Transportation Research Part C 126 (2021) 103104

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Combining immediate customer responses and car–passenger reassignments in on-demand mobility services

Marvin Erdmann *, Florian Dandl, Klaus Bogenberger

Chair of Traffic Engineering and Control, Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany

ARTICLE INFO

Keywords:
On–demand mobility
Car–passenger assignments
Immediate response
Dial-a-ride problem
Nearest neighbor policy
List–based assignments

ABSTRACT

This paper presents a two-step information system for operating an on-demand mobility (ODM) service using the advantages of both immediate responses based on heuristics and global (re)optimization of car–passenger assignments. Service providers using such a model can offer a better user experience due to shorter response times, while they also benefit from the increased profits made possible by the global optimization of assignments. This study compares two immediate response strategies (IRSs) in terms of their system performance and individual ability to correctly predict customers' pickup time windows. It also compares the key performance indicators (KPIs) of global reassignment optimization without immediate responses and several constrained cases in which customer acceptances and rejections are handled by IRSs.

Ten combinations of IRSs and service model variations are tested in simulations using the ODM demand data from New York City taxis obtained over one week for varying fleet sizes of between 1,000 and 6,000 vehicles.

The results show that in general, the list-based assignments (LBA) approach outperforms the nearest neighbor policy (NNP) as an IRS in most of the scenarios evaluated with respect to KPIs, such as requests served and profit generated for the service provider, while it also produces more empty vehicle mileage and longer customer waiting times. The pickup time window predictions of both LBA and NNP were correct in 68 % to 72 % of cases in scenarios in which no constraints are induced by the IRS.

It was also found that global (re)optimization of assignments helps to improve the profit generated for the service provider, especially if the decision as to which requests are accepted is made during global optimization rather by the IRS. However, such a service model would imply an average customer response time of half the optimization period, which was set to 30 s in this study compared to the immediate responses given when using an IRS.

1. Introduction

Worldwide urban traffic problems have become more and more obvious in recent years. Since the trend toward urbanization shows no sign of abating in the coming years (United Nations and Department of Economic and Social Affairs, Population Divison, 2018), cities are trying to find alternative mobility concepts that will reduce the number of cars on the road while still providing the required level of mobility.

On–demand mobility (ODM) services have emerged in recent years as such an alternative. The general concept is to enable users of the service, especially those in urban areas, to enjoy individual mobility without them actually needing to own a car.

* Corresponding author.
E-mail address: erdmann.marvin@tum.de (M. Erdmann).

https://doi.org/10.1016/j.trc.2021.103104
Received 21 October 2020; Received in revised form 19 March 2021; Accepted 19 March 2021
Available online 31 March 2021

There are different modes of ODM services, one of which is called 'ride hailing' (RH). Users of an RH service are able to request a ride from a pickup location to a destination of their choice. Unlike customers of a 'ride sharing' service, they are transported individually from A to B with no stops between pickup and drop-off points to serve other requests, and therefore no detour times either. This service mode enables quick and convenient individual transportation with travel times comparable to privately-owned vehicles and without the downsides of high fixed costs, parking problems and the exclusion of certain groups of people. On the other hand, some studies suggest that ODM fleets in general and those offering RH services in particular cause increased vehicle mileage leading to more traffic in urban areas due to empty rides between requests and trips substituted from modes like public transportation, cycling or walking (Henao and Marshall, 2018).

The strategies of companies such as Uber or Lyft in managing their ODM fleets, which comprise thousands of vehicles per city, are aimed towards optimization of their respective profits. To achieve this, they need to minimize their fleet costs while serving as many customers as possible. In a large-scale system, this can only be done effectively using algorithms to control which vehicles are assigned to new customers and which requests have to be rejected if no car is able to serve a particular user in time.

This optimization problem in its general form is known as 'dial-a-ride problem' (DARP) and has been the subject of research since it was first formulated by Psaraftis in 1980 (Psaraftis, 1980). Since then, new optimization techniques have resulted in increasingly better solutions. However, due to the highly dynamic nature of the problem, it is practically impossible in most cases to find an optimal solution (a static case solution if all information was available beforehand) as new information is constantly revealed over time. Moreover, the ability to fing close-to-optimal solutions is not the only performance indicator that needs to be taken into account. At least as important as finding good assignment solutions (which are defined by a so called objective function within the optimization process) is to find them quickly. This is not only to enable optimal navigation of the ODM fleet with only short delays, but also to be able to inform customers of the service whether and when they will be served. The time that users of an ODM service have to wait for a response is crucial in deciding which service provider they are willing to use.

Hence, there is currently a gap between the research itself and its practical application in ODM services. Providers often use heuristics in their algorithms such as the nearest neighbor policy (NNP). Such approaches are in general easier to implement and need less computation time to find feasible solutions to a problem compared to more sophisticated optimization methods. These benefits are possible because the principle of heuristics is not to globally optimize the whole system but instead to generate solutions based on simple rules for small parts of it. This means that solutions found with heuristics may be far less effective than the global optimum, which translates to fewer customers served and higher fleet costs in DARPs.

One way of avoiding this would be to make use of both heuristics — to enable quick responses to customers of an ODM service — and global optimization — to be able to benefit from more customers served and lower fleet costs. So far, there has been a lack of research into the capabilities and potentials of heuristics as immediate response strategies (IRSs). It is unclear what effect such an aspect of ODM services would have on the overall performance of the operational system. Consequently, neither has there been any investigation into the kind of heuristic that is best suited to this task.

The aim of this study is to find a system design that combines an immediate response to customers with the operational benefits of global car–passenger reassignment optimization. The object is therefore twofold:

1. To evaluate different methods that can be used as IRSs in ODM services in terms of their ability to predict a time window in which a customer is picked up based on global optimization.
2. To compare the system performance of global re-assignment optimization without immediate responses as well as the constrained cases with an IRS in which customer acceptances and rejections are handled by heuristics. Furthermore, the effect of communicated time windows as soft and hard constraints is also considered.

The remainder of this paper is structured as follows. An overview of related literature is presented in the next section. The contributions made by this study are then presented in more detail. Section 3 defines the evaluated service and introduces the simulation framework used to achieve it. Also, the problem formulation of our model is described, first in general, then in more detail for each specific system-design scenario examined in the simulations. The properties of each scenario are explained, as are the differences between them. The section closes with the definition of the two IRSs employed in the study. Section 4 presents the experimental design, including data and performance indicators along with the study's findings. The paper concludes with a short summary of the study, the central aspects of its methodology, and its key findings, along with a short subsection on the potential for future research.

2. Background

Vehicle automation and the prospect of shared vehicle fleets have sparked a significant amount of research in recent years, as it enables the main cost component, the driver, to be removed (Fagnant et al., 2015; Dandl and Bogenberger, 2019).

The operation of an RH system involves many aspects and a variety of mathematical problems (Hyland and Mahmassani, 2017; Narayanan et al., 2020). There are different approaches to driver scheduling and attracting of freelance drivers in two-sided platforms, while pricing and repositioning aim to balance the total amounts of demand and supply (Nourinejad and Ramezani, 2019; Bimpikis et al., 2019; Zhang and Pavone, 2016; Dandl et al., 2019b). Finally, car–passenger matching determines, which vehicle can and should (if available) be assigned to a customer request. The underlying vehicle routing problem is denoted as DARP (Cordeau and Laporte, 2007).

DARP represents a dynamic stochastic problem in which customer requests are displayed over time. For this reason, finding an optimal solution (given by the solution of a corresponding static problem with all information revealed ahead of time), is not

possible in most cases because of the curse of dimensionality (Zhang et al., 2016; Al-Kanj et al., 2020). Hence, the DARP is often split into solving the assignment problem for the revealed information and a repositioning problem to account for future demand and supply imbalance. Since there is plenty of literature on the aspect of repositioning, this paper focuses on the assignment aspect. There are several approaches to this facet of DARP, which can be classified into sequential local optimization and batched global optimization approaches with and without reassignments.

Sequential approaches attempt to insert a single new request into an existing solution before going on to the next request. NNP is a well-known heuristic that in general assigns incoming requests to the vehicles that are closest to the respective pickup location. Such an assignment problem is solvable in $O(n)$, where n is the number of vehicles in the problem. An example that does not allow reassignments is presented in Maciejewski and Bischoff (2015).

By including reassignments, more refined insertions can be made, thereby improving the current solution. In recent years, many different approaches have been studied besides NNP (Sheridan et al., 2013; Maciejewski et al., 2016). In Erdmann et al. (2020), the performance of list–based assignments (LBA) is compared to that of NNP. A combination of a genetic algorithm and NNP is used to solve a static version of DARP in Bergvinsdottir et al. (2007). Machine learning (ML) techniques are applied to problems closely related to the DARP in Nazari et al. (2018) and Syed et al. (2019a). Both of these ML approaches consider only static problem instances and small problem sizes but offer good scaling behavior once the training of the respective neural networks is completed.

Batched global optimization approaches collect requests for a certain time period and then go on to solve the assignment problem. In its most general form allowing pooling and reservations, these problems are NP-hard, which means that the computation time required for an optimal solution rises exponentially as the problem increases in size. Therefore, some approaches have sought an optimal solution to constrained versions of the problem (Alonso-Mora et al., 2017) while others use metaheuristics in order to find close-to-optimal solutions, such as ‘tabu search’ (Cordeau and Laporte, 2005; Pandi et al., 2018) and ‘large-neighborhood search’ (Syed et al., 2019b). Hyland and Mahmassani (2018) and Hörl et al. (2019) place additional constraints on RH to reduce its complexity and test several reassignment strategies to quantify their impacts and benefits over NNP in terms of traveler waiting time and empty vehicle mileage.

Although, the optimization potential of batching approaches is higher than that of sequential assignment techniques, the implied additional mean waiting time of half the batching time is inevitable. On top of that, the global assignment problem results in longer computation times compared to those of simpler sequential models, which would result in even later responses in real-time frameworks (Dandl et al., 2019a). Therefore, in Erdmann et al. (2019) and Erdmann et al. (2020) the advantages of both approaches are combined, using heuristics to enable quick responses to user requests and a tabu search metaheuristic to periodically solve a global optimization problem in order to find close-to-optimal assignments that minimize empty vehicle mileage and customer waiting times while serving as many requests as possible. However, the problem sizes considered in these studies were very small, comprising at most 5 % of the New York City taxi demand and fleet sizes of 300 vehicles. Also, no reassignments were made after requests were assigned after a global optimization.

The major contributions of this study are the following. First, the study successfully designs a system that combines the advantages of immediate response to customers of an ODM service and global car–passenger (re)assignment optimization. Previous comparable two-step approaches did not consider reassignments of customers already matched, which inherently increases the optimization potential. The effect of global optimization is quantified in terms of potential monetary profit for the service provider, the total driven mileage saved by the ODM fleet, the average customer waiting time, and the percentage of requests accepted.

Second, we define and analyze key performance indicators (KPIs) of various IRSs in simulation scenarios with varying hardnesses of time window constraints. The degree of hardness determines whether pickups outside of the communicated time windows are (a) not penalized, (b) penalized or (c) not allowed at all.

Third, the study presents a measurement, evaluation and discussion of the ability of these IRSs to accurately predict pickup time windows for customers at the time of the request in scenarios without time window constraints. A strong performance here enables a great improvement in the service quality experienced by users of an ODM service that provides such a feature.

Our findings provide insights for ODM operators with user-centric services that provide convenient instant responses to request queries. Moreover, to the best of our knowledge this study is the first to evaluate KPIs such as the prediction accuracy for pickup times for varying hardnesses of time window constraints and thereby contributes to the research area of modeling and optimizing ODM services.

3. Methodology

3.1. Service definition

The service investigated in this paper has several general properties, which do not change throughout the study. An overview of the notation is given in Table A.1 in the Appendix.

In all scenarios evaluated in this study, users of the service do not have the option to share their rides. The only available mode is referred to as ‘online ride hailing’, which also means that customers can only request a pickup as soon as possible and cannot schedule a ride further into the future.

The vehicles in the ODM fleet are all identical in terms of the car model and fare structure. Also, vehicles are assumed to be ‘autonomous’ in the sense that no human drivers are involved and they are available 24 h a day. These assumptions help to avoid additional operational constraints due to driver schedules and give the algorithm tested in the study full control over the fleet.

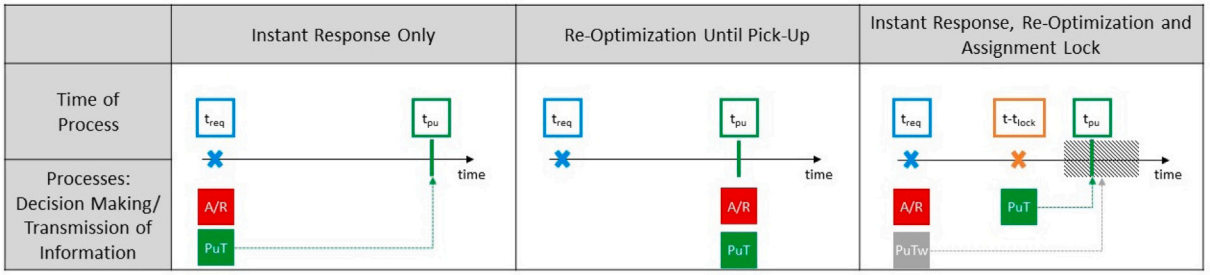


Fig. 1. Three ODM service model concepts. Left: If assignments are made very quickly and do not change later on, the acceptance/rejection (A/R) and pickup time (PuT) can be communicated to the customer immediately, but the service quality is potentially suboptimal. Center: If reassignments are allowed until the pickup actually takes place, the service quality might improve in terms of waiting times and the number of rejections, but customers have to wait longer for information. Right: The combination of instant decision making and reoptimization up until t_{lock} before the projected pickup time makes it possible to take advantage of both quick responses and improved solution qualities due to reassignments.

Let J be the set of all users of the service. A user $j \in J$ needs to declare both pickup location and destination when requesting a ride. Both locations have to lie within a service area defined by the operator of the service.

Various service models are examined and these are described in detail in the scenarios presented in Section 3.2.2. In all scenarios, the maximum waiting time $t_{\max}$ for each customer is set to six minutes, in line with studies that suggest that longer waiting times would lead to many users quitting the service (Spieser et al., 2016). $t_{\max}$ defines the maximum difference between the pickup time $t_{j,\text{pu}}$ and the request time $t_{j,\text{req}}$ of a customer:

$$t_{j,\text{pu}} - t_{j,\text{req}} \leq t_{\max} \quad \forall j \in J. \tag{1}$$

If a customer cannot be picked up before that time, the request is rejected and will not be served. The time of rejection depends on the service model in the scenario in question.

All service models examined in this study are based on a two-step information system. Once accepted, customers are informed of the pickup time in two ways: first, when they are informed that their request has been accepted, they are also sent a time window, in which they can expect to be picked up. A second notification informs them of the exact pickup time and the assigned vehicle. The scenarios introduced in Section 3.2.2 vary in the way when service users are contacted and which decision making process is used.

Fig. 1 presents the conceptual differences between three service models in respect of the timing of their respective decision making and the transmission of information to the customers. Service models that rely only on instant request responses (left) often only use heuristic approaches to determine whether a request is accepted or rejected (A/R) and when a request is picked up (PuT). Such methods do not use the potential of global optimization in order to improve their assignments and therefore often perform worse than more sophisticated approaches.

