Australasian Transport Research Forum 2023 Proceedings 29
November – 1 December, Perth, Australia
Publication website: http://www.atrf.info

Quantifying the benefits of short-term reservations in on-demand ridepooling

Nander Theodoridis[1], Andres Fielbaum[1,2,*], Javier Alonso-Mora[1], and Bilge Atasoy[1]

[1]Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, The Netherlands
[2]School of Civil Engineering, University of Sydney, Australia
[*]Corresponding author: andres.fielbaum@sydney.edu.au

Abstract

Grouping the users and assigning them to vehicles is one of the most complex and relevant problems in on-demand ridepooling systems. One of the main challenges is that as decisions are taken on-demand, there is a lack of knowledge about future users. In this paper, we anal- yse the benefits of getting some information shortly in advance, namely, having a percentage of the users requiring their trips some minutes (between 5-20) before their desired pickup. To this end, we modify the assignment algorithm by Alonso Mora et al. (2017) to admit passengers revealed shortly in advance, and propose a number of heuristics to deal with the increased complexity. We show that this little extra information can have a significant pos- itive impact on the system's KPIs, e.g., increasing the service rate from 86% to 92% in our simulations in Manhattan, without increasing the fleet size or vehicles-kilometres-travelled. Moreover, we show that this information is also beneficial even if i) Users that reserve in advance are prioritised when deciding whom to serve and ii) We consider an adversarial set of requests making the reservations. These results imply a relevant managerial conclusion for on-demand ridepooling: companies would be benefited if asking users to reveal their trips shortly in advance, in exchange for being prioritised.

1. Introduction

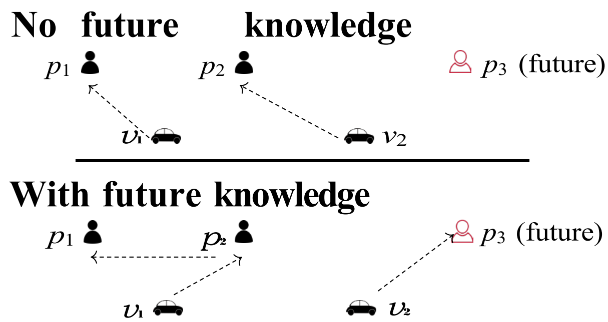
With increasing urbanisation and a need for reducing greenhouse gas emissions, new forms of urban mobility are being implemented and studied worldwide. Particularly, on-demand ride- hailing companies, such as Uber or Cabify, have become popular in many cities, but often increasing congestion by attracting users from more sustainable modes (Erhardt et al. 2019; Tirachini 2020). Its pooled counterpart (that receives different names in the literature, and we use "on-demand ridepooling" or just "ridepooling"), where people can share parts of rides with other passengers travelling in a similar direction in the same car, can be a promising mode to offer the virtues of on-demand mobility in a sustainable way. Ridepooling has been offered by some of the ride-hailing companies, but still in limited numbers, increasing the relevance to study how to improve their operational aspects. With current technology, large numbers of passengers can be efficiently allocated for ridepooling (Santi et al. 2014). For instance, if taxis were shared in Manhattan, 98% of their demand could be served with just 30% of the number of taxis, while staying within reasonable constraints on delays (Alonso- Mora et al. 2017a)[1]. The advantages of ridepooling could potentially be amplified by the

[1]Assuming that all taxi users are willing to share their vehicles is actually a strong and optimistic assumption. When mode choice is included in the models, the environmental benefits of ridepooling are less clear (Zwick et al. 2021). This aspect of the

adoption of autonomous vehicles (AVs), thanks to the decreased operational costs (Fielbaum 2020).

A key challenge for an efficient operation of ridepooling is deciding how to group the users and assign them to vehicles. One of the most popular algorithms to do this is the one by Alonso-Mora et al. (2017a), which accumulates batches of requests during a pre-defined interval (e.g. 30 seconds) and assigns them all together in an anytime optimal way. However, here the word "optimal" refers to the best possible decision *given the current information*; in other words, this is a *myopic* approach, that does not take the unknown future into account, which suggests that the assignment decisions could be improved if the future was known (at least partially), as exemplified in Figure 1.

Figure 1: Two passengers are waiting and two vehicles are to be assigned. Without future information (top row), the system assigns v_1 to p_1 and v_2 to p_2, rejecting p_3 when he emerges. If p_3 was known in advance, the assignment would be modified and everyone would be served



To overcome this issue, researchers have proposed anticipatory methods (e.g. Fielbaum et al. 2022; Alonso-Mora et al. 2017b; Li et al. 2020), i.e., techniques to try to somehow predict the future demand and incorporate those predictions into the assignment decisions. Such studies have confirmed that having better information bears the potential to make ridepooling more efficient, but are limited by the quality of their predictions. In this paper, we go one step further and analyse what are the potential gains if we had (partial) real information about the future. Concretely: Would it be possible to improve the system if some of the users requested their trips shortly before their desired pickup time? By this means, we are able to quantify what is the relevance of this (lack of) information.

We also identify an important trade-off: it is reasonable to assume passengers will only make reservations if doing so is associated with benefits, such as prioritising to serve the users that revealed their information in advance. These benefits impose constraints on the solution space and thus potentially reduce the quality of the passenger assignments. We quantify the trade-off and show that the gains of the extra information outweigh the costs of such benefits.

From a methodological perspective, the introduction of reservations can increase the com- plexity of the problem greatly. This happens because the number of feasible groups is usually limited by ensuring a given quality of service to all the users (e.g., capping the maximum waiting time or the detour); however, future requests can be easily combined with cur- rent requests respecting the corresponding constraints. Therefore, we introduce a number of heuristics to face this issue, and also show that this complexity imposes a limit on the benefits of reservations.

