WORKING PAPER SERIES

Evaluation of Anticipatory Decision-Making in Ride-Sharing Services

Jarmo Haferkamp/Jan Fabian Ehmke

Working Paper No. 4/2020

FACULTY OF ECONOMICS
AND MANAGEMENT

Impressum (§ 5 TMG)

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
Jarmo Haferkamp and Jan Fabian Ehmke
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

SPECIAL ISSUE ARTICLE

Evaluation of Anticipatory Decision-Making in Ride-Sharing Services

Jarmo Haferkamp[1]* | **Jan Fabian Ehmke**[2]*

[1]Management Science Group, Otto von Guericke University Magdeburg, Magdeburg, Germany

[2]Department of Business Decisions and Analytics, University of Vienna, Vienna, Austria

Correspondence
Jarmo Haferkamp, Management Science Group, Otto von Guericke University Magdeburg, Magdeburg, Germany
Email: jarmo.haferkamp@ovgu.de

Funding information
This research was supported by the German Federal Ministry of Transport and Digital Infrastructure, 16AVF2147E.

In recent years, innovative ride-sharing services have gained significant attention. Such services require dynamic decisions on the acceptance of arriving trip requests and vehicle routing to ensure the fulfillment of requests. Decision support for acceptance and routing must be made under uncertainty of future requests. In this paper, we highlight that state-of-the-art approaches focus on anticipatory decision-making for either acceptance or routing decisions. Our aim is to evaluate the potential of different levels of anticipation in ride-sharing services. Up to now, it is unclear how the value of information differs between none, partial, or fully anticipatory decision-making processes. To this end, we define and solve variants of the underlying dial-a-ride problem, which differ in the information available about future requests. Using a large neighborhood search, our experimental results demonstrate that ride-sharing services can highly benefit from anticipatory decision-making, while the favorable level of anticipation depends on particular characteristics of the service, esp. the demand-to-service ratio.

KEYWORDS
ride-sharing, dynamic vehicle routing, anticipation, dial-a-ride problem, large neighborhood search

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 recently launched on-demand ride-sharing services like UberPool [1], 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, on the one hand, lower fares compared to individual taxi services and, on the other hand, a more convenient travel experience compared to traditional local public transport through smaller transport cabins and direct trips. For on-demand ride-sharing services, successful bundling of requests is crucial to ensure their profitability. This poses a great challenge to operators, since requests arrive stochastically and decisions have to be made dynamically. Dynamic fleet management, which controls both the demand to be fulfilled as well as the allocation of the vehicle resources, is a key factor for the successful operation.

Dynamic fleet management comprises (1) acceptance, (2) routing and (3) execution of trip requests. In the *acceptance* step, requests are submitted by travelers – often via a mobile application – within digitized and automated booking processes, and travelers expect instant request confirmations. It must be ensured that all accepted requests can be fulfilled with respect to the given vehicle resources. Operators can also reject requests in favor of potential future requests. The following *routing* step addresses the optimized utilization of the fleet. This includes an efficient dispatching of vehicles in order to fulfill the current, already accepted and potentially future requests. Since accepted requests must be fulfilled at short notice, routing tries to include new requests in route plans that are currently executed. In the *execution* step, the fulfillment of accepted requests according to the incumbent route plan is carried out. The fulfillment is currently accomplished by drivers who are informed about their routes via mobile devices. In the future, automated fulfillment with fleets of autonomous vehicles is expected to improve the cost-efficiency of such services.

The three-step request fulfillment process involves two tasks of decision-making: acceptance and routing. Since requests arrive stochastically, decision support for acceptance and routing has to be made under uncertainty of future demand. In order to overcome the uncertainty, future requests can be anticipated. However, most contributions in dynamic routing do not anticipate future requests at all or establish anticipation for only one already challenging decision task, namely acceptance or routing. Focusing on either acceptance or routing implies that the anticipated information about the future demand is only partially explored, though. It is therefore unclear to what extent anticipation can contribute to increase the efficiency of the dynamic fleet management of ride-sharing services, especially how the value of information derived from anticipatory decision-making can contribute to the improvement of acceptance and routing decisions.

The aim of this paper is to evaluate the potential of anticipatory acceptance and/or routing for dynamic fleet management of ride-sharing services. To this end, we define different levels of anticipation – none, partial, or fully anticipatory –, review the related literature, and carry out a comprehensive computational study to analyze how the different levels of anticipation affect the performance metrics of a typical urban ride-sharing service as well as the service quality perceived by travelers. To this end, we model variants of the dynamic dial-a-ride problem (DDARP) representing acceptance and routing decisions faced by a ride-sharing service. The problem variants are solved by a large neighborhood search (LNS) under the objective of maximizing the acceptance rate defined by the number of incoming requests divided through the number accepted requests. The computational study investigates real trip data from the Yellow Caps operating in New York City, USA.

The paper is organized as follows. Section §2 provides a review of related literature on decision support for dynamic fleet management and anticipatory decision-making in the scope of the DDARP. In Section §3, the DDARP under consideration is presented and modeled as Markov decision process. Based on this problem description, Sec-

tion §4 differentiates the levels of anticipatory decision support. Section §5 covers the approaches to evaluate the potential of the different levels of anticipation as well as the presentation of the LNS. In Section §6, the computational experiments are presented including study design and computational results. Finally, Section §7 provides a conclusion and outlines future research directions.

2 | RELATED LITERATURE

In this section, we give an overview of the related research on the DDARP, in particular with regard to how acceptance and routing decisions are made. For a comprehensive literature review on the dial-a-ride problem in general we refer to Molenbruch et al. [2] and Ho et al. [3]. For an overview of the research on related dynamic vehicle routing problems (DVRP), see Psaraftis et al. [4] and Ritzinger et al. [5].

The first studies on fleet management of a ride-sharing service were conducted by Dial [6] and Madsen et al. [7] in 1995. Dial [6] decompose the problem into a set of travelling salesman problems, while Madsen et al. [7] suggest an insertion heuristic in order to solve the DDARP. These first contributions limit the problem to the routing decision since they assume that all incoming requests must be fulfilled. Routing is optimized re-actively after a new request has been received, and there is no anticipation of future demand. The algorithm proposed in Ma et al. [8] enables DDARPs to be solved for large-scale ride-sharing systems. The idea is to decompose the problem by means of a grid-based service area.

Many papers consider re-active acceptance and routing without anticipation of future demand. For this purpose, a static problem is solved and updated for each incoming request by applying well-known solution methods. Acceptance is made by means of a so-called feasibility check, which ensures that an incoming request can be integrated into the incumbent route plan. In case of acceptance follows a routing decision by re-optimizing the feasible route plan. Such a two-step procedure is proposed by Attanasio et al. [9] applying a parallel Tabu Search (TS) for both steps, by Coslovich et al. [10] through a two-stage insertion heuristic, by Beaudry et al. [11] through an insertion heuristic for the feasibility check and a TS for re-optimizing the route plan, and by Berbeglia et al. [12] proposing constraint programming for the feasibility check, extended in Berbeglia et al. [13] to the two-step procedure by a combination of TS and constraint programming. Constraint programming guarantees that a feasible solution can always be found, while the other heuristic approaches reject requests more likely if a feasible route plan is more difficult to be found.

There is only a small number of papers which consider anticipatory request acceptance in the DDARP. Corresponding policies accept requests if they are feasible and favourable with respect to the expected cumulative reward. The idea of actively rejecting unfavourable requests for a DDARP was first discussed in Horn [14], yet dismissed due to the potential unfairness towards requests with certain characteristics. Potential discrimination of particular requests is therefore one aspect that will be examined in our evaluation of the levels of anticipation. Further steps towards anticipatory acceptance were made by Xiang et al. [15] and Hosni et al. [16]. The policies proposed by these two papers include the pro-active rejection of unfavorable requests, yet without consideration of future demand. In both cases, the incremental costs caused by an incoming request are calculated in order to reject unprofitable requests, which are those whose costs exceed a certain threshold. Xiang et al. [15] implements acceptance by an insertion heuristic, and the subsequent routing decision through re-optimization by local search. Hosni et al. [16] introduces, in contrast, a model-based approach that integrates each incoming request into the incumbent route plan at minimal incremental costs.

More sophisticated policies that enable anticipatory acceptance are proposed for the common DVRP. For example, Azi et al. [17] introduce request acceptance based on a multiple-scenario approach for the planning of same-day

deliveries. They evaluate the favourability of a request based on a set of solutions generated through an adaptive large neighbourhood search (ALNS), taking into account pending and sampled expected future requests. A further example for an extensive anticipation of request acceptance is the value function approximation presented in Ulmer et al. [18] for the single vehicle case. In these examples, it is shown that anticipatory acceptance can be beneficial with regard to a DVRP; we will evaluate this in detail for DDARPs.