Models that allow reoptimization of assignments until a customer is eventually picked up (center) are conceptually able to use this potential. However, if the reoptimization of assignments is possible until pickup, customers of such services cannot be informed if and when they are going to be picked up.

The service models presented in this study take advantage of both the quick responses enabled by heuristics and the optimization potential of reassignments. As shown on the right, a user is informed very quickly of whether or not a request has been accepted. If accepted, customers also receive a pickup time window (PuTw) that indicates when they can expect to be picked up. After this initial decision, the operator is able to reassign requests until a defined time t_{lock} before the predicted pickup time t_{pu}.

At this time, exact pickup information is available and is sent to the user, because the request will be served by the vehicle it is currently assigned to. This improves the service quality experienced by users because they can plan when exactly they are going to be picked up and by which vehicle, which simplifies the search for their vehicle in situations in which several cars arrive at the same location at the same time.

3.2. Service models and problem formulation

3.2.1. General problem formulation

In an RH system, the dynamic fleet-operational problem can be summarized as follows: over time, exogenous information about customer requests are revealed to the operator, and the operator has to assign vehicles to serve the demand. Typically, this is modeled in a discrete time system, in which the operator can assign routes with stops depending on the system state. The stops are either related to customers' pickup and drop-off locations (request assignment) or they anticipate future demand (repositioning). The assigned actions are driven by a global objective function F_{obj}. In this study, the global objectives of the operator are to serve incoming requests and to maximize profit, where revenues are generated by fares with both a base and a distance-based component. On the other hand, vehicles generate fixed costs (e.g. investment, insurance) and operating costs (e.g. energy, wear & tear) costs:

$$F_{\text{obj}} = \sum_{j \in J} \left(p_{\text{base}} + p_{\text{dist}} \cdot d_j^{\text{od}} \right) - \sum_{i \in I} \left(c_{\text{fix}} + c_{\text{dist}} \cdot d_i \right) \tag{2}$$

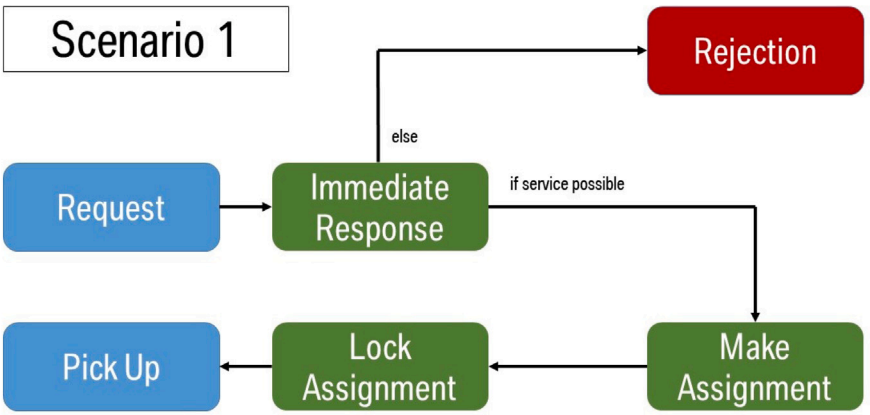


Fig. 2. Basic concept of Scenario 1.

where $i \in I$ is an index looping over all vehicles I and $j \in J$ is an index for customers J, p_{base} is the base fare associated with an accepted request, p_{dist} is the distance-dependent fare coefficient, d_j^{od} is the travel distance between the origin and destination of a customer j, c_{fix} represents fixed vehicle costs, c_{dist} is the distance-related coefficient of operating costs and d_i is the total driven mileage of vehicle i.

In this study, customer assignment and repositioning are treated as separate problems, with the focus on customer assignment. The repositioning strategy is based on the real–time rebalancing algorithm introduced in Zhang and Pavone (2016). It also makes use of forecasting methods presented in Dandl et al. (2019b). The concept of this approach is to periodically direct the movement of ODM vehicles according to a demand forecast for the near future. The repositioning intervals t_{rep} are not necessarily the same as the forecast horizon t_{hor}. Since repositioning is an important factor of an ODM service but not the focus of this study, please see the references stated above for further information.

By separating the DARP into two subproblems, the global objective function can be formulated as:

$$F_{\text{obj}} = \sum_{i \in I} \sum_{j \in J} \left(p_{\text{base}} + (p_{\text{dist}} - c_{\text{dist}}) \cdot d_j^{\text{od}} - c_{\text{dist}} \cdot d_{ij} \right) x_{ij} - \sum_{i \in I} (c_{\text{i,rep}} + c_{\text{fix}}) \tag{3}$$

where $x_{ij} = 1$ denotes an assignment of vehicle i to customer j. Here, the operating costs related to pickup trips ($c_{\text{dist}} \cdot d_{ij}$), to customer service ($c_{\text{dist}} \cdot d_j^{od}$) and repositioning $c_{\text{i,rep}} = c_{\text{dist}} \cdot d_{\text{i,rep}}$ are named separately.

The operator's decisions are guided by the goal of maximizing a control objective during the simulation, based on currently available information. Since this information and the optimal global solution according to Eq. (3) evolve over time, what is an optimal time-step solution according to the global objective does not necessarily have to be the best decision in the dynamic problem. Hence, the global objective F_{obj} and the control objective F_{con} for the periodic global optimization do not have to be the same. However, they are closely related.

Moreover, users of the RH service expect an instantaneous response from the operator. Different types of instant responses are possible, ranging from acceptance/rejection to non-binding information and will be examined in the scenarios with the various service models. Periodic control problems depend on the scenario and are described in the next section.

3.2.2. Scenario-specific problem formulation

Scenario 1: IRS Only (IRS–O)

The first service design version evaluated in this study represents the status quo of many RH services in practice. When a customer sends a request to the service provider, whether or not this request is accepted is decided in the first instance of decision making, referred to as the IRS. If a service is possible according to the heuristic acting as the IRS in the simulation, the request is accepted and a car–passenger assignment made. If the heuristic rejects the new request, the service provider will be able to inform the customer accordingly immediately after the request is sent, since the IRS renders the decision very quickly. An alternative view of the same model is that the operator communicates the waiting time to the customer but the customer rejects it if it is too long according to Eq. (1).

Once a request $j \in J$ is accepted by the IRS, an assignment is made and the car makes its way to the pickup location. The customer also receives a pickup time window of length t_{twl}. This time window indicates when the customer can expect to be picked up. The beginning of this time window is referred to as $t_{j,\text{tw}}$, thus the end is defined as $t_{j,j,\text{tw}} + t_{\text{twl}}$.

The unique feature of Scenario 1 is that the assignment decision is only based on IRS function calls. The service model is very simple, and decisions and responses can be made very quickly, which is an asset that cannot be underestimated. Fig. 2 illustrates the basic concept of Scenario 1.

Scenario 2: IRS Acceptance & Global Optimization with Hard Constraints (IRS–A & GO–HC)

In Scenario 2, the model contains all the aspects considered in this paper. Instead of basing a decision solely on an IRS, in this scenario a periodic global optimization is employed to make better use of the optimization potential of the dynamic DARP. It is performed at intervals of t_{per}, taking into account all unlocked assignments. This means that requests can be reassigned multiple times before they are finally matched to the vehicles which will pick them up. Depending on the optimization parameters, it is also possible that customer requests will be rejected after being initially accepted by the IRS. Such late rejections can be prevented by awarding high penalties in the objective function of the optimization, because they imply a bad user experiences, leading to potentially fewer requests in the long term. Both IRSs considered in this paper guarantee that all accepted customers will be included in at least one feasible solution of the optimization.

The global optimization of car–passenger assignments is an important instrument with which the operator of an ODM fleet can minimize fleet costs caused by driven mileage c_{dist}, while being able to serve as many requests as possible. The saving in vehicle mileage not only increases the profitability for the service provider but also means less traffic on the streets of the city hosting the service, which is often a crucial argument in relation to ODM services.

In this scenario, a time window given by the IRS when accepting a request is defined as a 'hard' constraint for the optimization. This means that solutions found during the optimization procedure that include customer pickups that do not take place within the respective time windows are infeasible and therefore not considered. Customers are guaranteed to be picked up within the time windows they are given, otherwise they are treated as late rejections. From a customer's perspective, a short time window is almost as good as an exact pickup time (especially considering that changes due to traffic congestion can happen anyway in reality). On the other hand, these time windows allow the operator to optimize assignments. The periodic optimization problem reads:

$$\max_{x_{ij}} F_{\text{con}}(x_{ij}) = \max_{x_{ij}} \sum_{i,j} \left(p - c_{\text{dist}} \cdot d_{ij}\right) x_{ij} \tag{4}$$

$$\text{s.t.} \sum_{i} x_{ij} \leq 1 \quad \forall j \in J \tag{5}$$

$$\sum_{j} x_{ij} \leq 1 \quad \forall i \in I \tag{6}$$

$$x_{ij} = 0 \quad \forall (i \in I, j \in J) : t + t_{ij} > t_{j,\text{req}} + t_{\max} \tag{7}$$

$$x_{ij} = 0 \quad \forall (i \in I, j \in J) : t_{j,\text{tw}} > t + t_{ij} \vee t + t_{ij} > t_{j,\text{tw}} + t_{\text{twl}} \tag{8}$$

where F_{con} denotes the control function, t is the current time, t_{ij} is the time until arrival of vehicle i at the pickup location of request j, and p is a reward for an assignment. This reward is either p_{base} if a customer has not yet been accepted or p_{acc}, which is set to a large value to avoid late rejections of requests already accepted. Constraint (5) limits the number of vehicles assigned to a request to 1 and Constraint (6) limits the number of non-locked user assignments to 1. Constraints (7) restrict car–passenger assignments to those that are within the globally accepted maximum customer waiting time $t_{\max}$. Finally, due to Constraints (8) solutions are only accepted for which all customers are picked up within their respective time windows.

Hard time window constraints imply a pruning of the solution space, unlike unconstrained problems, in which more solutions are feasible. In practice, constraints (5), (6), (7) and (8) are handled as part of the preprocessing of the optimization by only considering feasible assignments of vehicles and customers. Hence, the effective constraints used in the optimization are

$$\sum_{i \in I(j)} x_{ij} \leq 1 \quad \forall j \in J \tag{9}$$

$$\sum_{j \in J(i)} x_{ij} \leq 1 \quad \forall i \in I \tag{10}$$

where $I(j)$ is the subset of vehicles that are able to pick up customer j according to time constraints implied by Eqs. (7) and (8) and $J(i)$ represents the subset of requests that allows feasible assignments to vehicle i according to the same constraints. For a clear distinction between the service models introduced in this section, the constraints are mentioned within the respective optimization problem itself.

Compared to Scenario 1, the addition of global (re)optimization increases both the model's complexity and the computation time. On the other hand, this service model should be able to perform better with respect to crucial aspects of fleet management, which should outweigh the downsides of this approach. Fig. 3 shows the additional element of global optimization and how it influences the processing of a request.

The optimization problem represents a global bipartite matching problem that is widely used in literature, but with different control objectives depending on the study, e.g. Dandl and Bogenberger (2019), Hyland and Mahmassani (2018) and Hörl et al. (2019). The performance of the fleet varies with the control function chosen; for clarity, we decided to apply the control function derived from the global objective.

Scenario 3: IRS Acceptance & Global Optimization with Soft Constraints (IRS–A & GO–SC)

The differences between Scenarios 2 and 3 are rather small. Nonetheless, the conceptual impact they have is significant. In Scenario 3, instead of hard time windows, the constraints induced by the pickup time windows defined by the IRS are stated as 'soft'. This means that rather than not taking solutions into account whose assignments imply pickups outside of the time windows

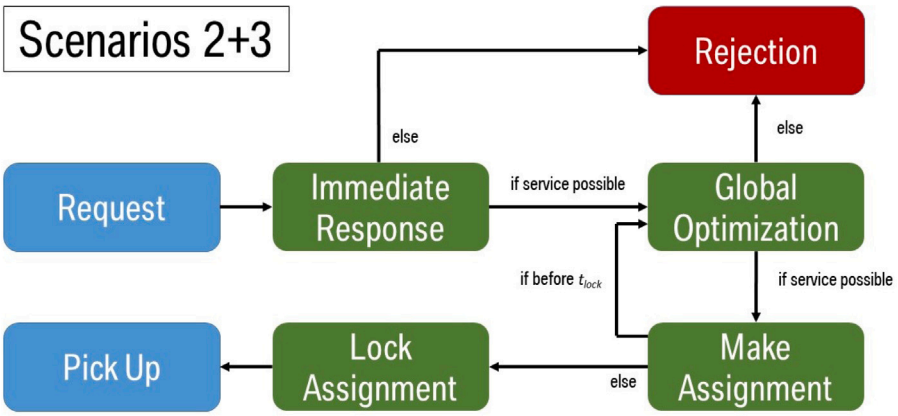


Fig. 3. Basic concept of Scenarios 2 and 3.

communicated by the IRS, such solutions are penalized in scenarios with soft constraints by way of a certain value P in the control function of the global optimization F_{con}.

$$\max_{x_{ij}} F_{\text{con}}(x_{ij}) = \max_{x_{ij}} \sum_{i,j} \left(p - c_{\text{dist}} \cdot d_{ij} - P \cdot \delta_{ij}\right) x_{ij} \tag{11}$$
$$\text{s.t. (5), (6), (7)}$$

where

$$\delta_{ij} = \begin{cases} 0, & \text{if } t_{j,\text{tw}} \le t + t_{ij} \le t_{j,\text{tw}} + t_{\text{twl}} \\ 1, & \text{otherwise} \end{cases} \tag{12}$$

is a parameter that is calculated in advance of the optimization problem, which equals 0 in case the assignment of vehicle $i \in I$ and request $j \in J$ implies a pickup time within the associated time window and 1 otherwise.

Compared to Scenario 2, in which the hard time windows defined by the IRS cause a substantial cut in the solution space, in Scenario 3, more solutions are considered feasible, allowing better assignments to be searched for according to F_{con}. This additional degree of freedom potentially increases the solution quality found during each global optimization even further.

Scenario 4: IRS Time Windows & Global Optimization with Soft Constraints (IRS–TW & GO–SC)

In all the scenarios introduced so far, whether a request is accepted or not is decided immediately by an IRS. This allows the service provider to inform customers quickly and, assuming the request is accepted, to issue the time window. However, these decisions are based on heuristic procedures that are not designed to take into account the global performance of the service. Therefore, an assignment made by them implies a solution quality that is not optimal with respect to the control function F_{con}. Even in scenarios in which such assignments are reoptimized by periodical global optimization, the selection of new requests that are accepted has already been made at that point, which reduces the optimization potential considerably.

This drawback of IRSs is avoided in Scenario 4. Rather than making an immediate decision, a request is either accepted or rejected based on the next global optimization to take place after the request is made, as illustrated in Fig. 4. Nonetheless, customers will receive two separate notifications. The timing of the first notification will vary, though. If the IRS projects that the request is accepted, the first response is sent immediately and contains the projected pickup time window. If not, the initial notification is sent when the first global optimization after the request has completed. This implies longer average response times as well as a relatively high probability of late rejections when customers first received a time window based on the projections of the IRS but are rejected after the global optimization. Both of these model properties are unattractive to service users, but the potential increase in profitability for the service provider could be worth the trade-off, which is the reason this service model is being investigated in this study.

