Unreliability in ridesharing systems: measuring changes in users' times due to new requests

Andrés Fielbaum*,[1] and Javier Alonso-Mora[2]

[1] TU Delft, Delft, The Netherlands, a.s.fielbaumschnitzler@tudelft.nl
[2] TU Delft, Delft, The Netherlands, j.alonsomora@tudelft.nl

Abstract. On-demand systems in which several users can ride simultaneously the same vehicle have great potential to improve mobility while reducing congestion. Nevertheless, they have a significant drawback: the actual realization of a trip depends on the other users with whom it is shared, as they might impose extra detours that increase the waiting time and the total delay; even the chance of being rejected by the system depends on which travelers are using the system at the same time. In this paper we propose a general description of the sources of unreliability that emerge in ridesharing systems and we introduce several measures. The proposed measures are related to two sources of unreliability induced by how requests and vehicles are being assigned, namely how users' times change within a single trip and between different realizations of the same trip. We then analyze both sources using a state-of-the-art routing and assignment method, and a New York City test case. Regarding same trip unreliability, in our experiments for different fixed fleet compositions and when reassignment is not restricted, we find that more than one third of the requests that are not immediately rejected face some change, and the magnitude of these changes is relevant: when a user faces an increase in her waiting time, this extra time is comparable to the average waiting time of the whole system, and the same happens with total delay. Algorithmic changes to reduce this uncertainty induce a trade-off with respect to the overall quality of service. For instance, not allowing for reassignments may increase the number of rejected requests. Concerning the unreliability between different trips, we find that the same origin-destination request can be rejected or served depending on the state of the fleet. And when it is served the waiting times and total delay are rarely equal, which remains true for different fleet sizes. Furthermore, the largest variations are faced by trips beginning at high-demand areas.

Keywords: Ridesharing; On-demand; Unreliability; Mobility; Ridepooling; Mobility-as-a-service.

1 Introduction

Mobility systems have been facing profound changes throughout the last years due to the emergence of new technologies that are able to coordinate massive numbers of users and vehicles online. Transportation network companies operate worldwide. In their most common service requests are consecutively assigned to the same vehicle, without pooling. This is, several passengers do not share the same car at the same time unless they travel in a group. Although these systems might be useful to decrease the number of cars in a city (as they can replace car ownership [7]), they fail to reduce the number of cars on the streets, i.e. congestion ([41, 16]), because they operate as private trips. More recently, these ideas have been extended to include shared[3] trips, which have considerable potential in reducing the required number of vehicles [37] to operate the system. This reduction in the number of cars on the streets may effectively reduce congestion if the size of the vehicles is appropriate [40], due to an increase in the vehicle's usage. The mathematical complexity of these systems is much higher, but previous works have proposed successful methods to assign sets of requests to vehicles ([3, 10, 42, 38]), showing that these systems might be efficient.

Nevertheless, on-demand shared systems might have a significant drawback. For each user, the actual realization of her trip depends on the other users in the vehicle. In a lucky instance, she might be the only passenger, and the trip will be fast, but when there are more users, it can affect the travel distance (detour), waiting times and walking distances, among other variables.

* Corresponding Author

[3] Throughout this paper, the word "shared" refers to several passengers from multiple requests using the same vehicle at the same time.

Therefore, on-demand ridesharing systems present new sources of *unreliability* , i.e., users cannot know in advance how their trips are going to be in an accurate way. As a comparison, non-shared systems (including taxis) always follow the shortest paths, while traditional public transport systems have fixed routes (as opposed to on-demand) and waiting times are known when timetables are available. Reliability is one of the most relevant quality factors for users both in public transport [13, 34, 6] and in different kinds of ridesharing or on-demand systems [14, 9, 1, 29]. As such, understanding unreliability due to new requests is crucial to succeeding in providing shared mobility systems in the future. Moreover, even in systems that do not share trips, reliability is seen as one of the major improvements that new technologies can introduce when compared with traditional taxis [33, 43].

Unreliability phenomena are inherent to the on-demand and shared nature of ridesharing systems, because the route that a vehicle will follow while serving a request evolves over time. Consider the example shown in Fig. 1. Passenger p_1 is being moved by the pink vehicle v from O_1 to D; when p_1 is close to her destination, a new request p_2 emerges, that shares the same destination D and whose origin O_2 is close to the current position of the vehicle, but requires a small detour. Assigning p_2 to v is efficient, as the system is serving an additional request with little extra effort. However, from the point of view of p_1, this sudden change might be quite disturbing, a phenomenon that is independent of total traveling and waiting times.

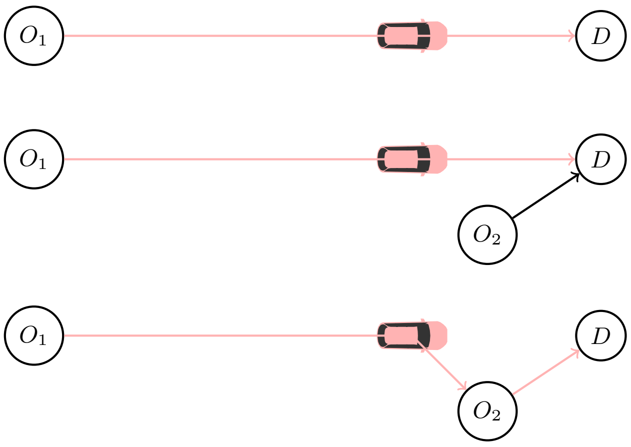


Fig. 1. An example of the inherent unreliability of on-demand ridesharing systems. A new request p_2 emerges just before the pink vehicle completes the trip of passenger p_1. In order to serve p_2 efficiently, p_1 must face a sudden detour.

In this paper, we first propose a general description of the sources of unreliability that emerge in on-demand ridesharing systems, to then measure those that can be controlled by the provider of the system, i.e., that are related to how the system assigns vehicles and requests together. We calculate these measures using an adapted version of the model proposed in [3], during two hours (about 20000 requests) in Manhattan. We study how the predicted times for each single request change from its first assignment until the drop-off, and also how different realizations of the same request obtain different results.

1.1 Related works

The design of centrally controlled ridesharing systems, which match on-demand groups of requests with vehicles, is an emerging topic in the literature. The underlying optimization problem is quite complex, as the on-demand assignment is related to the Dynamic Vehicle Routing Problem [30], and vehicle-sharing is related to the Dial-A-Ride problem [17], both problems being NP-Hard. Several works, such as [44, 15, 46, 10], deal with this complexity by using agent-based models with somewhat simple assignment rules, while others propose algorithmic approaches. For instance [3], the model that we use here, consider the set of requests that emerges at each interval (of some pre-defined duration) to identify the possible matches among them and the vehicle fleet and optimize the assignment. The method was modified by [38] to assign no more than one passenger to a vehicle each time, thus reducing the computational time. [42] propose a fast algorithm but limited to vehicles of capacity two. When these systems are assumed to replace taxis or private cars only, results show that the number of vehicles on the streets and the total Vehicles-Hour-Traveled (VHT) might be heavily reduced [3, 46], but when the modal share is also taken into account, VHT could increase [44, 15, 10, 40] because some passengers might come from public transport, whose large vehicles make a more efficient use of space.

Ridesharing systems have also been recently studied in a number of relevant directions other than unreliability. To improve their performance, rebalancing techniques and assignment methods that try to anticipate future requests have been proposed [45, 47, 39, 4]. The potential of reducing VHT suggests that integrating such systems in a public transport network could provide a better service. Several ideas to achieve this integration have also been proposed [32, 11, 48, 36]. These approaches will benefit from a better understanding of reliability issues in ridesharing systems.

Traditionally, reliability in transport systems have been understood as the uncertainty regarding waiting and traveling times. In public transport, for instance, waiting times might be uncertain if timetables are not being used, and traveling times might depend on the congestion on the streets. This phenomenon (defined as "daily unreliability" in Section 2) is also present in on-demand ridesharing systems. The unreliability that emerges due to traffic conditions in ridesharing systems has been studied by [26]. The relevance of uncertainty is measured by [2], who calculate the so-called value of reliability[4] VOR for a shared system, i.e., the willingness to pay to reduce the uncertainty (defined as the standard deviation of the waiting and traveling times), and compare it with the value of time VOT (waiting and in-vehicle), finding that VOR is about one half of VOT.

Reliability regarding total time is not the only concern in ridesharing systems; changes that occur during passengers' trips (as shown in Fig. 1, and here defined as "one-time unreliability" in Section 2) are also of interest. Its relevance has been indirectly analysed by [5], who estimate that demand could be increased by about 10% if waiting times could be better predicted.

Some works have addressed unreliability-related problems in ridesharing systems. [31] optimize the operation of a ridesharing system, according to reliability criteria, but in a simplified scheme in which all the origins and destinations are located over a single line. In these shared-on demand systems, users can also be a source of unreliability (defined as "unreliability induced by the users" in Section 2), if they don't arrive punctually at their pick-up point, which has been described and measured by [18, 22]. [28] focus on systems that are not centrally optimized, but in which drivers have their own routes and seek for passengers whose origins and destinations are compatible with their paths. In contrast, we focus on decisions taken by centrally operated on-demand systems with arbitrary origins and destinations.

1.2 Contribution

So far only partial aspects of unreliability in on-demand ridesharing systems have been studied. This paper defines and describes the novel sources of unreliability that emerge in on-demand ridesharing systems with pooled rides, when the operators centrally control how to assign users and vehicles. Subsequently, appropriate measures for the unreliability aspects that can be directly controlled by the provider of the system are proposed. The study provides qualitative explanations for these phenomena, defined quantitative indices to measure them, and describes the trade-offs between unreliability and other indicators of quality of service.

We then show how to apply these measures to gain insights about the operation of on-demand ridesharing over realistic scenarios. To do that, we experimentally study how the predicted times for each single request change from its first assignment until the drop-off, and how different realizations of the same request lead to different results, using an adapted version of the model proposed in [3], during two hours (about 20000 requests) in Manhattan.

[4] See [8, 25] for a general review on value of reliability.

The proposed measures enable the study of which requests are likely to receive a more unreliable service, and can be employed to design more reliable routing and assignment methods and to compare them.

1.3 Organization

The paper is organized as follows. Section 2 describes qualitatively the new sources of unreliability related to ridesharing systems and Section 3 discusses how to measure unreliability. Section 4 describes the assignment model that matches requests with vehicles and that is employed to generate results the results over Manhattan, presented in Section 5. Section 6 studies how controlling unreliability affects the other quality-of-service indices of the system. Finally, Section 7 concludes and proposes some avenues for future research.