The counterpart to anticipatory acceptance is anticipatory routing. Anticipatory routing has often been considered already in DDARP solution approaches. A first idea was presented in Horn [14]. In this case, anticipation is limited to the relocation of idle vehicles to areas with an expected high future demand. More sophisticated anticipatory routing policies were introduced in Ichoua et al. [19], Schilde et al. [20] and Alonso-Mora et al. [21]. Here, anticipatory routing considers future demand through dummy requests. Ichoua et al. [19] generates an initial route plan on the basis of the dummy requests by means of a TS, which is updated with each newly accepted request. Request acceptance is carried out through a simple feasibility check. Based on a variable neighborhood search, Schilde et al. [20] adapt a multiple scenario approach originally proposed by Bent and van Hentenryck [22] for the DVRP. Furthermore, in Schilde et al. [20] it is assumed that all requests must be fulfilled; request acceptance is therefore not considered.

Alonso-Mora et al. [21] integrate acceptance and routing in a two-stage process, which was first introduced in Alonso-Mora et al. [23] and then extended by incorporating the expected future demand. In the first step, all feasible combinations of anticipated and unfulfilled requests are determined independently of the available fleet, resulting in a set of all possible routes. Based on this set, in the second step, an assignment problem is solved in order to match each vehicle with a route. Since not all requests can necessarily be covered, request acceptance is made indirectly by solving the allocation problem. Alonso-Mora et al. [21] present a case study for the DDVRP using real taxi trip data from Manhattan, which serves as inspiration for our case study. These articles demonstrate that the incorporation of future demand can increase the performance of the service under consideration. However, anticipatory routing depends on the implementation of request acceptance. By examining acceptance and routing decisions separately, we will provide an in-depth analysis of what characteristics justify what level of anticipation.

3 | PROBLEM FORMULATION

In this section, we define the components of the DDARP under consideration. Then, we model the stochastic and dynamic problem of request acceptance and routing as Markov decision process.

3.1 | Problem components

Let $\mathcal{L}$ be a set of locations in the service area of a ride-sharing service. For each location $l \in \mathcal{L}$, it is assumed that a (deterministic) service time p_r for boarding or alighting of travelers 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 service 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 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 on the way to the destination and the service time required by them for boarding or alighting. In order 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.

3.2 | Markov decision process

The considered decision process consists of a series of decision epochs $k \in \mathcal{K}$, covering a temporally limited planning horizon of a DDARP. At the beginning of the planning horizon, 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, a degree of dynamism of one is considered, so that in the initial state s_0 no trips are pending for fulfillment. 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 vehicle 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, O_k^v), where $l_k^v \in \mathcal{L}$ specifies the current vehicle locations and $O_k^v \subset \mathcal{R}$ the set of accepted requests currently being executed for each vehicle. 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 pending for fulfillment. These three parts result in the state definition $s_k = (t_r, (l_k^v, O_k^v), (r_k, \mathcal{U}_k))$.

Based on the pre-decision state s_k, an action $A^\pi(s_k)$ is performed by a policy $\pi \in \Pi$. Each 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. Within the decision-making process, the acceptance decision x_k therefore controls the demands to be fulfilled. The second decision is the selection of a feasible route plan. Routing controls the vehicle resources in order to fulfill the requested trips efficiently. 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 $r \in O_k^v$, the drop-off at destination d_r is planned for unchanged 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. 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 the stochastic transition W_{k+1}, which reflects that the operator has received the next request $r_{k+1} \in \mathcal{R}$.

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 $k \in \mathcal{K}$. Let B_k be the partial reward 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 acceptance rate defined by the number of received requests divided by the number of accepted and thus fulfilled requests.

4 | LEVELS OF ANTICIPATION

The key challenge of the formulated problem is the uncertainty of future demand during the decision epochs $k \in \mathcal{K}$ caused by the stochastic nature of the requests $r \in \mathcal{R}$. This uncertainty means that the decisions are made based on incomplete information and thus the cumulative reward $v^{\pi}(s_0)$ is uncertain, too. In order to handle the uncertainty, future demands and their implications on the decision-making process can be anticipated. We define anticipation in the context of decision-making processes as the consideration of future stochasticity (e.g. via historical data or forecasts) in order to maximize the expected cumulative reward. In contrast, decision making that maximizes partial rewards based on confirmed information only is referred to as *myopic*. In the following, we will discuss different levels of anticipation in the context of the previously formulated problem to identify their inherent potential for dynamic fleet management. The distinction between anticipatory and myopic leads to the four levels of anticipation shown in Table 1.

With respect to the given problem, each decision epoch $k \in \mathcal{K}$ includes the request acceptance decision x_k and the routing decision y_k. We can make both decisions individually in an anticipatory or myopic way. For request acceptance x_k, in case of a myopic decision, a request is simply accepted when there is a vehicle available, since this will increase the immediate reward by B_k. In this case, acceptance only depends on the routing decisions made in the previous epochs. In the literature, this is known as *feasibility check*, which determines whether a feasible route plan can be found. This feasibility check is as well the basis for an anticipatory acceptance decision, but here, request acceptance is more complex since it comprises the proactive rejection of a current request in favor of potential future ones. This kind of rejection occurs when the expected resource savings result in a higher number of accepted requests over the planning horizon, maximizing the expected cumulative reward $v^{\pi}(S_0)$.

Regarding the routing decision y_k, the myopic variant corresponds to the selection of the route plan with the minimum resource utilization, i.e. the solution of a DARP associated with the pre-decision state s_k. The anticipatory variant implies that future demand will be considered in routing. However, the extent to which future demand is taken into account may differ significantly: this can range from anticipatory relocation of idle vehicles (e.g. Pureza and Laporte [24]) to sophisticated anticipatory routing that actively incorporates future demands into the fulfillment of requests (e.g. Ferrucci et al. [25]).

TABLE 1 Levels of anticipation

		Acceptance decision	
		Request acceptance if feasible	Request acceptance if feasible & favorable
Routing decision	Routing considers accepted requests	*None Anticipatory*	*Anticipatory Acceptance*
	Routing considers accepted & future requests	*Anticipatory Routing*	*Fully Anticipatory*

Each policy $\pi \in \Pi$ can be assigned to one of four anticipation levels presented in Table 1. *None Anticipatory* includes all policies that do not support anticipation in any form. Instead, both acceptance and routing decisions are made in a myopic manner. Reasons for the deployment of such a policy could be an unpredictable environment or insufficient amount of historical data. Moreover, the higher technical and computational effort for anticipatory approaches could favor the use of purely myopic policies. However, the main argument against such policies is the

risk of very unfortunate decision-making caused by insufficient information.

Anticipatory Acceptance and *Anticipatory Routing* address policies that anticipate only with respect to one decision. In particular, a policy of the level *Anticipatory Acceptance* restricts anticipation to the acceptance decision x_k. These policies improve acceptance decisions by assessing the long-term value of a request under consideration of future demand. By focusing on the less complex decision x_k, this kind of policy reduces the challenge of anticipatory decision making. Nevertheless, these policies require reliable information on future demand for a successful implementation. The focus of *Anticipatory Acceptance* policies is the acceptance of the most favorable requests that require a relatively small use of resources (e.g. requests which can be bundled more easily). However, every anticipatory policy carries the risk of unfortunate anticipatory decisions. For example, if the future demand is overestimated, too many requests may be rejected, so that the given vehicle resources are not fully utilized in the end. Moreover, anticipatory acceptance can bear drawbacks in terms of business considerations. For example, incomprehensible rejections as well as rejections perceived as proactive may lead to a dissatisfaction of travelers. Furthermore, the continuous rejection of certain trip requests identified as unfavorable may prevent such trips from being requested, irrespectively of whether the assessment might change over time.

In contrast, *Anticipatory Routing* limits anticipation to the routing decision y_k. Policies of this kind generally focus on the efficient current and future utilization of the fleet through the consideration of future demand. There are simple and more comprehensive policies according to requirements and capabilities of the service under investigation. In general, *Anticipatory Routing* has to deal with the complexity of the routing decision, which is already challenging without anticipation, esp. for large problem instances. In order to enable anticipatory routing during the fulfillment of requests, a policy requires precise temporal and spatial information on future demand. For instance, idle vehicles can be relocated in favor of future demand and anticipated requests can be considered in order to bundle more efficiently. However, misguided anticipation at this level can lead to an inefficient utilization of resources, e.g. by allocating vehicles to anticipated requests that never realize. From a business perspective, this may cause dissatisfaction among travelers through unnecessary detours or longer waiting times.

