ ) do
5        $a_k \leftarrow \text{createAssignment}(i, j, p_k, d_k, H^{kv})$ 
6       if (checkFeasibility( $a_k, m_{kv}, w_{kv}$ )) then
7         if ( $A_k[t_{p_k}^{a_k}] == \emptyset \vee \mathcal{E}(A_k[t_{p_k}^{a_k}]) > \mathcal{E}(a_k)$ ) then
8            $A_k[t_{p_k}^{a_k}] \leftarrow a_k$ 
9         end
10      end
11   end
12   end
13   return  $A_k$ 
    
```

Algorithm 1: Identification of Assignments

The algorithm works as follows. In the first line, the list of assignments A_k is initialized. As noted earlier, times are discretized in minutes so that the number of distinct pickup times, and hence the length of the list, is equal to the maximum lead time ψ . To fill the list considering all feasible assignments, the first for-loop iterates over the set of vehicles V . The second for-loop then systematically iterates over the stops of the corresponding vehicle route H^{kv} to check all combinations of insertion positions for the current pickup and drop-off locations. For each of these combinations, a new assignment a_k is created in line 5. We assume that for each created assignment a_k the associated pickup time $t_{p_k}^{a_k}$ is the earliest feasible one. Offering additional pickup times by having vehicles wait has been considered in preliminary experiments, but without yielding any advantages (see Appendix C.1 for the analysis). The feasibility of a created assignment a_k with respect to all previously defined time and capacity-related constraints is then checked in the if-clause (line 6). For a feasible assignment, it is next evaluated if no assignment with the same pickup time $t_{p_k}^{a_k}$ has been found yet or if the evaluation function $\mathcal{E}(a_k)$ constitutes an improvement. If either is true, the assignment becomes part of the candidate list A_k in line 8. Finally, in the last line, the list of best-evaluated assignments A_k is returned.

5.4.2 Cost function approximation

The design of multi-optional pickup offers requires a decision on the assignments to be used, taking into account their associated pickup times. Such a decision should ensure an acceptable offer for the currently requesting traveler and the effective utilization of the ride-sharing fleet in favor of future ones. To reflect this trade-off, we propose a parametric CFA following Powell (2022) that allows a deterministic decision objective to be manipulated to enable anticipatory decision-making. With respect to the problem at hand, the deterministic objective is to avoid a cancelled request. We model this objective in the CFA through the probability that a traveler chooses a pickup option and thus accepts the offer. Moreover, we extend this deterministic objective towards anticipatory decision-making by adding the approximate routing effort associated with an offer set. The tuning of an accompanying parameterization allows the offer design to be balanced between myopic maximization of acceptance probabilities and anticipatory minimization of routing effort.

The respective offer decision X_k is formally defined in Equation 5.3 given the set of assignments A_k and the balancing parameter θ . It states that the final pickup time offer X_k corresponds to the offer $O_k \in \mathcal{O}_k$ that maximizes the balanced sum of the acceptance probability $\mathcal{P}(O_k)$ and the routing effort approximated by the function $\bar{\mathcal{R}}(O_k)$.

$$X_k(A_k|\theta) = \arg \max_{O_k \in \mathcal{O}_k} \mathcal{P}(O_k) - \theta \bar{\mathcal{R}}(O_k). \quad (5.3)$$

To implement the equation, \mathcal{P} provides, for each combination of pickup time, the empirically observed acceptance probability. We assume that corresponding probabilities can be pre-computed, for example, by means of historical booking data (for the determination in the computational experiments, see Section 5.5.2). The function $\bar{\mathcal{R}}$, in turn, approximates the routing effort associated with an offer O_k by computing the weighted sum of the evaluation function $E(a_k)$ over all assignments $a_k \in O_k$. The weighting reflects the empirically observed probability that a pickup time $t_{p_k}^{a_k}$ will be chosen given the offer set O_k . Finally, the approximated routing effort is normalized to obtain a scaling comparable to the acceptance probability by using min-max values observed in preliminary experiments.

```

input : Assignments  $A_k$ , Balancing-parameter  $\theta$ ,
        Acceptance-probabilities  $\mathcal{P}(O_k)$ ,
        Routing-effort-approximation-function  $\bar{\mathcal{R}}(O_k)$ 
output: Offer set  $X_k$ 
1 Function getOffer( $A_k, \theta, \mathcal{P}, \bar{\mathcal{R}}$ )
2    $X_k \leftarrow \emptyset$ 
3    $O_k \leftarrow \text{powerset}(A_k)$ 
4   for ( $O_k \in O_k$ ) do
5     if ( $|O_k| \leq \zeta$ ) then
6       if ( $X_k == \emptyset$  ||
7          $\mathcal{P}(O_k) - \theta \bar{\mathcal{R}}(O_k) > \mathcal{P}(X_k) - \theta \bar{\mathcal{R}}(X_k)$ ) then
8          $X_k \leftarrow O_k$ 
9       end
10    end
11  end
12  return  $X_k$ 

```

Algorithm 2: Offer Design via CFA

Given the acceptance probabilities \mathcal{P} and the routing effort approximation function $\bar{\mathcal{R}}$, we further detail the offer design implementation in Algorithm 2 with the assignments A_k and the balancing parameter θ as additional inputs. Based on those inputs, the algorithm returns the offer set X_k provided to the traveler. In the first line of the algorithm, the offer X_k is initialized as an empty set. Next, based on the given set of assignments A_k , the set O_k is created, which contains all potential offer sets. Afterward, in lines 4-11, the offer decision is made by iterating over all offer sets $O_k \in O_k$. The first if-clause states that only offers O_k that include a maximum of ζ assignments, i.e., pickup options, will be considered. Note that this only restricts the maximum offer size, whereas offers with fewer options are feasible. In the following loop, an offer O_k is accepted whenever the incumbent offer X_k equals the empty set or is outperformed with respect to the sum of the acceptance probability $\mathcal{P}(O_k)$ and the θ weighted routing effort $\bar{\mathcal{R}}(O_k)$. The first condition ensures that, if feasible, at least one pickup option is provided to avoid the non-service case. The second reflects the offer decision according to Equation 5.3. Finally, in line 9, the incumbent offer set X_k is returned.

5.5 Experimental setup

In the computational experiments, we aim to evaluate the performance of the proposed multi-optional offer design by analyzing the capabilities of the CFA and comparing its effectiveness with alternative offer concepts. To evaluate the performance,

we consider the objective function value (i.e., the cancellation rate), as well as metrics that reflect the routing efficiency from a service operator’s perspective and the offer quality from the perspective of the transported travelers. With respect to routing efficiency, we analyze the average driving time per transportation request and the percentage of travelers sharing part of their transportation. Concerning the offer quality, we examine the distribution of deviations from agreed and preferred pickup time. As a basis for these analyses, in the following, we introduce the design of instances, the traveler’s choice modeling, and the considered benchmark policies.

5.5.1 Instances

The instances were generated in analogy to Haferkamp et al. (2023). The characteristics of the baseline scenario with regard to the defined service area, transportation demand, and ride-sharing fleet are presented in the following. Based on this baseline scenario, several sensitivity analyses were conducted to investigate the robustness of the results (see Appendix C.3 for details).

Service area

We consider a ride-sharing system operating in Brooklyn New York City (NYC). Brooklyn is a relatively large urban borough with a high-demand area slightly north of the center and low-demand areas on the periphery, resembling a prototypical city. For the simulations, 3000 unique locations in Brooklyn are taken from the January 2014 NYC taxi trip data (NYC Taxi and Limousine Commission, n.d.). Free-flow travel times between these locations are derived from OpenStreetMap and multiplied by two to mimic high traffic immanent in this area.

Demand creation

The planning period covers an 8-hour afternoon shift from 14:00 to 22:00. A total of 6400 incoming trip requests is processed per simulation run. Each request involves the transportation of one traveler. To obtain a realistic temporal and spatial distribution of demand, for each simulation run, samples are drawn from a pool of around 100,000 trips performed by Uber or Lyft within Brooklyn in September 2019. For a sampled trip, the reported pickup time is interpreted as the time of the request.

Furthermore, the exact locations are randomly selected according to the reported pickup and drop-off zone, subject to a direct travel duration of at least five minutes.

System parameterization

The parameterization of the ride-sharing system comprises the maximum lead time ψ , the maximum offer size ζ , the maximum pickup delay ω , and the maximum transportation duration defined by the direct travel time $\mathcal{T}(p_k, d_k)$ times ϵ .

The maximum lead time ψ is assumed to be 30 minutes. This is long enough to ensure that more than 99% of the requesting travelers receive a pickup offer, while clearly excluding long-term reservations. Based on the pickup time horizon, the maximum offer size ζ has been set to 3. This allows for both sufficient coverage and concise display in a mobile application (see Section 5.6.2 for an analysis).

The service quality is further defined by the maximum pickup time deviation ω and the maximum transportation duration factor ϵ . Regarding the deviation from the agreed pickup time, we assume a maximum of $\omega = 1$ minute. Thus, an additional stop can only be inserted before a planned pickup if located on the current shortest path. Finally, for the maximum transportation duration, $\epsilon = 1.5$ is considered.

Ride-sharing fleet

The ride-sharing system is assumed to operate a fleet of 200 homogeneous vehicles, with each four traveler seats. At the beginning of the planning period, the vehicles are randomly distributed among the service area. With respect to vehicle routing, it is assumed that vehicles are able to divert from their next stop to serve a new assignment. To this end, in the experiments, out of the 3000 locations considered, those located on a traveled shortest path are considered as possible deviation points.

5.5.2 Travelers choice modeling

In the following, we describe the modeling of the utility functions for travelers choice on an offer decision X_k , their characteristics assumed in the baseline scenario, and the determination of the travelers' choice probabilities as input for the proposed CFA.

Utility function modeling

We assume that a traveler makes a choice upon receiving an offer X_k , based on a utility function U_k . The function $U_k(t_{p_k})$ defines for the requesting traveler the utility for each expected pickup time $t_{p_k} \in \mathcal{T}$. For the experiments, we keep the modeling of the utility functions straightforward by making the following assumptions. First, we assume that each traveler has exactly one preferred pickup time, which they would state if they could freely choose. Following this assumption, we secondly assume that the utility continuously decreases with increasing deviation from the preferred pickup time until it equals zero. These assumptions yield the following properties of the utility functions:

- (1) Each traveler has one preferred pickup time $t_{p_k}^*$, with $U_k(t_{p_k}^*) > U_k(t_{p_k})$ for all pickup times $t_{p_k} \in \mathcal{T} \setminus \{t_{p_k}^*\}$.
- (2) With increasing preference deviation $|t_{p_k} - t_{p_k}^*|$, the utility $U_k(t_{p_k})$ decreases continuously.
- (3) All pickup times with a positive utility $U_k(t_{p_k}) > 0$ are within a time window $[t_{p_k}^{min}, t_{p_k}^{max}]$.

Given this modeling of the utility functions, determining the pickup time window $[t_{p_k}^{min}, t_{p_k}^{max}]$ as well as the preferred pickup time $t_{p_k}^*$ enables the complete definition of the travelers' pickup time requirements.

Assumptions on traveler requirements

The distributions of traveler pickup time requirements assumed in the baseline scenario are presented in Figure 5.3. The figure shows the proportion of the requests for the minute interval between (a) request time t_k and earliest pickup time $t_{p_k}^{min}$, (b) earliest pickup time $t_{p_k}^{min}$ and preferred pickup time $t_{p_k}^*$, and (c) earliest pickup time $t_{p_k}^{min}$ and latest pickup time $t_{p_k}^{max}$.

Graph (a) indicates that most travelers are assumed to be available for pickup within 10 minutes after requesting transportation, which corresponds to the on-demand nature of the considered ride-sharing system. Graph (b) illustrates that the proportion of requests decreases as the time interval between earliest availability $t_{p_k}^{min}$ and preferred pickup time $t_{p_k}^*$ increases, which is consistent with the prevailing

assumption in the literature that travelers want to be picked up as soon as they are available. Finally, Graph (c) shows that the length of the pickup time window $[t_{p_k}^{min}, t_{p_k}^{max}]$ is 15 minutes on average. Thus, it is assumed that the majority of travelers are fairly flexible in terms of their pickup time, which is plausible since ride-sharing focuses rather on affordable transportation through consolidation than on an individually tailored service.

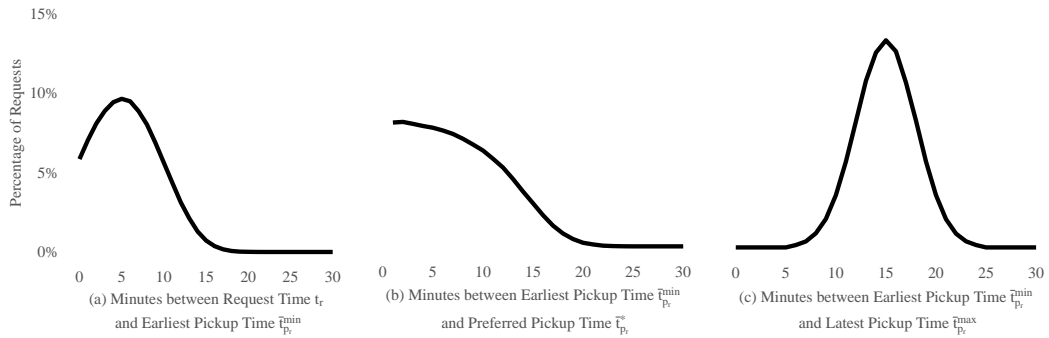


Figure 5.3: Characteristics of Traveler Requirements

Traveler choice probabilities

The traveler choice probabilities are assumed to be pre-computed as inputs to the CFA. This refers to both the probability that an offer contains an acceptable pickup time option and the probability that an option will be chosen given a certain set of pickup time options. To determine these probabilities for the computational experiments, 100 demand scenarios have been evaluated in advance. For each transportation request included in these scenarios, all eligible sets of pickup times were evaluated to determine if any option would have been chosen and which one so that the average choice probabilities could be obtained.

