TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Elmi, A., Moylan, E., Alonso-Mora, J., & Fielbaum, A. (2026). Improving ridepooling reliability using shareability shadows. *Transportation Research Part C: Emerging Technologies*, *192*, Article 105897. https://doi.org/10.1016/j.trc.2026.105897

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

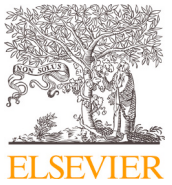
Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Transportation Research Part C 192 (2026) 105897



Contents lists available at ScienceDirect

Transportation Research Part C

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



Improving ridepooling reliability using shareability shadows

Amir Elmi [a], Emily Moylan [a], Javier Alonso-Mora [b], Andres Fielbaum [a,*]

[a] School of Civil Engineering, University of Sydney, Australia
[b] Department of Cognitive Robotics, Delft University of Technology, The Netherlands

ARTICLE INFO

Keywords:
Ridepooling
Reliability
Shareability shadow
Shareability score
On-demand
Ridesharing

ABSTRACT

A major limitation of ridepooling systems is travel-time unreliability: users may receive updated arrival times when vehicle itineraries are modified to accommodate additional passengers. This paper introduces a passenger-specific predictive model to estimate this unreliability. The model uses shareability shadows and expected demand rates to compute a *Shareability Score*, which captures the expected shareable demand along each passenger's trip and indicates their likelihood of experiencing an increase in the initially estimated arrival time. Using this prediction, we apply targeted operational rules for each passenger, and we guarantee no travel-time updates for low-score passengers. As a benchmark, we consider a system-wide policy that manages unreliability uniformly across all passengers. Simulation results using data from Manhattan and Utrecht show that the proposed approach improves both reliability and efficiency compared with the benchmark. Furthermore, at the same rejection rate of 29% between the proposed and benchmark methods, the approach reduces the percentage of passengers experiencing updates to travel time by up to 47.8% and guarantees a fully reliable trip for 83.9% of passengers at the time of assignment. The model performs consistently across different fleet sizes, vehicle capacities, user tolerance levels, and networks, highlighting the value of managing ridepooling unreliability at the individual passenger level.

1. Introduction

1.1. Ridepooling systems and unreliability

On-demand mobility has become a popular mode of transportation due to its ability to combine comfort and reduced prices. However, such systems have been widely employed as a private door-to-door service, which is accompanied by an increase in vehicles-kilometres-travelled and higher congestion in the network. Therefore, the notion of ridepooling - a system in which passengers share the same vehicle simultaneously to make their trips - has been studied in the past few decades to provide a more efficient and environmentally-friendly mode of transportation (Tirachini and Gomez-Lobo, 2020; Liu et al., 2021; Hyland and Mahmassani, 2020).

However, despite the operational, environmental, and user-related benefits of ridepooling, the willingness of users to share their trips remains low, leading to an underutilised system (Li et al., 2019; Alonso-González et al., 2020). In a study by Hansen and Sener (2023), travel time unreliability and trip cost are identified as the most important factors influencing users' decisions to share a ride. However, the lower fare of a pooled trip is often not sufficient to compensate for the longer and unreliable travel time, as well as other barriers discussed by Sajid et al. (2025), making users more inclined to book private trips. While users understand that sharing

* Corresponding author.
E-mail address: andres.fielbaum@sydney.edu.au (A. Fielbaum).

https://doi.org/10.1016/j.trc.2026.105897
Received 16 January 2026; Received in revised form 2 July 2026; Accepted 24 July 2026
Available online 4 August 2026
0968-090X/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