2 New sources of unreliability

In the following we focus on those unreliability sources whose cause is related to the specific characteristics of an on-demand ridesharing system. In this paper we do not discuss unreliability sources that are already present in traditional public transport systems, such as congestion or accidents on the streets[5], or in car-sharing systems in which rides are not shared[6].

The ultimate cause for the new sources of unreliability is that requests emerge dynamically and in a stochastic way [7], affecting the routes of vehicles and previous passengers in the system. We classify these sources in a more specific way, depending on whether they are caused (directly) by other users or by the operational rules of the system:

Definition 1. ***Unreliability induced by the users***: *Sometimes, users will not be there when vehicles arrive at their pick-up point, forcing the vehicle to wait for some (short) time, making users that are in the vehicle face a longer delay, and increasing waiting time for future users. Some previously assigned trips might now get rejected (by the system or by the user). Similar problems arise if a user cancels her trip when the vehicle is on its way.*

Definition 2. ***Unreliability induced by the operators***: *Each time the system receives a new request, it changes how it gathers several requests together, and also how it assigns requests to vehicles. Depending on the system's rules, such changes can affect routes, which in turn might increase waiting time for passengers that are not on-board yet, and delay for passengers that are on their way but that now have to pick-up a new request. Furthermore, a request that had been previously assigned may now be rejected, in order to prioritize new ones (for instance, because they are closer or they involve more users).*

Users' perceptions concerning these changes can also be classified, depending on whether it is a one-time variation or if it is systematic.

Definition 3. ***One-time unreliability***: *When a user executes a trip, the actual delay and waiting time might be different from the ones that were predicted at the beginning of the trip. These additional times directly impact short-term plans and decisions. Thus, one-time unreliability refers to changes that happen within a particular execution of a travel request.*

Definition 4. ***Daily unreliability***: *When deciding to request a trip (i.e., when choosing whether to travel or not, and in which mode), the user will consider not only the expected quality of service (i.e., delay, waiting time, etc.), but also the range of values that it might take, including the chance of being rejected by the system if there are no vehicles available. Thus, daily unreliability refers to changes that happen to multiple executions of the same travel request across multiple instances of the request, e.g., across multiple days.*

[5] See [24, 27] for an extensive review on unreliability in public transport.

[6] See [23] for an extensive review on unreliability in car-sharing systems in which rides are not shared.

[7] See [35] for a survey on assignment methods under these dynamic and stochastic conditions.

Most mobility systems do not have flexible routes. Public transport's vehicles are characterized by following consecutively always the same fixed path, while non-shared systems always follow shortest paths between origins and destinations. Therefore, unreliability regarding time in the vehicle (beyond congestion) is specific to flexible ridesharing systems. When this flexibility is on-demand (as opposed to previously arranged systems), one-time unreliability emerges because times cannot be predicted in advance. Daily unreliability, on the other hand, deals with the uncertainty of the total travel time (not with its predictions), i.e., this is the type of unreliability usually considered when measuring the value of reliability VOR. Uncertainty regarding waiting times is also present in public transport, traditional taxis and non-shared on-demand systems; however, the uncertainty regarding the route that the vehicle will take is specific to on-demand ridesharing.

How to measure unreliability should depend on these classifications. In this paper, we focus on the unreliability induced by the operators. In fact, operators might take decisions that control unreliability, at the cost of degrading some other quality-of-service indicators of the system. For instance, they could decide that once a request r has been assigned to a vehicle v, then every future request can only be served by v after completing r, which would reduce the uncertainty to a minimum. Nevertheless, such a rule would reduce the number of available vehicles in the system, inducing other undesirable effects such as an increase in waiting times or in the number of rejected requests. Therefore, there is a trade-off between reliability and traditional quality-related indices of the system.

3 Measures of Unreliability

Let us consider a user that poses a request r to the ridesharing system. The following concepts will be needed to provide formal definitions of the unreliabilty measures[8]:

- *Request time* t_r, which is the moment in which the trip was requested to the system.
- *Origin* o_r and *destination* d_r of the trip.
- *Real waiting time* $t_w(r)$, which is the difference between the pick-up time and the request time.
- *First-announced waiting time* $t_w^1(r)$. "First-announced" measures require a more detailed explanation, as their existence is the core of the one-time unreliability sources studied in this paper. Soon after the request is posed, it is processed by the system and the user is assigned to a vehicle (unless it is rejected because there are not enough available vehicles). The first-announced waiting time is the waiting time that would be achieved if the vehicle's itinerary remains unchanged until picking the passenger up. However, $t_w^1(r)$ might be different from the real waiting time $t_w(r)$, because before the pick-up takes place, new requests might emerge, which could add extra detours to the assigned vehicle; or the passenger could get reassigned to another vehicle before pick-up.
- *Real in-vehicle time* $t_v(r)$, which is the time ellapsed between the pick-up and drop-off of the request.
- *Real detour* $t_d(r)$, which is the difference between $t_v(r)$ and the time required by the shortest path between o_r and d_r.
- *First-announced in-vehicle time* $t_v^1(r)$ and *first-announced detour* $t_d^1(r)$, which are defined analogously as $t_w^1(r)$: they would be the achieved in-vehicle time and detour if the vehicle does not change its itinerary from the first announced assignment until the passenger is dropped-off.
- *Real delay* $D(r)$, which is the difference between the real arrival time provided by the system and the arrival time achieved if traveling by private car. Real delay can be decomposed into the waiting time and the detour, both nil when traveling privately:

$$D(r) = t_w(r) + t_d(r)$$

[8] A glossary with all the symbols used throughout the paper is provided in the Appendix.

- *First-announced delay*:
$$D^1(r) = t_w^1(r) + t_d^1(r)$$
- For all indices, we define the difference between the real and the first-announced ones, denoted by Δ:
$$\Delta t_w(r) = t_w(r) - t_w^1(r)$$
$$\Delta t_v(r) = t_v(r) - t_v^1(r)$$
$$\Delta t_d(r) = t_d(r) - t_d^1(r)$$
$$\Delta D(r) = D(r) - D^1(r)$$

  Note that $\Delta t_d(r) = \Delta t_v(r)$, because the difference between the detour and the in-vehicle time is the shortest distance between o_r and d_r, which is fixed, i.e., the in-vehicle time increases only due to the detour.

These last indices Δt_w, Δt_v, Δt_d, and ΔD are the crucial ones for studying one-time unreliability, as they measure exactly how the quality of service provided by the operator is degraded when admitting new requests while serving previous ones. These indices may take negative values, which imply a better performance of the system than expected. For example, consider the case that traveler r_1 is waiting for a vehicle v that was en route to pick up r_2 before her, but r_2 is suddenly assigned to another vehicle, reducing the time required by v to arrive at o_{r_1}. In the remainder of the paper we consider the positive values for these indices, and truncate to zero those that are negative, to study the unreliability effects that are negative for the users

Daily unreliability, on the other hand, deals with the different realizations of the indices when repeating the same request, i.e., how t_w and t_d can take different values depending on the circumstantial co-travelers for a same request.

An example of these concepts is shown in Fig. 2, representing the update of a vehicle's itinerary. For simplicity, we assume that all arcs have unitary length. We will focus on the brown passenger b. When she and the blue passenger first request their trips, the vehicle announces that it will move from its current position (CP) to pick-up the blue passenger first, then it will pick-up the brown passenger, and then it will drop-off both in the inverse order. Therefore, for the brown passenger:

$$t_w^1(b) = 2, \quad t_d^1(b) = 0, \quad t_v^1(b) = 1, \quad D(b) = 2$$

These values are explained as follows: the vehicle needs to traverse two arcs before the pick-up takes place, but then it goes directly to its drop-off, which takes 1 time unit. However, when the red request emerges, it is picked-up just before b and it is dropped-off also just before b, which increases the real waiting time and the detour by 1, so the delay is increased by 2:

$$t_w(b) = 3, \quad t_d(b) = 1, \quad t_v(b) = 2, \quad D(b) = 4$$

Yielding:

$$\Delta t_w(b) = 1, \quad \Delta t_d(b) = 1, \quad \Delta t_v(b) = 1, \quad \Delta D(b) = 2$$

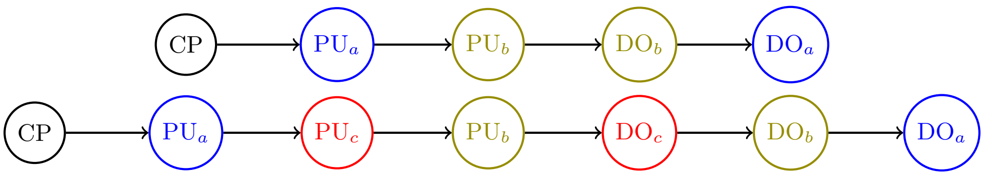


Fig. 2. The updated itinerary of a vehicle. In the upper line, the vehicle has been assigned to serve two passengers only (blue and brown), so it moves from its current position (CP) to serve both trips from the pick-ups (PU) to drop-offs (DO). With this itinerary, the brown request needs to wait for the vehicle to traverse two arcs before the pick-up, and it is in the vehicle during one arc. When the red request emerges (below), the vehicle's itinerary is updated, increasing both the waiting time and the detour of b.

3.1 One-time unreliability

In this subsection we propose and analyze indices that characterize the changes that a user might face *while* her trip is taking place[9]. As exemplified with Fig. 1, the phenomena are present in any on-demand shared system, and the indices proposed here are equally general. However, the specific values that these indices take depend not only on the demand and network, but also on the operational rules and algorithms that define the ridesharing system.

We propose five definitions to analyze the one-time unreliability, that compares the actual realization of a trip with its first announcements. We will compute these measures over each request r, and also over the set of requests that depart[10] from each zone x, whose size will be denoted $Q(x)$, to analyze whether there is a relationship between the number of passengers (in some zones of the transport network) and the unreliability measures.