Our methods and conclusions are not only theoretical. They could easily be applied in an on-demand ridepooling system where passengers can indicate whether they want to be served as soon as possible, or intend to leave in the near future (e.g., in 10 minutes).

problem is beyond the scope of this paper, as increasing the efficiency of ridepooling is desirable regardless of the underlying users' behaviour.

This paper is structured as follows. In Section 2, relevant background literature is dis- cussed. The proposed methodology is described in Section 3. Section 4 explains the ex- perimental setup for the research and the obtained results: here the potential benefits and trade-offs concerning reservations are analyzed. Finally, Section 5 consists of conclusions, discussions and recommendations for further research.

2. Related literature

On-demand ridepooling systems have been extensively studied in the last few years. One of the main aspects is the proposal of new algorithms to decide how to group the users and to route the vehicles to serve them. For surveys on this topic, see Mourad et al. (2019), Danassis et al. (2022), and Zardini et al. (2022). The assignment of passengers to vehicles can be performed statically (i.e., assuming the demand to be known in advance) or dynamically (i.e., users are served as they emerge).

In a *static* assignment, all decisions are made beforehand, which allows for additional computation time available and predictable service, avoiding the relevant unreliability-related issues that emerge when the assignment is not static (Fielbaum & Alonso-Mora 2020; Alonso- González et al. 2020). This version of the problem is usually called "dial-a-ride", and has been studied since decades ago (Ho et al. 2018). However, the massive emergence of *dy- namic* on-demand mobility has highlighted the attractiveness of systems where users can request the vehicle just when they need it. The dynamic nature of the problem demands efficient algorithms as large problems have to be computed in reasonable time to allow for re-assignment as new information becomes available. The assignment is usually performed in batches to allow for better matching between requests and vehicles (Yan et al. 2020). The method proposed by Alonso-Mora et al. (2017a), used in the methodology of this paper, combines requests into groups that are collectively assigned to vehicles. The method allows for anytime optimal assignment in large-scale ridesharing systems. A modification is made by Simonetto et al. (2019), where requests are assigned individually to vehicles in shorter batch times. This allows for a speedup in computation time but offers a lower level of service. One of the main problems of dynamic assignments is that decisions must be made without knowing the future demand. Significant research has been done to estimate future demand and use it to improve assignment decisions. These predictions can be used in two different ways: i) To instruct idle vehicles to move towards areas where more vehicles are expected to be required (a step usually called *rebalancing* in the literature), as done by Vosooghi et al. (2019), Sayarshad & Chow (2017), Liu & Samaranayake (2020), and Tsao et al. (2019), or ii) To modify the way vehicles are routed and assigned to the users, in order to leave them better prepared for the expected future demand (Fielbaum et al. 2022; Alonso-Mora et al. 2017b; Huang & Peng 2018; Van Engelen et al. 2018). These methods differ in how do they predict the demand, such as using historical demand (Alonso-Mora et al. 2017b) or considering the current demand as a proxy for the future one (Fielbaum et al. 2022), and in the specific ways these predictions are leveraged. However, the overall result is consistent, namely that including some anticipatory ideas improves the results.

The incorporation of reservations in an on-demand ridesharing system is, to the best of our knowledge, only studied by Engelhardt et al. (2022). In their method, reservations are placed a day ahead and confirmed or rejected directly. Acceptance of a reservation is considered binding to the operator. Oppositely, this paper focuses on reservations that are made shortly in advance, which implies a significant difference from the users' perspective, as these short- term reservations still enable the users to enjoy the flexibility of on-demand systems; from a methodological perspective, while Engelhardt et al. (2022) deal with reservations in a static way, for us they are dynamic because the short times involved imply that they might get combined immediately with passengers currently being served by the system.

3. Methodology

3.1. Problem description

We now describe briefly the problem and the method proposed by Alonso-Mora et al. (2017a) that we use as a basis. The goal of on-demand ridepooling is to assign a set of travel requests R to a set of vehicles V with minimal overall costs and within predetermined constraints. The problem takes place on a directed graph $G = (N, E)$. The edges are weighted based on the required travel time of the road segment it represents. Every $r \in$ R is a triplette $r = (o_r, d_r, t_r)$, representing its origin, destination (which we assumed to be nodes in the graph), and emerging time, respectively. Crucially, in the original problem the requests become known exactly when they emerge, and want to be picked up as soon as possible.

In the method proposed by Alonso-Mora et al. (2017a), requests are collected in batches during Δ (we use Δ = 30 seconds) and assigned collectively. Let us now explain how one batch is assigned, denoting by R_t the set of requests waiting to be assigned. The objective of the assignment is to serve as many requests as possible, offering the best possible quality of service, measured by the *waiting time* and the *detour* experienced by each user. The waiting time for request r, denoted w_r, is the time between it emerges t_r and when it is picked up pu_r

$$w_r = pu_r - t_r \tag{1}$$

The detour time is the extra in-vehicle time experienced by request r to a non-shared vehicle. The detour time of request r, denoted det_r, is given by the drop off time do_r minus the pickup time (which gives the in-vehicle time), minus the minimum travel time between o_r and d_r denoted $t_v(o_r, d_r)$, hence

$$det_r = do_r - pu_r - t_v(o_r, d_r) \tag{2}$$

The waiting time and detour time combined give the total delay time for a request δ_r, $\delta_r = w_r + det_r$. The goal of the assignment is to minimize Equation (3). In this equation, the first term represents the extra detour for all passengers that were already being transported by the system (P_v are the users currently in vehicle v), the second term is the total delay δ_r for all requests that get assigned R_{ok} and the third one is fixed a rejection penalty Π for all requests that are not assigned R_{no}. Note that $R_{ok} \cup R_{no} = R_t$.

$$\sum_{v \in V} \sum_{r \in P_v} det_r + \sum_{r \in R_{ok}} \delta_r + \sum_{r \in R_{no}} \Pi \tag{3}$$

Assignment of requests to vehicles is done via an integer linear program (ILP) method explained below. To do this, requests are first combined into *trips*. A trip $T \subset R_t$ is a group of requests that can be assigned together to the same vehicle.