5.5.3 Benchmarks

In the computational experiments, we consider three benchmark policies that differ in the level of information on travelers' utility function U_k . However, for all three policies, the considered assignments A_k are determined according to Section 5.4.1.

For the first policy, called no information (*NI*), it is assumed that not even probabilistic information on travelers' utility function U_k is available. Therefore, to max-

imize the acceptance probability, offer sets O_k are designed to systematically cover the pickup time horizon. This means for the baseline scenario, with a max offer size $\zeta = 3$ and an offer horizon $\psi = 30$ minutes, that the three assignments $a_k \in A_k$ are selected for which the gap between request time t_k and planned pickup time $t_{p_k}^{a_k}$ is closest to 7.5, 15, and 22.5 minutes, respectively.

For the second policy, called preferred pickup information (*PPI*), the traveler is assumed to communicate their favorable pickup time $t_{p_k}^*$. The policy attempts to accommodate these preferences by selecting the assignment $a_k \in A_k$ that minimizes the time gap between the offered pickup time $t_{p_k}^{a_k}$ and the preferred pickup time $t_{p_k}^*$. Consequently, this policy is in a myopic sense traveler-oriented, as routing effort is neglected to match the preference of the current requesting traveler.

The last policy, perfect information *PI*, is a theoretical benchmark as it assumes complete information on travelers utility function U_k . The policy aims at minimizing cancellations by always selecting the most routing-efficient evaluated assignment associated with an acceptable pickup time, i.e., the feasible assignment $a_k \in A_k$ with $t_{p_k}^{min} \leq t_{p_k}^{a_k} \leq t_{p_k}^{max}$ that minimizes evaluation function $\mathcal{E}(a_k)$.

5.6 Computational results

In presenting the computational results, we first focus on parameter tuning and then analyze the performance of the proposed *CFA* against the three benchmark policies.

5.6.1 Parameter tuning

In the following section, we focus on the parameters crucial for the performance of the *CFA*: the balancing parameter θ and the maximum offer size ζ . The waiting strategy and the assignment evaluation parameters α and β are treated in Appendix C.1 and C.2, since both have only a marginal impact on the performance of the *CFA* and the benchmark policies.

Balancing parameter

Tuning the θ parameter is crucial for the well-balanced offer design of the *CFA*. A too-low value for θ results in maximizing acceptance probabilities, with initially very convenient offers at the expense of rapid congestion of the ride-sharing fleet.

Conversely, a too-high value for θ results in minimizing the approximated routing effort and thus in an excessive number of cancellations due to unacceptable pickup time offers.

To find a well-balanced θ , we analyze values between 0 and 4.0 in increments of 0.25. The results in Figure 5.4(a) show the θ -value on the x-axis and on the y-axis as well as the corresponding average cancellation rates over 100 baseline instances. The graph shows that between $\theta = 0$ and $\theta = 0.75$, the average cancellation rate decreases significantly from 9.6% to 7.2% and then increases continuously until it reaches a cancellation rate of 9.2% at $\theta = 4.0$. The development of the curve illustrates the importance of the balance parameter θ on the performance of the *CFA*, with $\theta = 0.75$ providing the best trade-off with respect to minimizing cancellations.

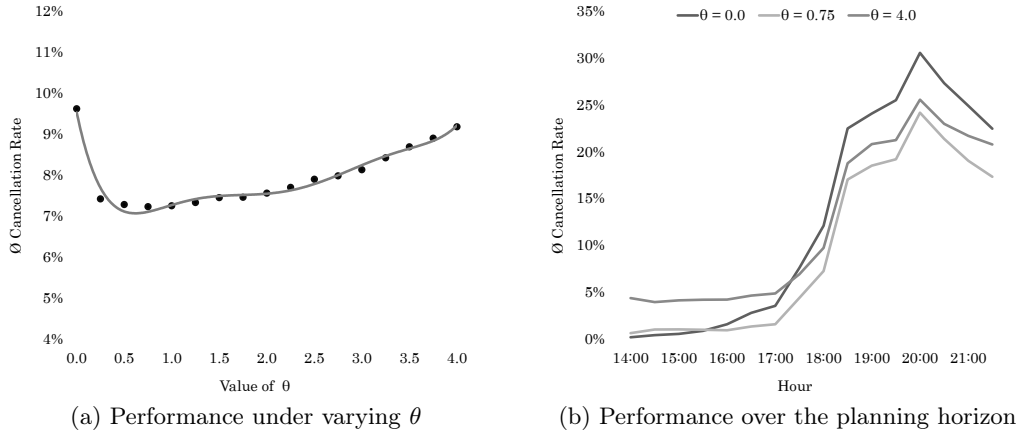


Figure 5.4: Tuning of the Balancing Parameter

In addition to the overall cancellation rate, we analyze whether the best value of θ varies as a function of the demand volume. Accordingly, Figure 5.4(b) illustrates the cancellation rate over the planning horizon for the minimum, maximum, and best evaluated θ value.

All three curves clearly show that cancellations, and hence demand, are comparatively low at the beginning of the planning horizon and have a peak in the second half. While the three curves are structurally similar, it is noticeable that at $\theta = 0$ and $\theta = 0.75$ the cancellation rates are comparably low at the beginning of the planning horizon. However, as the planning horizon progresses, they increase less sharply

at $\theta = 0.75$ and $\theta = 4$. This result indicates that a well-chosen balancing parameter ensures both acceptable offerings during low-demand periods and the effective utilization of the fleet to minimize cancellations during high-demand periods.

In summary, balancing acceptance probabilities and routing effort with a fixed parameter significantly improves the performance of a ride-sharing system even under temporally varying demand volume.

Offer size

In the following, we analyze how the restriction of the maximal offer size affects the performance of the *CFA*. For this purpose, again 100 baseline instances with a maximum offer size ζ of 1, 2, and 3 have been solved. The results are shown in Figure 5.5(a) in terms of average cancellation rates and in Figure 5.5(b) for $\zeta = 3$ in terms of average realized offer sizes across the sensitivity analyses presented in Appendix C.3.

Focusing first on the average cancellation rates presented in Figure 5.5(a), it can be observed that an increase of the maximum offer size from one to two options reduces the cancellation rate by 5.1%, while a further increase to three options only reduces it by 0.4%. This result indicates that three pickup time options are sufficient to cover the considered pickup time horizon.

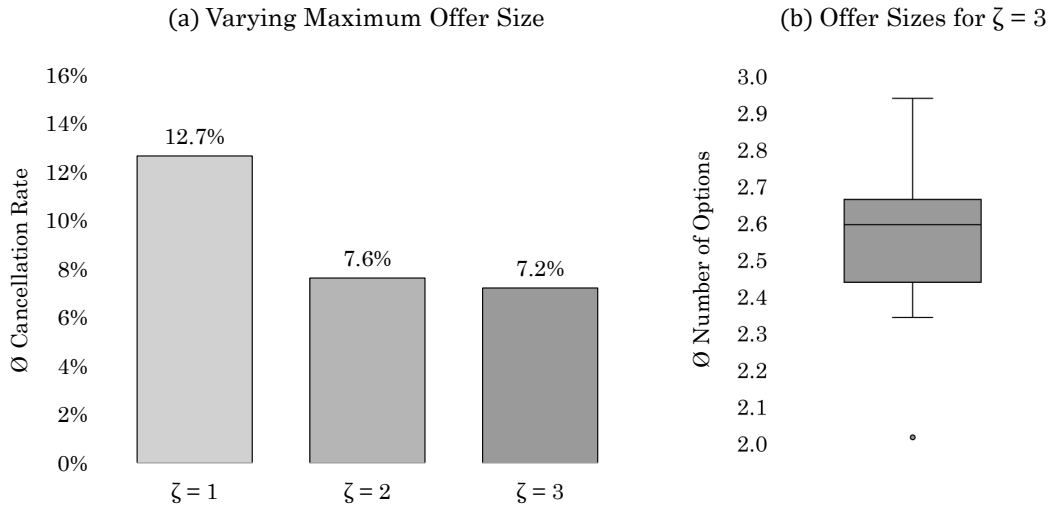


Figure 5.5: Tuning of the Maximum Offer Size ζ

Turning to the average realized offer sizes for $\zeta = 3$, shown as a boxplot in Figure 5.5(b), it becomes apparent that the average provided number of options varies greatly depending on the scenario. For example, the outlier with only two pickup time options on average belongs to a scenario assuming more flexible travelers. The offer design via the *CFA* thus ensures that fewer options than feasible are offered when advisable. However, apart from the offer design, the limited availability of alternative assignments may also contribute to a lower average offer size.

In summary, a maximum of three options is sufficient to cover the considered pickup time horizon, with the offered number of options often being lower in favor of better-rated offers or due to a lack of assignment alternatives.

5.6.2 Performance evaluation

In the following, the performance of the proposed *CFA* is evaluated against the benchmark policies using 100 newly generated instances of the baseline scenario. The additionally conducted sensitivity analyses yielded structurally comparable results and are therefore reported in Appendix C.3.

Regarding the evaluation metrics introduced in Section 5.5, we first analyze the performance in terms of the objective function value, i.e., the cancellation rate. Secondly, we focus on the routing efficiency by analysing the average driving time per transportation request and the average percentage of shared transports. Finally, we consider the offer quality via the distributions of the deviation of agreed and preferred pickup time.

Analysis of the objective function values

The objective function value is represented by the average cancellation rate, which indicates how frequent travelers received an inconvenient respectively infeasible pickup time offer. The corresponding results for the four considered policies are shown in Figure 5.6(a) for the baseline scenario as well as in Figure 5.6(b) across the sensitivity analyses presented in Appendix C.3.

Focusing first on the results of the baseline scenario, the average cancellation rate is lowest for *PI* at 5.7%. For *CFA*, it increases by only 1.5%, while for *PPI* and *NI* by 5.4% and 4.2%, respectively. A similar pattern can be observed in Figure 5.6(b) with respect to the median cancellation rates across the sensitivity analyses. Fur-

thermore, the interquartile ranges and the whiskers indicate a lower variance in the average cancellation rate for *PI* and *CFA* compared to *PPI* and *NI*. These results demonstrate that *PI* and *CFA* outperform *PPI* and *NI* with respect to both the level of cancellations and the robustness to different scenarios.

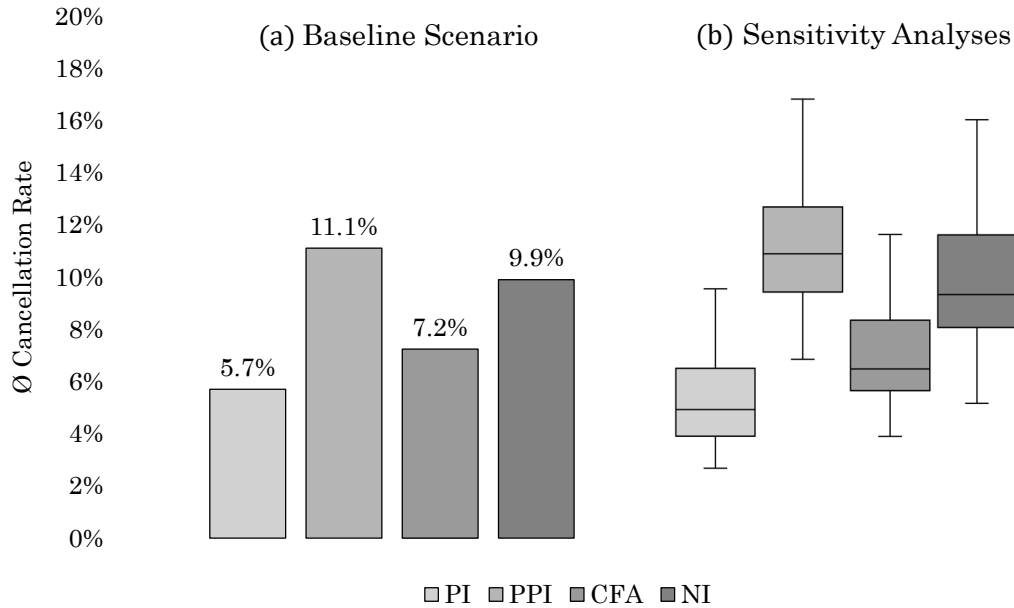


Figure 5.6: Comparison of Objective Function Values

Summarized, the results indicate that the *CFA* performs close to the theoretical benchmark *PI*. Moreover, it significantly outperforms both *PPI*'s single-option offers based on preferred pickup time and *NI*'s systematic multi-optional offer design. The results thus highlight the value of state-dependent multi-optional pickup time offers.

Analysis of the vehicle routing efficiency

In the following, we analyze the average driving time per transportation request and the average percentage of shared transports to evaluate how efficiently the vehicle resources are utilized by the four policies.

For the average driving time per transportation request shown in Figure 5.7(a), the *PI* again performs best with an average of 12.3 minutes per transport. For *CFA*,

the value increases by only 0.5 minutes, while for *PPI* and *NI* it increases by 1.4 and 2.5 minutes, respectively. This implies that the lower cancellation rates for *PI* and *CFA* are associated with a more time-efficient fulfillment of transportation requests.

Turning to the average percentage of shared transports, shown in Figure 5.7(b), we can notice that the rate for *CFA* is the highest being improved by 2% compared to *PI* as well as by 3.3% and 4.5% compared to *PPI* and *NI*. The *CFA* thus exploits given consolidation opportunities in comparison more extensively.