Definition 5. ***Unreliability in waiting time****: For a passenger r, this refers to the difference $\Delta t_w(r)$ between the first-announced waiting time $t_w^1(r)$ and the actual waiting time $t_w(r)$. Recall that this difference emerges when r is assigned to a vehicle v, but prior to the pick-up of r, one of the following happens: i) either new passengers are assigned to v and being picked-up before r, inducing a longer vehicle's detour that increases the waiting time faced by r; or ii) new requests yield a reassignment process that changes the assigned vehicle to another one with a longer time to arrive at the pick-up node.*

For each request r, we measure the difference $\Delta t_w(r)$ between the announced and the actual waiting time. For a zone x we calculate the average between the requests departing there

$$U_{t_w}(x) = \frac{\sum_{r:o_r=x} \Delta t_w(r)}{Q(x)} \tag{1}$$

We can interpret this definition as conditional expected values, noting that:

$$U_{t_w}(x) = \frac{\sum_{r:o_r=x} \Delta t_w(r)}{|\{r : o_r = x \wedge \Delta t_w(r) > 0\}|} \cdot \frac{|\{r : o_r = x \wedge \Delta t_w(r) > 0\}|}{Q(x)} \tag{2}$$

Which can be rewritten, considering probabilities with respect to requests that depart from x, as:

$$U_{t_w}(x) = E(\Delta t_w(r) | \Delta t_w(r) > 0) \cdot P(\Delta t_w(r) > 0) \tag{3}$$

Then, for each zone, we have two forces that explain unreliability in waiting times: how likely is to face an increase in waiting times while the vehicle has not arrived yet, and the expected magnitude of that increase.

Definition 6. ***Unreliability in detour****: For a passenger r, this refers to the difference $\Delta t_d(r)$ between the detour first-announced by the system $t_d^1(r)$ and the actual one $t_d(r)$. Recall that this difference emerges when, prior to dropping-off r, one of the following happens: i) the vehicle that is carrying her is assigned to new requests that induce new stops with r on board (i.e., the stops take place after r's pick-up but before her drop-off); or ii)new requests yield a reassignment process (before pick-up) that changes the assigned vehicle to a new one that requires a longer in-vehicle time to carry r.*

[9] We measure changes with respect to the first-announced indices defined above. These indices are "naive" when assuming that no changes will take place. They are relevant, however, because they show to which extent understanding these changes is indeed a relevant task for both researchers and practitioners. Moreover, proposing better first-announcements (predictions) is a very complex challenge, as they depend on each specific request and on the general state of the system. Up to our knowledge, there are no public methods to provide these predictions effectively

[10] Analogous destination-related indices can be defined.

For each request r we compute the difference $\Delta t_d(r)$ between the announced and the actual detour. For each zone we define:

$$U_{t_d}(x) = \frac{\sum_{r:o_r=x} \Delta t_d(r)}{Q(x)} \tag{4}$$

Which can be re-written as:

$$U_{t_d}(x) = E(\Delta t_d(r)|\Delta t_d(r) > 0) \cdot P(\Delta t_d(r) > 0) \tag{5}$$

I.e., for each zone the two forces explaining unreliability in detours are how likely is to face an increase in detours, and the expected magnitude of that increase.

Definition 7. ***Unreliability in delay*** *For a passenger r, this refers to the difference $\Delta D(r)$ between the total delay first-announced by the system $D^1(r)$ and the actual one $D(r)$. Recall that this definition includes both waiting time and detour: $\Delta D(r) = \Delta t_w(r) + \Delta t_d(r)$.*

For each request r we compute the difference $\Delta D(r)$ between the announced and the actual delay, which is a condensed way of describing if a completed request faced any changes at all. At a zone level, we obtain

$$U_D(x) = \frac{\sum_{r:o_r=x} \Delta D(r)}{Q(x)} \tag{6}$$

Which can be re-written as:

$$U_D(x) = E(\Delta D(r)|\Delta D(r) > 0) \cdot P(\Delta t_d(r) > 0) \tag{7}$$

Unreliability in delay is explained by analogous two forces: how likely is to face an increase in total delay, and the expected magnitude of that increase. Note that because the delay of a request includes its waiting time and detour, we can conclude that $\Delta t_w(r) \leq \Delta D(r)$ and $\Delta t_d(r) \leq \Delta D(r)$ for each request r, which implies that $P(\Delta t_w(r) > 0) \leq P(\Delta D(r) > 0)$ and $P(\Delta t_d(r) > 0) \leq P(\Delta D(r) > 0)$. That is to say, unreliability in delay is always the largest, which is caused by the first of the said forces, i.e., the probability of facing an increase.

Definition 8. ***Unreliability in rejections****: Each time the system processes all the upcoming requests, it does it together with the old requests that have been assigned to a vehicle but have not been completed yet, to allow for reassignments that might increase the global efficiency. When reassigning, the system might decide to reject some request(s) that had been previously accepted but not picked up yet, if this rejection(s) improves the global efficiency of the system.*

For each request we compute if this “becoming rejected” (after being originally accepted) happens or not; as there is no magnitude associated, we shall look only at how likely is that this happens to a request emerging from x. Defining $\Delta Rej(x)$ as the set of requests that face become rejected and whose origin is in x, we calculate the unreliability in rejections for the zone x as:

$$U_R(x) = \frac{|\Delta Rej(x)|}{Q(x)} \tag{8}$$

Definition 9. ***Number of changes faced per request****: Each request r might face several changes, as assignments are updated iteratively. For each request, we compute the expected waiting time at each assignment, since the first one until it is picked-up, and count how many times this prediction increases; analogously, we count how many times a request's expected detour and delay increase, since its first assignment until it is dropped-off. They will be denoted, respectively, $NC_{t_w}(r)$, $NC_{t_d}(r)$ and $NC_D(r)$.*

Analysis of these measures

The concrete dynamics that every request experience depend directly on the specific operational rules of the ridesharing system. However, some unreliability patterns can be described qualitatively and are valid in general.

Changes in waiting times occur only while the user has already made a request but the vehicle has not arrived yet, whereas changes in detours can take place until the drop-off. Therefore, the probabilities of facing changes are higher when the vehicle has just been assigned, and decrease afterward.

However, the contrary might happen with the probability of getting rejected if the system does not penalize unreliability (i.e., it does not put any special priority on previously assigned requests). To see this, consider a passenger p that has been waiting a long time T to be picked-up, and a passenger p' that has just requested a trip with similar origin and destination. Also consider that the system has to choose between both (due to the availability of seats). With the cost function utilized in this work, which maximizes the quality of service for those requests that are satisfied, then the new request should be selected. This is because its lower real waiting time implies that she is receiving a better quality of service. Such selection can be undesired when studying unreliability, as it implies that the chances of getting rejected are higher precisely for those passengers that have already been waiting for longer times.

Zones that are located in different parts of the network are expected to present different unreliability rates. Requests that are located in the high-demand areas face larger competition for the vehicles, which increases the probability of any change (either increasing delay or becoming rejected). Similarly, requests that connect distant origins and destinations face a higher chance of an increased detour, because they spend more time in the vehicle.

3.2 Daily unreliability

Daily unreliability is defined as the different outcomes that a same trip faces when repeated under varying circumstances. More precisely, if a same trip is repeated ℓ times, we look how many of those are rejected, and for those that are served, we calculate the usual deviation indices within the set of real waiting times, detours and delays. In particular, we are looking at the box-plots and the standard deviation (we could also look at the coefficient of variation, but using it to measure reliability has been "largely disregarded in recent studies" according to [2]). We will consider that two repetitions correspond to a same trip if the origin and destination are the same, and the request times belong to a same "period", i.e., where conditions remain similar (in our experiments, we will use requests emerging between 1 pm and 3 pm -afternoon off-peak- in weekdays excluding Friday).

However, in a real dataset it may be not possible to observe different realizations of the same trip, because it is unlikely that the exact same origin and destination are repeated several times every day. To overcome this issue, one may measure daily unreliability over a small number of artificial extra trips, defined by their origins and destinations and with similar request times. In our experiments, we insert them over the original set of requests, and we repeat each artificial trip several times[11] in such a way that they do not overlap (to prevent those different repetitions to share the same vehicle) and that their request times occur while the general demand pattern remains similar. The assignment method is applied over the whole set of requests (the real and the artificial ones), and the outcomes are measured for each of the artificial trips, so that we can analyze how they vary. Inserting a low number of artificial trips prevents inducing a significant impact on the system as a whole, so the results for each of these artificial trips can be studied in isolation.

[11] For instance, in Section 5 we repeat each artificial trip ten times. There are only three origin-destination pairs that reach this number of repetitions in the considered set of real requests; moreover, one specific node is present in the three of them either as origin or destination, which narrows seriously the analysis that could be done from them.

4 The assignment model

In the remainder of this paper we will compute the proposed measures in a real-life ridesharing system employing a state of the art method to assign passengers and vehicles. For this section, let us formally define a "request" as a single user call, defined by its origin, destination, time of request and number of passengers, and a "group" as a set of requests that can be served together by the same vehicle. The set of requests will be denoted as R, and the set of feasible groups is G (which is a subset of $\mathcal{P}(R)$, the power set of R). These requests have to be assigned to the set of vehicles V, taking into account their current positions and the passengers that they are serving. This assignment process is done iteratively each δt (here we use $\delta t = 1$ minute) with R containing the requests that have accumulated during that lapse and those that can be reassigned, i.e., that were assigned earlier and have not been picked-up yet.

To decide how to assign requests to vehicles and how to match different requests together, we will use the model proposed by [3] with a slight modification that is helpful to study unreliability: we drop the requirement that waiting time for reassigned passengers can't increase. This model can be synthesized in the following three steps:

1. Search for feasible trips (combinations of vehicles, requests and the route to serve them) and build the so-called RGV graph (RGV stands for "requests-groups-vehicles"): This is a bipartite graph, with nodes $G \biguplus V$, such that an arc (trip) τv exists iff the group τ can be conducted by the vehicle v, without exceeding the capacity of the vehicle and without violating constraints that deal with the maximum admissible waiting times and total delay (the numeric values for all the parameters are shown in Table 6 in the Appendix). Each arc has an associated cost that depends on the waiting and in-vehicle time for each user in the group, on the extra time induced to the current users of the vehicle, and on the extra length that the vehicle has to tour (operators' costs). The route in which requests are served is optimized with respect to these costs.

$$c(\tau, v) = \sum_{r \in \tau} [p_w t_w(r, \tau, v) + p_v t_v(r, \tau, v)] + \tag{9}$$

$$\sum_{r \in v} [\Delta p_w t_w(r, \tau, v) + p_v \Delta t_v(r, \tau, v)] + c_0 \Delta L(\tau, v)$$

   Eq. 9 shows the explicit expression for the cost for v to serve τ. The first term stands for the waiting and in-vehicle time for the requests in τ (p_w and p_v are the respective unitary costs), the second term represents the additional times for those requests that were already being served by v, and the third term represents the operators' costs (ΔL is the extra length of the route of the vehicle, and c_0 is the respective unitary cost)[12].
2. Solve an ILP with binary variables, that selects some arcs (i.e., trips) from the RGV graph and some requests to be rejected (with a corresponding penalization in the objective function), such that each request is either rejected or in one selected trip, and each vehicle is at most in one selected arc.
3. A rebalancing process, in which unused vehicles are assigned to the set of rejected requests (because there is a lack of vehicles near to the origin of those requests), without sharing. These requests are not actually being served by those vehicles, but this process just tells the idle vehicles in which direction to move, so the system gets better prepared for the next iteration.