Finally, *Fully Anticipatory* includes policies that make anticipatory decisions with respect to both, acceptance and routing. While being the most computationally challenging techniques, they naturally show the largest potential of dynamic fleet management.

5 | EVALUATION FRAMEWORK

This section describes our framework to evaluate the impact of the different levels of anticipation in dynamic fleet management of ride-sharing services. We first discuss how request acceptance and routing decisions are made to enable the evaluation of the anticipation levels and address then how we employ an established LNS to conduct the computational evaluation.

5.1 | Evaluation approaches

The four presented levels of anticipation are evaluated by solving variants of the DDARP that differ in the level of information available for the acceptance decision x_k as well as the routing decision y_k. In the following, the problem variants as well as the policies applied are discussed for each level.

***None Anticipatory*:** For this level, as a benchmark, the presented problem is solved in a myopic manner. In particular, the corresponding policy solves the DDARP through a feasibility check and re-optimization following ideas of the

none-anticipatory approaches presented in Section 2. 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. If the insertion is successful, the request will be accepted. The route plan obtained by a successful check is then re-optimized in the scope of the routing decision y_k. For this purpose, a static DARP is solved for all accepted requests under the objective of minimizing the total travel time. Note that for both acceptance and routing decisions already fulfilled requests as well as locations approached by a vehicle cannot be rescheduled. This means that vehicles are not diverted on their way to a location $l \in \mathcal{L}$, which decreases flexibility of planning, but also computational effort. Moreover, it allows drivers and travelers to be reliably informed about the next stop, avoiding frequent diversions of vehicles.

Anticipatory Acceptance: In order to evaluate the potential of *Anticipatory Acceptance*, it is assumed that complete information on future demand is available for the acceptance decision x_k. This decision 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 *none anticipatory*). 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 profitable locations are visited. The objective is to find the optimal set of visited locations which maximizes the operator's benefit [26]. As input for the TOP serve all requests of the incumbent route plan y_{k-1}, the current request r_k as well as all expected future requests. All requests have equal scores, representing that the route plan found maximizes the number of scheduled requests. All requests of the incumbent route plan y_{k-1} must be covered, and the TOP identifies favorable requests among the current and all 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 the idea of *None Anticipatory*.

Anticipatory Routing: Here, the decisions on request acceptance x_k are made dynamically by carrying out a feasibility check for each incoming request $r_k \in \mathcal{R}$ through an insertion heuristic, similar to the procedure of *None Anticipatory*. However, it is assumed that all request related time windows $[b_r, e_r]$ refer to the fulfillment on a subsequent day, so that an incumbent route plan y_{k-1} can still be rescheduled flexibly. This enables a dynamic acceptance decision x_k without information on future demand combined with a routing decision y_k with complete information regarding all requests to be fulfilled. After a successful feasibility check, no further re-optimization of the route plan is performed, yet a final routing decision y_k is investigated once all decisions on request acceptance have been made. The finale route plan to be executed is then determined for the set of accepted requests by solving the resulting static DARP.

Fully Anticipatory: We assume that perfect information on future demand is given, allowing all dynamic decisions to be made in advance. For this purpose, the problem is solved as 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 covered requests such that the requests to be accepted and the routes to be taken can be optimized accordingly.

5.2 | Large Neighborhood Search

In the following, we describe the LNS applied for the evaluation of the impact of the different anticipation levels. We apply the same heuristic for all levels and decisions to ensure the comparability of the computational experiments. The developed LNS is based on the ALNS proposed by Ropke and Pisinger [27]. 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 short run times, which is important especially for request acceptance.

5.2.1 | Overview

The basic idea of a LNS in general is to destroy and repair solutions iteratively [28]. For the problem at hand, a solution w is represented by a route plan n_w and a set of unplanned trip 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 planned request fulfillments $|n_w|$ and/or to minimize the required total travel time $c(n_w)$.

```
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) then
11                 w_best = w_new
12             end
13         end
14     end
15     return w_best
```

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 better solution is performed until a termination criterion is met. As termination criterion, a 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 (line 5 to 7). For this purpose, the incumbent solution w is saved as 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 this 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 with $\{q \in \mathbb{N} \mid (\gamma_1 \times |n_{w_{new}}|) \leq q \leq (\gamma_2 \times |n_{w_{new}}|)\}$. 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, 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. Removal operators correspond to those used in Ropke and Pisinger [27]. 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 and Pisinger [27], we use the formula $q_1^{\delta} \times |list|$ to determine the list position of the next request to be removed. In this formula, q stands for a random value with $\{q \in \mathbb{Q} | 0 \leq q \leq 1\}$ and δ_1 for the parameter that controls the degree of noise.

Shaw-Removal: Originally introduced by Shaw [29], 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 $\Delta(z_{a_{r_1}}, z_{a_{r_2}}) + \Delta(z_{d_{r_2}}, z_{d_{r_2}})$. Before the geographically and temporally 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 only discuss the most promising Regret-2 operators, one with and one without noise.

Regret-2-Insertion: The Regret-Insertion heuristic was first proposed by Potvin and Rousseau [30] for the vehicle routing problem with time windows. The idea is to insert requests at the position where the regret would be greatest if the best found insertion position was no longer feasible. The regret is calculated for the Regret-2 variant by the difference between the most and the second most cost-effective feasible insertion position. The costs correspond to the additional travel time which would result if the request was inserted in the position of the route plan. In case that only one feasible insertion position 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 accordingly 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 compared to the incumbent solution w (see line 8). Since mostly fully utilized services are investigated which often shiw limited routing flexibility, this acceptance criterion has the advantage of allowing a maximum diversification with respect to the overall travel time and prevents a deterioration of the number of planned requests. After accepting and saving s_{new} as incumbent solution s, it is checked whether it is an improvement compared to the best known solution s_{best} (line 10 to 12). This is the case if the number of planned requests $|n_{w_{new}}|$ is increased or remains equal by a decreased 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).

5.2.2 | Implementation

In the following, we briefly describe how the LNS is implemented according to the different levels of anticipation described in Section 5.1.

None Anticipatory: Here, the LNS is applied as an insertion heuristic for the feasibility check of the acceptance decision and for re-optimization of the routing decision. For the insertion heuristic, the initial solution w_0 is provided with the set of unplanned requests m_{w_0} including the new request $r_k \in \mathcal{R}$. The route plan n_{w_0} only covers locations $l \in \mathcal{L}$ whose planned arrival times z_l^v plus service time p_l are greater or equal to the time of request t_{r_k}. The first

location of each plan contained in n_{w_0} thus represents the current respectively next location of a vehicle and cannot be rescheduled. Based on this input, the LNS searches for a solution w_{new} where all requests are inserted in the route plan $n_{w_{new}}$. The search is terminated when either such a solution could be found or a maximum of β iterations has been performed. Note that in case of an unsuccessful feasibility check, the returned solution is discarded, while the initial solution w_0 is reused as initial solution for the feasibility check of the next request r_{k+1}. In case of a successful feasibility check, the found solution is used as initial solution for the re-optimization performed by the LNS under the objective of minimizing the total travel time in β iterations.

Anticipatory Acceptance: This level requires as well a feasibility check for the acceptance decision and a re-optimization for the routing decision, and generally follows the ideas of *None Anticipatory*. However, for solving the TOP in the additional favourability check of the acceptance decision, the initial solution w_0 consists of the same route plan as in case of the feasibility check. Then, 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. Based on this input, the LNS maximizes 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 planned 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} where all in the search considered requests could be inserted in the route plan 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 it has passed the favourability check.

Anticipatory Routing: In this case, the LNS is primarily used to solve the feasibility check for the acceptance decisions. The feasibility check is initialized with a solution w_0 that consists of an empty route plan n_{w_0} or a route plan that results from the last successful feasibility check and a set of unplanned requests m_{w_0} including the new request $r_k \in \mathcal{R}$. After completion of all feasibility checks, the LNS is applied in the final routing decision to minimize the travel time of the solution returned by the last successful feasibility check.

Fully Anticipatory: Here, the LNS is applied to solve the TOP. 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 trip 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 $c(n_w)$.

6 | COMPUTATIONAL EVALUATION

In this section, we analyze the potential of anticipation for the quality of service and the performance of ride-sharing services. We introduce our instances, justify the parameterization of the LNS, and present the results of the computational study.

6.1 | Experimental design

The potential of different anticipation levels is evaluated in a case study 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 Cap taxi fleet in the year 2014 [31]. 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 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 (between 17:30 and 20:30) in the area of Manhattan. Furthermore, only trips with a distance greater than zero are considered.