Trips computation A trip-vehicle (T, v) combination is feasible if it is possible to serve all the requests in that trip within waiting and detour time constraints for passengers in T and in P_v, i.e., every request has to wait less than Ω_w and face a detour lower than Ω_d (in our simulations, we use $\Omega_w = \Omega_d$ = 5 minutes). Furthermore, the vehicle capacity η_v can not be exceeded at any time during the trip.

The cost of adding a trip to a vehicle is given by Equation (4). If multiple feasible routes (sequence of pick up and drop off actions) are found for a trip, the one with the lowest cost is selected.

$$cost(T, v) = \sum_{r \in P_v} d_r + \sum_{r \in T} \delta_r \tag{4}$$

As the number of feasible trip-vehicle combinations can be very large, they are computed for trips of increasing size, leveraging the property that for the combination (T, v) to be feasible, all (T', v) with $T' \subset T$ have to be feasible as well. To include reservations in this framework, they are simply added to the set of requests to be assigned R_t. The set of feasible (T, v) assignments is denoted T .

ILP assignment: Once all the feasible trip-vehicle combinations have been identified, we need to decide which of them are taking place. The actual assignment is decided via an ILP, which has two sets of binary variables. First, $\epsilon_{T,v}$ for every feasible combination between a trip T and a vehicle v, which takes the value 1 if this combination takes place. Second, variables X_r, defined $\forall r \in R_t$, take the value 1 if the request r is not assigned.

The constraints in the ILP formulation ensure that i) each vehicle will have at most one trip assigned (Equation 6), and ii) each request is either assigned to one vehicle or ignored (Equation 7). Three subsets are used in the ILP formulation: $T_{v=j}$ is the set of trips that can be served by vehicle j; $T_{R=r}$ is the set of trips that contain request r; $V_{T=i}$ is the set of vehicles that can serve trip i.

$$\min_{\epsilon, X} \sum_{(T,v) \in T} cost(T, v) \cdot \epsilon_{T,v} + \sum_{r \in R} \Pi \cdot X_r \tag{5}$$

$$\text{s.t.} \sum_{i \in T_{v=j}} \epsilon_{i,j} \leq 1 \forall j \in V \tag{6}$$

$$\sum_{i \in T_{R=r}} \sum_{j \in V_{T=i}} \epsilon_{i,j} + X_r = 1 \forall r \in R \tag{7}$$

Once the assignment procedure is done, it is communicated to the vehicles and they update their itinerary. If a request is not picked up before the next time step, it is put back into the set of requests to be assigned, which enables the system to perform reassignments when this improves overall efficiency (which is crucial when dealing with reservations). Note that this implies that some requests can be first assigned to a vehicle but then become rejected in a subsequent iteration before being picked up.

Rebalancing: The only non-myopic step in the original method is as follows: those vehicles that are idle (i.e., that had no passengers before the assignment and received none through the ILP) are sent towards the origins of the requests that were rejected. We do this exactly as in the original method, so we refer the reader to Alonso-Mora et al. (2017a) for the details.

3.2. Including short-term reservations

As discussed above, including reservations in this method might be straightforward, as it is possible to just include them in the set of requests to be assigned. In this subsection, we explain the two main modifications we need to incorporate:

1. First, we explain the benefits for users that reserve. This is not required by the method *per se*, as the assignments can be decided just as in the no-reservations scenario. How- ever, if a ridepooling company is to expect some users to provide their information some minutes in advance, it is reasonable to expect that they need to stimulate this through some sort of benefit.

2. Second, we show that the inclusion of reservations can imply a great increase in the required computational time, and propose some heuristics to tackle this.

Benefits: We study two alternative ways to provide benefits to reservations. The first and simplest one is to increase the rejection penalty Π for reservations: in our simulations, we use $\Pi = 50$ minutes for normal requests, and we double its value for reservations.

The second one is called "Pledged service": We modify the ILP by including an additional constraint that states that any request that has been assigned to a vehicle before, and thus expect to be served, cannot be rejected. This constraint will only be applied to reservations and has two main implications:

1. Each user that makes a reservation receives an immediate response on whether she is going to be serviced. Although this could be desirable for everybody, it is much more important in the case of reservations, as it is unlikely that someone would be willing to make a reservation to find out that she is being rejected just before the desired pickup time.

2. In practice, this implies that the chances of being served increase significantly for reser- vations. In fact, the first time a reservation r appears it is usually simple to find a feasible combination to serve her, as a vehicle could arrive at the pickup point even before t_r yielding zero waiting time. Therefore, most reservations are assigned to some vehicle the first time they appear, and then the pledged service constraint ensures that they are never rejected. This argument only fails if there are too many reservations so that vehicles are not enough for everybody. All of this will be verified in the numerical experiments (Section 4.2).