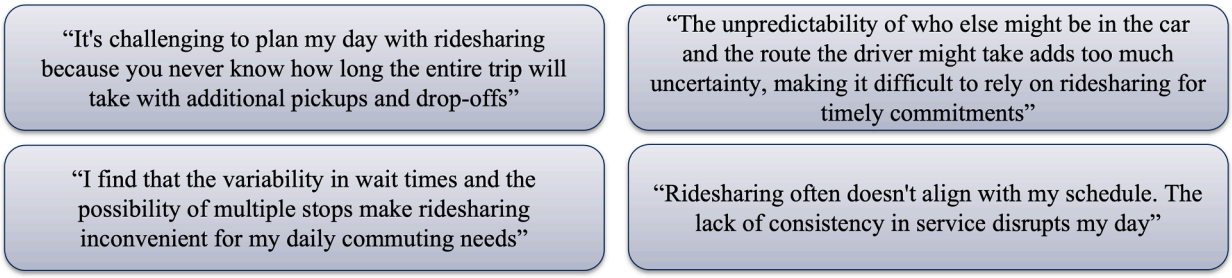


Fig. 1. Passenger perceptions of travel-time unreliability in ridepooling (source: adapted from Sajid et al., 2025). Uncertainty in realised travel time reduces the attractiveness of ridepooling services.

a ride may lead to longer travel times, they remain uncertain about whether their own trip will actually be shared and to what extent. Alonso-González et al. (2020) show that users explicitly trade off travel time, reliability, and cost, and that although the value of reliability is lower than the value of travel time (with a ratio of around 0.5), it remains a significant factor in decision-making. Fig. 1 illustrates users' perception of travel time in a shared mode (Sajid et al., 2025):

Furthermore, from the user's perspective, the unreliability of how much longer the trip will take, rather than just knowing it will take longer, is particularly significant (Zhang et al., 2024; Sajid et al., 2025; Pellegrini and Fielbaum, 2025; Alonso-González et al., 2020). On-demand mobility trips may connect to time-sensitive, post-drop-off activities, e.g., arriving at work on time, catching a train, etc., and inconsistency in the travel time can interrupt those activities.

Travel-time unreliability is not specific to ridepooling and appears in all transport modes in different ways. In general, the travel time of a trip may be updated due to various external factors, such as traffic signals, congestion, or unexpected incidents. As a result, travellers often face uncertainty not only about the final travel time, but also about how that time may evolve during the trip. In this sense, unreliability is a multidimensional concept: it can relate to the likelihood that a travel-time update occurs, the size of the resulting change, and how frequently such updates take place. These aspects of unreliability shape the user experience and can affect the perceived quality of service, especially in on-demand and dynamically operated transport systems (Fielbaum and Alonso-Mora, 2020).

While travel-time unreliability in all transport modes, including ridepooling, can originate from many different factors - such as late arrival of passengers (Kucharski et al., 2021) and different service times for different people (Harmann et al., 2025) (unreliability caused by the *users*), or uncertainty in traffic conditions and stochastic travel times on network arcs (Li et al., 2021; Hörl and Zwick, 2022) (unreliability caused by the *network*) - there is one factor that is specific to ridepooling systems: the detours a vehicle must undertake to accommodate new passengers while serving previously assigned ones (unreliability caused by the *operator*). Even in a perfect network where no network- or user-driven factors affect travel times, users of a ridepooling system are still likely to experience unreliable travel times due to this sharing feature. In contrast to the other examples listed above, this source of unreliability is caused by the operator's decisions, namely how passengers are matched and routes are constructed, and is therefore more directly controllable. This interdependence between different agents is common across modes where the assignment between users and vehicles is performed online (such as ride-hailing or ridepooling), which typically increases their unreliability (Fielbaum, 2025). In this paper, we do not consider unreliability caused by the users or the network, in order to isolate and study this operator-induced source of unreliability.

This ridepooling-specific unreliability can appear in two distinct forms: one-time unreliability and daily unreliability, as introduced by Fielbaum and Alonso-Mora (2020)[1]. One-time unreliability occurs *within a single trip* and refers to updates in travel time after assignment, for example when a passenger is informed that their originally announced travel time has increased due to the arrival of a new passenger. Daily unreliability, in contrast, arises *across repeated trips* of a similar type, where a passenger experiences different realised travel times on different days. In the latter, passengers form expectations based on past experience and care primarily about consistency in the final travel time across trips, rather than updates within a single trip. This paper focuses on reducing one-time unreliability.