Another disadvantage of this approach is that the mean waiting time until pickup also potentially increases, because idle vehicles do not start moving towards the pickup location right away, as they would in Scenarios 1 to 3. Hence, not only is the service quality experienced by the user worse, but some requests are also rejected that would have been acceptable had the vehicle been on the move sooner.

Scenario 5: IRS Time Windows & Global Optimization (IRS–TW & GO)

Scenario 5 is based on the same service model as Scenario 4. Hence, the IRS is not the decisive factor when it comes to the acceptance or rejection of customer requests, but it generates time windows for each request it assumes to be feasible. The ultimate goal would be to design an IRS that estimates the correct time windows for customers and sends this information to the respective customers instantaneously while the operator waits for the optimization to make actual customer–vehicle assignments. Also, idle vehicles do not start moving towards newly assigned requests until the next global optimization takes place.

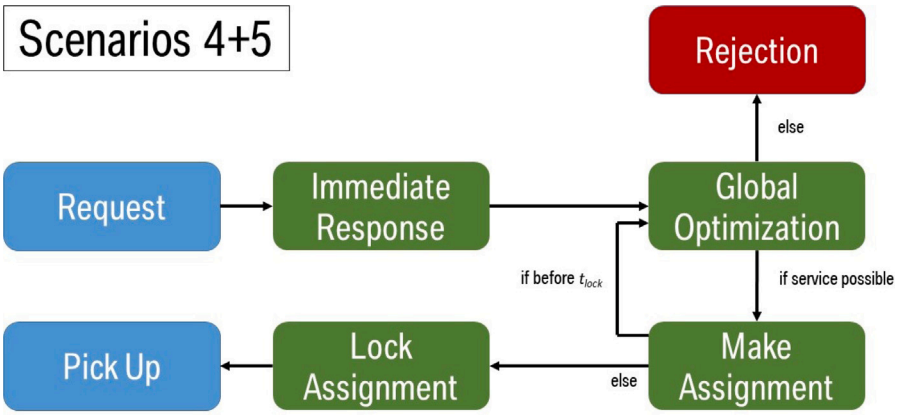


Fig. 4. Basic concept of Scenarios 4 and 5.

The difference between the two scenarios is the hardness of the constraint implied by the time windows generated by the respective IRS. While in Scenario 4, a penalty is associated with solutions that include assignments that involve pickup times outside of the respective time windows, in Scenario 5, these time windows do not imply any impact on global optimization. This means that the solution space is not limited in any way by time window constraints, and global optimization is allowed to search for solutions that are optimal for fleet management and include the highest number of accepted requests.

The IRS is still active in this service model to evaluate the ability of the heuristic used to predict the pickup time windows that are consistent with the assignments the global optimization produces. If an IRS could predict a time window including the actual pickup time of the assignment, the shorter time window would not constrain the global optimal solution while improving the customer experience. Reliable predictions are important elements of two-step information systems, allowing customers of the service to estimate their pickup times shortly after sending their request. Hence, being able to provide predictions that are sufficiently accurate while using the benefits of global optimization without obstructive constraints by time windows not matching the posterior global optimization solution would be a very appealing prospect for both operators and users of the service.

3.3. Immediate response strategies

The methods used to quickly respond to customer requests in the ODM service are defined as IRSs. In this study, two heuristics are used as IRSs, and the respective performances are compared in simulations outlined in Section 4. The two approaches are described in detail in the following.

3.3.1. Nearest neighbor policy

The nearest neighbor policy (NNP) is a simple heuristic designed to find feasible solutions of the DARP very quickly. Each time a new request occurs, the distances from each of the vehicles' next idle locations to the pickup location of the new customer are calculated and the earliest possible pickup times derived.

In Algorithm 1 the procedure of finding assignments A for all new requests joining the system at a certain simulation step is presented in the form of pseudocode. The routing problems of finding the fastest path between vehicle and customer pickup locations is solved independently. The resulting travel time matrices are filtered by the time constraints of the respective scenario. For the remaining assignments, the control objective values (COV) associated with each vehicle-request trajectory are calculated, which provides *vehicle-routing-infos*. These lists are then ordered by COV and limited to the v_{elig} most profitable assignments for each request in order to accelerate the iteration process for requests that could be served by an excessive amount of vehicles. This method of reducing the search space has been used before, for example in Engelhardt et al. (2020) and Sarma et al. (2020). Reassignments are not allowed; hence, vehicles that are already assigned to an unlocked request cannot be part of *vehicle-routing-infos*, to ensure that the respective pickup takes place in time.

As shown here, the search iterates over all considered *new-vr-infos*. If a vehicle is already assigned to one of the other new requests, this vehicle cannot be assigned again. New requests that are not yet assigned to any vehicle are assigned immediately. If an assignment was already found in an earlier iteration, the profit from that assignment is compared with the current iteration and the existing assignment is updated in the event that the new assignment is better. Once all *new-vr-infos* have been checked, the preliminary assignments are filtered by the waiting times implied for the respective customer. If the maximum waiting time $t_{\max}$ is surpassed, the request is rejected. Otherwise, a time window is generated around the expected time of arrival at the pickup location. Depending on the scenario, the request is also immediately added to the task queue of the assigned vehicle and the user is informed about whether the request has been accepted or not.

Algorithm 1: Pseudocode of the Nearest Neighbor Policy

```
Input: new-vr-infos
// vehicle-routing-infos=[(vehicle1,request1,COV),(vehicle1,request2,COV)...]
A = {};                                   // Assignments={request1:[vehicle,COV],request2...}
for vr-info ∈ new-vr-infos do
    (v, r, c) = vr-info;
    v-in-a = False;
    for a ∈ A do
        if v = a[0] then
            v-in-a = True;
            break;
    if v-in-a then
        continue;
    if r ∉ A then
        A[r] = [v, c];
    else
        if c > A[r][1] then
            A[r] = [v, c];
return A
```

Using NNP in the service model introduced in Scenario 1 'IRS Only' represents the concept of 'instant response only' introduced on the left in Fig. 1. This combination of service model and IRS is the only scenario considered in this work that does not include any reassignments at all.

The benefit of NNP is its ability to produce feasible solutions without any appreciable delay, since the computation time needed to find solutions with it scales linearly with the fleet size. As it only considers the very next request, it tends to fail when it comes to finding optimal solutions for the system.

3.3.2. List–based assignments

The idea of list–based assignments (LBA) is to improve the initial solution by enhancing the simple NNP heuristic while decreasing the size of the solution space to be searched by global optimization, without eliminating any feasible solutions. A brief example is given to illustrate this approach.

As described in Section 3.3.1, the NNP searches for vehicles that are able to pick up customers before they have waited longer than $t_{\max}$ and which imply the highest profits, as soon as new requests are submitted. Other cars' earliest pickup times for each particular request are not stored. If another request r_2 is submitted later, occupation of a vehicle is considered until the drop-off time of the first preliminarily assigned request r_1, in order to guarantee that r_1 is picked up in time. If no other car is able to pick up r_2, this request is rejected.

However, with the LBA method, all cars that are able to pick up a customer i before $t_{\max}$ after the request is sent, are added to a list L_i. This list is sorted by the COV related to an assignment in respect of the control function F_{con}. The vehicle enabling the most profitable assignment is preliminarily matched with the request. Thus, the same preliminary match is initially made for a customer as would have been done using the NNP.

When r_2 occurs, vehicle v_1, which was preliminarily matched to r_1, is not considered occupied. This is different from NNP, where a vehicle is considered unavailable until it has dropped off r_1. If there is another vehicle v_2 in L_1, r_1 can be instantly reassigned to that vehicle and is still guaranteed to be picked up in time. By doing this, v_1 is able to serve r_2 and the request is accepted.

Every time a request is preliminarily assigned to a vehicle, this vehicle is removed from all existing lists L_i to avoid cases in which a request i may be reassigned to a vehicle on L_i that has meanwhile been preliminarily assigned to a new request j. This ensures that only cars with no unlocked assignments are eligible for preliminary assignment to new requests.

Algorithm 2 shows the generalized procedure. The *new-vr-lists* are the collection of lists L_i of all new requests r_i joining the system at a certain simulation step. They are based on profits calculated in an independent routing problem and limited in the same way that *new-vr-infos* are limited in Section 3.3.1. An important difference to *new-vr-infos*, however, is the fact that unlocked assignments do not block potential assignments as is the case in Section 3.3.1. *old-vr-lists* are the stored lists of all unlocked requests in the system. Iterating over all requests r in *new-vr-lists* and over all considered assignments i in the respective list L_i, the algorithm first checks if the current vehicle v already has an unlocked request j in its task queue. If so, it checks if this request could be served by another vehicle $\tilde{v}$ out of the list of potential assignments for j. The profits associated with assignments of j to v and $\tilde{v}$ are referred to as p_1 and p_2 respectively. Should only v be available to pick up j on time, this assignment remains unchanged and the next vehicle in L_i is checked in the same way. If an alternative assignment for an already assigned request j is eventually found, the difference Δp between the profits p_1 and p_2 associated with the assignments to the current vehicle v and the alternative $\tilde{v}$ is calculated. In the event that the task queue of v does not contain an unlocked assignment, this difference is defined as zero. If no assignment was found in a previous iteration, a preliminary assignment a is made, containing vehicle v and the associated profit plus

Algorithm 2: Pseudocode of the List–Based Assignment Approach

```
Input: new-vr-lists,old-vr-lists
// vehicle-request-lists={request1:[[vehicle1,COV],[vehicle2,COV]...],req.2...}
A = {};                                          // Assignments={request1:[vehicle1,COV],request2...}
Ã = {};                                          // Re-assignments={request1:vehicle1,request2...}
for r ∈ new-vr-lists do
    a = {};
    L = new-vr-lists[r];
    for i ∈ L do
        (v, c) = i;
        ṽ, c1, c2 = False,False, False;
        for j ∈ v.tasks do
            if not j.locked then
                for k ∈ old-vr-lists[j] do
                    (c̃, c̃) = k;
                    if v = ṽ then
                        c1 = c̃;
                    else
                        if not c2 then
                            c2 = c̃;
                    if c1 then
                        if c2 then
                            break;
                break;
        if ṽ then
            if c2 then                           // if another car is on the list of j, c1 always bigger than c2
                Δc = c2 − c1;
            else                                 // if no other car on the list of j, try next car in list of r
                continue;
        else                                     // if v has no other task, there are no difference in profit to consider
            Δc = 0;
        if r ∈ a then
            if c + Δc > a[r][1] then
                a[r] = [v, c + Δc];
                if c2 then
                    ã = [j : ṽ]
                else
                    ã = False;
        else
            a[r] = [v, c + Δc];
            if c2 then
                ã = [j : ṽ]
            else
                ã = False;
    if r ∈ a then
        A[r] = a[r];
        if ã then
            Ã[ã[0]] = ã[1]
return A, Ã
```

the difference Δp due to the potential reassignment. If such a reassignment is necessary, it is saved as $\tilde{a}$. If later iterations find better assignments, both a and $\tilde{a}$ are changed accordingly. After all iterations of all new requests r are completed, the new assignments and corresponding reassignments are returned to the fleet management algorithm.

In this way, more requests can be accepted in a particular optimization period than would be possible using NNP, without losing the guarantee of accepted customers being picked up within a shorter time than the maximum waiting time $t_{\max}$ and without considerable additional computational effort. Furthermore, as only vehicles in the list L_i are able to pick up a customer i, the global optimization can neglect solutions in which the customer may be matched to a vehicle that is not in L_i.

4. Case study

This section describes the experimental design of the study and presents the results. It concludes with a brief discussion of the findings.

4.1. Experimental design

4.1.1. Simulation framework

To evaluate the relative performance of the various approaches described in the previous section, this study employs a framework that simulates the different approaches. The requirements of the framework are that the simulations must be as realistic as possible while avoiding random external aspects that might unduly blur the results. Additionally, the simulations have to be executed in a reasonable amount of time in order to generate a sufficient number of runs to allow confidence in the results.

The following simplifying assumptions apply to the simulation framework:

- Vehicles in the fleet move within a network at constant travel times throughout the simulation. Free-flow travel times multiplied by a factor of 3 were found to represent the average approximate travel times during the simulated dates.
- Vehicles do not need to be refueled or maintained during a simulation. Therefore, there are no maximum distance constraints.
- The customer response time, maximum boarding time, boarding time and disembarking time are set to constant values.

It was found that assuming constant travel time improved our understanding of some performance indicators during the simulations, as dynamic travel times result in differences between computed routing schedules and vehicle movements. Methodologically, a varying speed would be relatively easy to implement, but this element is neither essential to the aspects focused on in this study nor would it change the complexity of the problem at hand.

The second assumption relates to the focus of this study, which is to match customers of an ODM service with vehicles of the service's fleet. As we expect refueling and maintenance to be scheduled during off-peak hours, this should not affect the performance of the fleet.

Heterogeneous customer characteristics would have enormously increased the framework's complexity while not shedding any light on service design aspects that are relevant to the investigation. This could confuse findings and relationships between the simulation parameters that this study actually has set out to examine.

Besides these assumptions, the framework allows individual vehicles to be repositioned according to a precalculated demand-forecast data set. The deployment of a profound repositioning algorithm (RA) is vital in order to prevent parts of the ODM fleet from being scattered within low-demand localities of the service area, which would effectively exclude them from the actual assignment problem because they are too far away to reach most of the requested pickup locations before $t_{\max}$. The RA used in the simulation framework combines the real–time rebalancing policy (Zhang and Pavone, 2016) with forecasts of vehicle supply and demand (Dandl et al., 2019b). The RA decision time step t_{rep} and the RA time horizon t_{hor} are set to 15 min and 30 min, respectively. This means that repositioning tasks are assigned to the vehicles every 15 min while taking into account the supply and demand forecast of the next 30 min.

The time period between one global optimization and the next is set to $t_{\text{per}} = 30\,\text{s}$. In Scenario 1 (IRS only), this period marks the time frame in which requests assigned by the LBA can potentially be reassigned. The maximum number of vehicles eligible v_{elig} for each assignment is set to 50, which means that even if there are more than v_{elig} cars able to pick up a customer within the respective maximum waiting time, only the 50 closest cars are considered for assignment. This reduces the computational effort needed to find an assignment without significantly cutting the solution space. The maximum waiting time $t_{\max}$ (defined in Eq. (1) in Section 3.1) is set to 6 min, and the assignment lock time t_{lock} (defined in Scenario 1 in Section 3.2.2) to 3 min. Both values were chosen to ensure a high service quality for customers while allowing a reasonable degree of optimization potential. They are based on realistic assumptions for real-world service users' expectations. This also applies to the values of boarding time t_{boa} and disembarking time t_{dis}, which are 45 s and 15 s respectively.