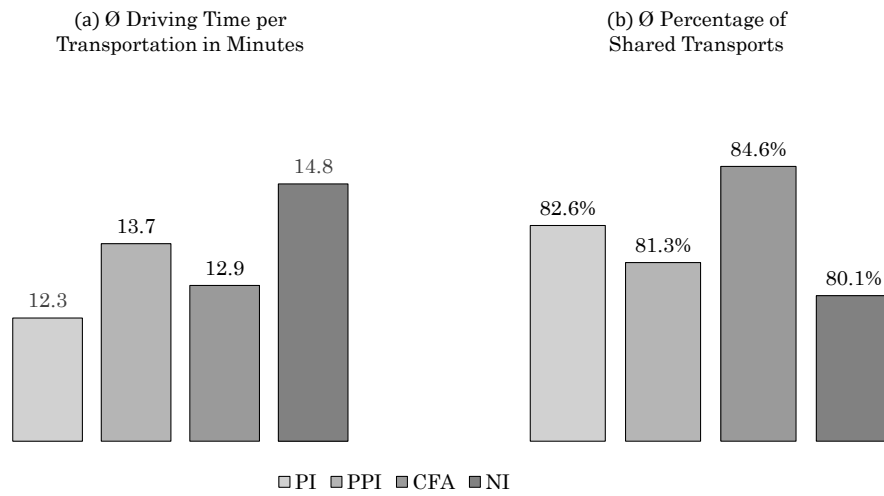


Figure 5.7: Comparison of the Routing Efficiency

In summary, the proposed *CFA* ensures a time-efficient vehicle routing by the successful consolidation of transportation requests. In contrast, *PPI* and *NI* perform significantly worse on both routing efficiency metrics. This indicates that both offering preferred pickup times and offering three pickup time options systematically is very resource-intensive and results in inefficient utilization of the ride-sharing fleet.

Analysis of offer quality

Finally, we analyze to what extent preferred pickup times are met as an indicator of the offer quality from the perspective of transported travelers. To this end, Figure 5.8 shows the distribution of the deviation between agreed and preferred pickup time in minutes for the four policies.

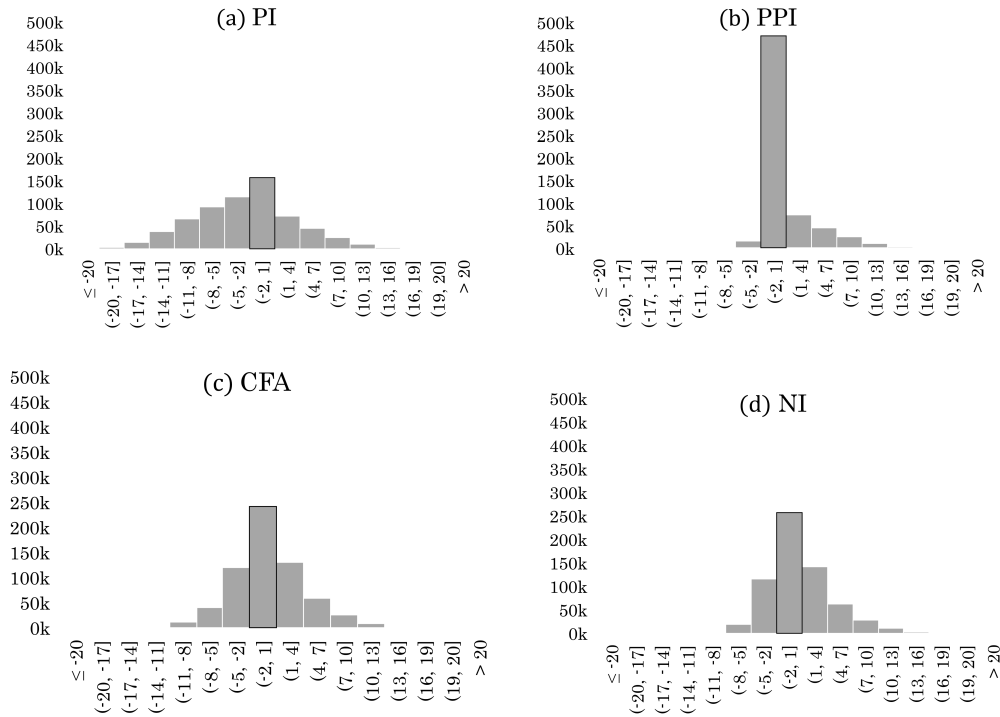


Figure 5.8: Comparison of the Preferred Pickup Time Deviation per Transported Traveler

First, it is remarkable that for *PPI*, the overwhelming majority of the travelers have been offered approx. their preferred pickup time (± 1 minute). In contrast, the distributions of *CFA* and *NI* resemble a normal distribution with a deviation of up to ± 10 minute at the tails. Lastly, for *PI*, the distribution is again considerably flatter, with the majority of pickups occurring before the preferred time, with a deviation of up to 17 minutes.

It can be concluded that accommodating preferred pickup times, as in *PPI*, results in travelers receiving either a very convenient pickup time offer or an infeasible one. In contrast, exploiting information on travelers' individual pickup time requirements, as in *PI*, allows for minimizing cancellations at the expense of more inconvenient offerings. The proposed *CFA* offers a trade-off by accommodating travelers' preferred pickup time better than *PI* while offering more travelers an acceptable pickup time than *PPI*.

5.7 Final remarks

In this paper, we have introduced the new stochastic and dynamic problem of designing multi-optional pickup time offers in ride-sharing systems. We have proposed a CFA to balance offer acceptance probability and the associated routing effort. In comprehensive computational experiments, we have shown that the CFA provides a powerful trade-off between minimizing canceled requests and accommodating travelers preferred pickup time. The results further highlight the advantage over a simplistic offer design that systematically covers the pickup time horizon. Moreover, we have found that exploiting information about travelers' individual pickup time requirements either allows for further minimization of the cancellation rate at the expense of less convenient pickup times or the offering of more convenient pickup times at the expense of more cancellations in the long run.

In this work, we have addressed the trade-off between acceptance probability and routing effort and presented an effective method to find their balance. To this end, we have relied on a straightforward choice model based on utility functions. Future work may derive more sophisticated discrete choice models from real data, perhaps considering heterogeneous classes of travelers.

Furthermore, we have focused on pickup times as a decisive feature of transportation offers. Future work might extend the offer design through dynamic pricing to nudge travelers' choice toward resource-efficient options. Alternatively, the offer design could be enhanced in complexity by additional non-monetary features, such as different service modes, vehicle types, and long-term reservations.

Finally, in our experiments, we shown the effectiveness of multi-optional pickup time offers based on probabilistic information about travelers' pickup time requirements compared to benchmarks that exploit different levels of information. Future research could build on our work by comparing the CFA with other sophisticated approaches that originate, for example, in the fields of machine learning or assortment optimization.

Chapter 6

Conclusion

In this thesis, we have developed non-monetary approaches for demand management and vehicle routing in dynamic ride-sharing systems that involve travelers or drivers in the decision-making processes of demand management and vehicle routing. In the following, we summarize the contributions of the chapters and conclude with an outlook on potential future work.

6.1 Summary

We summarize the four main chapters and highlight their contribution to the three research questions defined in the introduction.

Chapter 2 focuses on the efficient integration of myopic feasibility checks and re-optimization by means of a large neighborhood search. The computational results show that the proposed approach is able to improve the level of service in comparison to a travel-time minimizing insertion heuristic, even with very limited computational time. However, the results also reveal that these improvements decrease as the service quality requirements increase. Moreover, re-optimization has the disadvantage that travelers and drivers cannot receive reliable information about expected pickup and drop-off times or the upcoming route plan.

With respect to *RQ1*, the presented approach provides a first non-monetary means for demand management and vehicle routing in dynamic ride-sharing systems. While it shows some merits with respect to an efficient improvement of the system performance, the disadvantage of myopic decisions in general and regarding re-optimization in particular highlight the need for more advanced approaches.

Chapter 3 provides a corresponding classification of the related literature and a comprehensive computational study to analyze the opportunities and implications of advancing either demand and/or fulfillment control. The computational results demonstrate that advanced demand control through selective request acceptance affects ride-sharing systems quite differently than advanced fulfillment control through

anticipating future transportation requests. With advanced demand control, the opportunities to reduce request cancellations increase with an increase in demand surplus. In other words, for selective acceptance to be effective, the number of rejections must be sufficiently high. In this respect, area-wide improvements come at the detriment of travelers in peripheral areas, where the availability of service decreases drastically. In contrast, for advanced fulfillment control, increasing opportunities have been observed with decreasing demand surplus. It is thus particularly effective when demand can be largely satisfied. The potential improvements here come at the cost of longer detours for travelers, i.e., along with a more comprehensive consolidation. The combination of both advancements yields a considerably improved service level regardless of the demand surplus, however, the disadvantages for travelers are likewise accumulating.

With respect to *RQ1*, the literature classification highlights the variety in which non-monetary demand management and vehicle routing can be integrated. Considering *RQ2*, the results of the computational study indicate that opportunities for operators and the implications for travelers differ considerably between demand and fulfillment control. In particular, it can be concluded that demand management through selective acceptance leads to a systematic disadvantage of travelers in certain areas, while it cannot be effectively applied to satisfy a very high percentage of demand. In contrast, corresponding opportunities have been identified for advanced vehicle routing, associated with an improved consolidation of transportation requests.

Chapter 4 builds on the opportunities identified for advanced vehicle routing in Chapter 3 by proposing heatmap-based decision support for repositioning in decentralized ride-sharing systems. The heatmaps indicate repositioning opportunities for individual drivers by combining adaptively learned information about the expected demand with state-dependent information about the distribution of competing idle drivers and the location of the requesting driver. In a comprehensive computational evaluation, the merits of this approach have been demonstrated. In a scenario with compliant drivers, it was first shown that heatmaps can help minimize cancellations and ensure a more reliable service availability throughout the service area while reducing variability in driver income. Furthermore, several scenarios with non-

compliant drivers were investigated, showing potential for and against deviating from heatmap recommendations.

With respect to *RQ1*, the carefully designed heatmaps provide a non-monetary approach to balance supply and demand in decentralized ride-sharing systems. In terms of *RQ2*, the computational results demonstrate that the repositioning heatmaps can be beneficial to operators, travelers, and drivers. Finally, with regard to *RQ3*, the results suggest that engaging drivers by providing them with an overview of repositioning opportunities could in particular benefit inexperienced drivers and could thus help to keep them operating for the ride-sharing system.

Lastly, Chapter 5 focuses on an alternative to the selective acceptance considered in Chapter 3 by proposing a non-monetary approach to demand management in which travelers are offered multiple pickup options to choose from. To design these offer sets, a parametric cost function approximation has been proposed that balances acceptance probability with the approximated routing effort. In an extensive computational study, the advantages of this approach were demonstrated in comparison with benchmarks that exploit different levels of information about travelers' pickup time requirements. The results indicate that the proposed offer design provides a favorable balance between minimizing cancellations and meeting travelers' pickup time preferences.

Regarding *RQ1*, multi-optional pickup offers are proposed as a soft approach to effectively managing demand in dynamic ride-sharing systems. With respect to *RQ2*, it has been shown that those offers are able to improve both system performance and the service received by travelers. As to *RQ3*, it highlights the potential benefits of involving travelers in the decision-making process by offering a well-selected set of options.

6.2 Outlook

In this thesis, we have shown how non-monetary approaches to advanced demand management and vehicle routing can help improve both the performance of dynamic ride-sharing systems and the service/working conditions experienced by travelers/drivers. The developed approaches thus present alternatives to the related literature that relies on centralized decision-making ignoring the interests of travel-

ers and drivers. Based on our findings, three primary areas for potential future work can be identified.

First, in this thesis, we have modeled traveler and driver decisions based on some reasonable assumptions. Future work may develop more sophisticated choice models based on real-world data. Such models, in turn, would create new opportunities for the design of the decision policies. In this context, it would be interesting to investigate to what extent the consideration of the circumstances of a request together with the corresponding choice probabilities allows for individually tailored decisions. This would in turn raise the question of whether such an advancement would be used to improve the performance of the ride-sharing system and/or to better meet the individual preferences of the requesting traveler/driver.

Second, for the sake of clarity, we have restricted ourselves to constant travel times, a homogeneous fleet, and transportation requests that differ only in terms of request time and trip. However, future work may further increase the complexity by considering additional features to better reflect the real-world requirements and the diverse mobility needs. This could include, for example, stochastic travel times, multiple vehicle types with, for example, different numbers of seats or spaces for wheelchairs as well as more diverse transportation requests with respect to vehicle or transportation requirements. However, given the diverse potential extensions, future work may also involve analyzing which features actually change the decision problems to the extent that new innovative solution approaches are required.

Last, like most of the related literature, we focused on the management of ride-sharing systems facing peak demand in dense urban areas. In contrast, future work might consider reducing the reliance on private cars in rural areas. The recent funding of mobility projects with a focus on rural areas by the German Federal Ministry of Digital and Transport has shown that there is still a great need for corresponding solutions (BMDV, 2022). The emphasis in this research might be less on very resource-intensive on-demand systems, but rather on increasing flexibility in public transport through sophisticated reservation systems, for example. Such systems would in turn require non-monetary approaches that integrate demand management and vehicle routing with the aim of providing passengers with reliable and more flexible public transport in a resource-efficient manner.

Bibliography

- Abrams, Z. (2022). *Uber and Lyft customers hate surge pricing — but here’s why it won’t go away*. <https://www.businessofbusiness.com/articles/uber-and-lyft-customers-hate-surge-pricing-but-heres-why-it-wont-go-away/>. Accessed 26-January-2023.
- Alnaggar, A. (2021). *Optimization under uncertainty for e-retail distribution: From suppliers to the last mile*. <http://hdl.handle.net/10012/17336>. Accessed 31-August-2023.
- Alonso-Mora, J., S. Samaranayake, A. Wallar, E. Frazzoli & D. Rus (2017a). “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment”. In: *Proceedings of the National Academy of Sciences of the United States of America* 114.3, pp. 462–467.
- Alonso-Mora, J., A. Wallar & D. Rus (2017b). “Predictive routing for autonomous mobility-on-demand systems with ride-sharing”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3583–3590.
- Ashkrof, P., G. H. d. A. Correia, O. Cats & B. van Arem (2020). “Understanding ride-sourcing drivers’ behaviour and preferences: Insights from focus groups analysis”. In: *Research in Transportation Business & Management* 37, p. 100516.