Examples of one-time unreliability, i.e., the increase in the travel time, are illustrated in Fig. 2. As depicted in Fig. 2 (top row), passenger A is waiting to be picked up by the vehicle. When passenger B puts their request into the system, the vehicle is scheduled to pick up passenger B before picking up passenger A. This will impose **additional waiting time** on passenger A (3 min in this example). Another example is shown in Fig. 2 (bottom row), where the newly inserted passenger B will impose 5 min of **additional detour time** on passenger A while they are travelling towards their destination. The increase of travel time in both examples can also be expressed as **additional delay time**.

[1] We adopt the term "unreliability" following Fielbaum and Alonso-Mora (2020) to maintain consistency with the existing literature. Alternative terms, such as "ambiguity" (Zhang et al., 2024) or "disruption" (Zhu and Levinson, 2011) used in related studies, have also been proposed, but they may not clearly convey variations in travel time in this context.

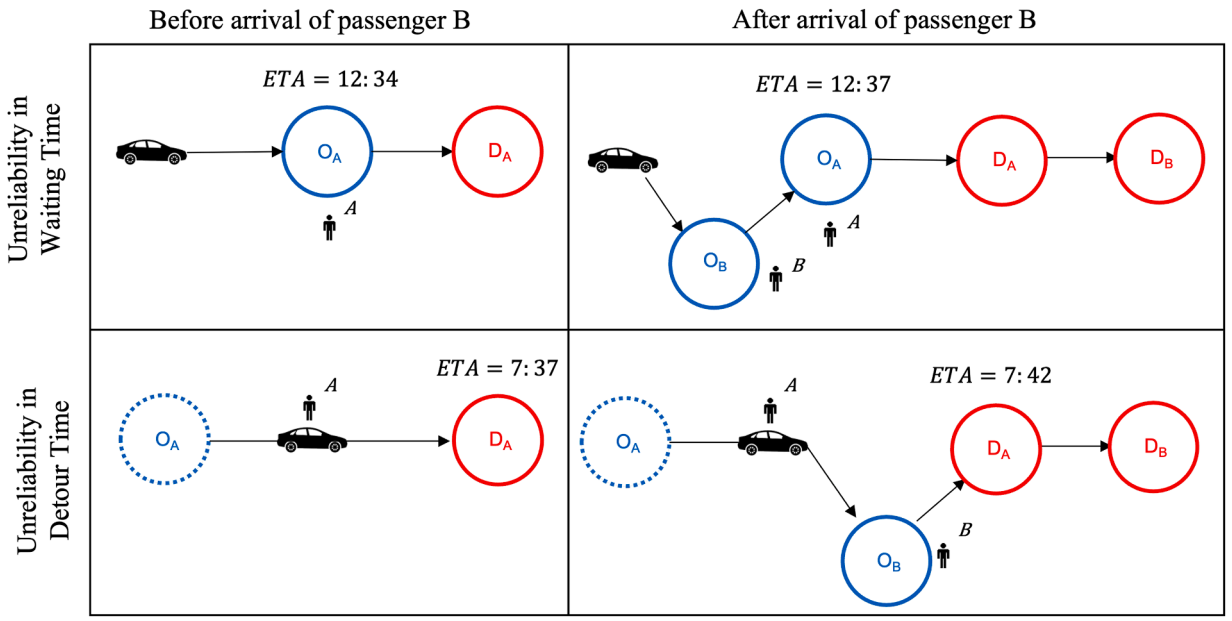


Fig. 2. Unreliability instances in ridepooling systems. Fig. 2 (top row) demonstrates a situation where the user faces a change in their waiting time, and Fig. 2 (bottom row) shows when the user encounters a change in their detour time. These changes make ridepooling trips unpredictable, resulting in lower user satisfaction.