Provided that the ODM fleet consists of autonomous vehicles which do not require a driver who the operator would need to pay, the daily fixed costs c_{fix} per vehicle would include investment, insurance, wear &tear and periodic maintenance costs, estimated in this study to amount to $25. The costs of energy consumption attributable to driven mileage is given as $c_{\text{dist}} = 0.25\,\$/\text{km}$. When a customer is transported, these costs have to be compensated for. The study assumes that the fare paid by a customer also depends on the distance driven and is set to $p_{\text{dist}} = 0.50\,\$/\text{km}$, in addition to a base fare of p_{base}, which is set to $1 in the simulations.

4.1.2. Data

Demand R is based on the New York City taxi data set (NYC Taxi & Limousine Commision, 2018). This open-source data has been used previously in many studies (e.g. Alonso-Mora et al., 2017; Hyland and Mahmassani, 2020). For this study, trips are filtered to include only those that start and finish within the Manhattan area. The exact pickup and drop-off locations were set by randomly choosing nodes of the Open Street Maps network within the taxi zones recorded for each request in the data set. The demand data (from November 12 to November 18 2018) is split into days, each with between 169,356 and 244,288 requests (mean: 215,266; std: 24,226). The pickup time in the data set is used as the request time t_{req} in the simulations. The simulations proceed in one-second steps, corresponding to the increment in the data set.

All simulation and optimization parameters that remain constant throughout all simulations are summarized in Table 1.

Table 1
Constant simulation and optimization parameters of scenarios.

Parameter	Symbol	Value
Simulation duration	t_{sim}	24 h
Demand	R	Manhattan taxi data, Nov.12 to Nov.18 2018
Fleet sizes	V	1000–6000 vehicles
Simulation step	t_{step}	1 s
Optimization period	t_{per}	30 s
Maximum vehicles eligible	v_{elig}	50 vehicles
Maximum waiting time	t_{max}	6 min
Assignment lock time	t_{lock}	3 min
Pickup time window length	t_{twl}	2 min
Boarding time	t_{boa}	45 s
Disembarking time	t_{dis}	15 s
Repositioning decision time step	t_{rep}	15 min
Repositioning time horizon	t_{hor}	30 min
Vehicle fixed costs	c_{fix}	$25
Service base fare	p_{base}	$1
Distance costs	c_{dist}	0.25 $/km
Distance fare	p_{dist}	0.50 $/km
Accepted request assignment reward	p_{acc}	$10,000

4.1.3. Key performance indicators

To evaluate the central aspects of the algorithms tested in the various scenarios, KPIs need to be defined. On the one hand, the focus of the study is to find an approach that is able to provide a high quality service to users and enable the profitable operation of the fleet. For this reason, the percentage of customers served and the mean waiting times are measured and compared, as is the empty mileage driven by the vehicles and the overall profit of the service. The four KPIs are used for evaluation purposes in all of the five scenarios introduced in Section 3.2.2 and are considered standard KPIs when evaluating use cases in the DARP or related problems.

It should be mentioned that for the purposes of this study, profit is defined as the total objective function value F_{obj} including all vehicles in the fleet and all requests accepted during each simulation, as described in Section 3.2.1. Rejected customers do not reduce the profit of the service provider, they just do not actively increase it. In real–world applications, rejected customers could lead to fewer service users in the long term, reducing the profitability of the service. This effect is not taken into account in this study, however.

Another KPI of relevance to scenarios without hard time window constraints is the 'pickup-in-time-window rate' (PUITWR). This indicates the percentage of time windows correctly predicted by the respective IRS. If a customer pickup takes place within the time window initially projected when the user made the request, this pickup prediction is considered to be correct. If the pickup occurs earlier or later or if the customer would be rejected by the IRS but is in fact accepted and picked up (only possible in Scenarios 4 and 5, where the request acceptance decision is made by the global optimization), the pickup prediction is considered to be wrong. Requests rejected by the IRS do not count towards the PUITWR as they are not covered by the global optimization. This KPI comes into play in Scenarios 3, 4 and especially 5, where no time window constraints prevent the optimization from finding solutions in which pickups can happen outside of the projected time windows.

4.2. Results

This section is divided into subsections, the first four of which present a comparison between the service models introduced in Section 3.2.2. All of the scenarios are simulated once per day of demand data, fleet size and IRS, which accounts for 420 total simulations. At the end of the section an evaluation of the computation times is presented as well.

As found during evaluation of the results, the variance in the demand data also causes significant deviations in the total values found for the KPI described in Section 4.1.3 when averaging the results over all considered days. As is already apparent from the literature, fleet size also has a significant impact on the results. Since the purpose of this study is to detect differences between various service models and IRSs, the results are presented in two ways. To visualize the scale of a KPI, the respective average total value is presented with error bars indicating the standard deviation calculated by averaging over all dates. In addition, KPIs of one simulated day and one specific fleet size are compared for two scenarios and both evaluated IRSs. Out of these four values, a mean value is calculated and the difference (delta) to each of the four single values is determined, indicating the performance of the respective scenario/IRS combination compared to the others for the specific day. This delta varies much less over the course of the considered simulation days, which illustrates that the visualization format may be more suitable for examining the results of this study. Therefore, most of the figures presented in the following section will depict deltas instead of total values averaged over simulated dates. Moreover, it should be noted that even rather small deltas on a daily basis can accumulate to considerable values over the course of years a service might be running.

In the following, error bars within figures indicate the standard deviation of the respective value, both in figures showing total values and deltas. Also, the mean value X and the standard deviation Y of the KPI in the above scenarios are given as $(X \pm Y)$.

4.2.1. Global optimization

The first service models to be compared are Scenario 1 'IRS only' (IRS–O) and Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC). The main difference between these two models concerns the aspect of global optimization with unlocked assignments. While in both scenarios the decision as to whether a request is initially accepted or not is based on the IRS, in Scenario 2 these initial assignments are subject to global optimization within hard time window constraints defined by the IRS. Both versions of the service are evaluated using the NNP and the LBA. It should be mentioned again that the LBA approach allows reassignments of vehicle–request pairs until a request is eventually locked to a vehicle, while the concept of NNP does not allow any reassignments whatsoever.

Fig. 5a shows the total values of profit and served customer requests averaged over all simulation dates, with respect to fleet sizes. As mentioned above this format is suitable for identifying the rough total scale of KPIs rather than differences between individual scenarios, as the standard deviation implied by averaging over the simulation dates clearly outweighs the variances between the models. Nonetheless, a maximum in overall profit is identified in scenarios with 3000 to 4000 vehicles, indicating that larger fleets may lead to more accepted requests but even more idle cars, contributing to the fleet costs in the form of fixed costs c_{fix}, which negates the positive effect of the paying customers.

This tendency is even clearer in Fig. 5b, where the distinct maximum in profit made per customer is made even more obvious by the proportionality of demand and profit. Also, the percentage of served customers increases progressively in scenarios with larger fleet sizes as would be expected. However, the added benefit of more vehicles being able to pick up more customers becomes progressively smaller the larger the ODM fleet is. In scenarios with more than 3000 vehicles, the acceptance percentages are higher than 90 % in all simulations considered, leaving very little scope for improvement by further increasing fleet sizes, ultimately leading to a drop in profits.

Empty vehicle miles traveled (VMT) as a percentage of total mileage are shown on the left of Fig. 5c, including empty miles due to repositioning. Since repositioning is an optimization problem that is solved independently of the assignment problem and not evaluated in detail in this study, the more interesting KPI is the empty VMT without considering VMT due to repositioning, shown on the right. While on the left, the empty VMT increases with bigger fleet sizes, on the right the opposite effect is observed. This is due to the increased potential for relocating parts of the fleet in more promising areas of the city in scenarios with more vehicles. The more idle vehicles are available, the more repositioning is conducted to balance the density of available vehicles, leading to more empty VMT. The empty mileage caused by trips to pickup locations is smaller in scenarios with bigger fleets, though, because the average distances between pickup locations and available vehicles are smaller.

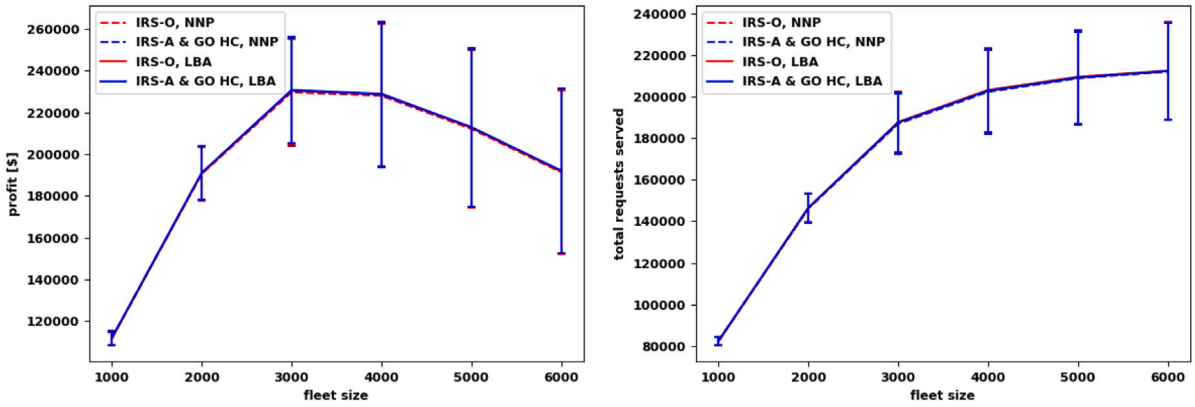
The differences between the scenarios become clearer in Fig. 6 where the deltas between the scenarios' KPIs to the mean values are depicted as described earlier.

The differences in profit generated according to the objective function introduced in Section 3.2.1 are presented in Fig. 6a on the left. The first observation that can be made is that there is a gap between the service models. With all considered fleet sizes, the average profit is higher in scenarios with IRS acceptance and global optimization (blue) compared to those with IRS only (red) for both IRSs. This trend is particularly evident in scenarios with fleet sizes of 3000 vehicles or more, in which most of the requests can be served and the optimization potential of reassignments can be put to better use.

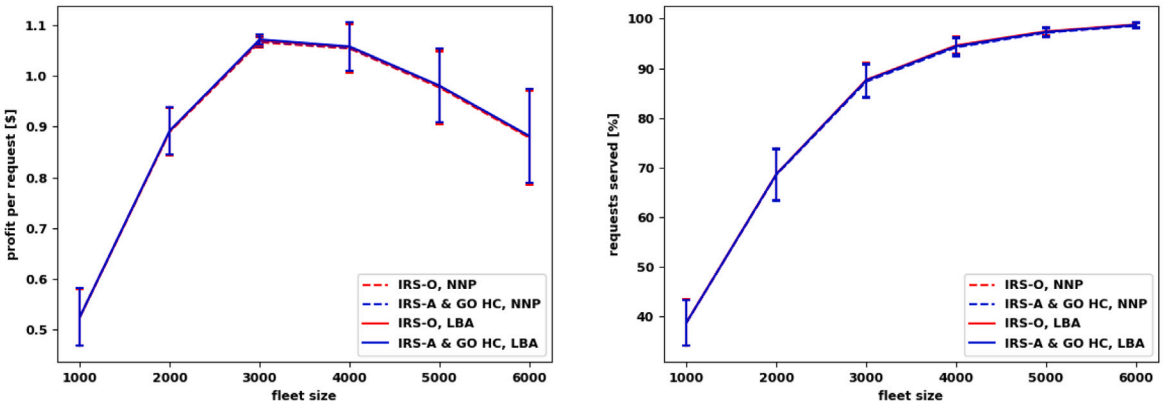
The benefit of reassignments is also apparent when comparing scenarios using either NNP (dashed lines) or LBA (solid lines) as the IRS. In scenarios with more than 1000 vehicles, the strategies using LBA outperform those using NNP with respect to the generated profit (for both service models). This effect is comparable in scale to the difference between the two compared service models, indicating that the optimization potential of LBA compared to NNP when deciding about whether to accept a request is approximately as great as the potential of global reoptimization of these accepted requests. In simulation scenarios with 3000 vehicles, the differences in profit generated between LBA and NNP are $(806 ± 233) and $(653 ± 366) respectively for the service model scenarios IRS–O and IRS–A & GO–HC, while the switch of service models when using the same IRS accounts for $(503 ± 343) (NNP) and $(350 ± 253) (LBA).

As described in Section 3.2.1 the profit for the service provider depends on the number of accepted requests and the driven vehicle mileage of the vehicles in the fleet. Hence, the results shown in Fig. 6a are mainly based on those presented in the rest of Fig. 6. The request percentages served are compared in Fig. 6a on the right. The differences between LBA and NNP remain similar to those observed with the profits, namely LBA outperforms NNP in scenarios with more than 1000 vehicles, as it is expected due to its intrinsic ability to reassign requests in order to allow more customers to be accepted.

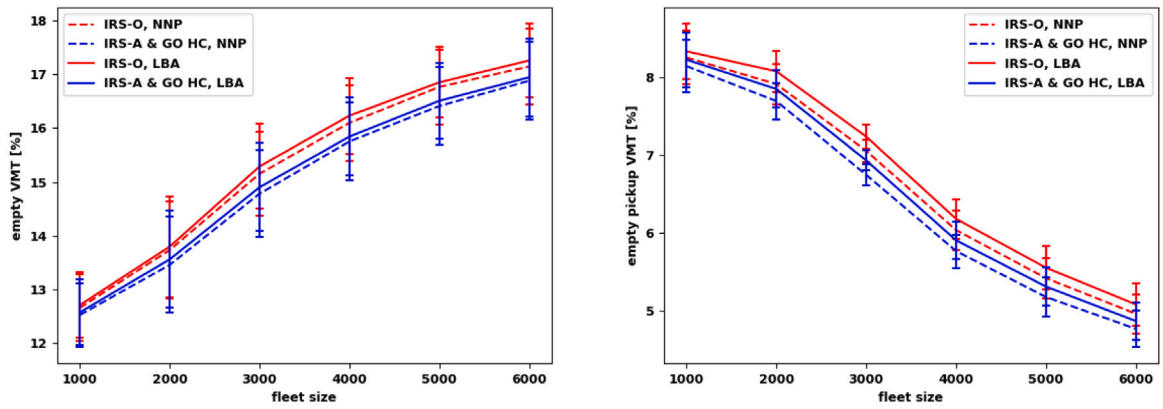
Meanwhile, the average percentage of customers served is lower in scenarios with periodic global optimization compared to scenarios with IRS decision making only. This effect is likely caused by the dynamic nature of the assignment problem. Solutions found by global optimization are optimal with respect to the current situation, covering the best available vehicles for requests accepted by the respective IRS. The optimization control function is not aware of future demand, therefore the optimized solution at a certain point in time might leave the system in an overall worse state regarding the availability of vehicles for upcoming demand. Since the set of accepted requests is determined by the IRS, the only goal of the global optimization is to minimize the empty mileage driven by the fleet according to the control function in Eq. (4). During peak-demand such a control function tends to prefer solutions in which vehicles in areas with the most demand are fully utilized because the distances between their respective tasks are short. At the same time, some vehicles remain idle in areas with lower demand because their respective distances to the pickup locations in the areas of high demand are longer. Occasional repositioning of vehicles can only partly counter this imbalance and eventually a lack of available vehicles occurs in high demand regions of the network, ultimately leading to more rejected requests. This phenomenon is referred to as dynamic optimization effect (DOE).