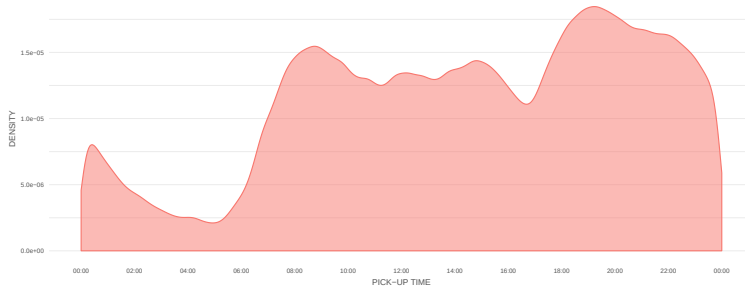


FIGURE 1 Pick-up time distribution

Given the taxi trip data, we derive the characteristics for our ride-sharing service as follows. First, potential initial vehicle locations have to be 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 have to be 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 in order to enable trip-specific evaluations across all anticipation levels. All selected locations are visualized in Figure 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 GraphHopper [32]. Free flow travel times are varied by a factor ϵ to account for longer travel times during congestion.

(a) Initial vehicle locations **(b)** Origins **(c)** Destinations

FIGURE 2 Location distributions

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 horizon from 17:30 to 20:30 (180 minutes), a travel time time factor $\epsilon = 3$, and a maximum arrival delay for each request of

15 minutes.

In our analysis, we are interested in the following variants of the baseline scenario. First, we vary the fleet size to analyze different levels of demand coverage. Second, we analyze the impact of temporally varying demand density. To this end, the length of the planning horizon 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 horizon. 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.

TABLE 2 Values for the sensitivity analysis

Sensitivity analysis	Varying characteristic	Values				
Demand Coverage	Fleet size	2	6	**10**	14	18
Temporal Demand Density	Planning horizon	36 min	108 min	**180 min**	252 min	324 min
Geographical Demand Density	Factor on travel time	0.6	1.8	**3**	4.2	5.4
Fulfillment Time Window	Maximum arrival delay	3 min	9 min	**15 min**	21 min	27 min

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 2). With these parameter intervals, we can cover a wide range of the possible objective function values and at the same time create deep insights into where and when what level of anticipation is beneficial.

6.2 | Parameter tuning

The parameter tuning of the LNS is based on the *Demand Coverage* sensitivity analysis. This represents a compromise between parameter tuning for a particular scenario and all scenarios. The 10 instances generated for parameter tuning are solved five times, each time with an adapted fleet size. For insertion and re-optimization in the scope of *None Anticipatory* and *Anticipatory Acceptance* the tuning of the parameters is based on *None Anticipatory*. For *Anticipatory Routing*, a separated tuning is performed, since considerably more requests have to be handled during an insertion and the final re-optimization due to the postponed fulfillment. Regarding the TOP, the parameter tuning is based on *Fully Anticipatory*. The resulting values are mostly applied to solve the TOP as favorability check within *Anticipatory Acceptance*. However, the number of required iterations β and thus the computational effort is determined separately.

The number of iterations as termination criterion has a particularly 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 by rounding up to the next number divisible by 100 resp. 1000. The results of this procedure are summarized in Table 3. At the beginning, the percentage of trips removed per iteration is set to $\gamma_1 = 0.3$ and $\gamma_2 = 0.4$ following Ropke and Pisinger [27], and the noise for the operators is set to a medium level of $\delta_1 = 4$ and $\delta_2 = 4$.

From the table 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 *Anticipatory Acceptance*. 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

TABLE 3 Number of iterations β

Anticipation Level	Case	Final β	Test β	⌀ Last successful iteration per fleet size				
				2	6	10	14	18
None Anticipatory	Insertion	100	1000	1	2	5	5	**11**
	Re-optimization	200	1000	0	1	8	24	**126**
Anticipatory Acceptance	TOP	3000	3000	2895	2995	2995	2996	**2999**
Anticipatory Routing	Insertion	1000	2000	260	531	728	**910**	719
	Re-optimization	10000	10000	3	1245	6283	9713	**9778**
Fully Anticipatory	TOP	30000	40000	4770	15195	15288	**27235**	15306

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 *None Anticipatory* and *Anticipatory Acceptance*, $\gamma_1 = 0.7, \gamma_2 = 0.8$ is applied. For both TOP as well as the insertion and re-optimization of *Anticipatory Routing*, 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 ($\delta_1 | \delta_2 = 0$), medium noise ($\delta_1 | \delta_2 = 4$) and a high degree of noise ($\delta_1 | \delta_2 = 8$) 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 ($\delta_1 | \delta_2 = 4$), this value is selected for the further experiments. The detailed results of the tuning of γ_1, γ_2, δ_1 and δ_2 are reported in the appendix.

6.3 | Computational results

First, we analyze acceptance rates for different levels of anticipation. Then, further metrics that describe the operational performance of the ride-sharing service are discussed. This provides insights into the nature of such services and contributes to a better understanding of the context-related effects of anticipatory decision-making. Last, we investigate the effect of the anticipation levels on the service quality perceived by travelers through a detailed trip-specific evaluation. We will discuss the results of all four sensitivity analysis presented in Table 2 with regard to their impact on acceptance rates. For further investigations, we focus on *Demand Coverage* and leave detailed results for *Temporal Demand Density*, *Geographical Demand Density* and *Fulfillment Time Window* for the appendix, since the findings are structurally similar.

6.3.1 | Acceptance rates

We begin with analyzing the potential of anticipation with respect to achievable acceptance rates. We particularly analyze the value of information on future demand in acceptance and routing decisions. Results are presented in

Figure 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 anticipation levels. The different levels of anticipation are separated by color. The points represent the numeric results and the trend is highlighted by connecting lines. Additionally, the standard deviations of the acceptance rates are illustrated by a lighter color range around the lines.

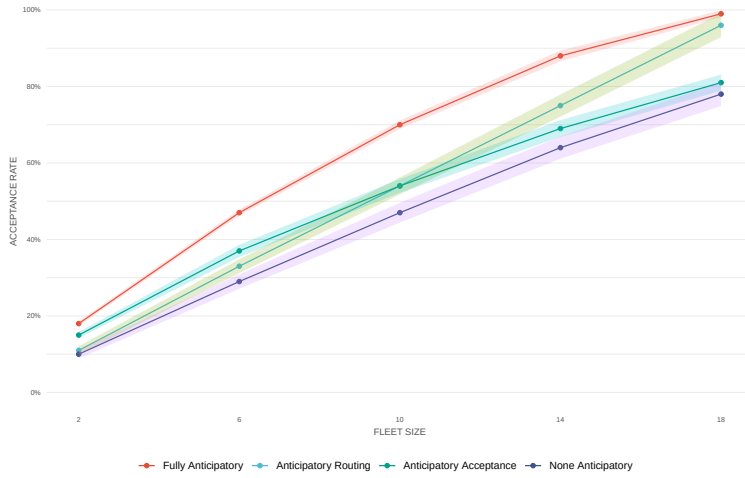


FIGURE 3 *Demand Coverage*: Acceptance rate with standard deviation

Generally, with increasing fleet size, achievable acceptance rates increase as well. As expected, *None Anticipatory* leads to the smallest acceptance rates, while *Fully Anticipatory* creates the best acceptance rates with a gap of about 10 – 20%. Hence, there is significant potential for anticipation. Interestingly, for smaller fleet sizes, *Anticipatory Acceptance* yields better results, while for larger fleet sizes, *Anticipatory Routing* can create significantly higher acceptance rates. The standard deviations increase with increasing fleet sizes. They are negligible for *Fully Anticipatory*.

We now analyze the results of further sensitivity analyses (see Figure 4). We begin with *(a) Temporal Demand Density*, where we manipulate the demand through temporal variation of the booking period. Generally, results are similar to those obtained for the *Demand Coverage* analysis. For the same fleet size, a relatively larger booking period allows to accommodate more requests, with a high potential of anticipatory routing for a large temporal spread of requests and a high potential of anticipatory acceptance for a small temporal spread of requests. For *(b) Geographical Demand Density*, instead of the time of the booking 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 *Fully Anticipatory* is about 20% higher than for *None Anticipatory*. Anticipation via either routing or acceptance can improve this by about 5% only. Here, a high geographical density diminishes the benefits of anticipatory acceptance and increases those of anticipatory routing. However, when the geographical density decreases, unfavorable requests from remote regions may automatically be infeasible to fulfill.