[12] This definition of the costs entails that the system considers directly all the agents involved in the process, i.e., users and operators. For-profit companies might use slightly different cost functions depending on how they price each trip, but they are also interested in minimizing their own costs and in providing high quality of service to attract more users. Therefore, eq. 9 can approximate a cost function for the for-profit case as well.

When these steps are completed, the system begins accumulating new requests. In the following iteration, these new requests are assigned together with the old non-rejected requests that have not been picked-up yet, in order to allow the system to reassign them when it is efficient to do so. As we aim to understand how unreliable the system can be, here we maximize the flexibility of the system by admitting any change, as long as it does not violate the bounds on waiting times and delay of every request. Furthermore, requests that have already been assigned but have not been picked-up yet might be reassigned to a different vehicle (maybe increasing their waiting times) or rejected, if that reduces the total cost of the system. We only drop some of these rules when analyzing directly the trade-off between reliability and flexibility in Section 6.

5 Computing the measures on a real-life case

In this section, we compute the unreliability measures proposed in Section 3 over a real dataset of requests from Manhattan, using the assignment method explained in Section 4. This process illustrates the values of the said unreliability measures, which is achieved twofold:

1. We obtain several general conclusions about on-demand ridesharing systems. In short, we show that they can be very unreliable, because users can face very relevant changes to their schedules.
2. We obtain several specific conclusions, whose validity might be constrained to the scenarios we are studying, but that shows how policy-makers can use the proposed indices to describe and understand better pooled on-demand systems in order to improve them.

We solve the assignment problem over a subset of the publicly available dataset of taxi trips in Manhattan's network (4091 nodes and 9453 edges)[13]. The assignments are computed over the sets of requests that emerge between 1-3 pm on 15/01/2013 for one-time unreliability. For daily unreliability, we consider ten weekdays starting on Monday 14/01/2013. During the off-peak period, congestion is not a very relevant problem, i.e., traveling times are somewhat stable, which allows us to disregard unreliability due to traffic conditions, isolating the effects we are aiming to study in this paper. A period of two hours is selected to keep the demand pattern stable (within the afternoon off-peak), while being long enough to allow vehicles to update their itineraries many times (so one-time unreliability phenomena can show up) and to insert ten copies of the artificial requests (to study daily unreliability). Friday is not considered as a weekday, because it usually presents different travel demand than Monday-Thursday (see [21], for instance, for the case of New York).

5.1 One-time unreliability

The basic scenario considers 2000 vehicles capable of carrying 4 passengers at a time, as in [3]. Removing all requests that have more than 4 passengers leaves a total of 19610 requests for the two hours period.

As we will show in Table 1, the basic scenario has a service rate of 66.5%. This number is lower than what is achieved in [3], because we are using a lower rejection penalty, to allow for a trade-off with the other quality measures, and to see how this is related to unreliability; moreover, we also consider operators' costs (recall that they are assumed proportional to VHT), so requests that require long distances are more likely to be discarded because they induce a higher cost to the system. To obtain sound results, we consider five alternative scenarios, that analyze the impact of different design variables on one-time unreliability: fewer vehicles (FV, 1000 vehicles of capacity 4), smaller vehicles (SV, 2000 vehicles of capacity 3), same total capacity with larger fewer vehicles (LFV, 1600 vehicles of capacity 5), mixed-capacity vehicles (MCV, 2000 vehicles of capacity $\in \{3, 4, 5\}$, that begin evenly

[13] We use this dataset because it provides enough detail to run the simulations, as shown by [3].

distributed across the network), and a scenario with 5000 vehicles and doubled rejection penalty, that achieves a service rate greater than 97% (Table 1), which allows us to study unreliability when rejecting passengers is a less relevant problem (FR, few rejections)[14]. The first three scenarios FV, SV, and LFV, are expected to offer worse quality of service to the users than the basic scenario. On the one hand, fewer or smaller vehicles offer a lower total capacity without any advantages for the users. On the other hand, fewer larger vehicles offering equal total capacity should be worse for the users but better for the operators for slightly different reasons: Users are affected because it is less likely to find an available vehicle nearby, and operators are benefited due to costs that are fixed per vehicle, such as part of the capital costs or drivers' wages (this trade-off between users' and operators' costs has been thoroughly studied in public transport systems, see [19, 12]).

Measure	**Basic scenario**	**FV**	**SV**	**LFV**	**MCV**	**FR**
Rejection rate	34.5%	48.1 %	37.8 %	36.8 %	34.6%	2.29%
Av. Waiting time [min]	1.01	1.21	1	1.12	1.04	1.42
Av. Detour [min]	0.65	0.73	0.52	0.82	0.67	1.39
Av. Delay [min]	1.66	1.94	1.52	1.95	1.72	2.8

Table 1. Measures of quality of service that are not related to unreliability, for the basic scenario and the five alternative scenarios.

Table 1 shows the quality-of- service indicators that are considered when computing the optimal assignments, i.e., without including reliability. As predicted, the first three alternative scenarios provide a worse quality of service than the basic one. Using vehicles of mixed capacities that provide the same total yields almost the same results than uniform capacities (a little worse, though). Last column is interesting, as the higher penalty rates have a clear impact on worsening the quality of service for the served passengers, as it exhibits the largest average delays despite of having more vehicles (a larger fleet is expected to reduce waiting times and detours, something that is shown empirically to happen in the first two columns and in [3]).

Unreliability in rejections and delay

Let us begin analyzing unreliability by looking at the indices that condense the most relevant information. In Table 2 we show unreliability in rejections -the one that affects requests that were originally assigned but are never picked-up- and unreliability in delay -the one that affects requests that were served by the system and that faced any sudden change. In the basic scenario, more than one quarter of the total requests faces some change, and this number increases to 36.1% when we take the requests that were immediately rejected out of the picture, because they cannot face any change. Note that 36.1% is calculated as 26.1% (the percentage of requests facing any change) divided by the percentage of first-accepted requests 72.35%, which in turn results from the sum of 65.5% (the number of requests that are served by the system, from Table 1) and 6.85% (requests that become rejected, first row in Table 2). This figure (in bold in Table 2) unifies the most relevant information: if you request to be transported by the ridesharing system, and you are accepted with some first predictions, how likely is that these predictions are not fulfilled?

Moreover, the last three rows of the basic scenario are also informative: if a passenger's delay increases with respect to its original prediction ($E(\Delta D|\Delta D > 0)$, from equation 7), her extra delay will be (on average) as large as the average extra delay of the whole system; the average passenger faces 0.35 changes on the

[14] Of course, this low rejection rate could be achieved with even more vehicles and without increasing the rejection penalty. However, a very large fleet would offer non-shared trips to almost every user, preventing the emergence of the unreliability phenomena we are studying

predictions, but the maximum number of changes is 6, which makes any prediction barely useful.

These conclusions are robust, as shown by the other columns: the number of passengers facing changes is quite large, and the average extra delay is very similar to the average total delay of the system. When we compare across columns, two remarkable conclusions emerge: more vehicles imply more reliability, but the opposite happens with respect to the size of the vehicles. The comparison with the "smaller vehicles" scenario is illustrative: although it is supposed to be a worse system (same number of vehicles of a smaller size), it offers better reliability both regarding the chance of staying accepted and of keeping the first-announced delay.

These relationships between fleet's conditions and unreliability can be interpreted: having more vehicles increases the chance of assigning idle vehicles to new requests, which keeps the conditions for previous requests. On the other hand, when a vehicle becomes full, its passengers will face no new changes, which happens more often with small vehicles. This is why the scenario with larger fewer vehicles provides the worst total results.

The scenario with mixed capacities have better results than the basic one regarding rejections, but worse regarding delay, when measuring either the chance of getting an extra delay or its magnitude. The last column reveals that even a system with a very low rejection rate can be quite unreliable. In this case, changes are related mostly with increasing total delay: these changes are necessary to achieve the low rejection rate, as serving most of the new requests requires updating the itineraries of a large portion of the vehicles.

In all, Table 2 verifies that on-demand ridesharing systems can be very unreliable if they rely on first predictions, because many users face changes due to the upcoming requests. The bold row exhibits numbers that are always at least almost one third, and that can be higher than one half. The high variance among these numbers can be syntehsized by saying that unreliability is always an issue, but its relevance depends heavily on the fleet conditions. First predictions are of little use if changes are so usual, so developing techniques to provide better predictions or to make these changes less frequent is crucial for these systems.

Measure	**Base**	**FV**	**SV**	**LFV**	**MCV**	**FR**
Requests that become rejected	6.85%	10.7%	5.25%	6.3%	4.83%	1.14%
$P(\Delta D > 0)$	19.2%	22.2 %	17.2%	22.3 %	19.6%	31.3%
Requests that face changes	26.1 %	32.9 %	22.5%	28.6%	24.4%	32.4%
First-accepted requests that face changes	**36.1%**	**52.6%**	**33.3%**	**41.2%**	**34.7%**	**32.8%**
$E(\Delta D \vert \Delta D > 0)$ [min]	1.68	1.87	1.7	1.74	1.83	2.1
Av. NC_D	0.35	0.46	0.32	0.41	0.35	0.61
Max. NC_D	6	7	6	7	6	8

Table 2. Synthesis of the most relevant unreliability measures for each scenario. We highlight the measure that synthesizes how likely is to face a change for requests that are first assigned to some vehicle.

Unreliability in waiting time and detour

We now analyze how changes in total delay split between waiting times and detour (Table 3). The first apparent conclusion is that detours are much more unreliable than waiting times, both in magnitude and in their chance to occur, which can be explained as users spend more time in the vehicle than waiting for it. Moreover, when an increase in the detour occurs, its magnitude is much larger than the average detour of the system, which is explained because many users can have zero detour. When we compare the different scenarios, one conclusion differs from the deductions explained in the previous paragraph: regarding waiting times, smaller vehicles are less reliable. This change on the conclusion is coherent with the interpretation we already provided, that rests on vehicles being full: if a passenger is waiting for a vehicle, it means the vehicle is not full yet, so it has room for adjusting its route to include more passengers. However, the number of vehicles has a much larger impact, as shown by the two scenarios with fewer vehicles, whose indices are all worse than the results obtained by the basic one.