- Attanasio, A., J.-F. Cordeau, G. Ghiani & G. Laporte (2004). “Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem”. In: *Parallel Computing* 30.3, pp. 377–387.
- Ausseil, R., J. A. Pazour & M. W. Ulmer (2022). “Supplier menus for dynamic matching in peer-to-peer transportation platforms”. In: *Transportation Science* 56.5, pp. 1304–1326.
- Beaudry, A., G. Laporte, T. Melo & S. Nickel (2010). “Dynamic transportation of patients in hospitals”. In: *OR Spectrum* 32.1, pp. 77–107.
- Bent, R. W. & P. van Hentenryck (2004). “Scenario-based planning for partially dynamic vehicle routing with stochastic customers”. In: *Operations Research* 52.6, pp. 977–987.
- Berbeglia, G., J.-F. Cordeau & G. Laporte (2010). “Dynamic pickup and delivery problems”. In: *European Journal of Operational Research* 202.1, pp. 8–15.

- Berbeglia, G., J.-F. Cordeau & G. Laporte (2012). “A hybrid tabu search and constraint programming algorithm for the Dynamic dial-a-ride problem”. In: *INFORMS Journal on Computing* 24.3, pp. 343–355.
- Berbeglia, G., G. Pesant & L.-M. Rousseau (2011). “Checking the feasibility of dial-a-ride instances using constraint programming”. In: *Transportation Science* 45.3, pp. 399–412.
- BerlKönig (n.d.). *BerlKönig reigns here*. <https://www.berlkoenig.de/en/>. Accessed 21-October-2019.
- Bertini, M. & O. Koenigsberg (2021). *The pitfalls of pricing algorithms*. <https://hbr.org/2021/09/the-pitfalls-of-pricing-algorithms>. Accessed 26-January-2023.
- Bertsimas, D., P. Jaillet & S. Martin (2019). “Online vehicle routing: The edge of optimization in large-scale applications”. In: *Operations Research* 67.1, pp. 143–162.
- Bimpikis, K., O. Candogan & D. Saban (2019). “Spatial pricing in ride-sharing networks”. In: *Operations Research* 67.3, pp. 744–769.
- Bischoff, J., M. Maciejewski & K. Nagel (2017). “City-wide shared taxis: A simulation study in Berlin”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 275–280.
- BMDV (2022). *BMDV fördert sieben Innovationen mit Schwerpunkt im ländlichen Raum*. <https://bmdv.bund.de/SharedDocs/DE/Pressemitteilungen/2023/001-wissing-modellprojekte-oepnv.html>. Accessed 28-August-2023.
- Branke, J., M. Middendorf, G. Noeth & M. Dessouky (2005). “Waiting strategies for dynamic vehicle routing”. In: *Transportation Science* 39.3, pp. 298–312.
- Braverman, A., J. G. Dai, X. Liu & L. Ying (2019). “Empty-car routing in ridesharing systems”. In: *Operations Research* 67.5, pp. 1437–1452.
- Brinkmann, J., M. W. Ulmer & D. C. Mattfeld (2019). “Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems”. In: *Computers & Operations Research* 106, pp. 260–279.
- Campbell, A. M. & M. Savelsbergh (2006). “Incentive schemes for attended home delivery services”. In: *Transportation Science* 40.3, pp. 327–341.
- Campbell, A. M. & M. W. P. Savelsbergh (2005). “Decision support for consumer direct grocery initiatives”. In: *Transportation Science* 39.3, pp. 313–327.

- Castillo, V. E., D. A. Mollenkopf, J. E. Bell & T. L. Esper (2022). “Designing technology for on-demand delivery: The effect of customer tipping on crowdsourced driver behavior and last mile performance”. In: *Journal of Operations Management* 68.5, pp. 424–453.
- Chao, I.-M., B. L. Golden & E. A. Wasil (1996). “The team orienteering problem”. In: *European Journal of Operational Research* 88.3, pp. 464–474.
- Chen, C., F. Yao, D. Mo, J. Zhu & X. Chen (2021). “Spatial-temporal pricing for ride-sourcing platform with reinforcement learning”. In: *Transportation Research Part C: Emerging Technologies* 130, p. 103272.
- Conger, K. (2021). *Prepare to pay more for Uber and Lyft rides*. <https://www.nytimes.com/article/uber-lyft-surge.html>. Accessed 26-January-2022.
- Cook, C., R. Diamond, J. V. Hall, J. A. List & P. Oyer (2021). “The gender earnings gap in the gig economy: Evidence from over a million rideshare drivers”. In: *The Review of Economic Studies* 88.5, pp. 2210–2238.
- Cordeau, J.-F. & G. Laporte (2007). “The dial-a-ride problem: models and algorithms”. In: *Annals of Operations Research* 153.1, pp. 29–46.
- Coslovich, L., R. Pesenti & W. Ukovich (2006). “A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem”. In: *European Journal of Operational Research* 175.3, pp. 1605–1615.
- Cwioro, G., P. Hungerländer, K. Maier, J. Pöcher & C. Truden (2019). “An optimization approach to the ordering phase of an attended home delivery service”. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Ed. by L.-M. Rousseau & K. Stergiou. Theoretical Computer Science and General Issues. Cham: Springer International Publishing, pp. 208–224.
- Dayarian, I. & M. Savelsbergh (2020). “Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders”. In: *Production and Operations Management* 29, pp. 2153–2174.
- Dholakia, U. M. (2015). *Everyone hates Uber’s surge pricing – Here’s how to fix it*. <https://hbr.org/2015/12/everyone-hates-ubers-surge-pricing-heres-how-to-fix-it>. Accessed 26-January-2023.
- Dial, R. B. (1995). “Autonomous dial-a-ride transit introductory overview”. In: *Transportation Research Part C: Emerging Technologies* 3.5, pp. 261–275.

- Ehmke, J. F. & A. M. Campbell (2014). “Customer acceptance mechanisms for home deliveries in metropolitan areas”. In: *European Journal of Operational Research* 233.1, pp. 193–207.
- Ermagun, A. & A. Stathopoulos (2018). “To bid or not to bid: An empirical study of the supply determinants of crowd-shipping”. In: *Transportation Research Part A: Policy and Practice* 116, pp. 468–483.
- Fleckenstein, D., R. Klein & C. Steinhardt (2023). “Recent advances in integrating demand management and vehicle routing: A methodological review”. In: *European Journal of Operational Research* 306.2, pp. 499–518.
- Gdowska, K., A. Viana & J. P. Pedroso (2018). “Stochastic last-mile delivery with crowdshipping”. In: *Transportation Research Procedia* 30, pp. 90–100.
- Geržinič, N., O. Cats, N. van Oort, S. Hoogendoorn-Lanser, M. Bierlaire & S. Hoogendoorn (2023). *An instance-based learning approach for evaluating the perception of ride-hailing waiting time variability*. <http://arxiv.org/pdf/2301.04982v1>. Accessed 12-June-2023.
- Goncharova, M. (2017). *Ride-hailing drivers are slaves to the surge*. <https://www.nytimes.com/2017/01/12/nyregion/uber-lyft-juno-ride-hailing.html>. Accessed 26-January-2023.
- GraphHopper (n.d.). *GraphHopper routing engine*. <https://github.com/graphhopper/graphhopper>. Accessed 07-April-2021.
- Greening, L. & A. Erera (2021). *Effective Heuristics for Distributing Vehicles in Free-floating Micromobility Systems*. https://www.researchgate.net/publication/344503514_Effective_Heuristics_for_Distributing_Vehicles_in_Free-floating_Micromobility_Systems. Accessed 24-January-2023.
- Guda, H. & U. Subramanian (2019). “Your Uber is arriving: Managing on-demand workers through surge pricing, forecast communication, and worker incentives”. In: *Management Science* 65.5, pp. 1995–2014.
- Haferkamp, J. & J. F. Ehmke (2022). “Effectiveness of demand and fulfillment control in dynamic fleet management of ride-sharing systems”. In: *Networks* 79.3, pp. 314–337.
- Haferkamp, J., M. W. Ulmer & J. F. Ehmke (2023). “Heatmap-based decision support for repositioning in ride-sharing systems”. In: *Transportation Science*.

- Heitmann, R.-J. O., N. Soeffker, M. W. Ulmer & D. C. Mattfeld (2023). “Combining value function approximation and multiple scenario approach for the effective management of ride-hailing services”. In: *EURO Journal on Transportation and Logistics* 12, p. 100104.
- Henao, A. & W. E. Marshall (2019). “An analysis of the individual economics of ride-hailing drivers”. In: *Transportation Research Part A: Policy and Practice* 130, pp. 440–451.
- Ho, S. C., W. Y. Szeto, Y.-H. Kuo, J. M. Leung, M. Petering & T. W. Tou (2018). “A survey of dial-a-ride problems: Literature review and recent developments”. In: *Transportation Research Part B: Methodological* 111, pp. 395–421.
- Holler, J., R. Vuorio, Z. Qin, X. Tang, Y. Jiao, T. Jin, S. Singh, C. Wang & J. Ye (2019). “Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem”. In: *19th IEEE International Conference on Data Mining*. Ed. by J. Wang, K. Shim & X. Wu. Piscataway, NJ: IEEE, pp. 1090–1095.
- Horn, M. E. (2002). “Fleet scheduling and dispatching for demand-responsive passenger services”. In: *Transportation Research Part C: Emerging Technologies* 10.1, pp. 35–63.
- Horni, A., K. Nagel & K. W. Axhausen (2016). *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press.
- Hosni, H., J. Naoum-Sawaya & H. Artail (2014). “The shared-taxi problem: Formulation and solution methods”. In: *Transportation Research Part B: Methodological* 70, pp. 303–318.
- Hu, B., M. Hu & H. Zhu (2022). “Surge pricing and two-sided temporal responses in ride hailing”. In: *Manufacturing & Service Operations Management* 24.1, pp. 91–109.
- Hyytiä, E., A. Penttinen & R. Sulonen (2012). “Non-myopic vehicle and route selection in dynamic DARP with travel time and workload objectives”. In: *Computers & Operations Research* 39.12, pp. 3021–3030.
- Ichoua, S., M. Gendreau & J.-Y. Potvin (2006). “Exploiting knowledge about future demands for real-time vehicle dispatching”. In: *Transportation Science* 40.2, pp. 211–225.
- Iglesias, R., F. Rossi, K. Wang, D. Hallac, J. Leskovec & M. Pavone (2018). “Data-driven model predictive control of autonomous mobility-on-demand systems”.

- In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6019–6025.
- Jacob, J. & R. Roet-Green (2021). “Ride solo or pool: Designing price-service menus for a ride-sharing platform”. In: *European Journal of Operational Research* 295.3, pp. 1008–1024.
- Jiao, G. & M. Ramezani (2022). “Incentivizing shared rides in e-hailing markets: Dynamic discounting”. In: *Transportation Research Part C: Emerging Technologies* 144, p. 103879.
- Jiao, Y., X. Tang, Z. Qin, S. Li, F. Zhang, H. Zhu & J. Ye (2021). “Real-world ride-hailing vehicle repositioning using deep reinforcement learning”. In: *Transportation Research Part C: Emerging Technologies* 130, p. 103289.
- Ke, J., F. Xiao, H. Yang & J. Ye (2022). “Learning to delay in ride-sourcing systems: A multi-agent deep reinforcement learning framework”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.5, pp. 2280–2292.
- Klein, V. & C. Steinhardt (2023). “Dynamic demand management and online tour planning for same-day delivery”. In: *European Journal of Operational Research* 307.2, pp. 860–886.
- Larsen, A., O. B. G. Madsen & M. M. Solomon (2004). “The a priori dynamic traveling salesman problem with time windows”. In: *Transportation Science* 38.4, pp. 459–472.
- Lei, Z., X. Qian & S. V. Ukkusuri (2020). “Efficient proactive vehicle relocation for on-demand mobility service with recurrent neural networks”. In: *Transportation Research Part C: Emerging Technologies* 117, p. 102678.
- Li, X., J. Gao, C. Wang, X. Huang & Y. Nie (2021). “Driver guidance and rebalancing in ride-hailing systems through mixture density networks and stochastic programming”. In: *2021 IEEE International Smart Cities Conference (ISC2)*, pp. 1–7.
- Liu, C., C.-X. Chen & C. Chen (2021). “META: A city-wide taxi repositioning framework based on multi-agent reinforcement learning”. In: *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–6.
- Liu, Y., P. Bansal, R. Daziano & S. Samaranayake (2019). “A framework to integrate mode choice in the design of mobility-on-demand systems”. In: *Transportation Research Part C: Emerging Technologies* 105, pp. 648–665.

- Lowalekar, M. & P. Jaillet (2019). “ZAC: A zone path construction approach for effective real-time ridesharing”. In: *Proceedings of the International Conference on Automated Planning and Scheduling* 29, pp. 528–538.
- Lund, K., O. Madsen & Rygaard J.M. (1996). *Vehicle routing problems with varying degrees of dynamism*. https://www.researchgate.net/publication/248313097_Vehicle_Routing_Problems_with_Varying_Degrees_of_Dynamism. Accessed 25-August-2023.
- Luo, X., L. Li, L. Zhao & J. Lin (2022). “Dynamic Intra-Cell Repositioning in Free-Floating Bike-Sharing Systems Using Approximate Dynamic Programming”. In: *Transportation Science* 56.4, pp. 799–826.
- Ma, H., F. Fang & D. C. Parkes (2022). “Spatio-temporal pricing for ridesharing platforms”. In: *Operations Research* 70.2, pp. 1025–1041.
- Ma, S., Y. Zheng & O. Wolfson (2013). “T-share: A large-scale dynamic taxi ridesharing service”. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, pp. 410–421.
- Mackert, J. (2019). “Choice-based dynamic time slot management in attended home delivery”. In: *Computers & Industrial Engineering* 129, pp. 333–345.
- Madsen, O. B. G., H. F. Ravn & J. M. Rygaard (1995). “A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives”. In: *Annals of Operations Research* 60.1, pp. 193–208.