(a) Total profits (left) and customers served (right)



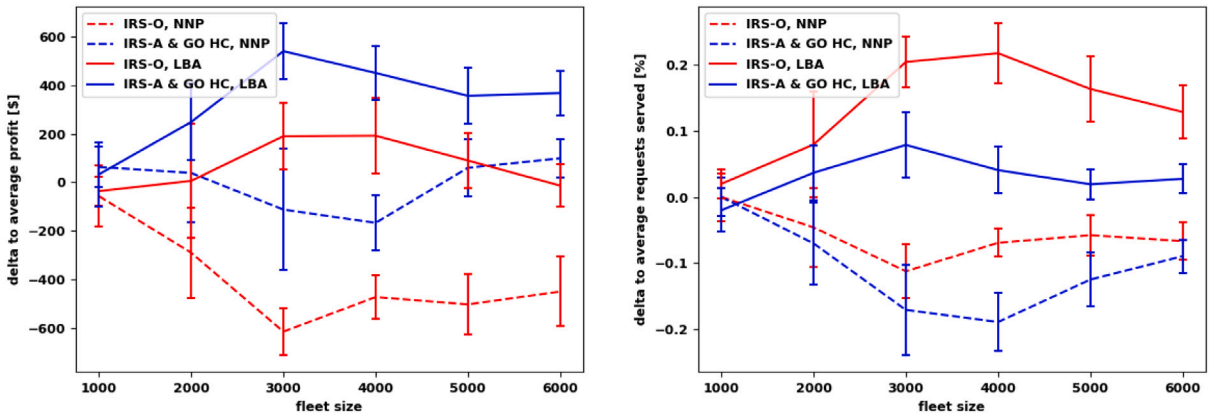
(b) Relative profits (left) and customers served (right) with respect to the total number of requests



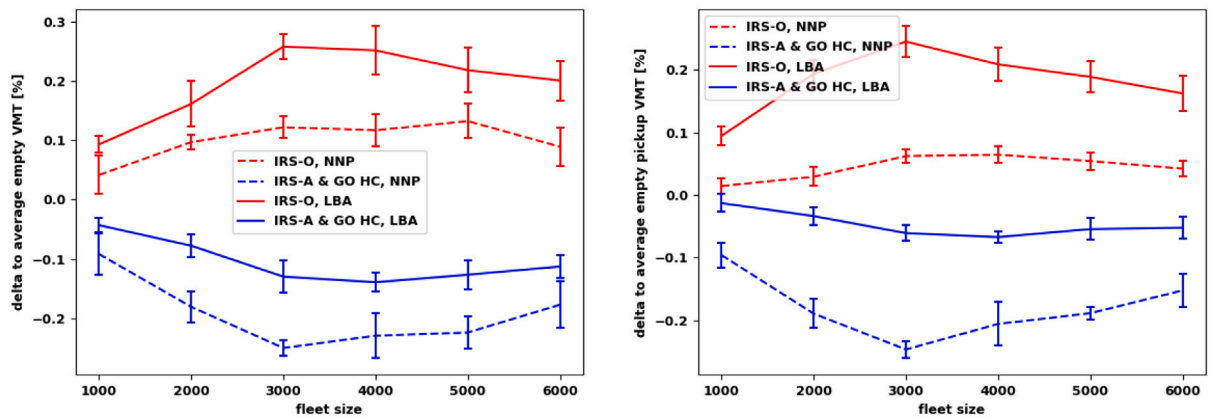
(c) Empty vehicle miles travelled (VMT) as a percentage of total driven mileage, with (left) and without (right) the share caused by repositioning

Fig. 5. KPI of Scenario 1 'IRS only' (IRS–O, red) and Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC, blue) using NNP (dashed lines) and LBA (solid lines) as IRS, averaged over all simulated days.

The delta in empty VMT as a percentage of total mileage is shown on the left of Fig. 6b, while on the right is the slightly different delta as a percentage of empty VMT, caused only by pickup trips. Comparing both KPIs, the similarity of deltas indicates that most of the differences in empty VMT between the various scenarios stems from the varying empty VMT of assigned pickup trips. In both figures, two observations can be made: (1) In scenarios that make use of LBA, more empty mileage is produced compared to those



(a) Delta to averages in total profit and percentage of customers served



(b) Delta to average in empty vehicle miles travelled with and without mileage due to repositioning

Fig. 6. KPIs of Scenario 1 'IRS only' (IRS–O, red) and Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC, blue) using NNP (dashed lines) and LBA (solid lines) as IRS, delta to averages per simulated day.

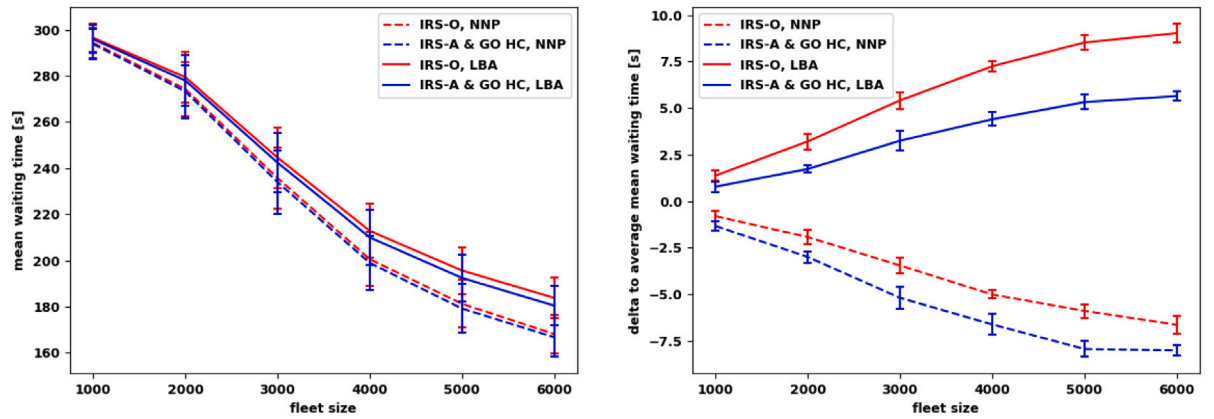
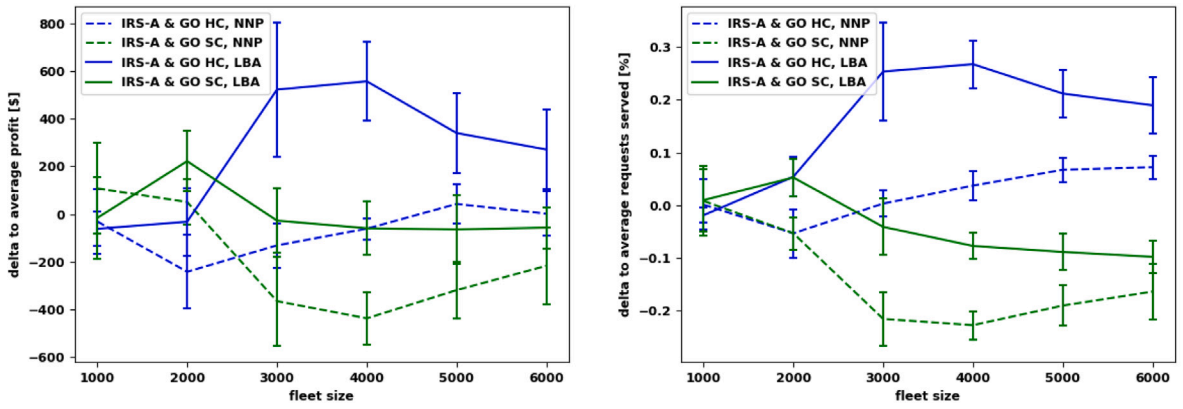


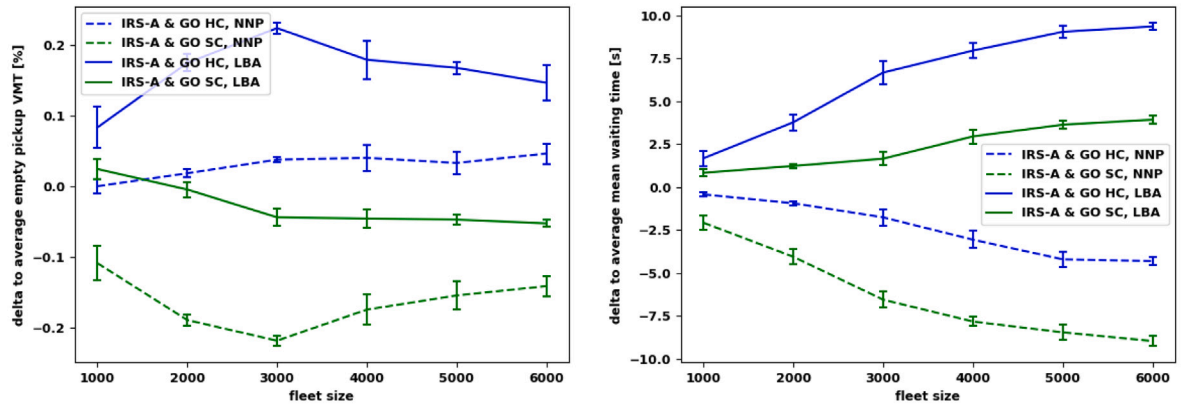
Fig. 7. Mean waiting times in Scenario 1 'IRS only' (IRS–O, red) and Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC, blue) using NNP (dashed lines) and LBA (solid lines) as IRS, averaged over all simulated days (left) and delta to averages per simulated day (right).

using NNP; (2) The global optimization in Scenario 2 (IRS–A & GO–SC) reduces the empty pickup VMT, making up for the lack of optimization potential in respect of served requests.

Fig. 7 presents the mean waiting times for service users. From the total scale on the left, it is obvious that the time between sending a request and the eventual pickup decreases with growing fleet sizes in all considered scenarios, dropping from around



(a) Delta to average in total profits and relative requests served



(b) Delta to averages in relative empty vehicle miles travelled (without repositioning) and mean customer waiting times

Fig. 8. KPIs of Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC, blue) and Scenario 3 'IRS acceptance & global optimization with soft constraints' (IRS–A & GO–SC, green) using NNP (dashed lines) and LBA (solid lines) as IRS, delta to averages per simulated day.

5 min for cases with 1000 vehicles to between (167 ± 8) s and (184 ± 8) s for 6000 vehicles, depending on the service model and IRS used.

These differences are even more obvious when examining the deltas on the right. First, there is an apparent gap between the two service models, in which the mean waiting times in the scenarios with global optimization (blue) are between (0.5 ± 0.5) s and (3.4 ± 0.8) s shorter than in scenarios using only IRS (red). The gap grows with increasing fleet sizes, especially in scenarios using LBA as the IRS. Even clearer differences can be seen between the two evaluated IRSs, however. In scenarios using NNP (dashed lines) the mean waiting times are shorter than in those using LBA (solid lines), ranging from (2.1 ± 0.5) s in Scenario IRS–A & GO–SC with 1000 vehicles to (15.7 ± 1.0) s in Scenario IRS–O with 6000 vehicles.

This observation indicates that reassignments of requests and vehicles due to LBA and the potential increase in the number of accepted requests implied by this cause longer individual waiting times for customers. This effect outweighs the impact on waiting times due to periodic global reoptimization and grows with increasing fleet sizes.

4.2.2. Time window hardness

In both Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC) as in Scenario 3 'IRS acceptance & global optimization with soft constraints' (IRS–A & GO–SC) solutions are reoptimized periodically, while the initial decision as to whether a customer request is accepted or not is made by the IRS. The difference between the two service models is in the hardness of time window constraints. Unlike in Scenario 2 where solutions are strictly forbidden from containing assignments in which customers are picked up outside of their respective time window, in Scenario 3, such solutions are penalized, i.e. the objective function value is reduced as outlined in Section 3.2.2.

On comparing these two models, the results shown in Fig. 8 reveal on the one hand, that the empty mileage due to pickup trips in Scenario 3 is reduced compared to the more constrained global optimization in Scenario 2. On the other hand, fewer requests are served, which substantiates the observation made in Section 4.2.1 that the increased degrees of freedom from using the optimization

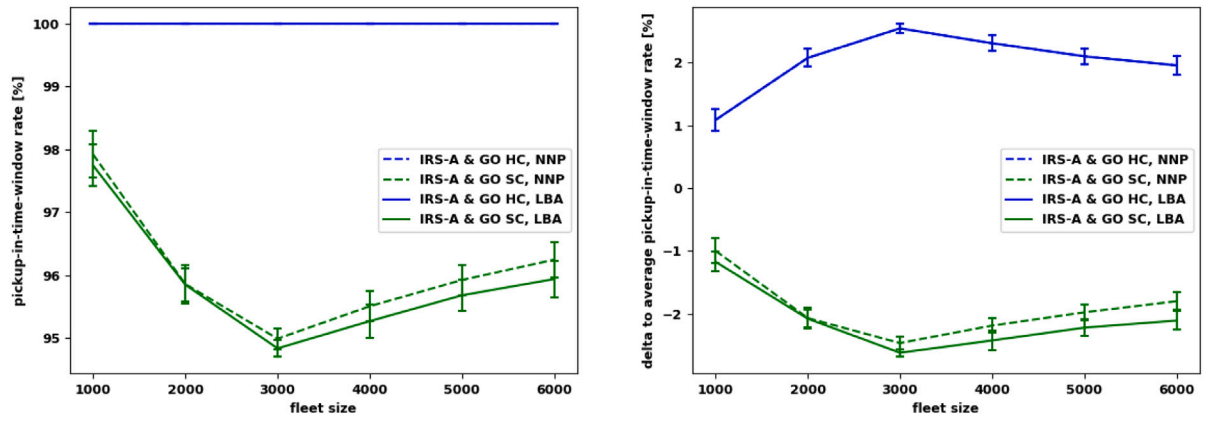


Fig. 9. Pickup-in-time-window rates of Scenario 2 'IRS acceptance & global optimization with hard constraints' (IRS–A & GO–HC, blue) and Scenario 3 'IRS acceptance & global optimization with soft constraints' (IRS–A & GO–SC, green) using NNP (dashed lines) and LBA (solid lines) as IRS, averaged over all simulated days (left) and delta to averages per simulated day (right).

potential of the dynamic problem tends to lead to solutions with fewer accepted customer requests overall due to the DOE. Unlike in Section 4.2.1, the benefit due to saved empty mileage cannot make up for the loss of paying customers in respect of generated profit in general. While for fleet sizes of up to 2000 vehicles, the service models generate very similar profits when the supply of ODM vehicles is overloaded with requests and the optimization potential is rather small, with fleet sizes of 3000 to 4000 vehicles, the difference in profit is up to $(618 ± 208) per day.

Both the decrease in average distances to pickup locations and the smaller number of served requests are also reflected in shorter average customer waiting times in Scenario 3 compared to Scenario 2. The gap widens with increasing fleet sizes using both IRSs, starting at (1.7 ± 0.5) s and (0.9 ± 0.6) s with NNP and LBA as IRS respectively for fleet sizes of 1000 vehicles, going up to (4.7 ± 0.5) s (NNP) and (5.5 ± 0.5) s (LBA) with 6000 vehicles.