Measure	Basic scenario	FV	SV	LFV	MC	FR
$P(\Delta t_w > 0)$	5.9 %	8.9 %	6.5 %	6.7 %	6.2 %	7.73%
$E(\Delta t_w \| \Delta t_w > 0)$ [min]	1.22	1.36	1.35	1.35	1.3	1.28
Av. NC_{t_w}	0.13	0.22	0.14	0.15	0.14	0.19
Max. NC_{t_w}	4	4	4	4	4	4
$P(\Delta t_d > 0)$	16.1 %	17.6 %	13.5 %	19.4 %	16.7 %	29.3%
$E(\Delta t_d \| \Delta t_d > 0)$ [min]	1.67	1.83	1.64	1.78	1.69	2.08
Av. NC_{t_d}	0.27	0.31	0.22	0.33	0.22	0.55
Max. NC_{t_d}	6	6	6	6	5	8

Table 3. Decomposition of unreliability regarding delay into waiting time and detour for each scenario.

Spatial distribution of the unreliability indices

Let us analyze now how these results distribute in space, using the basic scenario (that presents intermediate results among all the scenarios). To do this, we first divide the network into zones, each one being a set of nodes. The partition is computed following the method proposed by [45], based on finding a "center" for each zone: define an upper bound (here we consider 150 [s]) to the distance between any node and the center that corresponds to its zone. Then, find the smallest set of centers that respect these bounds, when every node is linked to its closest center. This set is found through an ILP, that divides the Manhattan graph into 167 zones in our case.

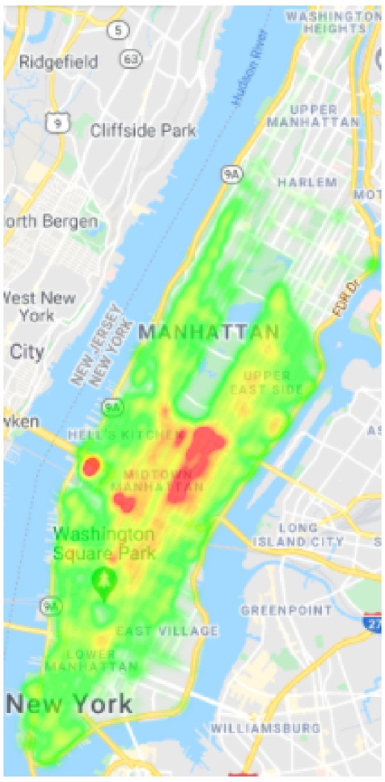


Fig. 3. Departure rate at each node in the graph. Red dots generate more requests.

Our aim now is to study $U_{t_w}(z), U_{t_d}(z), U_D(z)$ and $U_R(z)$, for every zone z. For the first three indices, we use the relationships explained by Eqs. (3), (5) and (7), i.e., we split the analysis into the probability of facing changes and the magnitude of these changes when they occur. To do this, first we show Figs. 3 and 4. Fig. 3 is a heatmap showing where are the origins of the requests located. There is a clear

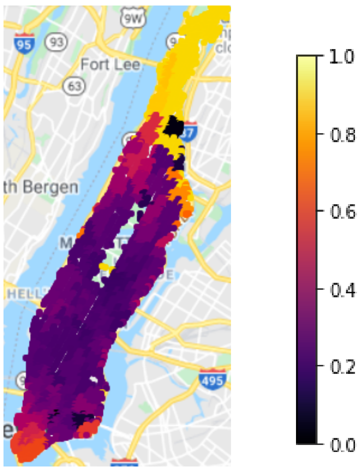


Fig. 4. Average length of the trips departing from each zone. Indices, that lie between 0.75 and 14.1 km, are normalized and the colorbar is at the right.

red area in the "center" of the network, that generates most of the trips; this area will be used as a reference in what follows. Fig. 4 shows the average length of trips departing from each zone, which will prove useful to analyze one-time unreliability in detours. Note that requests that travel from the center present, on average, shorter trips.

We begin studying the probabilities of facing changes at all, i.e., the probability that a request emerging at each zone either become rejected or its total delay increases, after being accepted. These probabilities are shown in Fig. 5, in which darker colors represent lower probabilities of facing changes. The probability of becoming rejected after accepted (Fig. 5 left) is clearly higher for requests that depart from the high-demand area, which fits intuition as the competition for a vehicle is more intense there; moreover, most zones in the south of the network, as well as all the zones in the north of the network, present almost zero chance for becoming rejected once accepted. The analysis regarding the chance of facing an extra delay (Fig. 5 center) is similar but less conclusive, as the zones with high probability are a bit more spread.

The spatial distribution of the probabilities of facing a delay are better understood recalling that that extra delays are caused either by extra waiting time or by extra detours (or by both). Fig. 6 exhibits the probabilities of facing an increase in waiting times (Fig. 6 left) and in detour (Fig. 6 center). Waiting times are much more unstable for users that depart from the center of the network, which is again related with the competition for a vehicle: while waiting for the vehicle to arrive, the chances that another request emerges nearby are high (notably, there is a large region in the north of the city in which requests never face extra waiting time). Extra detours, on the other hand, are more evenly distributed across the city, reflecting that two explanations are acting simultaneously: requests that depart from the center have more requests nearby, but their trips are shorter, as shown by Fig. 4. As extra detours are more common than extra waiting times (see Table 3), the somewhat even distribution of extra delay is mostly explained by extra detours.

Finally, Fig. 7 reveals that the average increase in any of these measures, constrained to the users that face changes, is distributed evenly across the city.

In all, we can synthesize this analysis by noting that this system is less reliable for users that depart from the most demanded zones in the city, which is explained mostly by the chance of getting rejected or by facing a waiting time longer than the first-announced one. However, if a user faces a change, the magnitude of this change does not depend on the user's origin. This synthesis could be used, for instance, to modify the algorithm to aim for a fairer spatial distribution of the unreliability indices.

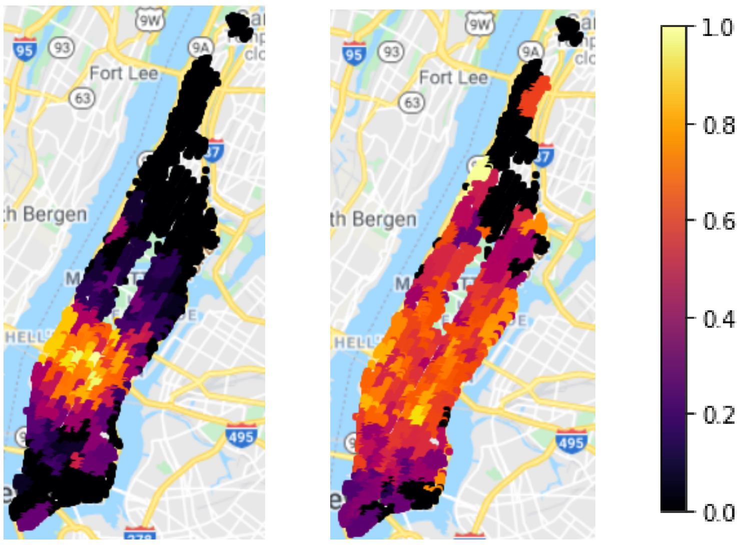


Fig. 5. Spatial distribution of the probability of becoming rejected after being accepted (left), and of the probability of facing an extra delay (center). Indices are normalized and the colorbar is at the right.

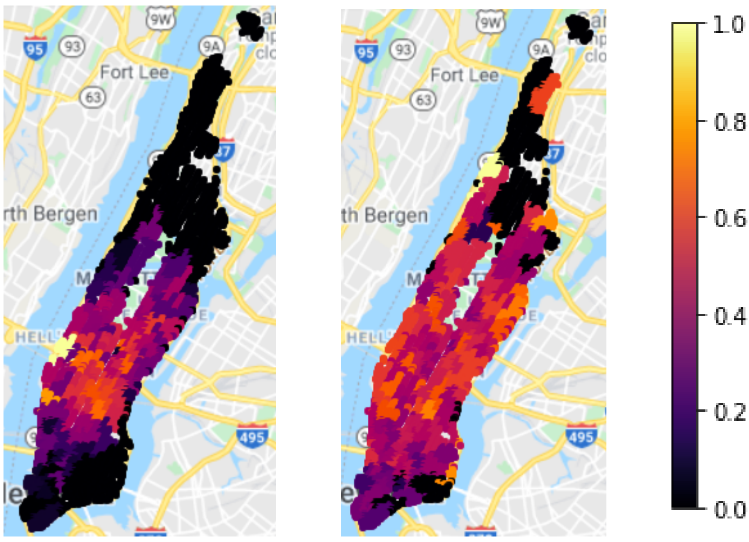


Fig. 6. Spatial distribution of the probabilities of facing extra waiting time (left) or extra detour (center). Indices are normalized and the colorbar is at the right.

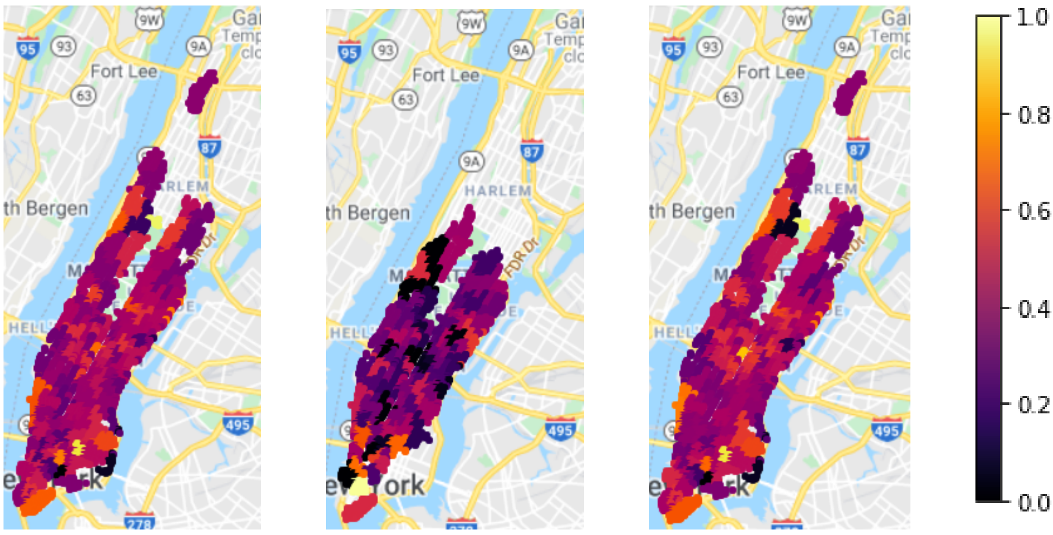


Fig. 7. Spatial distribution of the average increase in delay (left), waiting time (center-left), and detour (center-right), for those users that face changes. Indices are normalized and the colorbar is at the right.