- Martin, L., S. Minner, D. Poças & A. S. Schulz (2021). “The competitive pickup and delivery orienteering problem for balancing car-sharing systems”. In: *Transportation Science* 55.6, pp. 1232–1259.
- Mercedes-Benz (2019). *BerlKönig Ridesharing in Berlin*. <https://niederlassungsmagazin.mercedes-benz.de/lifestyle/berlkoenig>. Accessed 21-October-2019.
- Mitrović-Minić, S., R. Krishnamurti & G. Laporte (2004). “Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows”. In: *Transportation Research Part B: Methodological* 38.8, pp. 669–685.
- Mitrović-Minić, S. & G. Laporte (2004). “Waiting strategies for the dynamic pickup and delivery problem with time windows”. In: *Transportation Research Part B: Methodological* 38.7, pp. 635–655.

- Möhlmann, M., L. Zalmanson, O. Henfridsson & R. W. Gregory (2021). “Algorithmic management of work on online labor platforms: When matching meets control”. In: *MIS Quarterly* 45.4.
- Molenbruch, Y., K. Braekers & A. Caris (2017). “Typology and literature review for dial-a-ride problems”. In: *Annals of Operations Research* 259.1-2, pp. 295–325.
- Narayanan, S., E. Chaniotakis & C. Antoniou (2020). “Shared autonomous vehicle services: A comprehensive review”. In: *Transportation Research Part C: Emerging Technologies* 111, pp. 255–293.
- Nielsen, J. (1994). *Usability engineering*. [Nachdr.] Amsterdam: Kaufmann.
- Nourinejad, M. & M. Ramezani (2020). “Ride-Sourcing modeling and pricing in non-equilibrium two-sided markets”. In: *Transportation Research Part B: Methodological* 132, pp. 340–357.
- NYC Taxi and Limousine Commission (n.d.). *Trip record data*. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>. Accessed 07-April-2021.
- Pavone, M., S. L. Smith, E. Frazzoli & D. Rus (2012). “Robotic load balancing for mobility-on-demand systems”. In: *The International Journal of Robotics Research* 31.7, pp. 839–854.
- Pisinger, D. & S. Ropke (2010). “Large neighborhood search”. In: *Handbook of Metaheuristics*. Ed. by M. Gendreau & J.-Y. Potvin. Vol. 146. International series in operations research & management science. Boston, MA: Springer US, pp. 399–419.
- Potvin, J.-Y. & J.-M. Rousseau (1993). “A parallel route building algorithm for the vehicle routing and scheduling problem with time windows”. In: *European Journal of Operational Research* 66.3, pp. 331–340.
- Pouls, M., N. Ahuja, K. Glock & A. Meyer (2022). “Adaptive forecast-driven repositioning for dynamic ride-sharing”. In: *Annals of Operations Research*, pp. 1–34.
- Pouls, M., A. Meyer & N. Ahuja (2020). “Idle vehicle repositioning for dynamic ride-sharing”. In: *Computational Logistics*. Ed. by E. Lalla-Ruiz, M. Mes & S. Voß. Theoretical Computer Science and General Issues. Cham: Springer International Publishing and Imprint: Springer, pp. 507–521.
- Powell, W. B. (2022). *Reinforcement learning and stochastic optimization: A unified framework for sequential decisions*. Hoboken, New Jersey: Wiley.

-
- Powell, W. B. & S. Ghadimi (2022). *The parametric cost function approximation: A new approach for multistage stochastic programming*. <http://arxiv.org/pdf/2201.00258v1>. Accessed 15-June-2023.
- Psaraftis, H. N., M. Wen & C. A. Kontovas (2016). “Dynamic vehicle routing problems: Three decades and counting”. In: *Networks* 67.1, pp. 3–31.
- Qiu, H., R. Li & J. Zhao (2018). *Dynamic pricing in shared mobility on demand service*.
- Rai, H. B., S. Verlinde & C. Macharis (2021). “Who is interested in a crowdsourced last mile? A segmentation of attitudinal profiles”. In: *Travel Behaviour and Society* 22, pp. 22–31.
- Riley, C., A. Legrain & P. van Hentenryck (2019). “Column generation for real-time ride-sharing operations”. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Ed. by L.-M. Rousseau & K. Stergiou. Vol. 11494. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 472–487.
- Riley, C., P. van Hentenryck & E. Yuan (2020). “Real-time dispatching of large-scale ride-sharing systems: Integrating optimization, machine learning, and model predictive control”. In: *Twenty-Ninth International Joint Conference on Artificial Intelligence* 5, pp. 4417–4423.
- Ritzinger, U., J. Puchinger & R. F. Hartl (2016). “A survey on dynamic and stochastic vehicle routing problems”. In: *International Journal of Production Research* 54.1, pp. 215–231.
- Ropke, S. & D. Pisinger (2006). “An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows”. In: *Transportation Science* 40.4, pp. 455–472.
- Savelsbergh, M. W. & M. W. Ulmer (2022). “Challenges and opportunities in crowdsourced delivery planning and operations”. In: *4OR* 20.1, pp. 1–21.
- Sayarshad, H. R. & J. Y. Chow (2015). “A scalable non-myopic dynamic dial-a-ride and pricing problem”. In: *Transportation Research Part B: Methodological* 81, pp. 539–554.
- (2017). “Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem”. In: *Transportation Research Part E: Logistics and Transportation Review* 106, pp. 60–77.

- Sayarshad, H. R. & H. Oliver Gao (2018). “A scalable non-myopic dynamic dial-a-ride and pricing problem for competitive on-demand mobility systems”. In: *Transportation Research Part C: Emerging Technologies* 91, pp. 192–208.
- Schilde, M., K. F. Doerner & R. F. Hartl (2011). “Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports”. In: *Computers & Operations Research* 38.12, pp. 1719–1730.
- Shah, S., M. Lowalekar & P. Varakantham (2020). “Neural approximate dynamic programming for on-demand ride-pooling”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01, pp. 507–515.
- Sharif Azadeh, S., B. Atasoy, M. E. Ben-Akiva, M. Bierlaire & M. Y. Maknoon (2022). “Choice-driven dial-a-ride problem for demand responsive mobility service”. In: *Transportation Research Part B: Methodological* 161, pp. 128–149.
- Shaw, P. (1998). “Using constraint programming and local search methods to solve vehicle routing problems”. In: *Principles and Practice of Constraint Programming — CP98*. Ed. by G. Goos, J. Hartmanis, J. van Leeuwen, M. Maher & J.-F. Puget. Vol. 1520. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 417–431.
- Simonetto, A., J. Monteil & C. Gambella (2019). “Real-time city-scale ridesharing via linear assignment problems”. In: *Transportation Research Part C: Emerging Technologies* 101, pp. 208–232.
- Soeffker, N., M. W. Ulmer & D. C. Mattfeld (2017). “On fairness aspects of customer acceptance mechanisms in dynamic vehicle routing”. In: *R. O. Large, N. Kramer, A.-K. Radig, M. Schäfer, A. Sulzbach (Hg.) 2017 – Proceedings of Logistikmanagement*, pp. 17–24.
- (2022). “Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review”. In: *European Journal of Operational Research* 298.3, pp. 801–820.
- Spielman, F. (2021). *Alderman accuses Uber, Lyft of ‘predatory fares,’ wants price cap imposed*. <https://chicago.suntimes.com/city-hall/2021/5/24/22451667/uber-lyft-ride-share-hailing-surge-pricing-cap-city-council-ordinance-alderman-reilly-taxi-cabs>.
- Taylor, T. A. (2018). “On-demand service platforms”. In: *Manufacturing & Service Operations Management* 20.4, pp. 704–720.

- Thomas, B. W. (2007). “Waiting strategies for anticipating service requests from known customer locations”. In: *Transportation Science* 41.3, pp. 319–331.
- Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.
- Uber Pool (n.d.). *How Uber Pool works*. <https://www.uber.com/us/en/ride/uberpool/>. Accessed 07-April-2021.
- Ulmer, M. W. & M. W. Savelsbergh (2020). “Workforce scheduling in the era of crowdsourced delivery”. In: *Transportation Science* 54, pp. 1113–1133.
- Ulmer, M. W., B. W. Thomas, A. M. Campbell & N. Woyak (2021). “The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times”. In: *Transportation Science* 55.1, pp. 75–100.
- Ulmer, M. W. (2020). “Dynamic pricing and routing for same-day delivery”. In: *Transportation Science* 54.4, pp. 1016–1033.
- Ulmer, M. W., J. C. Goodson, D. C. Mattfeld & M. Hennig (2019). “Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests”. In: *Transportation Science* 53.1, pp. 185–202.
- Ulmer, M. W., L. Heilig & S. Voß (2017). “On the value and challenge of real-time information in dynamic dispatching of service vehicles”. In: *Business & Information Systems Engineering* 59.3, pp. 161–171.
- Ulmer, M. W., D. C. Mattfeld & F. Köster (2018). “Budgeting time for dynamic vehicle routing with stochastic customer requests”. In: *Transportation Science* 52.1, pp. 20–37.
- Urata, J., Z. Xu, J. Ke, Y. Yin, G. Wu, H. Yang & J. Ye (2021). “Learning ride-sourcing drivers’ customer-searching behavior: A dynamic discrete choice approach”. In: *Transportation Research Part C: Emerging Technologies* 130, p. 103293.
- Wallar, A., M. van der Zee, J. Alonso-Mora & D. Rus (2018). “Vehicle rebalancing for mobility-on-demand systems with ride-sharing”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4539–4546.
- Wang, G., H. Zhang & J. Zhang (2022). “On-demand ride-matching in a spatial model with abandonment and cancellation”. In: *Operations Research*.
- Wang, H. & H. Yang (2019). “Ridesourcing systems: A framework and review”. In: *Transportation Research Part B: Methodological* 129, pp. 122–155.

- Wang, H. & X. Bei (2022). “Real-time driver-request assignment in ridesourcing”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.4, pp. 3840–3849.
- Wang, Q., Y. Huang, S. Jasin & P. V. Singh (2023). “Algorithmic transparency with strategic users”. In: *Management Science* 69.4, pp. 2297–2317.
- Wang, X., W. Liu, H. Yang, D. Wang & J. Ye (2020). “Customer behavioural modelling of order cancellation in coupled ride-sourcing and taxi markets”. In: *Transportation Research Part B: Methodological* 132, pp. 358–378.
- Waßmuth, K., C. Köhler, N. Agatz & M. Fleischmann (2023). “Demand management for attended home delivery—A literature review”. In: *European Journal of Operational Research*.
- Wen, J., J. Zhao & P. Jaillet (2017). “Rebalancing shared mobility-on-demand systems: A reinforcement learning approach”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 220–225.
- Xi, J., F. Zhu, Y. Chen, Y. Lv, C. Tan & F. Wang (2021). “DDRL: A decentralized deep reinforcement learning method for vehicle repositioning”. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3984–3989.
- Xiang, Z., C. Chu & H. Chen (2008). “The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments”. In: *European Journal of Operational Research* 185.2, pp. 534–551.
- Yang, H., X. Qin, J. Ke & J. Ye (2020). “Optimizing matching time interval and matching radius in on-demand ride-sourcing markets”. In: *Transportation Research Part B: Methodological* 131, pp. 84–105.
- Yang, X. & A. K. Strauss (2017). “An approximate dynamic programming approach to attended home delivery management”. In: *European Journal of Operational Research* 263.3, pp. 935–945.
- Yang, X., A. K. Strauss, C. S. M. Currie & R. Eglese (2016). “Choice-based demand management and vehicle routing in e-fulfillment”. In: *Transportation Science* 50.2, pp. 473–488.
- Yu, X. & S. Shen (2020). “An integrated decomposition and approximate dynamic programming approach for on-demand ride pooling”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.9, pp. 3811–3820.

- Yu, Z. & M. Hu (2021). “Deep reinforcement learning with graph representation for vehicle repositioning”. In: *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14.
- Zhang, R., F. Rossi & M. Pavone (2016). “Model predictive control of autonomous mobility-on-demand systems”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1382–1389.
- Zhang, R. & M. Pavone (2016). “Control of robotic mobility-on-demand systems: A queueing-theoretical perspective”. In: *The International Journal of Robotics Research* 35.1-3, pp. 186–203.
- Zhu, Z., J. Ke & H. Wang (2021). “A mean-field Markov decision process model for spatial-temporal subsidies in ride-sourcing markets”. In: *Transportation Research Part B: Methodological* 150, pp. 540–565.

Appendix

A An efficient insertion heuristic for on-demand ride-sharing services

A.1 Parameter tuning

The parameter tuning of the LNS is based on the *Resource Demand Ratio* sensitivity analysis. 10 instances generated for parameter tuning are solved five times, each time with an adapted fleet size. For feasibility check and re-optimization in the scope of *Basic Control*, and *Advanced Demand Control* the tuning of the parameters is based on *Basic Control*. For *Advanced Fulfillment Control*, a separated tuning is performed, since considerably more requests have to be handled during a feasibility check and the final optimization. Regarding the TOP, the parameter tuning is based on *Advanced Control*. The resulting values are mostly applied as well to solve the TOP as favorability check within *Advanced Demand Control*. However, the number of required iterations β and thus the computational effort is determined separately.

The number of iterations as termination criterion has a particular impact on the solution quality and the computing time. We define a reasonable maximum number of iterations β as follows. We begin with an overly large number and then check the last iteration yielding a new best solution. The final number of iterations is then determined in dependence of its magnitude by rounding up to the next number divisible by 100, 1000, or 10000. The results of this procedure are summarized in Table A.1. At the beginning, the percentage of trips removed per iteration is set to $\gamma_1 = 0.3$ and $\gamma_2 = 0.4$ following Ropke & Pisinger (2006), and the noise for the operators is set to a medium level of $\delta_1 = 4$ and $\delta_2 = 4$.