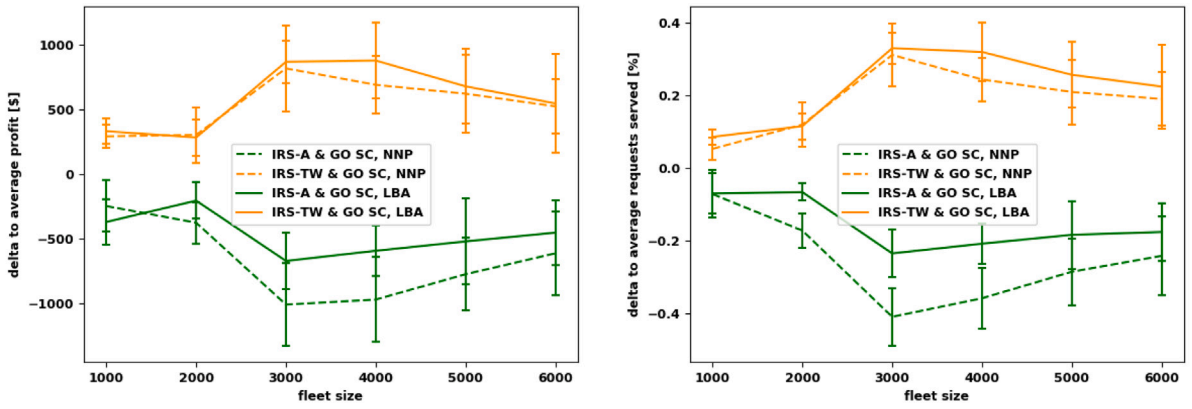
Both IRSs perform differently in these scenarios as well. Except for scenarios with 1000 vehicles, in which the generated profits are all similar and within the standard deviation of the evaluated cases, the LBA approach outperforms NNP, producing up to $(653 ± 373) more in scenarios with hard time window constraints ('HC') and up to $(368 ± 222) more in scenarios with soft time window constraints ('SC'). Solutions found with LBA as the IRS also include more served requests (up to (0.25 ± 0.12)% more in Scenario IRS–A & GO–HC and up to (0.17 ± 0.10)% in Scenario IRS–A & GO–SC) while also producing more empty mileage (up to (0.19 ± 0.01)% more in Scenario IRS–A & GO–HC and up to (0.17 ± 0.02)% in Scenario IRS–A & GO–SC) due to the fact that vehicles are reassigned and therefore make detours on their way to pickup locations. This also implies longer mean customer waiting times, leading to differences of between (2.9 ± 0.6) s for fleet sizes of 1000 vehicles in scenarios with soft constraints and (13.7 ± 0.5) s for fleet sizes of 6000 vehicles in scenarios with hard constraints.

Since with soft time window constraints solutions are allowed to include assignments with pickups outside of the respective pickup time windows, the PUITWR drops from a hundred percent in all simulations using hard constraints to between approximately 95 % and 98 % in such soft constraint scenarios as shown in Fig. 9 on the left. The minimum PUITWR is reached in scenarios with 3000 vehicles, indicating that the higher the optimization potential and, in turn, the profit generated, the lower the PUITWR tends to be, because more requests are reassigned, increasing the chances of being assigned to a vehicle that picks up the request outside of the respective time window. Fig. 9 shows the deltas on the right, but for this specific KPI, there is no deeper insight other than the adjusted error bars based on the standard deviations. The results show that the NNP performs slightly better than LBA, albeit generally within the standard deviations.

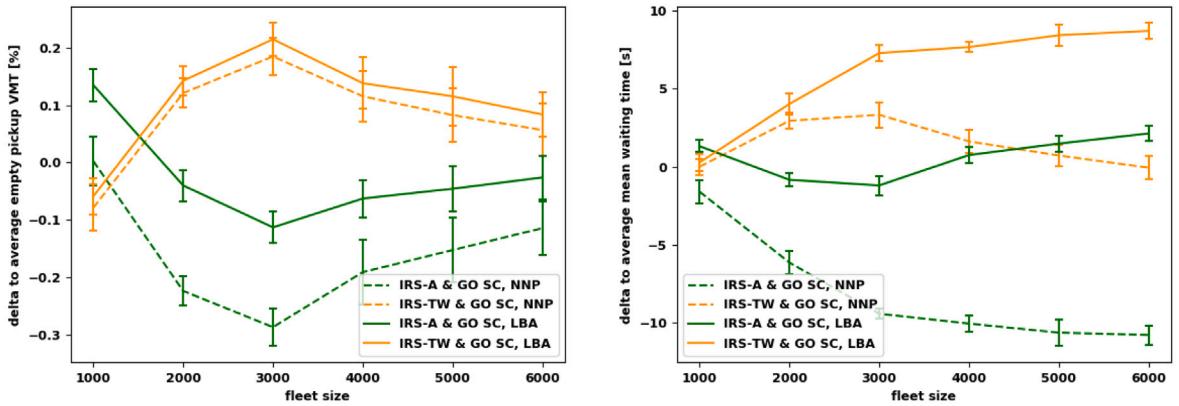
4.2.3. Acceptance decision

Scenario 3 'IRS acceptance & global optimization with soft constraints' (IRS–A & GO–SC) and Scenario 4 'IRS time windows & global optimization with soft constraints' (IRS–TW & GO–SC) both use soft time window constraint in the global (re)optimization of assignments. The difference between these two scenarios lies in the decision making process by which customer requests are accepted. In the case of IRS–A & GO–SC the IRS is responsible for selecting those requests that are accepted. The respective customers are then immediately informed about the decision. In the service model used in IRS–TW & GO–SC, customers are only informed about their time windows immediately, if the IRS forecasts that they will be accepted, i.e. the time window satisfies the maximum waiting time constraint. If this is not the case, customers have to wait until the next global optimization takes place before receiving a response. In real-world applications, longer response times would be unacceptable to many users, eventually leading to fewer people using the service eventually. The time window found by the IRS defines the time frame in which assignments are expected to take place according to the respective IRS used in the scenario. Any pickup that does not take place within its respective expected time frame, implies a defined penalty as in Scenario 3, described in Section 3.2.2.

On comparing the two service models in the scenarios, the most obvious difference is the increased percentage of accepted requests and the resulting increase in profit, both of which are shown in Fig. 10a. In scenarios with fleet sizes of 3000 vehicles, the



(a) Delta to average in total profits and relative requests served



(b) Delta to averages in relative empty vehicle miles travelled (without repositioning) and mean customer waiting times

Fig. 10. KPIs of Scenario 3 'IRS acceptance & global optimization with soft constraints' (IRS–A & GO–SC, green) and Scenario 4 'IRS time windows & global optimization with soft constraints' (IRS–TW & GO–SC, orange) using NNP (dashed lines) and LBA (solid lines) as IRS, delta to averages per simulated day.

increase in accepted requests of (0.72 ± 0.16)% and (0.56 ± 0.11)% in scenarios using NNP and LBA respectively leads to benefits of $(1830 ± 654) (NNP) and $(1543 ± 387) (LBA) respectively in terms of profit generated for the service provider.

The empty mileage due to pickup trips is slightly smaller in scenarios with IRS acceptance. The priority of global optimization to accept as many requests as possible according to the control function F_{con} is, however, still discernible. The total empty mileage might be higher, but the additional accepted customers outweigh the additional costs implied by this. This weighing of solutions is possible in Scenario 4, while in Scenario 3, the IRS decides which requests are part of the optimization, essentially only allowing the global optimization algorithm to minimize the driven mileage. In scenarios with 3000 vehicles or more, global optimization is not under so much pressure to find available vehicles that can be assigned to requests. The larger number of possible assignments implies an increased optimization potential, leading to an even bigger difference in accepted requests as well as a decreasing difference in empty pickup VMT, as depicted in Fig. 10b on the left. Note that the optimization potential is highest for fleet sizes of around 3000 vehicles. In scenarios with larger fleets, global optimization tends to find solutions similar to those found with IRS, because the supply of available vehicles is higher and more often than not, the closest vehicle is the global optimal solution.

Besides the aforementioned drawback of a longer time waiting for a response, the service design of Scenario 4 also induces longer customer waiting times between request time and pickup time. In Scenario 4, idle vehicles do not move before the global optimization assigns them to a request. In the event that a customer–vehicle assignment is not changed by global optimization, a vehicle could on average have driven half of the optimization period sooner. On the right of Fig. 10b the deltas to average in mean customer waiting times can be seen. Outside of scenarios with small fleet sizes (1000 vehicles), in which idle vehicles are very rare and the described effect is not obvious, the increase in mean waiting time between the service models of Scenarios 3 and 4 amounts to between (4.9 ± 1.1) s and (8.5 ± 1.1) s if using LBA as the IRS and between (9.1 ± 1.2) s and (12.7 ± 1.1) s if it is NNP.

When evaluating the differences between the two scenarios with respect to PUITWR, Fig. 11 gives a clear indication that when deploying the service model used in Scenario 4, the prediction accuracy declines. In particular, in scenarios with small fleet sizes the gap between Scenarios 3 and 4 is immense. Unlike Scenario 3, in which the PUITWR is lowest in scenarios with the highest

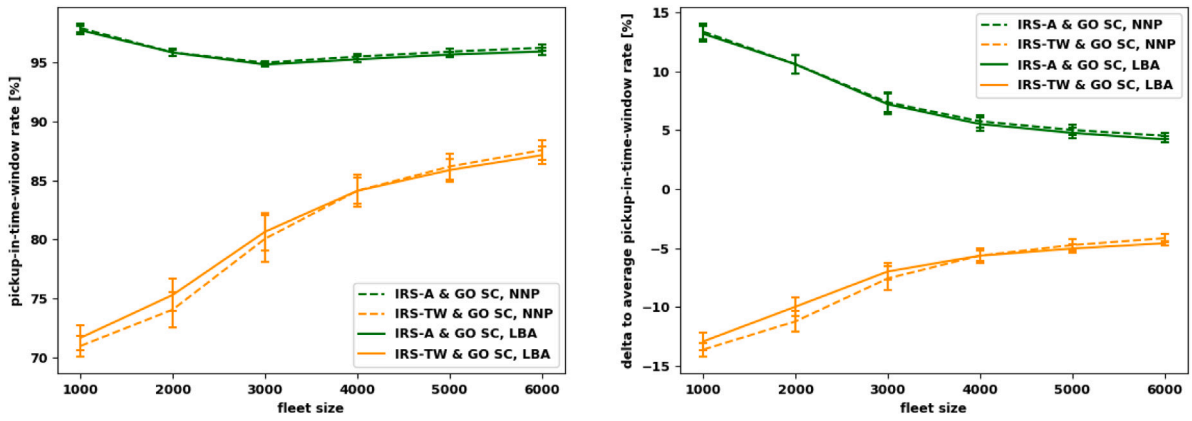


Fig. 11. Pickup-in-time-window rates of Scenario 3 'IRS acceptance & global optimization with soft constraints' (IRS–A & GO–SC, green) and Scenario 4 'IRS time windows & global optimization with soft constraints' (IRS–TW & GO–SC, orange) using NNP (dashed lines) and LBA (solid lines) as IRS, averaged over all simulated days (left) and delta to averages per simulated day (right).

optimization potential, with around 95 % for fleet sizes of 3000 vehicles, in Scenario 4 the smaller the fleet is, the worse is the ability of both IRSs to predict the correct time windows for customer pickups. In scenarios with 1000 vehicles, only about 72 % of pickups take place within the time windows predicted by the IRSs, while LBA performs slightly better than NNP in these instances, albeit within the standard deviations. This relatively low percentage indicates that in scenarios with small fleet sizes, both of the IRSs considered tend to fail to find feasible assignments for new requests. However, in Scenario 4 the decision whether requests are accepted is made based on global optimization. This leads to a different set of requests being accepted than anticipated by the respective IRS. Since the PUITWR–KPI considers requests that are falsely assumed to be rejected by an IRS as incorrect, as described in Section 4.1.3, the respective PUITWR drops. In scenarios with small fleet sizes, the sets of accepted requests may change even more than in scenarios with more vehicles, in which most of the requests are accepted, therefore the PUITWR is even lower in these cases.

Especially towards the end of optimization periods, earlier assignments projected by an IRS reduce the number of vehicles available for new requests. This effect is more prominent in the case of NNP, when vehicles already assigned are completely blocked from being assigned to another new request. However, the effect is also observable in the reduced number of feasible assignments when using LBA as the IRS. The rate of correctly predicted time windows increases steadily for the fleet sizes considered up to the point when around 87 % of all time window predictions are correct in scenarios with 6000 vehicles.

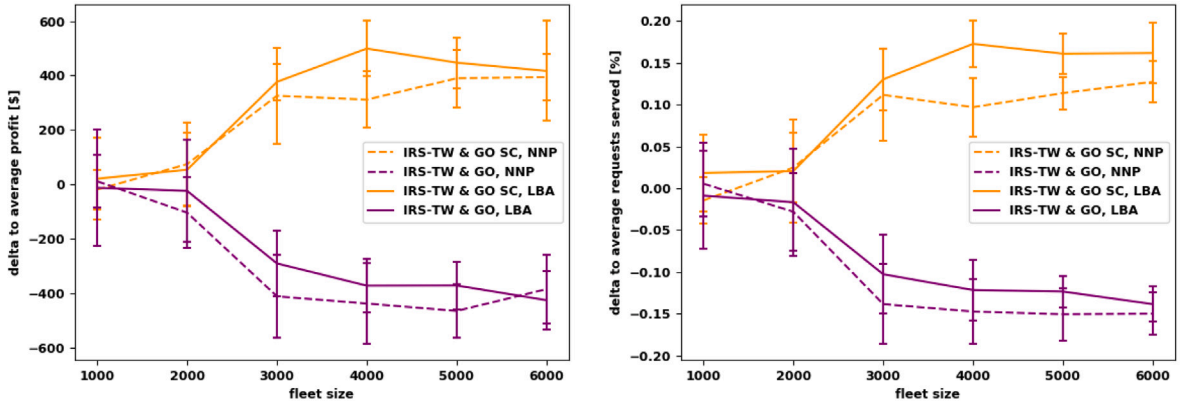
4.2.4. Unconstrained optimization

The evaluated aspect between Scenario 4 'IRS time windows & global optimization with soft constraints' (IRS–TW & GO–SC) and Scenario 5 'IRS time windows & global optimization' (IRS–TW & GO) is the difference between constrained and unconstrained global optimization. In Scenario 5, the time windows predicted by the IRS when a customer requests a ride do not have any impact when assignments are optimized globally, while in Scenario 4 assignments outside of the respective time windows imply a penalty in the control function F_{con} as described above. In theory, this allows the full optimization potential to be used when (re)assigning customers to vehicles and as well as clarification of which IRS is able to predict correct pickup time windows more often.