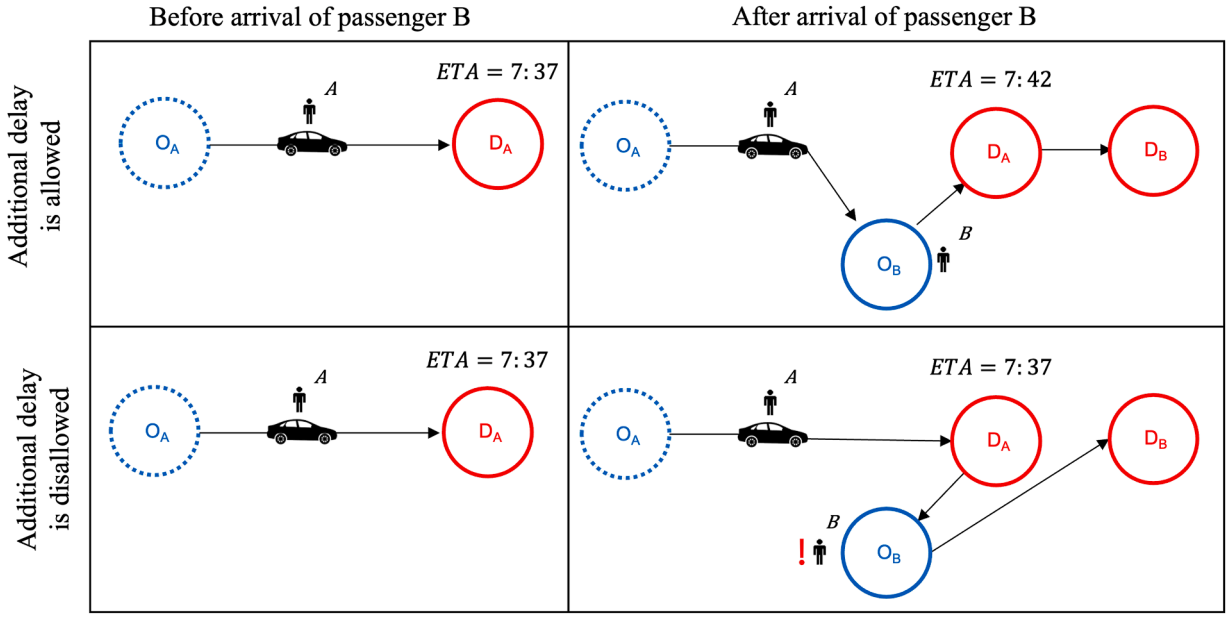


Fig. 3. Disallowing additional delay to tackle unreliability. Although maintaining the estimated arrival time solves the unreliability issue, more users are potentially rejected, and we also do not take advantage of the system's pooling feature (Fielbaum and Alonso-Mora, 2020).

1.2. Related studies

Fielbaum and Alonso-Mora (2020) studied the relevance and extent of the one-time unreliability. They showed that a significant percentage of passengers (36%-53%) experienced additional delay time, while the magnitude of the additional delay time was about the same as the initial delay time. To address this unreliability, they defined a scenario in which no additional delay is allowed, as shown in Fig. 3. However, the lower availability of vehicles led to a significant rise in rejection rate, suggesting a trade-off between the system's reliability and efficiency. Therefore, a more sophisticated approach is needed to tackle the unreliability while the inefficiency of the system is also taken into account.

Some papers have studied objectives that are indirectly related to the unreliability of the system. In a study by Liu et al. (2019b), on-time arrival has been prioritised in the assignment of passengers to vehicles. While this manages unreliability, it would also increase the rejection rate, as the on-time arrival makes the vehicle less available. In another line of work, some papers considered stochastic travel times (Zhou and Roncoli, 2022; Wolf et al., 2023; Bailey, 2022; Li and Chung, 2020; Xiang et al., 2008). While these methods might result in a more accurate prediction of the total travel time for users, they do not explicitly tackle the change (increase) in the travel time of each user.