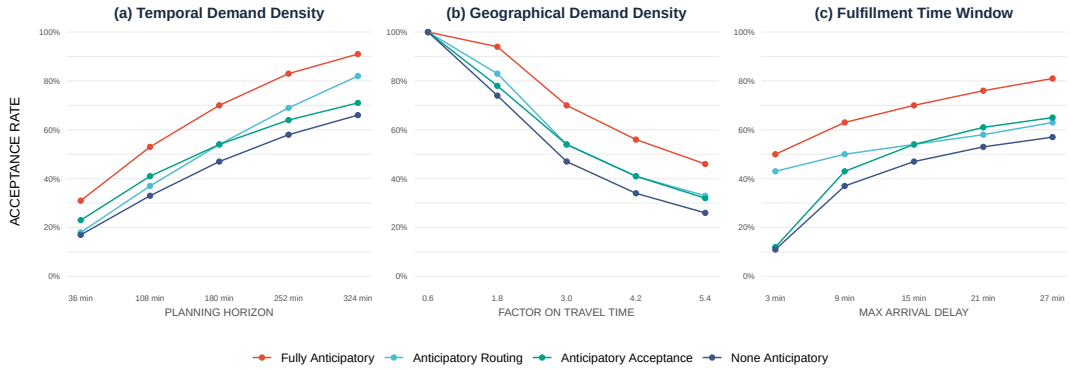


FIGURE 4 Acceptance rates per sensitivity analysis

Finally, we analyze for *(c) Fulfillment Time Window* how a variation of the maximum delay, consisting of waiting time and detour, effects the potential of the anticipation. As expected, acceptance rates increase for all anticipation levels with an increasing maximum delay. However, the gap between *None Anticipatory* and *Full Anticipatory* is very large for small maximum delays. In contrast, *Anticipatory Routing* yields quite stable results for all maximum arrival delays. The value of information about future demand is higher for acceptance when the maximum delay is higher and for routing when the maximum delay value is lower.

Above findings demonstrate that anticipatory decision-making has great potential to increase the acceptance rate of dynamic ride-sharing services. It is clear that the potential of both, anticipatory acceptance and routing, evolve differently in response to modification of the service under consideration. The value of information on future demand is particularly high for acceptance decisions, 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 the routing decision, the fulfillment time window analysis shows the importance of information on future demand for tight fulfillment time windows, 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 *Anticipatory Routing* is barely achievable. Overall, the results imply that the potential of policies from *Anticipatory Acceptance* and *Anticipatory Routing* vary greatly depending on the nature of the ride-sharing service.

6.3.2 | Operational performance

The aim of this subsection is to gain further insights into how anticipation impacts further performance metrics of a ride-sharing service. 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 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. *Boarding and Alighting Time:* Time required for pick-up or drop-off of travelers,
    5. *Idle Time:* Time a vehicle waits at a location for a traveler or the next assigned request.

The first metric examined is the average travel time per fulfilled request in minutes, plotted in Figure 5 against the varying fleet size. *None Anticipatory* creates constantly high average travel times per request even with increasing fleet size. Again, *Fully Anticipatory* is the counterpart, with travel time savings of 3 to 10 minutes on average, highlighting the potential of anticipation for ride-sharing services. *Anticipatory Acceptance* works almost as well as *Fully Anticipatory*; only for the largest fleet size, *Anticipatory Routing* becomes more efficient. Hence, the reduction of the average travel time per fulfillment is mainly rooted in anticipatory acceptance decisions.

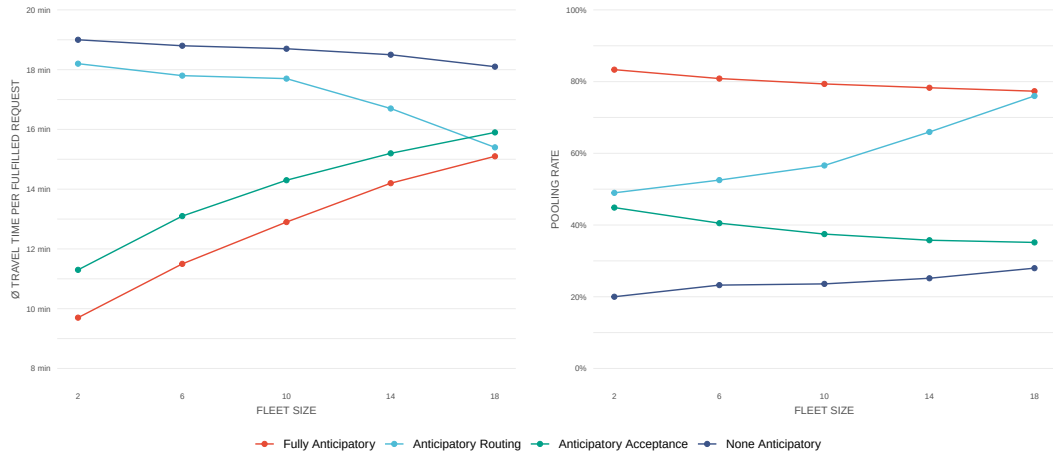


FIGURE 5 *Demand Coverage*: Average travel time per fulfilled request and pooling rate

The second metric of interest is the pooling rate shown in Figure 5. *None Anticipatory* and *Fully Anticipatory* define lower and upper bounds with a gap of 60%. *Anticipatory Routing* is the key for a good pooling rate; with increasing fleet size, it almost becomes as effective as *Fully Anticipatory*. However, if the fleet size is small, there is a similarly high potential for improving the pooling rate through anticipatory acceptance.

So far, we have seen that the effectiveness of anticipatory acceptance and anticipatory routing can vary quite a bit. *Anticipatory Acceptance* tends to achieve a reduced average travel time per fulfillment by accepting a set of favorable requests, while *Anticipatory Routing* 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 different levels of anticipation (see Figure 6). For all levels of anticipation and fleet sizes, a rather stable proportion of *Unoccupied Travel Time* as well as the relatively large share of *single travel time* is obvious. Interesting differences can be observed with respect to

the *Shared Travel Time* and *Idle Time*. For *Shared Travel Time*, again, *Anticipatory Routing* is the key. Interestingly, even with *Fully Anticipatory*, 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 *None Anticipatory*. Major differences are also apparent for the *Idle Time*. Especially for *Anticipatory Routing* and *Fully Anticipatory*, lower idle times can be observed. The lower idle times for anticipatory routing are rooted in proactively relocations in favor of future demands. In contrast, the share of the idle times is highest for *Anticipatory Acceptance*. Here, the higher idle times arise as anticipatory acceptance may have vehicles wait longer in idle mode instead of accepting an unfavorable request. Overall, these results show different strategies regarding the handling of idle time, whereby the combination of both strategies appears to be the most promising.

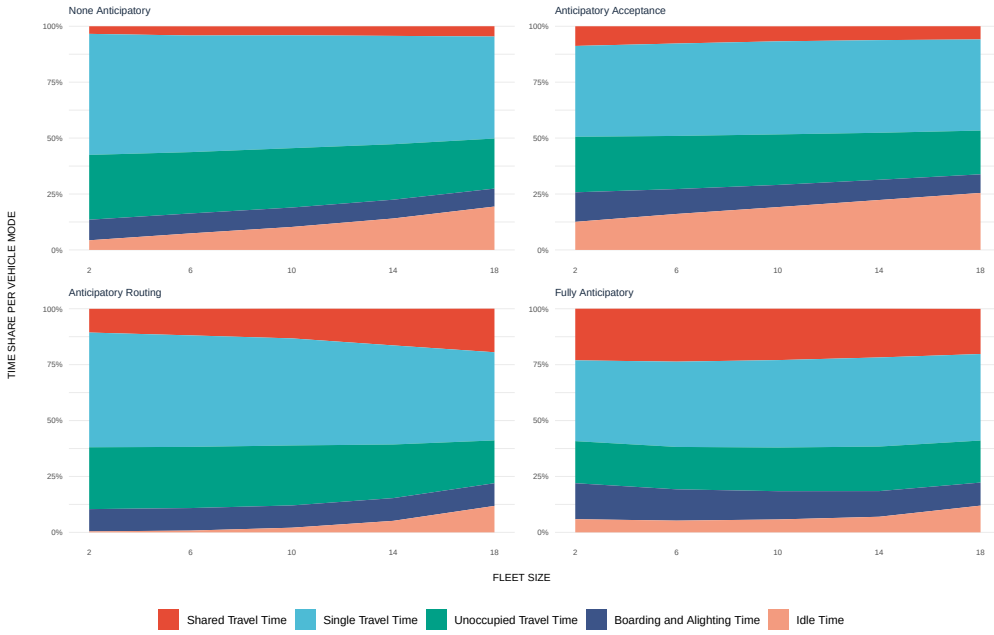


FIGURE 6 *Demand Coverage*: Time share per vehicle mode

6.3.3 | Quality of service per trip

Now we examine the impact of anticipatory decision-making on the quality of service experienced by travelers. Service quality metrics are derived for each of the trips and summarized per anticipation level. The first step is to investigate whether different service quality levels can be observed and if the trip-specific quality of service varies per anticipation level. We analyze the following metrics:

- The acceptance probability for each trip, represented by the number of times the trip is requested per 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.