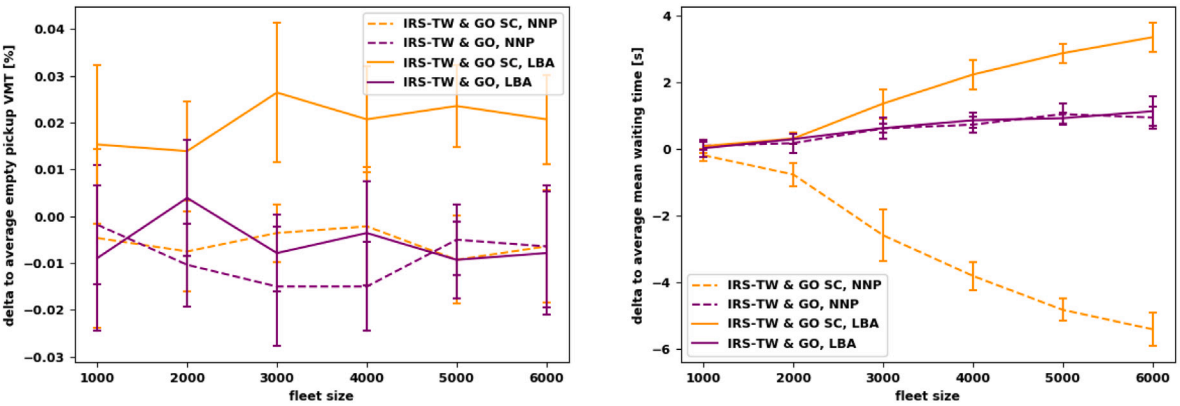
As shown in Fig. 12a the number of customers served and the profit generated are smaller in Scenario 5 than in Scenario 4, especially in instances with fleet sizes of 3000 or more vehicles. The difference in profit is up to \$(855 ± 203) (NNP) and \$(872 ± 201) (LBA) respectively, while the percentage of accepted requests is up to (0.28 ± 0.05)% (NNP) and (0.30 ± 0.06)% (LBA) higher in Scenario 4. This directly contradicts the assumption that the results would be better in terms of the objectives of the service provider in scenarios with unconstrained global optimization. On the other hand, these findings match those made in Section 4.2.2, where the performance in terms of profit and accepted customer requests was worse in scenarios with less constrained optimization (RS–A & GO–SC) than in scenarios with harder constraints (RS–A & GO–HC). This is another indicator that global optimization tends to lead to solutions that are optimal with respect to the current situation but leaves the fleet in a worse state for accepting future requests that it is not yet aware of. This aspect could probably be improved if the global control function F_{con} contained non-myopic terms and thus also enabled a non-myopic accept/reject decision by the global optimization.

Fig. 12b shows the deltas to average in empty pickup mileage and mean customer waiting time. Besides the observation that the empty mileage is very similar in both scenarios and the waiting times in Scenario 5 are between the waiting times of Scenario 4 with NNP and LBA as IRS respectively, it can be seen that in all instances of Scenario 5 differences between NNP and LBA in the considered KPIs are very small and within the respective standard deviations. This only makes sense, since the assignment of requests happens completely independently from the IRS in Scenario RS–A & GO.

When examining the ability to predict pickup time windows, the differences are more prominent. As in Scenario RS–A & GO–SC the PUITWR of both IRSs increases steadily with increasing fleet size. As shown in Fig. 13 on the left, for fleet sizes of 1000 vehicles,



(a) Delta to average in total profits and relative requests served



(b) Delta to averages in relative empty vehicle miles travelled (without repositioning) and mean customer waiting times

Fig. 12. KPIs of Scenario 4 'IRS time windows & global optimization with soft constraints' (IRS–TW & GO–SC, orange) and Scenario 5 'IRS time windows & global optimization' (IRS–TW & GO, purple) using NNP (dashed lines) and LBA (solid lines) as IRS, delta to averages per simulated day.

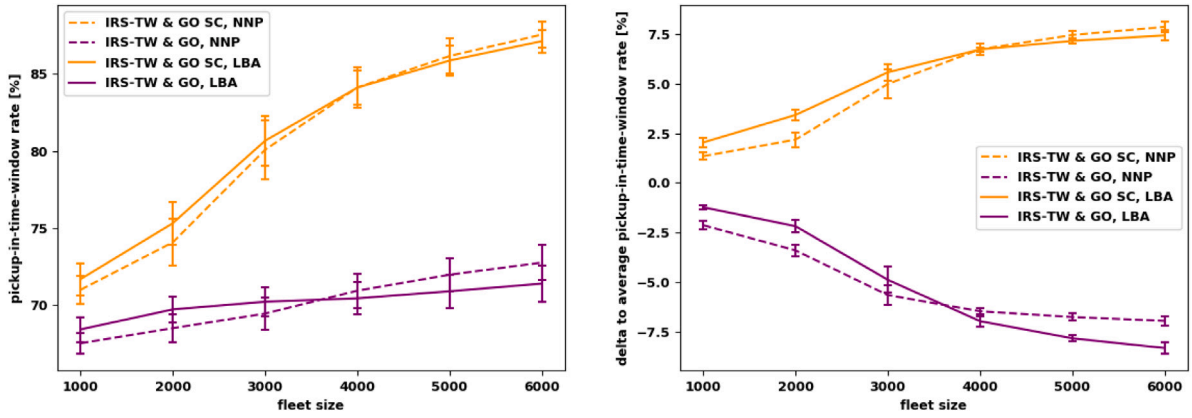


Fig. 13. Pickup-in-time-window rates of Scenario 4 'IRS time windows & global optimization with soft constraints' (IRS–TW & GO–SC, orange) and Scenario 5 'IRS time windows & global optimization' (IRS–TW & GO, purple) using NNP (dashed lines) and LBA (solid lines) as IRS, averaged over all simulated days (left) and delta to averages per simulated day (right).

it is about 68 %, while with 6000 vehicles it reaches approximately 72 %. Hence, as expected, the overall ability to predict correct pickup time windows is considerably lower if no penalties are implied to steer assignments such that solutions in which pickups take place within the respective time windows are favored. On the right, the deltas to average show a clearer picture with respect to the relative performances of NNP and LBA in the respective scenarios. In scenarios with both use cases, the LBA approach outperforms NNP if the fleet size considered is smaller than 4000 vehicles. In the case of scenarios with 2000 vehicles, the difference amounts to (1.25 ± 0.62)% (RS–A & GO–SC) and (1.22 ± 0.64)% (RS–A & GO) respectively. In instances with 4000 vehicles or more, though, the PUITWR is higher when using NNP as the IRS rather than LBA in both service models. When considering fleets of 6000 vehicles, in the case of Scenario 4 NNP outperforms LBA by (0.43 ± 0.52)%, in Scenario 5 the difference is (1.37 ± 0.52)%.

This observation implies that the LBA approach is better at predicting time windows in scenarios, in which demand is high compared to available vehicles. As outlined in Section 4.1.3, requests that are assumed to be rejected by the IRS but are later accepted by global optimization are treated as incorrect predictions, which implies that in these scenarios, the LBA approach is also more often able to predict which requests are going to be accepted at the end each optimization period. On the other hand, in scenarios in which there are enough vehicles to serve almost every request in time, the feature of LBA that allows it to reassign customers before global optimization takes place leads to more instances in which these reassigned customers are picked up outside of the initially generated time windows.

4.2.5. Computation time

In addition to the KPIs evaluated in the former subsections, another important indicator for system performance is the computation time associated with each service model. All simulations examined in this study are executed on a single central processing unit (CPU) of an Intel Xeon Silver 4114 processor with ten physical cores at 2.20 GHz and 64 GB random access memory (RAM).

Fig. 14 presents the mean computation times with standard deviations for all scenarios considered, in terms of total computation times (Fig. 14a) and the delta to average (Fig. 14b). The term ‘computation time’ in this context refers to the time it takes to finish one 24h-simulation, including optimization, but also data management and processing. Therefore, it scales almost linearly with increasing fleet size, not only because in each global optimization the solution space is potentially larger, but also because the movement of every single vehicle needs to be simulated.

In Scenario 1 ‘IRS-O’, no global optimization takes place, hence the computation time is the lowest out of all scenarios considered. In the case of NNP, such scenarios come with mean computation times of (2491 ± 160) s to (13 074 ± 1798) s for fleet sizes of 1000 and 6000 respectively, while simulations with LBA as IRS take slightly longer on average, ranging from (2666 ± 227) s (1000 vehicles) to (14 146 ± 2087) s (6000 vehicles). This is due to the additional processing necessary when using LBA as IRS, as explained in 3.3.2.

In scenarios with initial assignments based on an IRS and global optimization with hard (Scenario 2) and soft constraints (Scenario 3), additional computation time is spent on the periodic global optimization of assignments, which is reflected by longer total computation times compared to Scenario 1 for both IRSs. The scaling behavior in respect of the fleet size is similar for all of the scenarios, however, for larger problem sizes the difference between Scenario 2 and Scenario 3 increases, up to scenarios with 6000 vehicles, in which the delta to average amounts to (2425 ± 1064) s (NNP) and (2472 ± 1031) s (LBA). This implies that global optimization with soft time window constraints takes longer than those with hard constraints, which is plausible, since more vehicles are potentially allowed to be assigned to a request, which increases the number of potential assignments and calculations of associated costs as well as the solution space and therefore the time it takes to find the optimal solution.

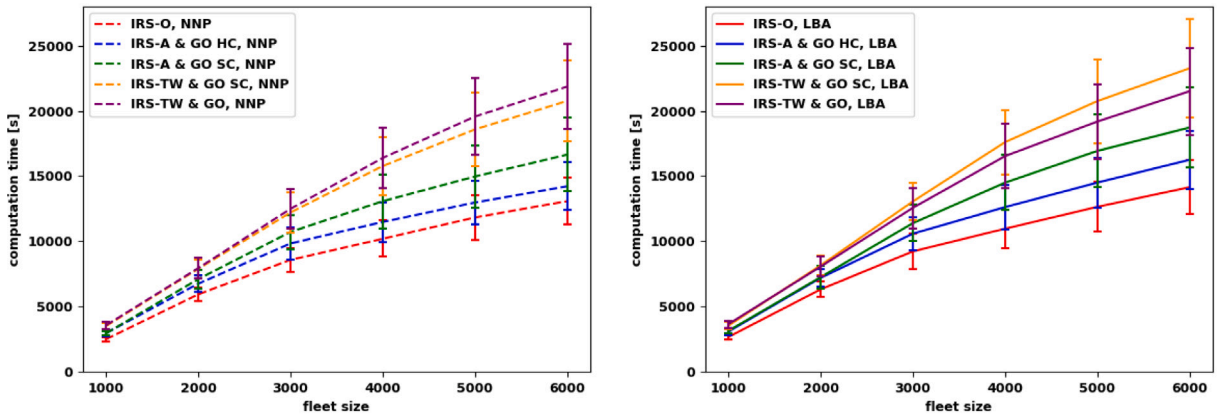
For the same reasons, the computation time is higher in scenarios in which the global optimization considers every request at least once. The respective IRS only generates pickup time windows but is not involved in the decision if a request is accepted or not. This increase in requests to consider causes longer computation times compared to the other scenarios, because more potential assignments imply a larger solution space and, again, longer computation times for global optimization. There is no clear trend in computation times between Scenarios 4 and 5: in scenarios with large fleet sizes in which NNP is used as IRS, the service model of Scenario 5 induces slightly longer computation times than the one of Scenario 4 albeit within the respective standard deviations, with a delta to average of (997 ± 1296) s for fleet sizes of 6000. In instances with LBA as IRS, the opposite is the case: the computation times of Scenario 4 surpass those of Scenario 5, with a delta to average of (1762 ± 1545) s for fleet sizes of 6000.

5. Conclusion

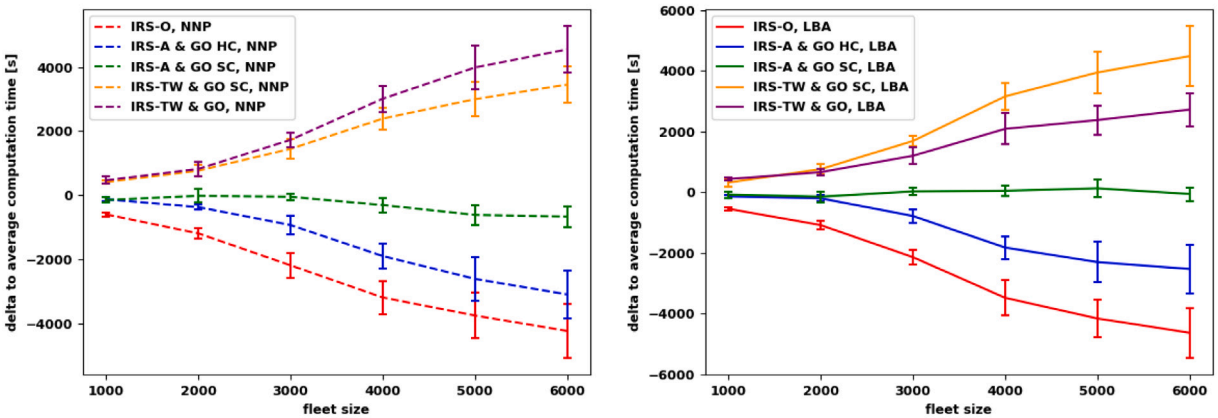
5.1. Summary

In today’s cities, it is recognized that traffic problems will only become bigger as the trend towards urbanization continues. It is anticipated that business models similar to that of Uber and Lyft will become part of the solution to these problems as it would lead to an overall reduction in the total numbers of private vehicles on the road. In order to improve the quality and profitability of such an on-demand mobility (ODM) service, assignments of vehicles from an ODM fleet and users of the service need to be made by an algorithm that is able to (a) find a globally close-to optimal solution and (b) quickly inform the customer of the match.

While the research areas of ODM and associated problems such as the dial-a-ride-problem have been investigated more and more in recent years, there is still a gap between research and the approaches used in real-world applications. Instead of complex optimization techniques, ODM operators often use fast heuristic methods to ensure that customers are informed about whether they will be picked up as soon as possible.



(a) Total computation times



(b) Delta to averages in total computation times

Fig. 14. Overview of computation times for 24-hour simulations by scenario and fleet size.

The present study addresses this gap by evaluating a two-step information system that is designed to make use of the optimization potential in the dynamic car–passenger matching problem formulated in it while quickly deciding whether or not a customer request is accepted and in which time window the pickup will take place. This paper examines two immediate response strategies (IRSs): the nearest neighbor policy (NNP) and the list-based assignments (LBA) approach. Both IRSs are tested in respect of (a) the respective system performances with varying degrees of hardness of the implied time windows and (b) their ability to predict the correct pickup time window in scenarios in which the final assignments are independent of the IRS-based time window constrains.

The simulation framework, the model and the scenarios examined in this study were all designed with the goal of representing close-to-real service applications. The data used in the simulations corresponds to the actual demand in Manhattan. Hence, the gap to use cases relevant for ODM service providers is considered to be small and the level of applicability relatively high.

It is shown that the LBA approach outperforms the NNP in terms of served requests and profit in most of the scenarios tested. At the same time, average empty vehicle mileage, customer waiting times and computation times are higher, indicating a worse user experience and a negative impact on traffic. It was found that with sufficiently large ODM fleets, the periodic global optimization process effectively decreases the empty mileage driven, resulting in a higher profit than in scenarios that depend solely on the IRS. The empty mileage was reduced even further in scenarios with soft time window constraints. The average profit was smaller due to an increase in rejected requests, though. This also explains the small margin of benefit between the examined scenarios proportional to the total profit generated.