For this purpose, an additional subset R_{rb} is defined for all reservations that have been assigned before, and the following constraint is added to the ILP:

$$X_r = 0 \quad \forall r \in R_{rb} \tag{8}$$

Extra computational time: The computational load from the inclusion of reservations into the set of requests to be assigned can grow exponentially if no provisions are taken.

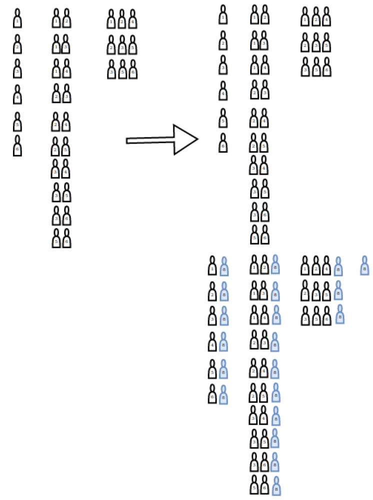
First of all, reservations could, in the worst case, be available for each feasible trip-vehicle combination. This phenomenon is shown for one reservation in Figure 2, where a blue reservation is added. This implies that the number of feasible trips might get multiplied by 2^{Res} where *Res* stands for the number of reservations. We remark that this worst-case analysis is also valid for on-demand requests, but in practice, the hard constraints on waiting and detour time prune many more combinations that in the reservations case.

Second, reservations stay in the request pool longer than on-demand requests. This means that every time an assignment is decided, the pool of requests to assign is larger than in the pure on-demand case.

Third, not only are there more trips, but the trips also have more requests on average. Finding the optimal route requires much more computational time when the size of the trip increases, as the number of feasible routes for a trip with k requests might be as large as $k!/2^k$.

Because of the scaling effects from reservations, countermeasures are required. First, and similar to Alonso-Mora et al. (2017a), an insertion heuristic is applied when searching for the optimal route. The insertion heuristic locks the order in which passengers already on board are dropped off. Two additional heuristics are considered: First, to limit the maximum trip size to a parameter κ_1 (we use $\kappa_1 = 3$), i.e., we do not assign more than κ_1 users to the same vehicle in the same iteration. Second (and similar to Fielbaum et al. 2021), for every request

Figure 2: Worst case growth in the number of feasible trips if a blue reservation is added



we consider only the κ_2 less costly vehicles in each iteration (we use $\kappa_2 = 20$); note that the accumulated effect is significant, because for a group of k users to be feasibly assigned to the same vehicle v, all those users need to have v among its best κ_2 vehicles.

4. Experiments

4.1. Experimental setup

The network for the experiments represents Manhattan, New York City. The graph describ- ing the network consists of 4,091 nodes and 9,454 weighted directional edges. A thousand vehicles with a capacity of four passengers are used for the simulation. The shortest route and distance between each pair of points in the graph are calculated beforehand using the Dijkstra algorithm and stored in a lookup table. A set of 10,548 requests are taken from actual data for taxi demand in Manhattan from 12 AM till 1 PM, on January 15th, 2013.

The standard test scenario consists of a warm-up phase and a "real" part. In the warm-up phase, 5250 requests are added to the system over the first half an hour. The set of warm-up requests is randomly drawn without redraw from all possible requests and equally spaced over the available time for warm-up. The service for the warm-up set is not considered in the KPIs evaluated.

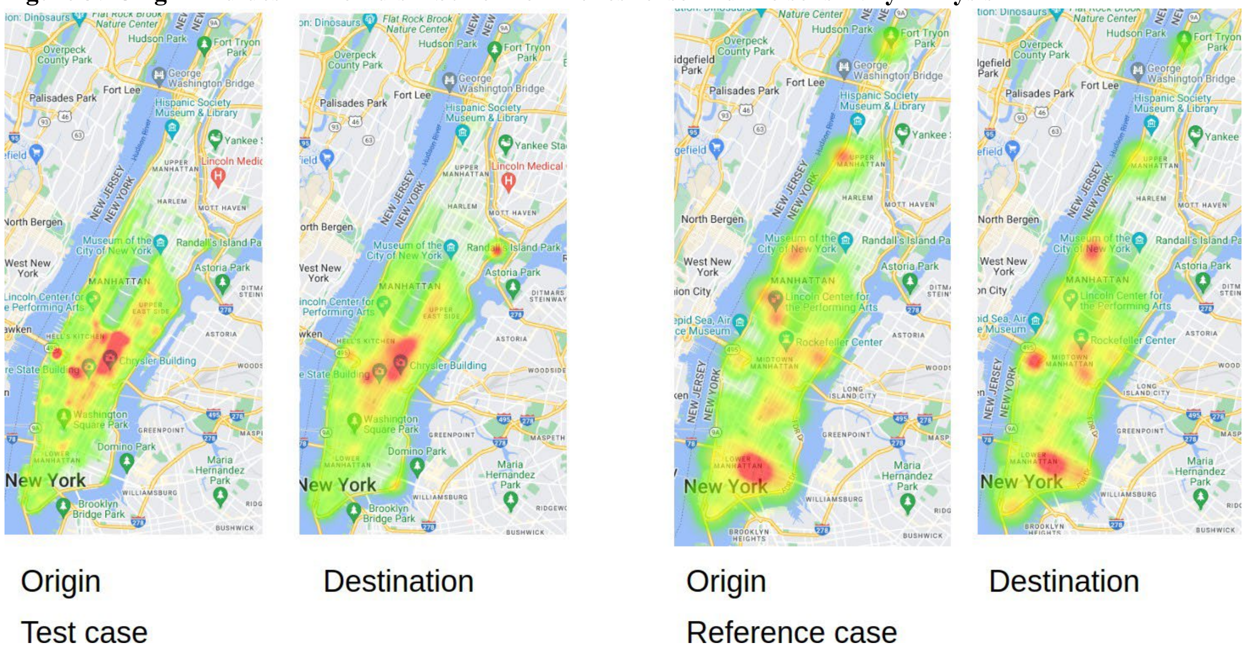
To understand the impact of reservations, we study different cases where we vary the number of reservations and how long in advance they are placed. Crucially, we consider two scenarios regarding which users are placing reservations:

1. **Random:** A random sample of all the requests is selected to be used as reservations.
2. **Adversarial:** We first simulate the whole sample without reservations. The users that are rejected in those simulations are taken as reservations to re-simulate. We remark those users are in general more difficult to be served (e.g., because they are located far from the high-demand zones), so having to prioritize them makes an adversarial scenario for our model; moreover, it is reasonable to expect that the users that will be more interested in making a reservation are the ones that know their chances of being served are lower than the average.

How long in advance reservations are placed can sharpen the two effects discussed above: on the one hand, earlier reservations entail better information to make decisions, but on the other hand, they increase the required computational time. This is why we run experiments modifying this parameter to be equal to 5, 10, 15, or 20 minutes in advance.

To examine the sensitivity towards the underlying request data, a second set of requests from Manhattan is used, taken from the morning peak, and which presents a significant difference in origin/destination distribution compared to the original test case. The distri- butions for both scenarios are shown in Figure 3. The morning request set consists of 28,030 requests. To be comparable with the noon data set, 10,548 unique samples are randomly drawn from this morning hour. As shown by Soza-Parra et al. (2022), different spatial de- mand patterns can have a strong impact on the efficiency of on-demand ridepooling, so this sensitivity analysis is meant to determine whether this is also the case when reservations are included.

Figure 3: Origin and destination distribution for the test case and the sensitivity analysis



4.2. Results

In this section, we show the results of our model, by comparing the scenarios with and without reservations, and also measuring the effect of the benefits. We take the service rate (i.e., the percentage of users that are served) as the main KPI to be analyzed.

First, the results are shown without heuristics in scenarios where it is possible to run the simulations to optimality. Second, we study the effects of the proposed heuristics. Third, the results for scenarios with both pledged service and the heuristics are evaluated. Last, the sensitivity of the results to the requests' data is analyzed.

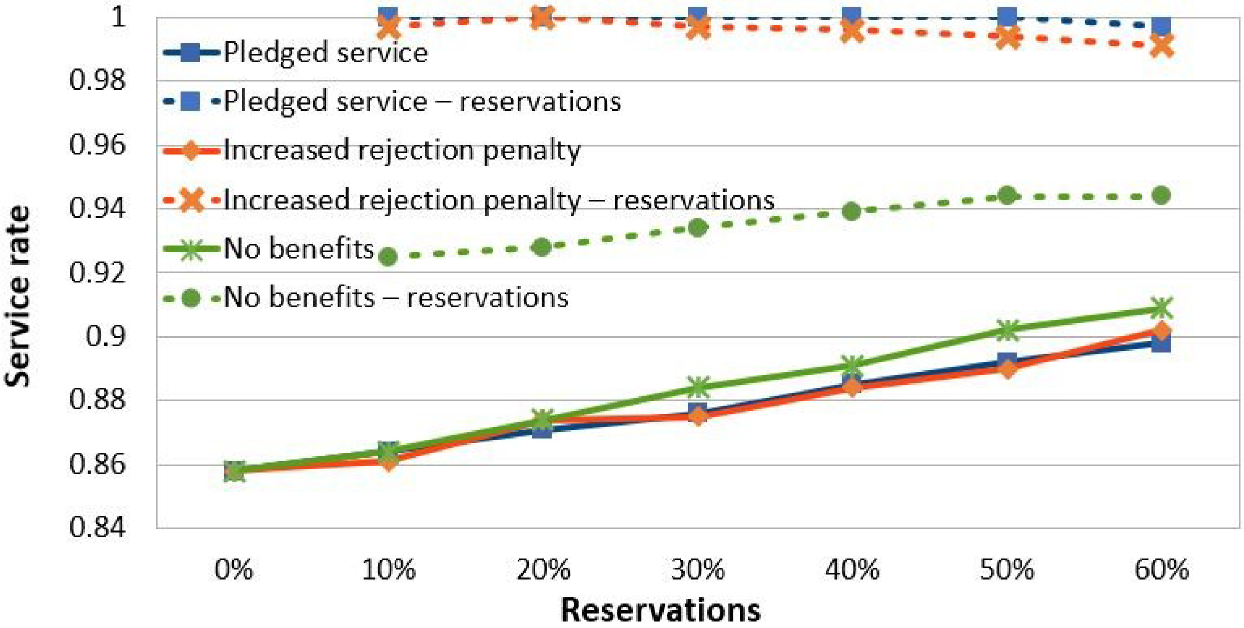
4.2.1. *Quantifying the trade-off*

The main results of the model are shown in Figure 4. For now we focus on the Random Reservations scenario, and we depict the service rate both overall (solid lines) and for reser- vations (dotted lines). The experiments are performed with reservations revealed only 5 minutes in advance.