5.2 Daily unreliability

Daily unreliability is studied inserting artificial requests during ten weekdays, which are enough to obtain a hundred repetitions per extra request, a number that enables making simple statistical analysis. Fig. 8 shows the number of requests per day (left) and the distribution of the requests' size (right). As expected by using only weekdays (excluding Fridays), daily requests present tiny variations. The pie chart reveals that although most requests are unitary, the number of requests of a larger size is significant, rising the average number of passengers per request to 1.26. The origins and destinations of the artificial requests are shown in Figs. 9, and described as follows:

- R_1 has its origin and destination within the center of the network. It is the shortest of the four requests, but in an area where there are many requests, inducing a higher degree of shared trips and a higher competition for the vehicles.
- R_2 connects two quite distant points, that are both located in very low-demand areas, but that require crossing the center to be linked. The nodes are chosen such that the operators' costs of serving the trip are lower than the rejection penalty.
- R_3 has nodes that are located at an intermediate distance. The origin of R_3 is in a low-demand area, which implies that not many vehicles will be around, and it goes to a high-demand area, which might induce sharing by the end of the trip.
- R_4 is a short trip, taking place out of the center.

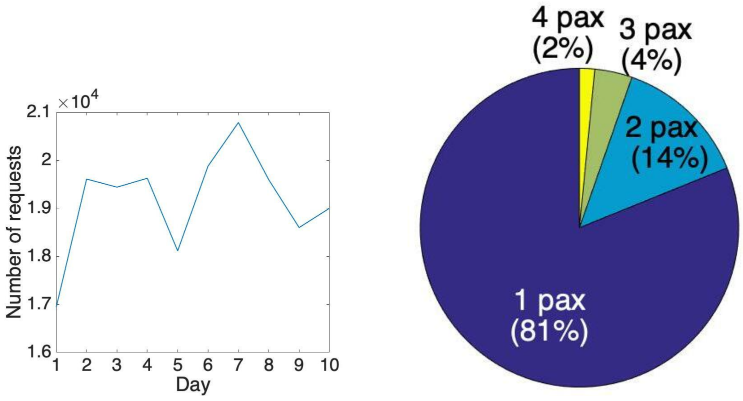
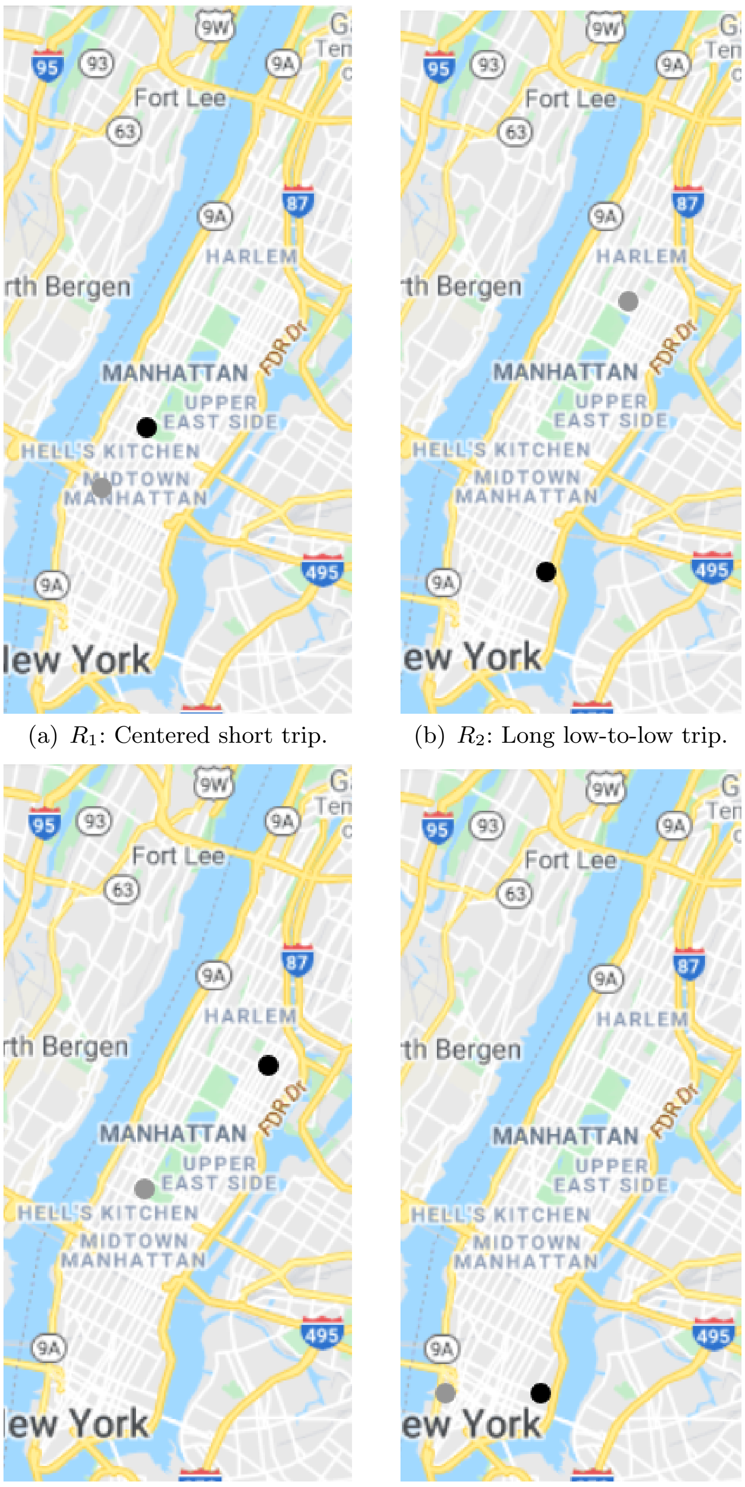


Fig. 8. Number of requests emerging between 1-3 pm on each of the ten days used for studying daily unreliability.

Results are shown in Table 4, which contains the results for all the requests in every scenario, and in Figure 10 that synthesizes the results obtained in the the basic scenario (2000 vehicles of capacity 4). Let us begin analyzing this scenario. A reliable system should respond equally each time a request is repeated. Regarding rejections, this means that a request should always be accepted, or always be rejected. A fully reliable system would always serve the same requests. In this case, results show that R_1 and R_4 (the two short trips) are very reliable, as they are almost never rejected. In the case of R_2, although its service rate is the worst, from a reliability point of view the situation is not bad, as the outcome is stable. In other words, users requesting such a trip (a long trip connecting two peripheral points) would probably never use the on-demand ridesharing system, because they know that the chance of finding a vehicle is too low. R_3 faces the most unreliable situation, as both being served and being rejected occur regularly. Note that R_3 is a trip of intermediate length, moving from a low-demand area (the north zone) to the center.



(a) R_1: Centered short trip.

(b) R_2: Long low-to-low trip.

(c) R_3: Medium-length low-to-high trip.

(d) R_4: Off-center short trip.

Fig. 9. Origins (black) and destinations (grey) of the additional requests.

Let us now analyze the conditions of the trips that are served. The magnitudes of the standard deviations and the averages of the achieved detours are comparable, meaning that all requests present quite some unreliability regarding detours. That is to say, each time a user travels with this system, she has to be willing to face quite different routes, sometimes the shortest ones and other times with many deviations. The situation regarding waiting times is more dependant on the type of request, yet variations are significant for all of them.

The request that travels within the center R_1 presents the largest variations in both indices. The trip that goes towards the center R_3, that was shown to be the most unstable one regarding rejections, presents waiting times that are quite unvarying. Variations for R_2 are harder to analyze, as there are only 16 successful rides; however, those rides show the lowest detours' variations among all the requests, and bounded changes with respect to the waiting times. Finally, R_4 presents intermediate results.

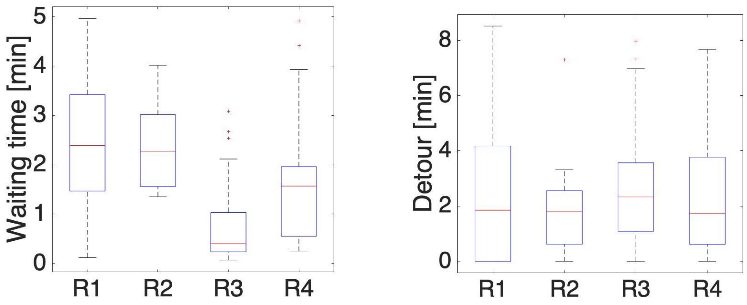


Fig. 10. Box-plots of the resulting waiting times (left) and detour (right) for each repetition of the four requests.

Scenario	**Measure**	R_1	R_2	R_3	R_4
Basic	Rejected repetitions	4%	84%	31%	0%
	Av. Waiting time [min]	2.36	2.37	0.59	1.38
	Std. deviation	1.26	0.86	0.69	1.09
	Av. detour [min]	2.45	1.91	2.64	2.31
	Std. deviation	2.36	1.77	2.23	2.1
FV	Rejected repetitions	23%	84%	57%	56%
	Av. Waiting time [min]	2.68	2.25	1.05	2.31
	Std. deviation	1.27	1.17	0.76	1.33
	Av. detour [min]	3.61	2.51	3.46	2.45
	Std. deviation	2.46	1.41	2.36	2.07
SV	Rejected repetitions	6%	82%	31%	0%
	Av. Waiting time [min]	2.36	1.15	0.57	1.36
	Std. deviation	1.48	0.57	0.67	1.14
	Av. detour [min]	2.3	2.03	2.24	2.04
	Std. deviation	2.32	1.27	1.96	1.89
LFV	Rejected repetitions	7%	90%	30%	13%
	Av. Waiting time [min]	2.42	1.92	0.78	1.73
	Std. deviation	1.27	1.06	0.81	1.17
	Av. detour [min]	3.52	3.63	2.78	2.48
	Std. deviation	2.52	3.15	2.19	2.23
MC	Rejected repetitions	5%	89%	29%	0%
	Av. Waiting time [min]	2.39	2.2	0.68	1.41
	Std. deviation	1.51	1.17	0.77	1.18
	Av. detour [min]	2.61	2.78	2.6	1.78
	Std. deviation	2.24	2.72	2.33	1.71
FR	Rejected repetitions	0%	0%	0%	0%
	Av. Waiting time [min]	1.53	2.62	0.74	1.2
	Std. deviation	1.43	1.36	0.67	1.3
	Av. detour [min]	1.76	2.6	2.23	1.53
	Std. deviation	2	2.23	2.09	1.8

Table 4. Global results for each artificial request in the different scenarios.