The results are shown in Figure 7 by means of density plots. With regard to acceptance probability, there are clear differences in the distributions. For *None Anticipatory* and *Anticipatory Routing* the diversification is relatively low, with a high density at about 50%. Distributions for *Anticipatory Acceptance* and *Fully Anticipatory* are very flat. This shows 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 relatively minor influence on whether it is feasible to accept a trip.

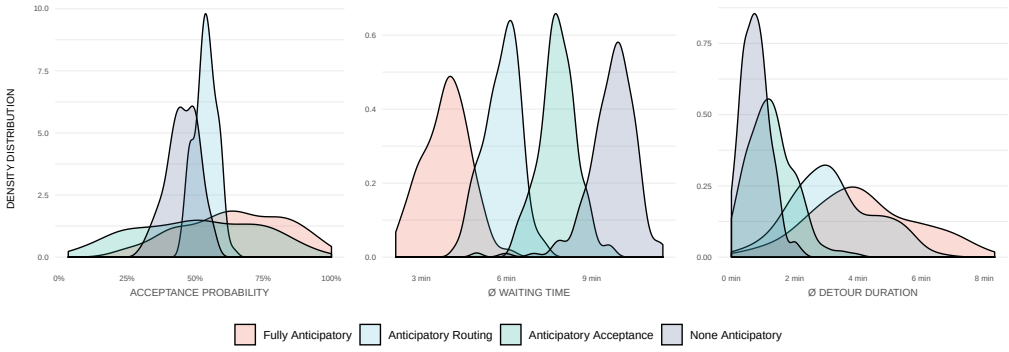
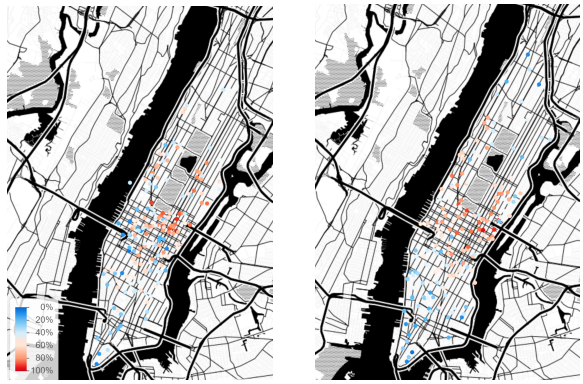


FIGURE 7 Quality of service per trip

The average waiting time for each trip is relatively constant per anticipation level, but the levels differ quite a bit from each other. *Anticipatory Routing* shows again its higher potential to transport travelers at shorter notice. For *Anticipatory Acceptance*, probably the rejection of requests with longer detours leads to shorter average waiting times.

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 *Anticipatory Routing* and *Fully Anticipatory*, 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 jointly limitation of both via the maximum delay parameter. The shorter waiting times achieved by anticipatory decision-making 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. [33] 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 *Anticipatory Acceptance*, Figure 8 illustrates this correlation separately for origin and destination of all trips, using a color scale that reflects the the acceptance probability. The blue dots indicate trips with a very low acceptance probability and red ones those with a very high acceptance probability. A preference for the regional center and the discrimination of upper and lower periphery is evident, indicating that, for anticipatory acceptance, there is a positive correlation between the acceptance probability of a trip and the geographical centrality of its origin and destination. In contrast, the analysis of average detour duration for *Anticipatory Acceptance* and *Anticipatory Routing* did not reveal any recognizable discrimination patterns, though.



(a) Trip origins **(b)** Trip destinations

FIGURE 8 *Anticipatory Acceptance:* Acceptance probability depending on the locations of the trips

As a further characteristic, we examine the trip distance in the light of acceptance probability and detour duration. Results are shown in Figure 9. It becomes evident that there is a distinct negative correlation in case of *Anticipatory Acceptance*. Implicitly, anticipatory acceptance utilizes the trip distance as a further criterion to assess trips. 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 *Anticipatory Routing*. Hence, anticipatory routing penalizes long-distance trips, yet in a way that limits the usability of the ride-sharing service for such trips not as strict as *Anticipatory Acceptance* does.

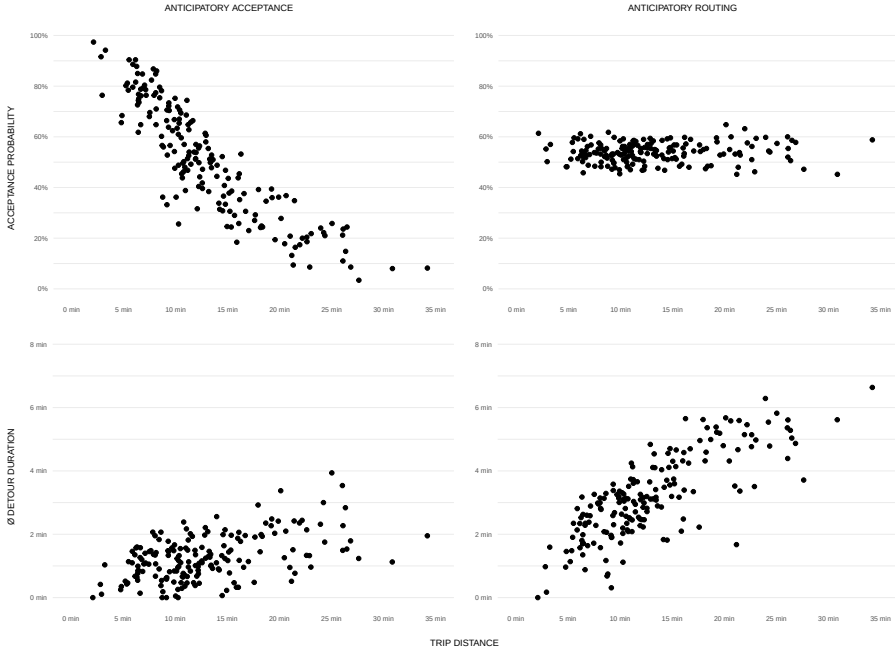


FIGURE 9 Acceptance probability and detour duration depending on the trip distances

In summary, anticipatory acceptance and routing have very different impact on the quality of ride-sharing services as experienced by travelers. For anticipatory acceptance, the quality depends significantly on the nature of the requested trip. A ride-sharing service 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 other characteristics are likely to switch to other mobility services. In contrast, for anticipatory routing, 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 may seem fair to the traveler. Finally, it should be noted that a fully anticipatory policy would not only incorporate the potentials as shown in the previous sections but also the unequal quality of service depending on the characteristics of the trip.

7 | CONCLUSION

In our paper, we investigated the value of anticipatory decision-making for dynamic fleet management of ride-sharing services. To this end, we defined four levels of anticipation, – none, anticipatory acceptance, anticipatory routing, and fully anticipatory –, which differ by how they consider information on future demand in the acceptance and/or routing decisions. The evaluation of these levels was accomplished in a comprehensive computational study. The results were analysed with regard to the impact of the anticipation levels from an operator's perspective as well as with their consequences for travelers. Overall, our results demonstrated a great potential for anticipatory decision-making in dynamic fleet management. Potential benefits range from increased acceptance and pooling rates to decreased travel and idle times. The advantages of anticipatory decision-making for service operators, however, go hand in hand with a varying quality of service perceived by travelers. In particular, acceptance probability and detour duration depend considerably on the nature of the requested trip.

A particular contribution of our paper is the differentiation of anticipation levels according to anticipatory routing and acceptance. This created insights about whether the value of information on future demand can be attributed to either acceptance or routing decisions or a reasonable combination of them. This is important since computational effort differs a lot for corresponding policies. For request acceptance, anticipation is especially beneficial if there is a sufficient surplus of demand. Anticipatory acceptance works well when there is a decent subset of favorable requests that can be selected from a larger pool of feasible requests. Furthermore, anticipatory acceptance 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.

Anticipatory routing is primarily associated with the acceptance rate and the promised fulfillment time window. Due to consideration of expected requests, anticipatory routing shows a rather stable performance despite increasingly narrow fulfillment time windows. However, the ability to anticipate unfavorable future requests can only be beneficial if the acceptance decision has a minor impact. Therefore, the value of information on future demand increases with an increasing acceptance rate. In particular, the performance improvement through anticipatory routing 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 services an orientation on which level of anticipation decision-making could be implemented. For instance, anticipatory acceptance could be more suitable for large services or services with 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 at special occasions. Anticipatory routing would be particu-

larly suitable for services that undergo few changes and rely instead on the transport of a rather fixed base of travelers. Besides, it can be limited to less ambitious anticipatory policies, e.g. the relocation of idle vehicles. Furthermore, we contribute to research on dynamic fleet management, in particular with regard to the DDARP, by providing a more differentiated view on acceptance and routing decisions and their implications towards anticipatory decision-making. We believe that this can be the basis for the development and benchmarking of new anticipatory approaches that incorporate a comprehensive view of acceptance and routing decisions.