Figure 4 shows the following:

- The extra information provided by reservations does improve the service rate. This can

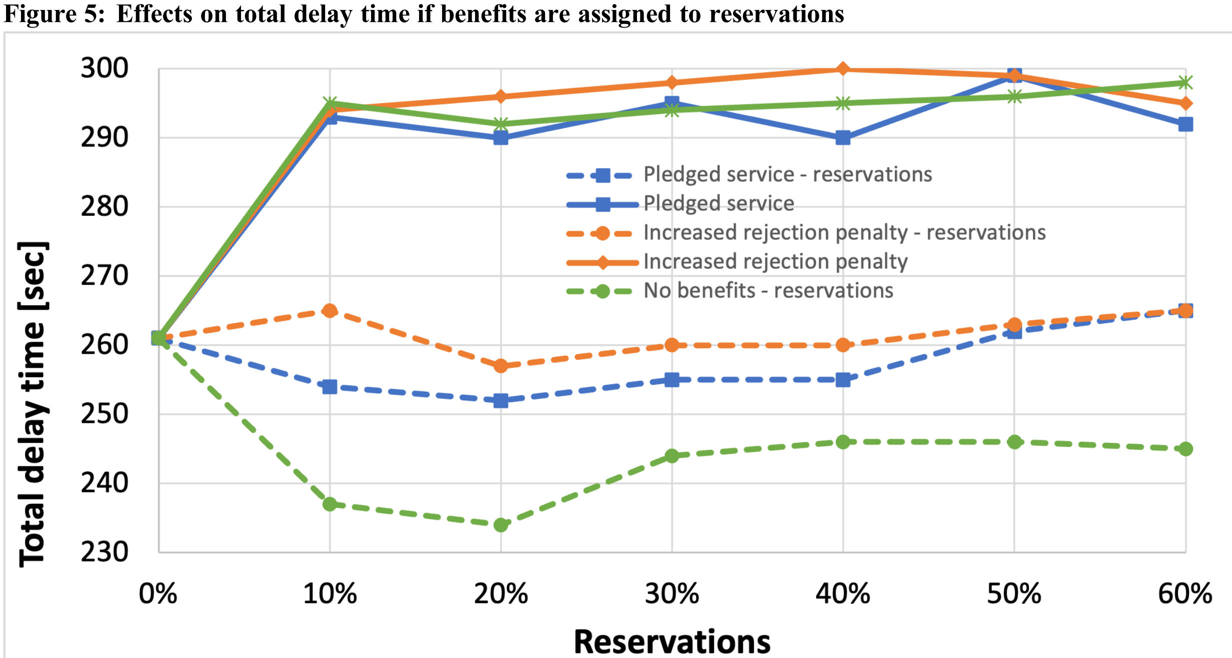
Figure 4: Effects on service rate if benefits are assigned to reservations



be seen as the three solid curves increase. In the most extreme case (60% of reservations with no benefits) the service rate increases from approx. 86% to 91%.

- The trade-off between having more information but less freedom to assign is real, as revealed by the fact that the solid blue and orange curves are below the green one, i.e., the service rate decreases when there are benefits. Crucially, **the overall effect is always positive**, that is, the service rate increases when there are reservations even if they receive benefits.

Figure 5: Effects on total delay time if benefits are assigned to reservations



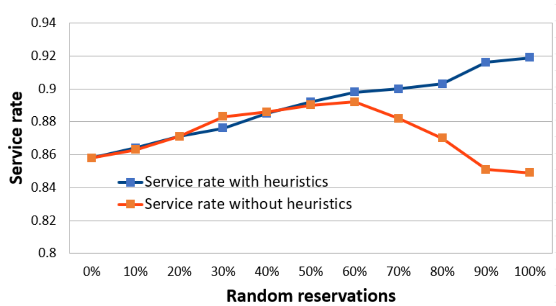
- The increase in service rate comes at the cost of increasing the average delay by about 40 seconds, as shown in Figure 5. This increase is expected as we are now serving more passengers with the same fleet, and has also been found in previous papers that leverage predictive techniques to increase service rate (Fielbaum et al. 2022). Users that reserve face a lower total delay than the rest, which is expected as vehicles can be sent towards them before the minimum pickup time.

- We remark that vehicles run continuously with or without reservations. In other words, the extra information enables serving more people without increasing vehicles-hours- travelled, effectively making the system more sustainable as discussed in Section 1.
- It is better to utilise the Pledged service as a benefit, rather than the Increased rejection penalty. The overall effect is similar, but the Pledged service is better for reservations: not only it gives an immediate response, but also provides a better service rate for reservations, which actually reaches 1 unless the percentage of reservations is too large. In the following sections, we assume that the Pledge service benefit is provided.

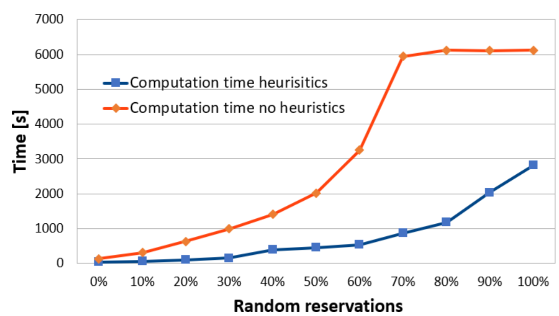
4.2.2. Heuristics

We now study the impact of utilising the heuristics proposed in section 3. The effect of more information on the service rate is shown in Figure 6a. The orange line shows the results without heuristics, and the blue line with them. In this experiment, all reservations are revealed to the system 5 minutes in advance. Crucially, we apply a time limit to the ILP as sometimes it is not possible to obtain the optimal solution.