| Policy | Case | Final β | Test β | Last successful iteration per fleet size | | | | |
|-------------------------------------|--------------------|---------------|--------------|--|-------|-------|--------------|-------------|
| | | | | 2 | 6 | 10 | 14 | 18 |
| <i>Basic Control</i> | Feasibility check | 100 | 1000 | 1 | 2 | 5 | 5 | 11 |
| | Re-optimization | 200 | 1000 | 0 | 1 | 8 | 24 | 126 |
| <i>Advanced Demand Control</i> | TOP | 3000 | 3000 | 2895 | 2995 | 2995 | 2996 | 2999 |
| <i>Advanced Fulfillment Control</i> | Feasibility check | 1000 | 2000 | 260 | 531 | 728 | 910 | 719 |
| | Final optimization | 10000 | 10000 | 3 | 1245 | 6283 | 9713 | 9778 |
| <i>Advanced Control</i> | TOP | 30000 | 40000 | 4770 | 15195 | 15288 | 27235 | 15306 |

Table A.1: Number of Iterations

It can be observed that the values vary considerably, which is due to the different number of replannable requests and the differences between single and repeated execution. Overall, a reasonable value of β could be determined for most of the cases. An exception is the TOP in case of *Advanced Demand Control*. Here, improvements are still found for all fleet sizes close to the last iteration. A further increase of the number of iterations was omitted, since the tested β values already induce significant computational effort. However, since this check is simply intended to determine whether a trip is favorable, i.e. whether it can be easily integrated together with current and future requests, there is no need to focus on exceptional solution quality.

Further parameter values are determined by the acceptance rate calculated across all instances. The first parameter values are γ_1 and γ_2 , which control the minimum and maximum percentage of requests to be removed per iteration. To determine these two parameters, values between $\gamma_1 = 0.1, \gamma_2 = 0.2$ and $\gamma_1 = 0.7, \gamma_2 = 0.8$ were tested for the same LNS cases as before. It turns out that in cases with a high number of replannable requests, lower values and thus smaller changes in the solution are advantageous. The acceptance rate for these cases differs up to 4%. In the opposite case, with only a few replannable requests, higher values are slightly advantageous, however, the differences are small. Based on these results, for the insertion and re-optimization in the case of *Basic Control* and *Advanced Demand Control*, $\gamma_1 = 0.7, \gamma_2 = 0.8$ is applied. For both TOP as well as the insertion and final optimization of *Advanced Fulfillment Control*, we set $\gamma_1 = 0.1, \gamma_2 = 0.2$. Regarding the noise parameters δ_1 and δ_2 , which are applied in the worst-removal and the regret-2 operator, no noise (δ_1 and $\delta_2 = 0$), medium noise (δ_1 and $\delta_2 = 4$) and a high degree of noise (δ_1 and $\delta_2 = 8$) are examined separately. However, a significant influence on the acceptance rate could not be determined. Since the results were best for all examined cases when using a medium noise (δ_1 and $\delta_2 = 4$), this value is selected for the experiments. For detailed results of the tuning of $\gamma_1, \gamma_2, \delta_1$, and δ_2 see Table A.2, A.3, and A.4.

| $\gamma_1 - \gamma_2$ | Advanced Control TOP | Advanced Fulfillment Control Feasibility check & final optimization | Basic Control Feasibility check & re-optimization |
|-----------------------|-------------------------|---|---|
| 10% - 20% | 64.4% | 54.0% | 44.4% |
| 30% - 40% | 63.9% | 53.3% | 44.5% |
| 50% - 60% | 61.8% | 51.5% | 44.5% |
| 70% - 80% | 61.4% | 50.5% | 44.6% |

Table A.2: Percentage of Requests Removed per Iteration

| Values | Advanced Control TOP | Advanced Fulfillment Control Feasibility check & final optimization | Basic Control Feasibility check & re-optimization |
|----------------|-------------------------|---|---|
| $\delta_1 = 0$ | 64.2% | 53.8% | 44.6% |
| $\delta_1 = 4$ | 64.4% | 54.0% | 44.7% |
| $\delta_1 = 8$ | 64.3% | 53.6% | 44.6% |

Table A.3: Noise Value Regret-2 Insertion

| Values | Advanced Control TOP | Advanced Fulfillment Control Feasibility check & final optimization | Basic Control Feasibility check & re-optimization |
|----------------|-------------------------|---|---|
| $\delta_2 = 0$ | 64.3% | 53.7% | 44.6% |
| $\delta_2 = 4$ | 64.4% | 54.0% | 44.7% |
| $\delta_2 = 8$ | 64.4% | 53.8% | 44.6% |

Table A.4: Noise Value Worst-Removal

A.2 Computational results

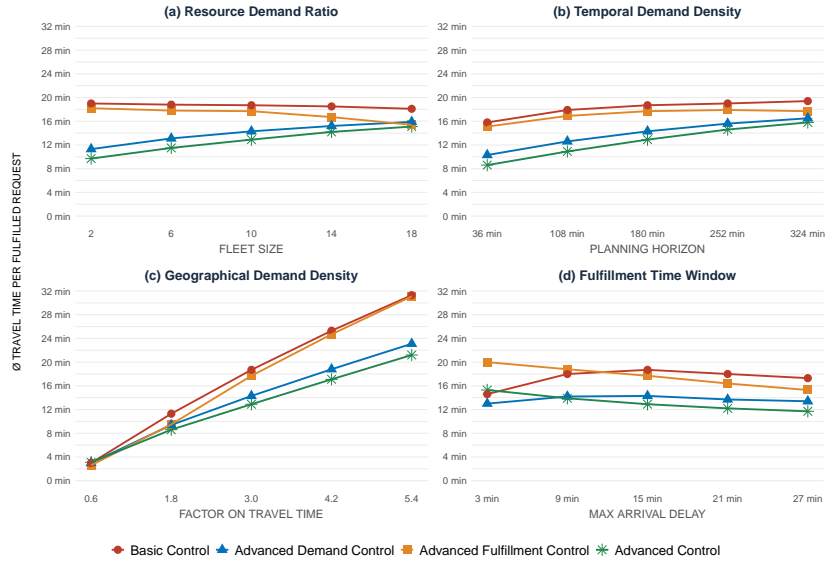


Figure A.1: Sensitivity Analyses: Average Travel Time per Fulfilled Request



Figure A.2: Sensitivity Analyses: Pooling Rate

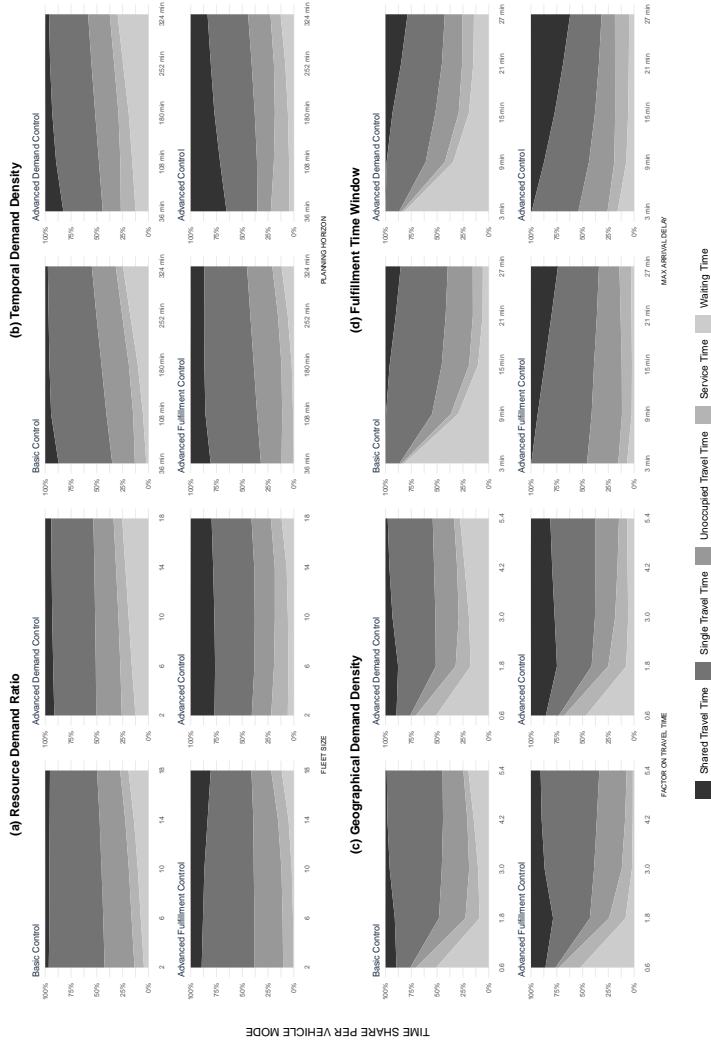


Figure A.3: Sensitivity Analyses: Time Share per Vehicle Mode

Appendix

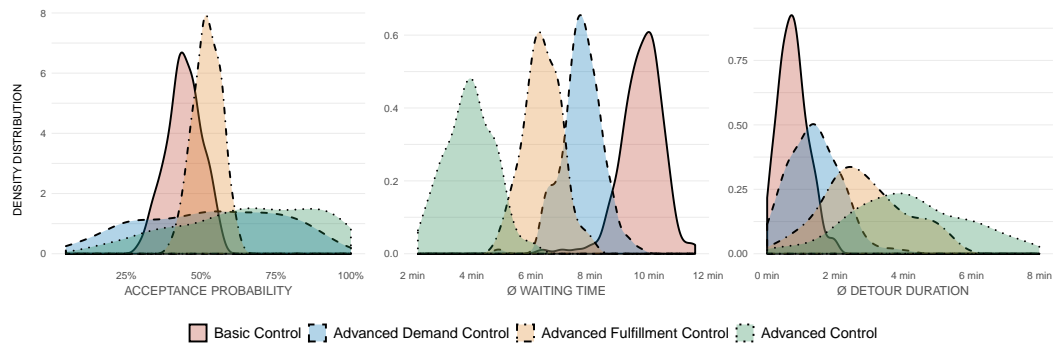


Figure A.4: *Temporal Demand Density: Quality of Service per Trip*

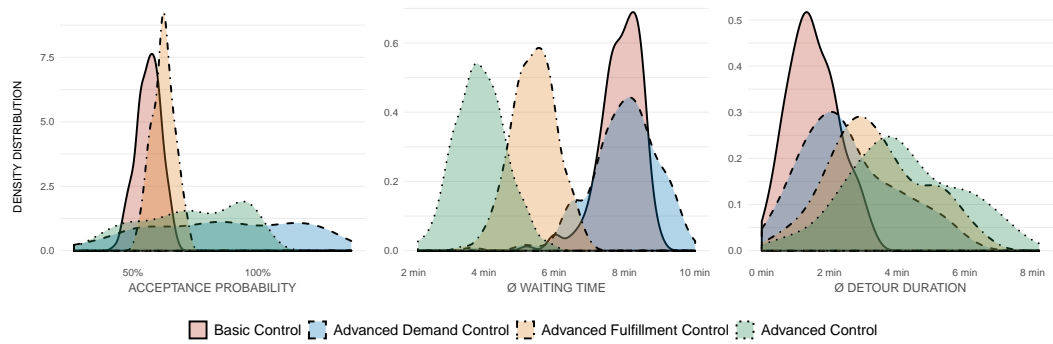


Figure A.5: *Geographical Demand Density: Quality of Service per Trip*

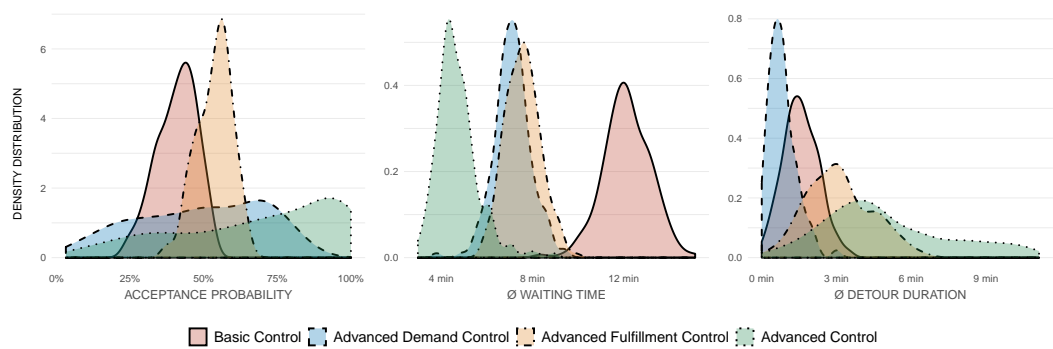


Figure A.6: *Fulfillment Time Window: Quality of Service per Trip*

B Heatmap-based decision support for repositioning in ride-sharing systems

B.1 Details of MPC benchmark

Even though the implementation of MPC is based on Pouls et al. (2020), there are some differences. The first concerns the representation of the service area: To ensure comparability with the other policies, we define subareas for each repositioning location instead of using a grid. Thereby, each location belongs to the area of the nearest repositioning location in terms of travel time. Another difference is the prediction of the expected demand. Instead of reactive or perfect demand, we determine the expected demand by taking the average of the demand that occurred in preliminary simulations. Finally, there is a difference in the execution of the repositionings determined by the model. This has been adapted so that drivers always move directly to the repositioning locations, while the assignment of which driver performs which repositioning is performed by a second travel-time minimizing model. Regarding the extensive parameterization of the first model, the original values are used, except for the two parameters that differ depending on the service area. These two parameters which indicate how many requests can be fulfilled between two repositioning periods and the distance over which a repositioning location can cover the demand of another repositioning location were tuned sequentially for Manhattan and Brooklyn as part of preliminary simulations.

B.2 Driver decision modeling details

With respect to *time savers* and *time investors*, the repositioning duration $\bar{\theta}_{rk}$ is reused for the driver-specific values u_{rjk} . Thus, for *time investors*, $u_{rjk} = \kappa_{rk} \times \bar{\theta}_{rk}$. In contrast, for *time savers* the normalized repositioning duration $\bar{\theta}_k$ is converted resulting in $u_{rjk} = \kappa_{rk} \times (1 - \bar{\theta}_{rk})$.