The general conclusion that trips might be very different each time they are performed continues to be true in all the other scenarios, as the standard deviations of both waiting times and detours are comparable with their averages. This conclusion is particularly true for detours, which is the type of unreliability specific to ridesharing systems as systems in which routes are not known a priori. A more detailed analysis reveals, however, that some conclusions are always valid and some are scenario-specific:

- All the requests might face uncertainty regarding they are going to be served or not. The case of R_4 is illustrative: although it presents the lowest rejection rates in most of the scenarios (many times reaching 0%), when the number of vehicles is low (FV) this is not longer true, as R_1 is better served, revealing that the system is focusing its few cars mostly on the center of the network.
- The requests R_2 and R_3 are the most rejected ones, but present the lowest variation on their waiting times, which is related to the fact that these requests' origins are placed in a very low-demand area. The standard deviation of detours, on the other hand, depends strongly on the scenario.

- There are high differences among the scenarios when analyzing rejection rates, waiting times and detours. However, standard deviations do not change that much, meaning that without policies that control uncertainty, **these systems are similarly unreliable regardless of the fleet conditions**. Even in the last scenario, that provides the very best service rate, unreliability in waiting times or detours is not reduced.
- The scenario with mixed capacities reveals another interesting property when compared with the basic scenario, as it presents larger standard deviations for all the requests' waiting times and for two of the requests' detours, with some differences being quite significant. This can be explained, as which type of vehicle is serving you is yet another source of variation: if you are assigned to a small vehicle, then you are sharing with fewer other passengers and your waiting times and detour are going to be low, whereas the contrary happens when a large vehicle is serving you. Note that most users will ride large-capacity vehicles (because there are the same number of vehicles per type, meaning that there are more chairs offered in the larger vehicles), which is why for most requests average waiting times and detours are slightly higher here than in the basic scenario.

All in all, for the given fleets considered here, the four inserted requests face relevant uncertainty when deciding if using the ridesharing system: many times they do not know if they are going to be accepted, and when they are, waiting times and detours can easily take values that are the double or the half of the average ones.

These indices might be used by policy-makers in order to modify the assignment rules to improve reliability for all: for instance, one might define a target waiting time and detour for each type of request, and penalize (or avoid) any assignment that yields results that are too far from that target. Note that such penalization requires the system to pre-define a better quality of service for some types of requests (something that already happens in public transport, that offers different frequencies depending on the line), which shows again the existence of a trade-off between reliability and other indices of quality of service, something that is studied in depth in Section 6.

As a sensitivity analysis, we have run the same experiment with the basic scenario, replacing each node with its closest neighbor. Results are robust, with the only significant differences occurring with R_2, that is rejected only 47% of the times, and presenting higher standard deviations (similar to the ones of R_1). This change happens because the distance between the original R_2 and its closest neighbour is more than 2 minutes, which shows that displacements that are local, but not so small, can have relevant impacts on the resulting quality of service and in its variations.

6 The trade-off between one-time unreliability and other quality-of-service indices

Previous analyses are valid because one-time unreliability was *admitted* as a design decision: the operating rules of the system consider the chance of rejecting passengers that were previously accepted, if these rejections increase the global efficiency of the system, and also admit that a vehicle includes a new detour when it is on its way to pick-up someone, increasing her waiting time[15]. What happens if these decisions are changed? Is it possible to eliminate unreliability just by forbidding these changes?

We study these ideas through slight modifications on the algorithm that create two new scenarios:

1. *Fully controlled scenario:* When a passenger gets assigned to a vehicle, it is marked as non-rejectable for future assignments, and her maximum waiting time

[15] The same happens with total delay, but if in-vehicle detours are not admitted, actual sharing would occur only at very specific circumstances, when two passengers are matched in the same iteration, i.e., when their request times are very similar. In other words, eliminating U_D by these means would turn the whole system in a (almost) non-shared one.

is updated to the predicted one. Note that we are still admitting for reassignments, if a different vehicle can pick her up in the same (or lower) time.

2. *Intermediate scenario:* Becoming rejected after being accepted might be more annoying than facing an extra waiting time. In this intermediate scenario, assigned passengers are also marked as non-rejectable, but their waiting time might increase (always bounded by the common maximum admissible waiting time).

These changes automatically make $U_R = 0$ for both scenarios, and $U_{t_w} = 0$ for the fully controlled one. Nevertheless, as the system becomes less flexible (the feasible set of solutions is now smaller), it is expected that the optimization process yields worse solutions, i.e., a worse combination of waiting time, detour and rejection rate.

Table 5 compares the global figures for the three strategies. The global costs are calculated using the same relative weights (see Table 6 in the Appendix) for the quality attributes that we use to optimize the assignment; the fleet is the one of the base scenario (2000 vehicles of capacity 4).

Scenario	Av. Waiting Time	Av. Detour	Av. Delay	Rejected Requests	Av. users' cost
Base	1.01	0.65	1.66	34.5%	9.56
Intermediate	1.51	1.01	2.52	31.2%	10.3
Fully controlled	0.77	0.94	1.71	45.6%	11.6

Table 5. Quality of service when unreliability is controlled. All times are in minutes.

Waiting times are reduced in the fully controlled scenario, which is a natural consequence of forcing waiting times to stick to their first announcements. However, detours increase because the system is forced to pick-up new passengers when previous ones are on board. The most relevant impact, however, is on the rejection rate, which increases dramatically due to the inflexible conditions.

The intermediate scenario exhibits different changes. Here, forbidding the rejections of previously accepted passengers reduces the rejection rate. Nevertheless, the quality of service for the accepted requests gets strongly degraded, as both waiting times and detours are increased in about 50% on average. This degradation happens due to the dynamic nature of the system: sometimes requests are first accepted because there are no other requests competing with it. When new requests emerge, removing a few of this previously accepted request might enable some vehicles to arrive quickly at the new origins, but such removals are forbidden in the intermediate scenario.

In all, looking at the last column of Table 5 is illuminating: the more controlled the unreliability, the higher the average users' costs. That is to say, there is indeed a trade-off between unreliability and the other traditional measures of quality of service. This reinforces the need of more sophisticated ideas to control unreliability.

7 Conclusions

In this paper, we have described and formulated the new sources of unreliability that emerge in on-demand systems whose vehicles are used simultaneously by different users. We focus on those sources that could be directly controlled by the system, namely the waiting times and detours induced by co-sharing passengers together with the chance of being rejected. We have noted that two types of unreliability phenomena can be identified: on the one hand, for a given origin-destination and departing time, a passenger faces uncertainty regarding waiting times (which also happens in public transport and in non-shared on-demand systems) and regarding the route that the vehicle will follow with its induced in-vehicle time (which does not happen in other systems); on the other hand, the appearance of new users might increase the waiting time or the detour for passengers that are already assigned to a vehicle, and it can make some of them to become rejected. The first type is defined as "daily unreliability" and the second one is the "one-time unreliability".

One-time unreliability was studied by simulating the operation of a ridesharing system during two hours over Manhattan, using a real dataset of taxi requests, and measuring for each passenger the difference between the first announced waiting times, detours and total delay, with the real ones. We also identified when some requests are accepted but then become rejected while the vehicle is on the way to pick them up, and for those that are served we counted how many times did the predictions change. We calculated all these measures globally, and we also analyzed their distribution in space, by taking averages over the requests that depart from each zone.

We found that, for different fleet conditions, a third or more of the assigned requests faces some change, either becoming rejected or increasing their total traveling times; when the latter happens, the average increase on total delay is similar to the average delay of the system. The average number of prediction changes is lower than one, but some users can face up to six to eight changes (depending on the fleet that is being used) on total delay's prediction, and up to four changes in waiting time's prediction. Smaller fleets provide a much more unreliable service. Regarding the spatial distribution of the measures, it is shown that requests that are originated at the most demanded zones (in the center of the network) have a larger probability of facing changes.

To study daily unreliability, we defined four representative origin-destination pairs, and we inserted several copies of each of them (with a time window that prevents matching them together), in order to compare the outcome of each copy. All these requests present some varying characteristics when they are served, which is true for different fleet conditions. Requests that depart from the most demanded areas present higher variations on their waiting times, but also a larger certainty of being served (not rejected).

Finally, we studied the relationship between unreliability and the other measures that define the quality of service of the system, namely average waiting times, delay and rejections rate. To do so, we compared the same simulations but with two different scenarios: in both once a passenger is assigned to a vehicle, she can't get rejected, and in one of them new assignments can't increase the waiting time of users that are already waiting to be picked-up. Doing so makes the system more reliable (considering the one-time unreliability), but results show that at the cost of degrading the other indices of quality of service.

Some of the specific conclusions explained above might depend on the chosen scenarios, i.e., on the demand, fleet conditions and on the assignment rules. However, the identification of these types of unreliability as a crucial issue is valid for any on-demand ridesharing system, as well as the trade-off between reliability and other aspects of quality-of-service. Moreover, such specific conclusions verify that the measures proposed in this paper help understanding the daily operation of these mobility systems, and can be used to modify and improve the assignment rules to serve different reliability-related purposes.

There is plenty of room for future research regarding unreliability in ridesharing systems. The most relevant one is how to control it, i.e., studying how to alter the assignment procedures to provide more accurate times without increasing the rejection rate or passengers' average delay and waiting times. Techniques to control unreliability are likely to be different if we aim to reduce one-time unreliability or daily unreliability. Sophisticated prediction tools, that are able to inform expected waiting times and detour by looking at the state of the system as a whole, is yet another way to face the same issues.

Acknowledgements

We want to thank the insightful comments of the anonymous referees. This research was partially funded by Didi Udian Technology (Shenzhen) Co. Ltd.