Figure 6: Effects of the heuristics



(a) Service rate with random reservations revealed 5 minutes in advance



(b) Computational time with random reservations revealed 5 minutes in advance

Initially, a steady increase in service rate is reached with more reservations. The reason behind the drop in performance of the orange line is given in Figure 6b, where the compu- tational time is shown with and without the heuristics. When the number of reservations is large, the experiment without heuristics reaches the computational time limits and de- teriorates in performance as a consequence. **Before that drop, the heuristics reduce significantly the computational time and obtain a similar service rate as the optimal solution**. Therefore, the heuristics will be used in the following results in this paper.

4.2.3. Different times revealed in advance

We summarize the results for different percentages of reservations, and different minutes in advance, in Table 1. We also show the results in the adversarial case. The results are with the described heuristics and the pledged service constraint. Each cell in the table shows the overall service rate, the service rate for reservations, and the computational time used for the simulation.

Table 1: Results from the incorporation of reservations in an on-demand ridepooling system for different percentages of random reservations revealed a certain time in advance

		Time revealed in advance [minutes] 5	10	15	20
Overall service rate (SR) Reservations' SR Simulation time [seconds]	0%	0.858 - 39	0.858 - 39	0.858 - 39	0.858 - 39
	10%	0.864 1 61	0.868 1 107	0.874 1 240	0.874 1 921
	20%	0.871 1 103	0.883 1 382	0.885 1 2136	0.834 1 2367
	30%	0.874 1 151	0.888 1 2677	0.8 1 2730	0.804 0.993 2775
	40%	0.885 1 395	0.836 1 2786	0.806 0.985 2890	0.808 0.976 3101
	50%	0.892 1 639	0.819 0.995 2932	0.816 0.965 3036	0.805 0.959 3435
Adversarial reservations	14.2%	0.864 1 75	0.871 1 325	0.877 1 1209	0.862 1 3242

Results follow the trend of the two previous subsections. Reservations do help the system, even when providing benefits to the users that reserve. However, if reservations are placed too much in advance, they can be harmful due to the computational limits. Crucially, **if reservations are done little in advance (5 minutes in our simulations[2]), the service rate can always be increased significantly; results are also positive in the adversarial scenario**, but worse than in the random one, as expected.

4.2.4. *Sensitivity analysis*

In Figure 7 we compare the results in the original scenario (midday) and in the morning peak. All reservations are revealed 5 minutes in advance.

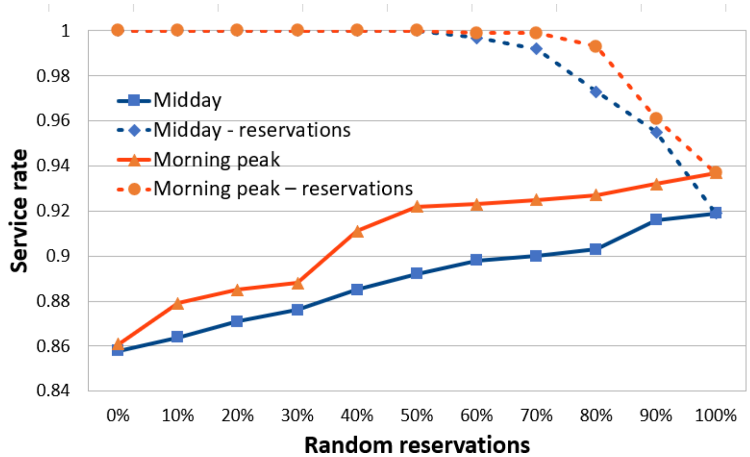
The service rate is slightly higher in the morning peak as there is a bit more overlap in desired trajectories for passengers. Trends in performance are similar and more reservations revealed shortly in advance increase the service rate significantly in both scenarios. In both scenarios, it becomes eventually unfeasible to serve all the reservations, so the dotted lines begin to decrease, but the solid lines (overall service rate) always increase. These results show that the merits of oure method do not depend on the spatial demand pattern.

5. Conclusion

One of the main difficulties when operating a ridepooling system, particularly when deciding how to group the users and assign them to the vehicles, stems from the lack of information about future demand. In this paper, we have analyzed and measured the benefits gained by

[2]As explained by Alonso-Mora et al. (2017a), the assignment algorithm can be easily distributed. For instance, each vehicle could compute all the feasible trips that involve it. The only centralised aspect of the algorithm is the solution of the ILP, which is not usually the bottleneck of the method. From a practical point of view, this means that some minutes of computational time are acceptable even if assignments are computed every 30 seconds.

Figure 7: Simulations for sensitivity data set compared to the test set. Requests are all revealed 5 minutes in advance



on-demand ridepooling systems if the said lack of information is somehow reduced, by some users making reservations shortly in advance.

To do this, we have identified a relevant trade-off: users would need some incentive (benefit) in order to place their requests in advance, and those benefits can indirectly harm the quality of service. We have proposed how to modify a state-of-the-art routing algorithm, in order to admit reservations and incorporate the said benefits. The inclusion of future demand increases greatly the computational burden of the algorithm, so we have proposed a number of heuristics.

We have run experiments considering the real-life network from Manhattan, and utilising a dataset containing the actual taxi trips occuring there. Considering the service rate as the main KPI, our results are promising: we have shown that reservations do improve the operation of the system, even when considering the benefits for the reserving users - namely, a better probability of being served. This increase in service rate is achieved without increasing the fleet size or the vehicles-kilometres-travelled, implying that the additional information provided by the reservations, and our method to handle it, make the system more sustainable and can help relieve congestion.

Moreover, the heuristics barely affect the service rate and reduce the computational time significantly. However, we have also shown that if too many users reserve, or if they do it too much in advance, the additional complexity precludes finding good solutions and the system degrades.