The evaluation of the impact of immediate responses on system performance in general confirmed the assumption that the number of requests served and the overall profit would rise if the decision as to whether a request is accepted or not is based on global optimization instead of a heuristic procedure. However, the average time between request and response consequently becomes half the optimization period, accompanied by considerably longer mean waiting times for customers. Finally, it was found that when no time window constraints were induced by an IRS, the time window predictions in scenarios with both LBA and NNP were correct in 68 % to 72 % of cases, with the figure slightly increasing with increasing fleet sizes. LBA outperforms NNP in scenarios with fewer ODM vehicles, while for bigger fleets it is the other way around. This outcome reflects the ability of LBA to predict the acceptance

decision produced by global optimization for more requests in scenarios with small fleet sizes. In scenarios with large fleet sizes, in which most of the requests are accepted anyway, its intrinsic feature that allows reassignments (unlike NNP) causes increased deviations from the correct pickup times.

5.2. Discussion

The results presented in this work suggest that there is only a small difference in monetary benefits between the service models considered in relation to the overall profits generated with ODM services in areas like Manhattan, where the demand is very high. However, we want to emphasize that a daily surplus of around $300 accumulates to over $100,000 per year. There are virtually no additional costs for the operator to run one of the presented service models compared to common methods or between themselves besides the differences in computation times, which may imply costs for hardware. Since the simulations in this study are performed on single processors, it is fair to assume that additional costs should be low. It is possible that the varying customer experiences and perceived service qualities related to the scenarios will have an impact on demand and thereby profitability.

The findings of this study estimate the impact of various service models. These will help ODM service providers to decide whether to use a service model based on a two-step information system and to determine what benefits and weaknesses each approach offers. Scenarios 1, 2 and 3 include shorter response times for the initial customer requests because they use a quick IRS in order to provide a pickup time window but tend to be less profitable than Scenarios 4 and 5. Scenarios with hard time windows outperform scenarios with soft ones, which indicates a strong dynamic optimization effect as explained in Section 4.2.1. This implies that it would be very beneficial for the global optimization to include information about predicted demand in the objective function or for the operator to make use of a more sophisticated repositioning algorithm alongside the global optimization of assignments between vehicles and customers of the service.

Additional to these operator-specific findings, transportation models can benefit from the advancement of IRS systems. The utilization of a simulation framework with immediate response by both operator and customer achieves an equilibrium of demand and supply (Dandl et al., 2021). In such models, the customer accept/reject decision is replaced by a traveler mode-choice model. As the operator's immediate response returns an offer based on the real-time state of the fleet, the ODM system will only generate more demand if it can provide supply with sufficient level-of-service.

5.3. Future work

It is left to future research to further investigate the dynamic effects of optimization that caused more rejected requests in scenarios with fewer time window constraints, e.g. by examining varying control functions with non-myopic terms and their implications on computation times and performance indicators. Also, a more realistic customer model that includes reservation requests and heterogeneous maximum waiting times would further increase the applicability of results to ODM operators. The simulation framework could be upgraded by using dynamic travel times and travel time predictions and varying routing methods (e.g. routes that are faster or more reliable) could be tested to improve real-life customer experience. The evaluation of different IRSs could include other heuristic approaches or machine learning techniques. Especially for Scenarios 4 and 5, data-based learning of time windows could be very interesting, also to include cancellations by customers. Immediate response strategies are even more interesting and challenging when applied to ride pooling systems, in which both pickup and detour times have to be predicted. Finally, the combination of an IRS and periodic optimization will help to build better transportation models, where quick responses of trip characteristics based on the current fleet state can be used for traveler mode choice models, while optimization procedures ensure high-quality fleet operation.

CRediT authorship contribution statement

Marvin Erdmann: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Florian Dandl:** Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Writing - review & editing. **Klaus Bogenberger:** Conceptualization, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix

See Table A.1.

Table A.1
Overview of notation.

Symbol	Parameter
c_{dist}	Distance costs
c_{fix}	Vehicle fixed costs
$c_{i,\text{rep}}$	Costs produced by vehicle $i \in I$ due to repositioning
d_i	Mileage driven by vehicle $i \in I$
d_{ij}	Empty distance required for vehicle $i \in I$ to pick up customer $j \in J$
$d_{i,\text{rep}}$	Distance driven by vehicle $i \in I$ due to repositioning
d_j^{od}	Distance between origin and destination of customer $j \in J$
F_{con}	Control function
F_{obj}	Global objective function
$I,\ I(j)$	Set of vehicles, subset that allows feasible assignments with request $j \in J$
$J,\ J(i)$	Set of customers, subset that allows feasible assignments with vehicle $i \in I$
P	Penalty for violations against soft constraints
p	Assignment reward
p_{acc}	Accepted request assignment reward
p_{base}	Service Base fare
p_{dist}	Distance fare
t_{ij}	Time to arrival of vehicle $i \in I$ at the pickup location of request $j \in J$
$t_{j,\text{pu}}$	Pickup time of customer $j \in J$
$t_{j,\text{req}}$	Request time of customer $j \in J$
$t_{j,\text{tw}}$	Beginning of time window associated with customer $j \in J$
t_{max}	Maximum waiting time
t_{twl}	Time window length
v_{ij}	Decision variable for an assignment of vehicle $i \in I$ and customer $j \in J$
δ_{ij}	Indicator if an assignment of vehicle $i \in I$ and customer $j \in J$ violates constraints

References

Al-Kanj, Lina, Nascimento, Juliana, Powell, Warren B., 2020. Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. European J. Oper. Res. (ISSN: 03772217) http://dx.doi.org/10.1016/j.ejor.2020.01.033.

Alonso-Mora, Javier, Samaranayake, Samitha, Wallar, Alex, Frazzoli, Emilio, Rus, Daniela, 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. 462–467. http://dx.doi.org/10.1073/pnas.1611675114.

Bergvinsdottir, Kristin Berg, Larsen, Jesper, Jorgensen, Rene Munk, 2007. Solving the dial-a-ride problem using genetic algorithms. J. Oper. Res. Soc. 58 (10), 1321–1331. http://dx.doi.org/10.1057/palgrave.jors.2602287.

Bimpikis, Kostas, Candogan, Ozan, Saban, Daniela, 2019. Spatial pricing in ride-sharing networks. Oper. Res. (ISSN: 0030-364X) 67 (3), 744–769. http://dx.doi.org/10.1287/opre.2018.1800.

Cordeau, Jean-Francois, Laporte, Gilbert, 2005. Tabu search heuristics for the vehicle routing problem. In: Sharda, Ramesh, Voß, Stefan, Rego, César, Alidaee, Bahram (Eds.), Metaheuristic Optimization Via Memory and Evolution: Tabu Search and Scatter Search. Springer US, Boston, MA, ISBN: 978-0-387-23667-4, pp. 145–163. http://dx.doi.org/10.1007/0-387-23667-8_6.

Cordeau, Jean-François, Laporte, Gilbert, 2007. The dial-a-ride problem: models and algorithms. Ann. Oper. Res. (ISSN: 0254-5330) 153 (1), 29–46. http://dx.doi.org/10.1007/s10479-007-0170-8.

Dandl, Florian, Bogenberger, Klaus, 2019. Comparing future autonomous electric taxis with an existing free-floating carsharing system. IEEE Trans. Intell. Transp. Syst. (ISSN: 1524-9050) 20 (6), 2037–2047. http://dx.doi.org/10.1109/TITS.2018.2857208.

Dandl, Florian, Bogenberger, Klaus, Mahmassani, Hani S., 2019a. Autonomous mobility-on-demand real-time gaming framework. In: 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, ISBN: 978-1-5386-9484-8, pp. 1–10. http://dx.doi.org/10.1109/MTITS.2019.8883286.

Dandl, Florian, Engelhardt, Roman, Hyland, Michael, Tilg, Gabriel, Bogenberger, Klaus, Mahmassani, Hani S., 2021. Regulating mobility-on-demand services: Tri-level model and Bayesian optimization solution approach. Transp. Res. C (ISSN: 0968090X) 125 (1), 103075. http://dx.doi.org/10.1016/j.trc.2021.103075.

Dandl, Florian, Hyland, Michael, Bogenberger, Klaus, Mahmassani, Hani S., 2019b. Evaluating the impact of spatio-temporal demand forecast aggregation on the operational performance of shared autonomous mobility fleets. Transportation (ISSN: 0049-4488) 46 (6), 1975–1996. http://dx.doi.org/10.1007/s11116-019-10007-9.

Engelhardt, Roman, Dandl, Florian, Bogenberger, Klaus, 2020. Speed-up Heuristic for an on-demand ride-pooling algorithm. arXiv:2007.14877 [eess.SY].

Erdmann, Marvin, Dandl, Florian, Bogenberger, Klaus, 2019. Dynamic car-passenger matching based on tabu search using global optimization with time windows. In: 8th International Conference on Modeling, Simulation and Applied Optimization.

Erdmann, Marvin, Dandl, Florian, Kaltenhaeuser, Bernd, Bogenberger, Klaus, 2020. Dynamic car-passenger matching of online and reservation requests. In: 99th Annual Meeting of the Transportation Research Board.

Fagnant, Daniel J., Kockelman, Kara., Bansal, Prateek, 2015. Operations of shared autonomous vehicle fleet for austin, texas market. Transp. Res. Rec. J. Transp. Res. Board (ISSN: 0361-1981) 2536, 98–106. http://dx.doi.org/10.3141/2536-12.

Henao, Alejandro, Marshall, Wesley E., 2018. The impact of ride-hailing on vehicle miles traveled. Transportation 46, 2173—2194. http://dx.doi.org/10.1007/s11116-018-9923-2.

Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K.W., 2019. Fleet operational policies for automated mobility: A simulation assessment for zurich. Transp. Res. C (ISSN: 0968090X) 102, 20–31. http://dx.doi.org/10.1016/j.trc.2019.02.020.

Hyland, Michael, Mahmassani, Hani S., 2017. Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility. Transp. Res. Rec. J. Transp. Res. Board (ISSN: 0361-1981) 2653, 26–34. http://dx.doi.org/10.3141/2653-04.

Hyland, Michael, Mahmassani, Hani S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transp. Res. C (ISSN: 0968090X) 92, 278–297. http://dx.doi.org/10.1016/j.trc.2018.05.003.

Hyland, Michael, Mahmassani, Hani S., 2020. Operational benefits and challenges of shared-ride automated mobility-on-demand services. Transp. Res. A (ISSN: 0965-8564) 134, 251–270. http://dx.doi.org/10.1016/j.tra.2020.02.017.

Maciejewski, Michal, Bischoff, Joschka, 2015. Large-scale microscopic simulation of taxi services. Procedia Comput. Sci. (ISSN: 1877-0509) 52, 358–364. http://dx.doi.org/10.1016/j.procs.2015.05.107.

Maciejewski, Michal, Bischoff, Joschka, Nagel, Kai, 2016. An assignment-based approach to efficient real-time city-scale taxi dispatching. IEEE Intell. Syst. (ISSN: 1541-1672) 31 (1), 68–77. http://dx.doi.org/10.1109/MIS.2016.2.

Narayanan, Santhanakrishnan, Chaniotakis, Emmanouil, Antoniou, Constantinos, 2020. Shared autonomous vehicle services: A comprehensive review. Transp. Res. C (ISSN: 0968090X) 111, 255–293. http://dx.doi.org/10.1016/j.trc.2019.12.008.

Nazari, Mohammadreza, Oroojlooy, Afshin, Takáč, Martin, Snyder, Lawrence V., 2018. Reinforcement learning for solving the vehicle routing problem. In: Advances in Neural Information Processing Systems 31. NIPS.

Nourinejad, Mehdi, Ramezani, Mohsen, 2019. Ride-sourcing modeling and pricing in non-equilibrium two-sided markets. Transp. Res. Procedia (ISSN: 23521465) 38, 833–852. http://dx.doi.org/10.1016/j.trpro.2019.05.043.

NYC Taxi & Limousine Commision, 2018. TLC trip record data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page [Accessed: September 2020].

Pandi, Ramesh Ramasamy, Ho, Song Guang, Nagavarapu, Sarat Chandra, Tripathy, Twinkle, Dauwels, Justin, 2018. Gpu-accelerated tabu search algorithm for dial-a-ride problem. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, ISBN: 978-1-7281-0321-1, pp. 2519–2524. http://dx.doi.org/10.1109/ITSC.2018.8569472.

Psaraftis, Harilaos N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. 14 (2), 130–154. http://dx.doi.org/10.1287/trsc.14.2.130.

Sarma, Navjyoth, Nam, Daisik, Hyland, Michael, de Souza, Felipe Augusto, Yang, Dingtong, Ghaffar, Arash, Verbas, Omer, 2020. Effective and efficient fleet dispatching strategies for dynamically matching AVs to travelers in large-scale transportation systems. In: 2020 23rd International Conference on Intelligent Transportation Systems(ITSC). IEEE.

Sheridan, Patricia Kristine, et al., 2013. The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. Transp. Res. A (ISSN: 0965-8564) 49, 178–194. http://dx.doi.org/10.1016/j.tra.2013.01.032, URL: http://www.sciencedirect.com/science/article/pii/S0965856413000396.

Spieser, Kevin, Samaranayake, Samitha, Gruel, Wolfgang, Frazzoli, Emilio, 2016. Shared-vehicle mobility-on-demand systems: Fleet operator's guide to rebalancing empty vehicles. In: TRB Annual Meeting.

Syed, Arslan Ali, Akhnoukh, Karim, Kaltenhäuser, Bernd, Bogenberger, Klaus, 2019a. Neural network based large neighborhood search algorithm for ride hailing services. In: Moura Oliveira, Paulo, Novais, Paulo, Reis, Luís Paulo (Eds.), Progress in Artificial Intelligence. In: Lecture Notes in Computer Science, vol. 11804, Springer International Publishing, Cham, ISBN: 978-3-030-30240-5, pp. 584–595. http://dx.doi.org/10.1007/978-3-030-30241-2{_}49.

Syed, Arslan Ali, Kaltenhäuser, Gaponova, Irina, Bogenberger, Klaus, 2019b. Asynchronous adaptive large neighborhood search algorithm for dynamic matching problem in ride hailing services. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 3006–3012. http://dx.doi.org/10.1109/ITSC.2019.8916943.

United Nations and Department of Economic and Social Affairs, Population Divison, 2018. World Urbanization Prospects: The 2018 Revision. United Nations, New York, ISBN: 978-92-1-148319-2, http://dx.doi.org/10.1007/0-387-23667-8_6.

Zhang, Rick, Pavone, Marco, 2016. Control of robotic mobility-on-demand systems: A queueing-theoretical perspective. Int. J. Robot. Res. (ISSN: 0278-3649) 35 (1–3), 186–203. http://dx.doi.org/10.1177/0278364915581863, URL: http://ijr.sagepub.com/content/35/1-3/186.full.pdf.

Zhang, Rick, Rossi, Federico, Pavone, Marco, 2016. Model predictive control of autonomous mobility-on-demand systems. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1382–1389. http://dx.doi.org/10.1109/ICRA.2016.7487272, URL: http://arxiv.org/pdf/1509.03985.pdf.