In the future, a comprehensive classification of the anticipatory approaches known for the DVRP could provide a better understanding of what types of anticipation are reasonable for dynamic fleet management. This would require a further differentiation of the presented levels, for example, by determining whether information on future demand should actively be incorporated by stochastic and dynamic solution approaches. Furthermore, in our study, only theoretical potentials were evaluated. An intuitive next step would be to investigate the realizability of the identified potentials on the basis of selected state-of-the-art approaches implementing anticipatory decisions for acceptance and/or routing. A first step in this direction could be the development of a fully anticipatory approach for the large-scale dynamic fleet management of ride-sharing services.

Acknowledgements

This research was supported by a grant from the German Federal Ministry of Transport and Digital Infrastructure (BMVI, Grant No. 16AVF2147E).

references

[1] UberPool; n.d. https://www.uber.com/de/de/ride/uberpool/.

[2] Molenbruch Y, Braekers K, Caris A. Typology and literature review for dial-a-ride problems. Annals of Operations Research 2017;259(1-2):295–325.

[3] Ho SC, Szeto WY, Kuo YH, Leung JMY, Petering M, Tou TWH. A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological 2018;111:395–421.

[4] Psaraftis HN, Wen M, Kontovas CA. Dynamic vehicle routing problems: Three decades and counting. Networks 2016;67(1):3–31.

[5] Ritzinger U, Puchinger J, Hartl RF. A survey on dynamic and stochastic vehicle routing problems. International Journal of Production Research 2016;54(1):215–231.

[6] Dial RB. Autonomous dial-a-ride transit introductory overview. Transportation Research Part C: Emerging Technologies 1995;3(5):261–275.

[7] Madsen OBG, Ravn HF, Rygaard JM. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Annals of Operations Research 1995;60(1):193–208.

[8] Ma S, Zheng Y, Wolfson O. T-share: A large-scale dynamic taxi ridesharing service. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE) IEEE; 4/8/2013 - 4/12/2013. p. 410–421.

[9] Attanasio A, Cordeau JF, Ghiani G, Laporte G. Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing 2004;30(3):377–387.

[10] Coslovich L, Pesenti R, Ukovich W. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European Journal of Operational Research 2006;175(3):1605–1615.

[11] Beaudry A, Laporte G, Melo T, Nickel S. Dynamic transportation of patients in hospitals. OR Spectrum 2010;32(1):77–107.

[12] Berbeglia G, Pesant G, Rousseau LM. Checking the Feasibility of Dial-a-Ride Instances Using Constraint Programming. Transportation Science 2011;45(3):399–412.

[13] Berbeglia G, Cordeau JF, Laporte G. A Hybrid Tabu Search and Constraint Programming Algorithm for the Dynamic Dial-a-Ride Problem. INFORMS Journal on Computing 2012;24(3):343–355.

[14] Horn MET. Fleet scheduling and dispatching for demand-responsive passenger services. Transportation Research Part C: Emerging Technologies 2002;10(1):35–63.

[15] Xiang Z, Chu C, Chen H. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. European Journal of Operational Research 2008;185(2):534–551.

[16] Hosni H, Naoum-Sawaya J, Artail H. The shared-taxi problem: Formulation and solution methods. Transportation Research Part B: Methodological 2014;70:303–318.

[17] Azi N, Gendreau M, Potvin JY. A dynamic vehicle routing problem with multiple delivery routes. Annals of Operations Research 2012;199(1):103–112.

[18] Ulmer MW, Goodson JC, Mattfeld DC, Hennig M. Offline–Online Approximate Dynamic Programming for Dynamic Vehicle Routing with Stochastic Requests. Transportation Science 2019;53(1):185–202.

[19] Ichoua S, Gendreau M, Potvin JY. Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching. Transportation Science 2006;40(2):211–225.

[20] Schilde M, Doerner KF, Hartl RF. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers & operations research 2011;38(12):1719–1730.

[21] Alonso-Mora J, Wallar A, Rus D. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) IEEE; 2017. p. 3583–3590.

[22] Bent RW, van Hentenryck P. Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. Operations Research 2004;52(6):977–987.

[23] Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences of the United States of America 2017;114(3):462–467.

[24] Pureza V, Laporte G. Waiting and Buffering Strategies for the Dynamic Pickup and Delivery Problem with Time Windows. INFOR: Information Systems and Operational Research 2008;46(3):165–175.

[25] Ferrucci F, Bock S, Gendreau M. A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. European Journal of Operational Research 2013;225(1):130–141.

[26] Chao IM, Golden BL, Wasil EA. The team orienteering problem. European Journal of Operational Research 1996;88(3):464–474.

[27] Ropke S, Pisinger D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. Transportation Science 2006;40(4):455–472.

[28] Pisinger D, Ropke S. Large Neighborhood Search. In: Gendreau M, Potvin JY, editors. Handbook of Metaheuristics, vol. 146 of International series in operations research & management science Boston, MA: Springer US; 2010.p. 399–419.

[29] Shaw P. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: Goos G, Hartmanis J, van Leeuwen J, Maher M, Puget JF, editors. Principles and Practice of Constraint Programming — CP98, vol. 1520 of Lecture Notes in Computer Science Berlin, Heidelberg: Springer Berlin Heidelberg; 1998.p. 417–431.

[30] Potvin JY, Rousseau JM. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research 1993;66(3):331–340.

[31] City of New York, 2014 Yellow Taxi Trip Data; n.d. https://data.cityofnewyork.us/Transportation/2014-Yellow-Taxi-Trip-Data/gn7m-em8n.

[32] GraphHopper; n.d. https://github.com/graphhopper/graphhopper.

[33] Soeffker N, Ulmer MW, Mattfeld DC. On Fairness Aspects of Customer Acceptance Mechanisms in Dynamic Vehicle Routing. In: R O Large, N Kramer, A -K Radig, M Schäfer, A Sulzbach, editor. Proceedings of Logistikmanagement; 2017.p. 17–24.

Appendix

TABLE 4 Percentage of requests removed per iteration

$\gamma_1 - \gamma_2$	Fully Anticipatory TOP	Anticipatory Routing Insertion & final re-optimization	None Anticipatory Insertion & 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 5 Noise value Regret-2 Insertion

Values	Fully Anticipatory TOP	Anticipatory Routing Insertion & final re-optimization	None Anticipatory Insertion & 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 6 Noise value Worst-Removal

Values	Fully Anticipatory TOP	Anticipatory Routing Insertion & final re-optimization	None Anticipatory Insertion & 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%