Walking to/from a pick-up/drop-off point is another interesting topic that is related to the unreliability (Dieter et al., 2023; Fielbaum et al., 2021; Li et al., 2025; Stiglic et al., 2015). While this has the potential to reduce the total travel time, which also means a reduction in the amount of increase in the additional travel time, users have shown reluctance towards walking. Moreover, the uncertainty regarding the pickup and drop-off locations can be another source of unreliability (Pellegrini and Fielbaum, 2025; Yan et al., 2025).

Reserving trip requests in advance is another way to increase both efficiency and reliability of the ridepooling system (Bilali et al., 2019; Theodoridis et al., 2023; Engelhardt et al., 2022; Su et al., 2024). Nevertheless, for the users who do not reserve, the system might remain as unreliable as in the purely on-demand one.

Analytical models have also been used to study ridepooling performance. In particular, Daganzo et al. (2020) and Wang et al. (2021) derive analytical expressions to predict system-wide passenger–to–passenger matching rates while assuming homogeneous maximum waiting and detour times for all passengers. More recently, Fayed et al. (2023) developed a macroscopic analytical framework to evaluate how allowing pooled ride-hailing vehicles to use dedicated bus lanes affects network-wide passenger delays, while Fayed et al. (2024) proposed a dynamic macroscopic pricing framework that adjusts the fare difference between solo and pooled rides to influence passengers' willingness to share and improve multimodal network performance. While such models provide useful insights into system-level matching and operational performance, they are designed for aggregate prediction and control rather than directly addressing unreliability at the level of individual passengers with different trip characteristics.

More futuristic approaches have also been proposed to address travel-time unreliability through system design. For example, Repoux et al. (2021) introduce a semi-autonomous on-demand system based on vehicle convoys, where passengers travel in individual trailers that can detach from a convoy. In this setting, when a trailer detours to pick up or drop off a passenger, the remaining trailers continue unaffected, limiting the propagation of travel-time updates to other users. While such designs can improve reliability by construction, they rely on advanced vehicle automation and coordination technologies and are therefore not yet practical for widespread deployment.

Another approach that can influence unreliability is increasing vehicle availability through the rebalancing of idle vehicles (Fielbaum et al., 2022; Wallar et al., 2018). Under this approach, new requests are more likely to be served by idle vehicles rather than by vehicles already in service, thereby avoiding travel-time updates for existing passengers. However, this outcome is not guaranteed and depends on system parameters; moreover, it might come at the cost of a lower sharing ratio.

1.3. Contribution

Existing studies have *measured* and *characterised* unreliability in ridepooling systems and its trade-off with efficiency, but they mainly focus on analysis rather than intervention. Our work takes a further step by actively *managing* and *improving* reliability through operational rules that are adapted to the characteristics of each trip. In this paper:

- We propose a predictive model based on **Shareability Shadows**, a concept initially introduced by Tachet et al. (2017) and Bilali et al. (2019), which we extend to define a new metric called the **Shareability Score**. Our model evaluates each passenger's trip individually and computes the shareability score as the real-time, state-dependent likelihood of the passenger experiencing an increase in travel time due to future requests.
- We incorporate this likelihood into the assignment and routing process to manage and reduce system unreliability, while maintaining efficiency, through a system parameter called **slack**.

The rest of the paper is organised as follows: Section 2 outlines the methodology used in this paper, including the concept of shareability shadow, shareability score, and how they are used to modify the routing and matching decisions. Section 3 describes the results of simulations developed over Manhattan and Utrecht. Finally, Section 4 concludes the work done in this paper and provides future insights about the further work that can be done in this field.

2. Methodology

In this section, we first introduce the concepts, variables, and definitions we need to build our model[2]. Then, we briefly describe the model from Simonetto et al. (2019) that we use as the basis for the matching and routing decisions. Next, we explain the computation of the shareability score for each passenger. Finally, using these scores, we present two ways to reduce or control the unreliability.