References

1. Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research 223*, 2 (2012), 295–303.
2. Alonso-González, M. J., van Oort, N., Cats, O., Hoogendoorn-Lanser, S., and Hoogendoorn, S. Value of time and reliability for urban pooled on-demand services. *Transportation Research Part C: Emerging Technologies 115* (2020), 102621.
3. Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences 114*, 3 (Jan. 2017), 462–467.
4. Alonso-Mora, J., Wallar, A., and Rus, D. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Sept. 2017), pp. 3583–3590.
5. Bansal, P., Liu, Y., Daziano, R., and Samaranayake, S. Can mobility-on-demand services do better after discerning reliability preferences of riders? *arXiv preprint arXiv:1904.07987* (2019).
6. Beirão, G., and Cabral, J. S. Understanding attitudes towards public transport and private car: A qualitative study. *Transport policy 14*, 6 (2007), 478–489.
7. Carranza, V., Chow, K., Pham, H., Roswell, E., and Sun, P. Life cycle analysis: Uber vs. car ownership. *Environment 159* (2016), 1–19.
8. Carrion, C., and Levinson, D. Value of travel time reliability: A review of current evidence. *Transportation research part A: policy and practice 46*, 4 (2012), 720–741.
9. Chen, X. M., Zahiri, M., and Zhang, S. Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach. *Transportation Research Part C: Emerging Technologies 76* (2017), 51–70.
10. Fagnant, D. J., and Kockelman, K. M. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation 45*, 1 (2018), 143–158.
11. Fielbaum, A. Strategic public transport design using autonomous vehicles and other new technologies. *International Journal of Intelligent Transportation Systems Research 18* (2020), 183–191.
12. Fielbaum, A., Jara-Diaz, S., and Gschwender, A. Beyond the Mohring effect: Scale economies induced by transit lines structures design. *Economics of Transportation 22* (2020), 100163.
13. Friman, M., Edvardsson, B., and Garling, T. Perceived service quality attributes in public transport: Inferences from complaints and negative critical incidents. *Journal of Public Transportation 2*, 1 (1998), 4.
14. Gargiulo, E., Giannantonio, R., Guercio, E., Borean, C., and Zenezini, G. Dynamic ride sharing service: are users ready to adopt it? *Procedia Manufacturing 3* (2015), 777–784.
15. Gurumurthy, K. M., and Kockelman, K. M. Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from Orlando, Florida. *Computers, Environment and Urban Systems 71* (2018), 177–185.
16. Henao, A., and Marshall, W. E. The impact of ride-hailing on vehicle miles traveled. *Transportation 46*, 6 (2019), 2173–2194.
17. Ho, S. C., Szeto, W., Kuo, Y.-H., Leung, J. M., Petering, M., and Tou, T. W. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological 111* (2018), 395–421.
18. Hyland, M., and Mahmassani, H. S. Operational benefits and challenges of shared-ride automated mobility-on-demand services. *Transportation Research Part A: Policy and Practice 134* (2020), 251–270.
19. Jara-Daz, S., and Gschwender, A. The effect of financial constraints on the optimal design of public transport services. *Transportation 36*, 1 (2009), 65–75.
20. Jara-Díaz, S., Fielbaum, A., and Gschwender, A. Optimal fleet size, frequencies and vehicle capacities considering peak and off-peak periods in public transport. *Transportation Research Part A: Policy and Practice 106* (2017), 65–74.
21. Jin, S. T., Kong, H., and Sui, D. Z. Uber, public transit, and urban transportation equity: A case study in New York City. *The Professional Geographer 71*, 2 (2019), 315–330.
22. Kucharski, R., Fielbaum, A., Alonso-Mora, J., and Cats, O. If you are late, everyone is late: Late passenger arrival and ride-pooling systems' performance. *Transportmetrica A: Transport Science Forthcoming*.
23. Lee, U., Kang, N., and Lee, I. Shared autonomous electric vehicle design and operations under uncertainties: a reliability-based design optimization approach. *Structural and Multidisciplinary Optimization* (2019), 1–17.
24. Levinson, H. S. The reliability of transit service: An historical perspective. *Journal of Urban Technology 12*, 1 (2005), 99–118.
25. Li, Z., Hensher, D. A., and Rose, J. M. Willingness to pay for travel time reliability in passenger transport: A review and some new empirical evidence. *Transportation Research Part E: Logistics and Transportation Review 46*, 3 (2010), 384–403.
26. Liu, Z., Miwa, T., Zeng, W., Bell, M. G., and Morikawa, T. Dynamic shared autonomous taxi system considering on-time arrival reliability. *Transportation Research Part C: Emerging Technologies 103* (2019), 281–297.
27. Muñoz, J. C., Soza-Parra, J., and Raveau, S. A comprehensive perspective of unreliable public transport services' costs. *Transportmetrica A: Transport Science 16*, 3 (2020), 734–748.
28. Nourinejad, M., and Roorda, M. J. Agent based model for dynamic ridesharing. *Transportation Research Part C: Emerging Technologies 64* (2016), 117–132.
29. Paquette, J., Cordeau, J.-F., and Laporte, G. Quality of service in dial-a-ride operations. *Computers & Industrial Engineering 56*, 4 (2009), 1721–1734.
30. Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. A review of dynamic vehicle routing problems. *European Journal of Operational Research 225*, 1 (2013), 1–11.

31. Pimenta, V., Quilliot, A., Toussaint, H., and Vigo, D. Models and algorithms for reliability-oriented dial-a-ride with autonomous electric vehicles. *European Journal of Operational Research 257*, 2 (2017), 601–613.
32. Pinto, H. K., Hyland, M. F., Mahmassani, H. S., and Verbas, I. Ö. Joint design of multimodal transit networks and shared autonomous mobility fleets. *Transportation Research Part C: Emerging Technologies* (2019).
33. Rayle, L., Dai, D., Chan, N., Cervero, R., and Shaheen, S. Just a better taxi? a survey-based comparison of taxis, transit, and ridesourcing services in san francisco. *Transport Policy 45* (2016), 168–178.
34. Redman, L., Friman, M., Gärling, T., and Hartig, T. Quality attributes of public transport that attract car users: A research review. *Transport policy 25* (2013), 119–127.
35. Ritzinger, U., Puchinger, J., and Hartl, R. F. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research 54*, 1 (2016), 215–231.
36. Salazar, M., Rossi, F., Schiffer, M., Onder, C. H., and Pavone, M. On the interaction between autonomous mobility-on-demand and public transportation systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018), IEEE, pp. 2262–2269.
37. Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., and Ratti, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences 111*, 37 (Sept. 2014), 13290–13294.
38. Simonetto, A., Monteil, J., and Gambella, C. Real-time city-scale ridesharing via linear assignment problems. *Transportation Research Part C: Emerging Technologies 101* (2019), 208–232.
39. Spieser, K., Samaranayake, S., Gruel, W., and Frazzoli, E. Shared-vehicle mobility-on-demand systems: a fleet operator's guide to rebalancing empty vehicles. In *Transportation Research Board 95th Annual Meeting* (2016), no. 16-5987, Transportation Research Board.
40. Tirachini, A., Chaniotakis, E., Abouelela, M., and Antoniou, C. The sustainability of shared mobility: Can a platform for shared rides reduce motorized traffic in cities? *Transportation Research Part C: Emerging Technologies 117* (2019), 102707.
41. Tirachini, A., and Gomez-Lobo, A. Does ride-hailing increase or decrease vehicle kilometers traveled (VKT)? a simulation approach for Santiago de Chile. *International Journal of Sustainable Transportation 14*, 3 (2020), 187–204.
42. Tsao, M., Milojevic, D., Ruch, C., Salazar, M., Frazzoli, E., and Pavone, M. Model predictive control of ride-sharing autonomous mobility on demand systems. In *Proc. IEEE Conf. on Robotics and Automation* (2019).
43. Tussyadiah, I. P., Zach, F. J., and Wang, J. Attitudes toward autonomous on demand mobility system: The case of self-driving taxi. In *Information and communication technologies in tourism 2017*. Springer, 2017, pp. 755–766.
44. Vosooghi, R., Puchinger, J., Jankovic, M., and Vouillon, A. Shared autonomous vehicle simulation and service design. *Transportation Research Part C: Emerging Technologies 107* (2019), 15–33.
45. Wallar, A., Van Der Zee, M., Alonso-Mora, J., and Rus, D. Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), IEEE, pp. 4539–4546.
46. Wang, Y., Zheng, B., and Lim, E.-P. Understanding the effects of taxi ride-sharing—a case study of Singapore. *Computers, Environment and Urban Systems 69* (2018), 124–132.
47. Wen, J., Zhao, J., and Jaillet, P. Rebalancing shared mobility-on-demand systems: A reinforcement learning approach. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (2017), IEEE, pp. 220–225.
48. Winter, K., Cats, O., Correia, G., and van Arem, B. Performance analysis and fleet requirements of automated demand-responsive transport systems as an urban public transport service. *International Journal of Transportation Science and Technology 7*, 2 (2018), 151–167.

Appendix

Parameter	**Value**
Max. admissible waiting time	5 [min]
Max. admissible delay	10 [min]
Weight of one in-vehicle minute for users (p_v)	1
Weight of one minute waiting for users (p_w)	2
Weight of rejecting a request	40
Weight of one minute with a vehicle in-motion for operators (c_0)	1.5

Table 6. Parameters used for the assignment process. The relative weights of in-vehicle time, waiting time and operators' costs due to a moving vehicle are adapted from [20]. Max. waiting times and delay are taken from [3]. Rejection weight is ours, adjusted to penalize rejections but to allow some of them if this is really beneficial for other users.

Symbol	**Meaning**
t_r	Time when request r is posed
o_r	Origin of request r
d_r	Destination of request r
$t_w(r)$	Real waiting time faced by request r
$t_w^1(r)$	First-announced waiting time for request r
$t_v(r)$	Real in-vehicle time faced by request r
$t_v^1(r)$	First-announced in-vehicle time for request r
$t_d(r)$	Real detour faced by request r
$t_d^1(r)$	First-announced detour for request r
$D(r)$	Real delay faced by request r
$D_w^1(r)$	First-announced delay for request r
$\Delta t_w(r)$	Increase in waiting time for request r since its first announcement
$\Delta t_v(r)$	Increase in in-vehicle time for request r since its first announcement
$\Delta t_d(r)$	Increase in detour for request r since its first announcement
$\Delta D(r)$	Increase in delay for request r since its first announcement
$Q(x)$	Number of requests departing from zone x
$U_{t_w}(x)$	Unreliability regarding waiting times for requests departing from x
$U_{t_d}(x)$	Unreliability regarding detours for requests departing from x
$U_D(x)$	Unreliability regarding delay for requests departing from x
$U_R(x)$	Unreliability regarding rejections for requests departing from x
$R_i(i = 1, ..., 4)$	Artificial requests inserted into the system
R	Set of requests
T	Set of trips (groups of requests)
δt	Time elapsed between consecutive iterations of the assignment process
p_w	Waiting time unitary cost
p_v	In-vehicle time unitary cost
c_0	Operating costs per vehicle-time unit

Table 7. Glossary of notation used throughout the paper.