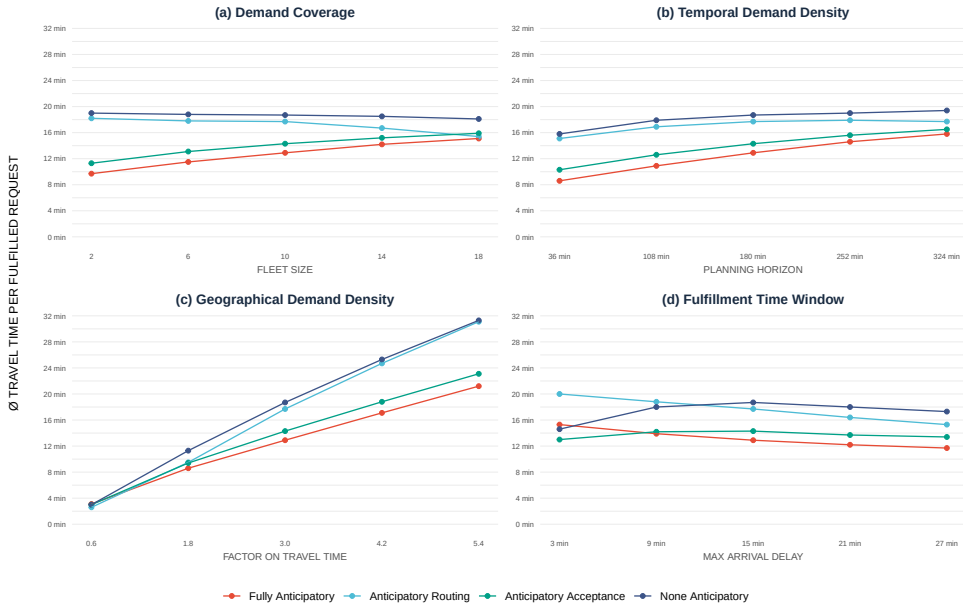


FIGURE 10 Sensitivity analysis: Average travel time per fulfilled request

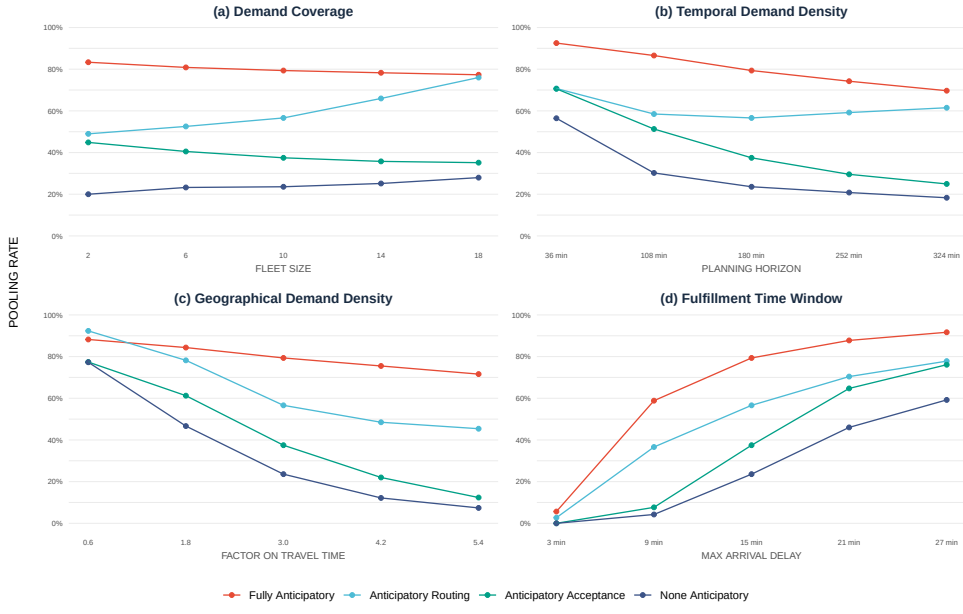


FIGURE 11 Sensitivity analysis: Pooling rate

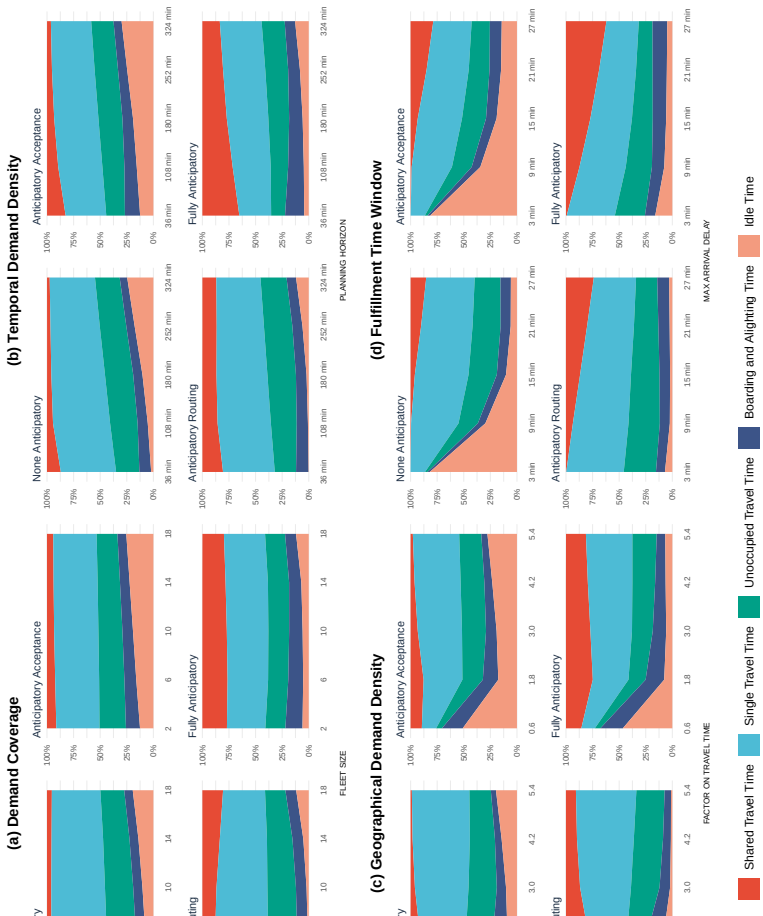


FIGURE 12 Sensitivity analysis: Time share per vehicle mode

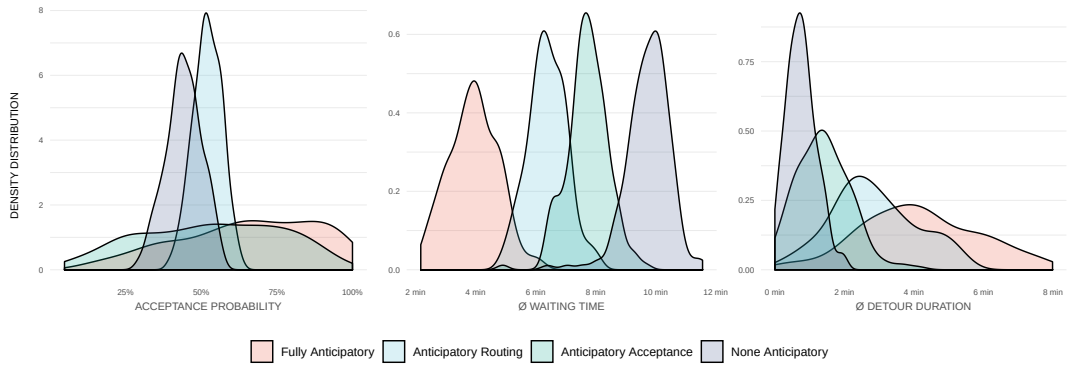


FIGURE 13 *Temporal Demand Density*: Quality of service per trip

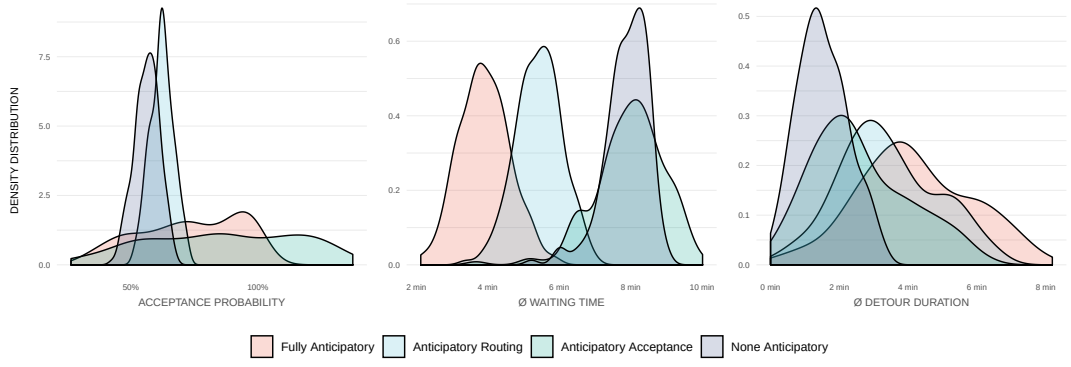


FIGURE 14 *Geographical Demand Density*: Quality of service per trip

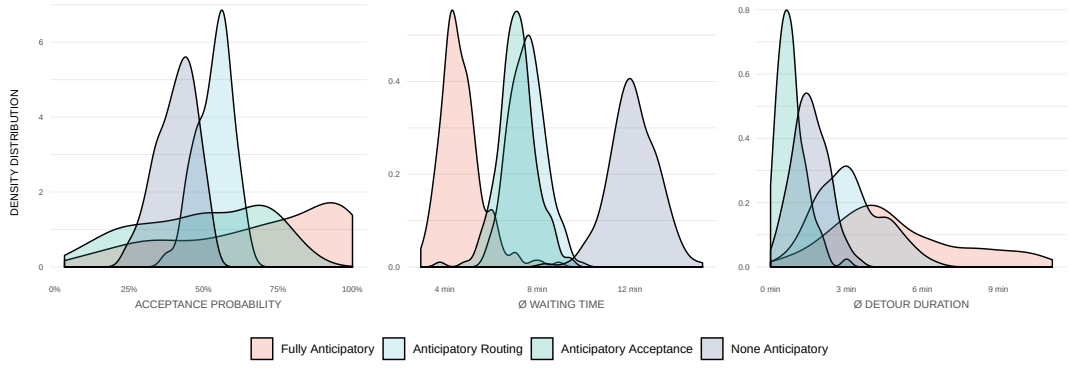


FIGURE 15 *Fulfillment Time Window*: Quality of service per trip

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274