For *supply-driven* and *demand-driven* drivers, the u_{rjk} values are approximated in a similar way as the distribution of unoccupied drivers \bar{b}_k used in RH. However, the expected supply distribution ι_{rk} captures the entire fleet by counting for each repositioning location $r \in R$ how many times it is nearest to a current vehicle location n_k . Likewise, for the expected demand distribution ι_{rk} , it is determined how many pickup requests are expected in the next 10 minutes in the vicinity of each

repositioning location $r \in R$. Furthermore, both supply distribution \bar{u}_{rk} and demand distribution \bar{d}_{rk} are min-max normalized to ensure comparable scales. Thus, for *demand-driven drivers* $u_{rjk} = \bar{d}_{rk} \times \bar{u}_{rk}$, whereas for *supply-driven drivers* the normalized values \bar{d}_{rk} are additionally converted, resulting in $u_{rjk} = \bar{d}_{rk} \times (1 - \bar{u}_{rk})$.

For *inexperienced drivers*, the u_{rjk} values are determined identically as for *time savers* ($u_{rjk} = \bar{d}_{rk} \times (1 - \bar{\theta}_{rk})$). In case of *experienced drivers*, the driver-specific values are defined based on *time investors* and *demand-driven drivers* as follows: $u_{rjk} = \bar{d}_{rk} \times \bar{\theta}_{rk} \times \bar{u}_{rk}$.

B.3 Cancellation rates over time

We have seen that our RH-based guidance reduces the average daily cancellation rates significantly. In the following, we analyze the rates over time. The average values are shown in Figure B.1. We observe that both RH and MPC reduce the cancellation rates at every point of time compared to NR. Furthermore, we see the high-demand rush-hour peak with increased cancellations around hours 18 to 20. Compared to NR, RH can reduce the peak substantially. This confirms the particular value of anticipation to prepare for demand peaks as also observed by Brinkmann et al. (2019) for ride-sharing systems.

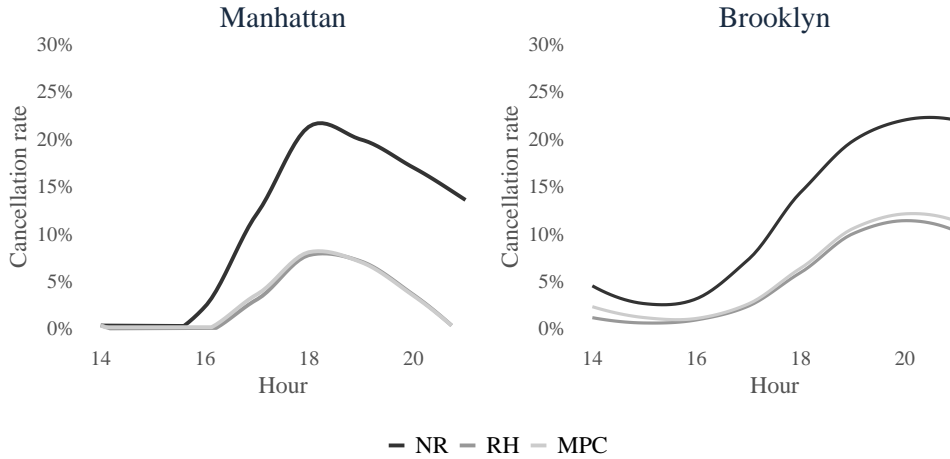


Figure B.1: Cancellations Rates per Hour

B.4 Sensitivity analysis

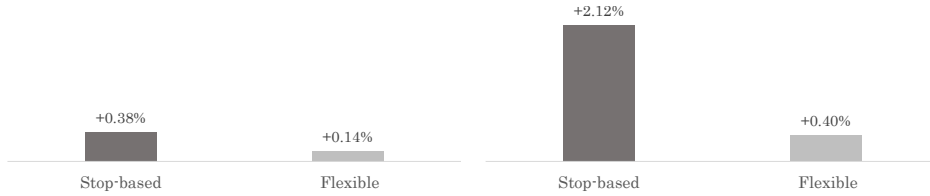
We present a sensitivity analysis for a set of instance parameters. We first analyze different routing and assignment procedures. We then have a look at different levels of user patience and conclude with results where the number of repositioning locations vary. For all settings, we report the cancellation rates, the empty driving time, the direct rides (without detours), the average waiting time, and the average detour time.

Routing In our experiments, we assumed that drivers can divert from their next stop to serve a new request. We denote this routing as “flexible”. We compare this to a routing where drivers can only receive an assignment at a stop, e.g., due to safety restrictions, and call this “stop-based”. The detailed results are shown in Table B.1 (bold entries indicate the best values in a setting, italic entries the best global values). We observe that our policy is superior regardless of the routing. However, with flexible routing, the cancellation rates can be further decreased, as expected and also observed by Ulmer et al. (2017). Notably, besides more services, also the service quality increases as the number of direct rides goes up while waiting time and detour decrease.

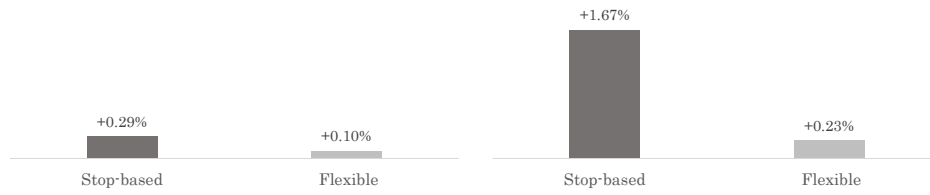
| | Stop-based | | | Flexible | | |
|--------------------------|--------------|--------------|-------|--------------|--------------|------------|
| | NR | RH | MPC | NR | RH | MPC |
| Manhattan | | | | | | |
| Cancellation rate | 13.6% | 4.6% | 5.0% | 13.4% | 3.6% | 3.7% |
| Empty driving time | 12.6% | 28.2% | 22.3% | 10.8% | 24.9% | 20.6% |
| Direct rides | 19.1% | 21.3% | 19.7% | 10.1% | 15.8% | 13.0% |
| Avg. waiting time in min | 5.7 | 5.2 | 5.4 | 5.4 | 4.7 | 4.7 |
| Avg. detour in min | 4.3 | 4.1 | 4.2 | 5.0 | 4.5 | 4.6 |
| Brooklyn | | | | | | |
| Cancellation rate | 13.8% | 8.1% | 9.4% | 13.0% | 6.0% | 6.5% |
| Empty driving time | 16.1% | 26.1% | 24.2% | 13.2% | 23.3% | 21.0% |
| Direct rides | 29.4% | 30.2% | 28.0% | 14.4% | 18.9% | 16.2% |
| Avg. waiting time in min | 6.0 | 5.9 | 6.2 | 5.4 | 5.2 | 5.4 |
| Avg. detour in min | 3.5 | 3.5 | 3.6 | 4.6 | 4.2 | 4.4 |

Table B.1: Performance under Alternative Routing Strategies

Regarding the evaluation under non-compliance, we analyze the impact of the routing strategy for split fleets with partially non-compliant drivers. Figure B.2 shows the system-wide increase in cancellations and Figure B.3 shows the change in earnings per driver group, both relative to the respective full-compliance case. In Figure B.2, it can be observed that the system performance decreases significantly more for stop-based routing. Furthermore, Figure B.3 shows that advantages and disadvantages in earnings are reversed depending on the routing strategy. For example, time investors benefit greatly from the possibility of deviation, while time savers do best under a stop-based routing. In summary, routing has a major impact on how non-compliance affect both system and driver performance.



(a) Time-Savers/-Investors



(b) Supply-/Demand-Driven

Figure B.2: Cancellation Rates under Alternative Routing Strategies Relative to the Full Compliance Case

Assignment Next, we analyze how our RH performs with different assignment strategies. In our experiments, we presented an assignment strategy that balances flexible assignments (minimum pickup time) with consolidation (minimize increase in travel time). The results can be found in Table B.2. We observe that our RH-based guidance is superior regardless the assignment. The results further confirm that assignment strategies should carefully balance a flexible fleet setup with efficient routes as also observed by Ulmer et al. (2021). Interestingly, the average waiting time

and the detour time are the smallest with the flexible assignment. The explanation is twofold. First, we serve fewer customers, thus the relative resources per service are higher. Second, in the flexible case, we send the nearest vehicle.



Figure B.3: Split fleets under Alternative Routing Strategies: Earnings for Compliant Drivers Competing with Partial Non-Compliant Drivers

In addition to full compliance, we evaluate non-compliance under different assignment strategies using the example of split fleets with partially non-compliant drivers. Figure B.4 shows the increase in cancellations relative to the corresponding full-compliance case. The increase is particularly low for the flexible strategy and comparable for the balanced and consolidation strategy. This can be attributed to the already significantly higher cancellation rates for the full-compliance case with flexible assignment. In addition, Figure B.5 illustrates the change in earnings per driver group, again relative to the full-compliance cases. Here it can be noticed that also the earning advantages and disadvantages are less pronounced for flexible assign-

Appendix

ment. Moreover, the trend is mostly contrary to consolidation, while, as expected, balance represents an intermediate strategy. In summary, the assignment strategy can play an important role in the evaluation of non-compliant decision-making.

| | Flexible | | | Balanced | | | Consolidation | | |
|--------------------------|--------------|--------------|-------------|--------------|--------------|------------|---------------|--------------|------------|
| | NR | RH | MPC | NR | RH | MPC | NR | RH | MPC |
| Manhattan | | | | | | | | | |
| Cancellation rate | 11.8% | 5.8% | 5.8% | 13.4% | 3.6% | 3.7% | 14.1% | 3.8% | 3.9% |
| Empty driving time | 11.1% | 23.8% | 18.6% | 10.8% | 24.9% | 20.6% | 10.8% | 22.5% | 20.0% |
| Direct rides | 15.7% | 20.0% | 17.4% | 10.1% | 15.8% | 13.0% | 7.1% | 9.9% | 8.6% |
| Avg. waiting time in min | 4.6 | 4.0 | 4.1 | 5.4 | 4.7 | 4.7 | 6.3 | 6.1 | 6.1 |
| Avg. detour in min | 4.5 | 4.1 | 4.2 | 5.0 | 4.5 | 4.6 | 5.2 | 4.9 | 5.0 |
| Brooklyn | | | | | | | | | |
| Cancellation rate | 12.9% | 8.5% | 8.7% | 13.0% | 6.0% | 6.5% | 13.8% | 6.2% | 6.7% |
| Empty driving time | 13.9% | 22.0% | 19.5% | 13.2% | 23.3% | 21.0% | 13.3% | 22.7% | 21.3% |
| Direct rides | 17.6% | 20.0% | 17.8% | 14.4% | 18.9% | 16.2% | 12.8% | 15.9% | 14.8% |
| Avg. waiting time in min | 4.8 | 4.6 | 4.8 | 5.4 | 5.2 | 5.4 | 6.1 | 6.2 | 6.2 |
| Avg. detour in min | 4.4 | 4.1 | 4.3 | 4.6 | 4.2 | 4.4 | 4.7 | 4.4 | 4.5 |

Table B.2: Performance under Alternative Assignment Strategies



(a) Time-Savers-/Investors



(b) Supply-/Demand-Driven

Figure B.4: Cancellation Rates under Alternative Assignment Strategies Relative to the Full Compliance Case



Figure B.5: Split fleets under Alternative Assignment Strategies: Earnings for Compliant Drivers Competing with Partial Non-Compliant Drivers

Maximum waiting time In our experiments, we assumed that users wait at most 10 minutes to be picked up. In the following, we analyze how the performance of the RH changes when users are more patient (15 min) or less patient (5 min). The results are shown in Table B.3. As expected, with increasing patience, the service level goes up and more users are consolidated. Again, regardless of the level of patience, our RH performs best and outperforms the NR-strategy significantly.

Appendix

| | 5 minutes | | | 10 minutes | | | 15 minutes | | |
|--------------------------|--------------|--------------|--------------|--------------|--------------|------------|--------------|--------------|-------------|
| | NR | RH | MPC | NR | RH | MPC | NR | RH | MPC |
| Manhattan | | | | | | | | | |
| Cancellation rate | 27.0% | 10.3% | 10.2% | 13.4% | 3.6% | 3.7% | 5.2% | 1.4% | 1.4% |
| Empty driving time | 9.6% | 26.7% | 22.1% | 10.8% | 24.9% | 20.6% | 11.7% | 23.7% | 20.2% |
| Direct rides | 16.5% | 22.3% | 19.1% | 10.1% | 15.8% | 13.0% | 8.6% | 13.7% | 11.6% |
| Avg. waiting time in min | 3.0 | 3.0 | 3.0 | 5.4 | 4.7 | 4.7 | 7.6 | 5.9 | 6.0 |
| Avg. detour in min | 4.2 | 3.7 | 3.9 | 5.0 | 4.5 | 4.6 | 5.2 | 4.7 | 4.9 |
| Brooklyn | | | | | | | | | |
| Cancellation rate | 31.0% | 18.4% | 21.3% | 13.0% | 6.0% | 6.5% | 6.0% | 2.6% | 2.7% |
| Empty driving time | 11.4% | 26.7% | 23.7% | 13.2% | 23.3% | 21.0% | 13.6% | 22.9% | 20.2% |
| Direct rides | 26.9% | 32.4% | 28.4% | 14.4% | 18.9% | 16.2% | 11.3% | 15.4% | 13.0% |
| Avg. waiting time in min | 3.0 | 3.0 | 3.0 | 5.4 | 5.2 | 5.4 | 7.5 | 6.9 | 7.0 |
| Avg. detour in min | 3.5 | 3.1 | 3.3 | 4.6 | 4.2 | 4.4 | 5.0 | 4.6 | 4.8 |

Table B.3: Sensitivity Analysis of the Maximum Waiting Time

Number of repositioning locations Finally, we analyze how the number of repositioning locations impacts the performance of our RH. In our experiments, we used 100 locations for potential repositioning. Now, we also analyze instances with 60 and 20 locations. The results are shown in Table B.4. Again, our method performs best regardless the setting. Further, the performance usually increases when the number of locations go up, as also repositioning flexibility increases.