As ridepooling is an emerging topic, there are plenty of directions for future research related to the inclusion of reservations. The most direct one emerging from our results is the need for stronger heuristics, or additional tailored methods, to deal with situations when the number of reservations is too large. Additionally, some users might want to ensure getting serviced by requesting long in-advance, so how to combine long-term and short- term reservations is a relevant methodological question, that would probably need to mix traditional dial-a-ride-related techniques with the ones presented here. Finally, ridepooling system present relevant issues related to unreliability, as the travelling times get constantly updated: utilising the future information to provide better predictions to the users is yet another promising direction to further investigate.

References

1. Alonso-González, M. J., van Oort, N., Cats, O., Hoogendoorn-Lanser, S. & Hoogendoorn, S. Value of time and reliability for urban pooled on-demand services. *Transportation Research Part C: Emerging Technologies* **115,** 102621 (2020).
2. Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E. & Rus, D. On-demand high-capacity ride- sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* **114,** 462–467 (2017).
3. Alonso-Mora, J., Wallar, A. & Rus, D. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 3583–3590 (2017).
4. Danassis, P., Sakota, M., Filos-Ratsikas, A. & Faltings, B. Putting ridesharing to the test: efficient and scalable solutions and the power of dynamic vehicle relocation. *Artificial Intelligence Review* **55,** 5781–5844 (2022).
5. Engelhardt, R., Dandl, F. & Bogenberger, K. Simulating Ride-Pooling Services with Pre-Booking and On-Demand Customers. *arXiv preprint arXiv:2210.06972* (2022).
6. Erhardt, G. D., Roy, S., Cooper, D., Sana, B., Chen, M. & Castiglione, J. Do transportation network companies decrease or increase congestion? *Science Advances* **5** (2019).
7. Fielbaum, A. Strategic public transport design using autonomous vehicles and other new technologies. *International Journal of Intelligent Transportation Systems Research* **18,** 183–191 (2020).
8. Fielbaum, A., Bai, X. & Alonso-Mora, J. On-demand ridesharing with optimized pick-up and drop-off walking locations. *Transportation Research Part C: Emerging Technologies* **126,** 103061 (2021).
9. Fielbaum, A., Kronmueller, M. & Alonso-Mora, J. Anticipatory routing methods for an on-demand ridepooling mobility system. *Transportation* **49,** 1921–1962 (2022).
10. Fielbaum, A. & Alonso-Mora, J. Unreliability in ridesharing systems: Measuring changes in users' times due to new requests. *Transportation Research Part C: Emerging Technologies* **121,** 102831 (2020).
11. Ho, S. C., Szeto, W. Y., Kuo, Y.-H., Leung, J. M., Petering, M. & Tou, T. W. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological* **111,** 395–421 (2018).
12. Huang, X. & Peng, H. Efficient mobility-on-demand system with ride-sharing. *2018 21st International Conference on Intelligent Transportation Systems (ITSC),* 3633–3638 (2018).
13. Li, X., Wang, C., Huang, X. & Nie, Y. A Data-Driven Dynamic Stochastic Programming Framework for Ride-Sharing Rebalancing Problem under Demand Uncertainty. *2020 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Commu- nications, Social Computing Networking,* 1120–1125 (2020).
14. Liu, Y. & Samaranayake, S. Proactive rebalancing and speed-up techniques for on-demand high capacity ridesourcing services. *IEEE Transactions on Intelligent Transportation Systems* (2020).
15. Mourad, A., Puchinger, J. & Chu, C. A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological* **123,** 323–346 (2019).
16. Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H. & Ratti, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences* **111,** 13290–13294 (2014).
17. Sayarshad, H. R. & Chow, J. Y. Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. *Transportation Research Part E: Logistics and Transportation Review* **106,** 60–77 (2017).
18. Simonetto, A., Monteil, J. & Gambella, C. Real-time city-scale ridesharing via linear assignment prob- lems. *Transportation Research Part C: Emerging Technologies* **101,** 208–232 (2019).
19. Soza-Parra, J., Kucharski, R. & Cats, O. The shareability potential of ride-pooling under alternative spatial demand patterns. *Transportmetrica A: Transport Science,* 1–23 (2022).
20. Tirachini, A. Ride-hailing, travel behaviour and sustainable mobility: an international review. *Trans- portation* **47,** 2011–2047 (2020).

21. Tsao, M., Milojevic, D., Ruch, C., Salazar, M., Frazzoli, E. & Pavone, M. Model Predictive Control of Ride-sharing Autonomous Mobility-on-Demand Systems. *2019 International Conference on Robotics and Automation (ICRA),* 6665–6671 (2019).

22. Van Engelen, M., Cats, O., Post, H. & Aardal, K. Enhancing flexible transport services with demand- anticipatory insertion heuristics. *Transportation Research Part E: Logistics and Transportation Review* **110,** 110–121 (2018).

23. Vosooghi, R., Puchinger, J., Jankovic, M. & Vouillon, A. Shared autonomous vehicle simulation and service design. *Transportation Research Part C: Emerging Technologies* **107,** 15–33 (2019).

24. Yan, C., Zhu, H., Korolko, N. & Woodard, D. Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics (NRL)* **67,** 705–724 (2020).

25. Zardini, G., Lanzetti, N., Pavone, M. & Frazzoli, E. Analysis and control of autonomous mobility-on- demand systems. *Annual Review of Control, Robotics, and Autonomous Systems* **5,** 633–658 (2022).

26. Zwick, F., Kuehnel, N., Moeckel, R. & Axhausen, K. W. Agent-based simulation of city-wide au- tonomous ride-pooling and the impact on traffic noise. *Transportation Research Part D: Transport and Environment* **90,** 102673 (2021).