| | #20 | | | #60 | | | #100 | | |
|--------------------------|--------------|--------------|-------|--------------|--------------|-------|--------------|--------------|------------|
| | NR | RH | MPC | NR | RH | MPC | NR | RH | MPC |
| Manhattan | | | | | | | | | |
| Cancellation rate | 12.2% | 5.0% | 6.4% | 13.5% | 5.4% | 6.1% | 13.4% | 3.6% | 3.7% |
| Empty driving time | 11.6% | 22.5% | 23.0% | 10.9% | 24.7% | 26.1% | 10.8% | 24.9% | 20.6% |
| Direct rides | 10.4% | 11.1% | 10.2% | 10.1% | 14.4% | 13.4% | 10.1% | 15.8% | 13.0% |
| Avg. waiting time in min | 5.3 | 5.1 | 5.4 | 5.4 | 4.8 | 5.1 | 5.4 | 4.7 | 4.7 |
| Avg. detour in min | 5.0 | 4.8 | 4.9 | 5.0 | 4.6 | 4.6 | 5.0 | 4.5 | 4.6 |
| Brooklyn | | | | | | | | | |
| Cancellation rate | 13.4% | 8.0% | 15.1% | 12.9% | 6.1% | 7.2% | 13.0% | 6.0% | 6.5% |
| Empty driving time | 13.2% | 22.3% | 22.4% | 13.4% | 24.4% | 23.0% | 13.2% | 23.0% | 21.0% |
| Direct rides | 14.3% | 15.9% | 12.9% | 14.5% | 19.0% | 16.8% | 14.4% | 18.9% | 16.2% |
| Avg. waiting time in min | 5.4 | 5.7 | 6.2 | 5.4 | 5.3 | 5.4 | 5.4 | 5.2 | 5.4 |
| Avg. detour in min | 4.6 | 4.4 | 4.7 | 4.6 | 4.2 | 4.4 | 4.6 | 4.2 | 4.4 |

Table B.4: Sensitivity Analysis of the Number of Repositioning Locations

B.5 Dynamic fleet size

In our main experiments, we assumed that the size of the fleet stays constant over the course of the day. This allowed a clean analysis of our methodology and different driver behavior. However, in reality, fleet sizes may change dynamically as drivers leave or enter the system. To analyze the performance of our algorithm under dynamic fleet size, we run an additional experiment. In this experiment, with a certain probability, a driver leaves and/or enters every minute. We vary the likelihood between 0% and 80% in steps of 20%. While the expected number of drivers available for assignments remains the same overall (200), during some times of the day, we observe significantly more drivers (up to 238 in our experiments) or fewer drivers (minimum 157 in our experiments). We assume that a driver completes the remaining assigned services before leaving the system. Furthermore, for a new driver entering the system, a location is drawn randomly from the set of locations in the service area. We compare the cancellation rates with the NR-policy and the MPC-policy.

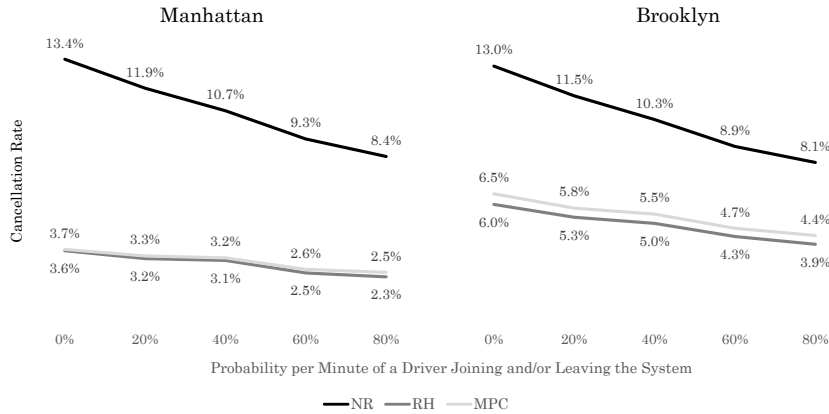


Figure B.6: Dynamic Fleet Size

The results for Manhattan and Brooklyn are shown in Figure B.6. The x-axis depicts the probability of a driver leaving and/or entering per minute. The y-axis depicts the performance of the three policies. We make two observations. First, the performance of our RH-policy is superior to NR and comparable to MPC regardless the instance setting. This indicates that our policy is also suitable for cases when the

fleet size changes dynamically. Second, the cancellation rate decreases with increasing probability of drivers leaving or entering the system. This can be explained as follows. On the one hand, drivers need to finish their assigned services before leaving the system, while new, unoccupied drivers are joining, resulting in more available driver resources overall. On the other hand, drivers leave the system at the location of the last completed service, while new drivers enter the system randomly over the service area causing a natural balancing of the fleet regardless of the applied policy. While the former causes performance to improve under all policies, the latter causes the gap between NR and the two actively repositioning policies MPC and RH to decrease as fleet volatility increases.

C Design of multi-optional pickup time offers in ride-sharing systems

C.1 Evaluation of vehicle waiting strategies

For the determination of potential pickup time offerings for the *CFA* and the three benchmark policies, two vehicle waiting strategies have been examined. For the first one, vehicle waiting is allowed to postpone pickup times, as long as no temporal constraint is violated. For the second one, vehicle waiting is excluded, so that only the earliest feasible pickup time is considered.

| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
|-----|--------------|---------------|--------------|--------------|
| Yes | 5.95% | 11.04% | 7.65% | 9.92% |
| No | 6.10% | 11.46% | 7.25% | 9.86% |

Table C.1: Waiting to Postpone Pickup Times

The results in terms of the average cancellation rate over 100 instances are compared for both variants in Table C.1. It can be observed that the influence of the waiting strategy is rather low for all four policies. However, allowing vehicles to wait to postpone pickups is slightly beneficial for the *PI* and *PPI* and slightly detrimental for the *CFA* and *NI*. This can be attributed to the fact that in *PI* and *PPI*, vehicles wait according to reliable information on travelers' requirements and preferences, respectively. Consistent with the results, waiting is allowed in all further experiments for *PI* and *PPI*, while excluded for *CFA* and *NI*.

C.2 Parameter tuning for the assignment evaluation

For the tuning of the assignment evaluation parameters α and β , values between 0 and 1 were tested, respectively.

The average cancellation rates are shown in Table C.2, with the best result highlighted in bold for each policy. For *PI*, $\alpha = 1.0$ and $\beta = 0.5$ is the most promising combination, focusing on a short fulfillment duration rather than on consolidating transportation requests. For *PPI* and *CFA*, the performance is best with an equal weighting. Finally, *NI* yields a marginally lower average cancellation rate if the consolidation of transportation requests is considered slightly more important. The corresponding parameter values are the basis for further computational experiments.

| | | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
|-----------------|----------------|--------------|---------------|--------------|--------------|
| $\alpha = 0.0$ | $\beta = 1.0$ | 6.79% | 11.21% | 7.60% | 9.95% |
| $\alpha = 0.25$ | $\beta = 1.0$ | 6.47% | 11.17% | 7.43% | 9.91% |
| $\alpha = 0.5$ | $\beta = 1.0$ | 6.14% | 11.12% | 7.36% | 9.87% |
| $\alpha = 0.75$ | $\beta = 1.0$ | 6.10% | 11.10% | 7.27% | 9.85% |
| $\alpha = 1.0$ | $\beta = 1.0$ | 5.95% | 11.04% | 7.25% | 9.86% |
| $\alpha = 1.0$ | $\beta = 0.75$ | 5.80% | 11.12% | 7.26% | 9.93% |
| $\alpha = 1.0$ | $\beta = 0.5$ | 5.71% | 11.18% | 7.47% | 9.99% |
| $\alpha = 1.0$ | $\beta = 0.25$ | 5.77% | 11.45% | 8.04% | 10.20% |
| $\alpha = 1.0$ | $\beta = 0.0$ | 6.65% | 12.19% | 9.55% | 10.72% |

Table C.2: Tuning of the Assignment Evaluation Parameters

C.3 Sensitivity analyses

Based on the baseline scenario, seven sensitivity analyses were conducted to examine the reliability of the results and the robustness of the policies under consideration. These analyses include, firstly, alternative demand scenarios based on a different service area and an increased respectively decreased average number of requests per hour. Secondly, they include the design of traveler requirements in terms of the distribution of the earliest possible pickup time, the pickup time window length, and the positioning of the preferred pickup time within the time window. Finally, with the maximum deviation from the agreed pickup time and the maximum travel duration factor, service quality defining parameters were varied. The results for the baseline scenario as well as for the sensitivity analyses are summarized in Tables C.3 to C.10. In the tables, the line *CR* corresponds to the average cancellation rate, *DT* to the average driving time per transport, *SR* to the average percentage of shared transports, and *PD* to the average deviation from the preferred pickup time.

| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
|----|-----------|------------|------------|-----------|
| CR | 5.7% | 11.1% | 7.2% | 9.9% |
| DT | 12.3 | 13.7 | 12.9 | 14.8 |
| SR | 82.6% | 81.3% | 84.6% | 80.1% |
| PD | 5.2 | 1.9 | 3.2 | 3.0 |

Table C.3: Baseline Scenario

| Service Area = Manhattan | | | | |
|--------------------------|-----------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 4.7% | 9.0% | 5.9% | 7.4% |
| DT | 10.1 | 11.2 | 10.5 | 12.6 |
| SR | 84.9% | 83.6% | 90.1% | 81.7% |
| PD | 5.3 | 1.7 | 3.1 | 2.9 |

Table C.4: Service Area

| | 700 | | | | 900 | | | |
|----|-----------|------------|------------|-----------|-----------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 3.0% | 7.0% | 4.2% | 5.4% | 9.6% | 15.9% | 11.5% | 14.9% |
| DT | 12.6 | 14.1 | 13.2 | 15.7 | 12.0 | 13.3 | 12.6 | 14.1 |
| SR | 77.6% | 75.3% | 81.6% | 73.0% | 86.8% | 86.0% | 87.5% | 85.5% |
| PD | 5.3 | 1.5 | 3.0 | 2.7 | 5.1 | 2.3 | 3.4 | 3.3 |

Table C.5: Transportation Requests per Hour

| | Approx. 10 Minutes | | | | Approx. 15 Minutes | | | |
|----|--------------------|------------|------------|-----------|--------------------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 5.0% | 11.1% | 6.6% | 9.5% | 4.3% | 10.6% | 6.0% | 8.7% |
| DT | 12.4 | 13.8 | 13.1 | 15.0 | 12.4 | 13.8 | 13.1 | 14.9 |
| SR | 82.5% | 80.5% | 84.8% | 80.2% | 82.7% | 80.7% | 85.7% | 81.0% |
| PD | 5.2 | 1.8 | 3.1 | 3.0 | 5.3 | 1.9 | 3.1 | 3.1 |

Table C.6: Mean Earliest Pickup Time

| | 10 Minutes | | | | 20 min | | | |
|----|------------|------------|------------|-----------|-----------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 10.7% | 15.6% | 13.4% | 15.7% | 3.0% | 8.4% | 4.4% | 7.4% |
| DT | 12.5 | 13.9 | 13.7 | 15.4 | 12.1 | 13.5 | 12.3 | 14.5 |
| SR | 80.0% | 77.9% | 80.3% | 74.7% | 84.0% | 83.1% | 87.6% | 82.1% |
| PD | 3.5 | 1.2 | 2.2 | 2.5 | 6.8 | 2.9 | 4.6 | 3.9 |

Table C.7: Mean Pickup Time Window Length

| | Middle of the Time Window | | | | End of the Time Window | | | |
|----|---------------------------|------------|------------|-----------|------------------------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 5.7% | 11.3% | 7.4% | 5.7% | 11.5% | 10.2% | 7.4% | 10.4% |
| DT | 12.3 | 13.7 | 12.9 | 14.9 | 12.3 | 13.8 | 12.9 | 14.9 |
| SR | 82.6% | 80.9% | 84.6% | 79.8% | 82.6% | 80.6% | 84.5% | 79.5% |
| PD | 5.2 | 1.8 | 3.1 | 2.9 | 5.2 | 1.7 | 3.1 | 2.8 |

Table C.8: Mean Preferred Pickup Time

| | $\omega = 3$ Minutes | | | | $\omega = 5$ Minutes | | | |
|----|----------------------|------------|------------|-----------|----------------------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 5.2% | 10.5% | 6.6% | 9.1% | 5.1% | 10.2% | 6.4% | 8.7% |
| DT | 12.2 | 13.6 | 12.7 | 14.6 | 12.1 | 13.5 | 12.6 | 14.6 |
| SR | 83.0% | 81.7% | 85.2% | 80.6% | 83.4% | 82.6% | 86.1% | 81.4% |
| PD | 5.2 | 1.8 | 3.1 | 3.0 | 5.2 | 1.7 | 3.1 | 2.9 |

Table C.9: Maximum Pickup Delay

| | $\epsilon = 1.2$ | | | | $\epsilon = 1.8$ | | | |
|----|------------------|------------|------------|-----------|------------------|------------|------------|-----------|
| | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> | <i>PI</i> | <i>PPI</i> | <i>CFA</i> | <i>NI</i> |
| CR | 10.9% | 16.5% | 12.6% | 15.4% | 4.0% | 9.6% | 5.6% | 8.4% |
| DT | 14.0 | 15.2 | 14.6 | 16.0 | 11.8 | 13.2 | 12.4 | 14.5 |
| SR | 58.7% | 50.7% | 57.7% | 51.2% | 89.2% | 90.1% | 91.2% | 88.8% |
| PD | 5.1 | 2.6 | 3.5 | 3.4 | 5.2 | 1.6 | 3.0 | 2.9 |

Table C.10: Maximum Transportation Duration Factor