[2] Refer to Table A.5 for the table of all notations used in this paper.

2.1. Preliminaries

2.1.1. Definitions and concepts

Let us define $G(\mathcal{N}, \mathcal{A})$ as the directed graph representing the urban network, which we assume is strongly connected. Each arc has a constant travelling time, and $t(i, j)$ represents the time it takes for the vehicle to move from node i to node j following the shortest path[3]. We define $\mathcal{R}$ as the set of trip requests (or passengers, as we use 'passenger' and 'request' interchangeably in this paper) and $\mathcal{V}$ as the set of vehicles. Each request $r \in \mathcal{R}$ is a 4-tuple, defined by its request time (t_r), origin (O_r), destination (D_r), and number of people involved κ_r. Note that a single request can include more than one person, but they all have to share the same origin and destination, and they need to be assigned to the same vehicle. In other words, they can be treated as a single passenger who occupies extra seats. Furthermore, for each request r, a direct travel time $T_r = t(O_r, D_r)$ is calculated, representing the time it takes to travel from the origin to the destination following the shortest path, as if the passenger were using a personal vehicle. Finally, each vehicle $v \in \mathcal{V}$ has a maximum number of seats, referred to as its capacity c_v.

2.1.2. Travel-time components

When a passenger uses a ridepooling system, there are several definitions indicating different components of the passenger's journey:

- **Pick-up Time (t^r_{PU})**, which is the moment the passenger is picked up at their origin O_r.
- **Drop-off Time (t^r_{DO})**, which is the moment the passenger is dropped off at their destination D_r.
- **Waiting Time (t^r_w)**, which is the time that the passenger has to wait until they are picked up at their origin, i.e., $t^r_w = t^r_{PU} - t_r$.
- **In-vehicle Time (t^r_v)**, which is the total time the passenger spends in the vehicle to reach their destination, i.e., $t^r_v = t^r_{DO} - t^r_{PU}$.
- **Detour Time (t^r_d)**, which is the time that the passenger spends onboard while the vehicle deviates from the direct route to serve other assigned passengers, i.e., $t^r_d = t^r_v - T_r$.
- **Delay Time (t^r_D)**, which is the sum of the waiting time and the detour time and indicates the total additional travel time compared to the direct travel time, i.e., $t^r_D = t^r_w + t^r_d$.

Given these, a maximum tolerable waiting time (Ω_r) and a maximum tolerable delay time (Φ_r) are defined for each request r to ensure user satisfaction.

2.1.3. Unreliability in travel time

As it was discussed in the example of Fig. 2, when a passenger is assigned to a vehicle, their components of travel time might get updated throughout the trip due to the insertion of new requests. Therefore, to study the unreliability of the travel time components, we define the *initial values* of the components at the *moment of assignment* by $t^r_{w^i}$, $t^r_{d^i}$, $t^r_{D^i}$, which can be different from the actual experienced values defined in Section 2.1.2. The other three components can also be defined similarly, but we use these three as the main ones to study unreliability. Unreliability, defined as an increase in the value of any component of travel time, is computed by Eq. (1).

$$\Delta t^r_w = t^r_w - t^r_{w^i}, \qquad \Delta t^r_d = t^r_d - t^r_{d^i}, \qquad \Delta t^r_D = t^r_D - t^r_{D^i} \tag{1}$$

If $\Delta t^r_h > 0$ (for $h = w, d, D$), then passenger r has experienced an increase in their travel time, i.e., unreliability. This is because the passenger accepts the ride with full knowledge of the initial travel time. What remains unknown–and thus unreliable from the passenger's perspective–is subsequent updates to this delay. It is worth mentioning that a passenger can experience several changes throughout a single trip, each of which holds a unique value, but Eq. (1) is the result of all changes, i.e., the final values at the moment of drop off.

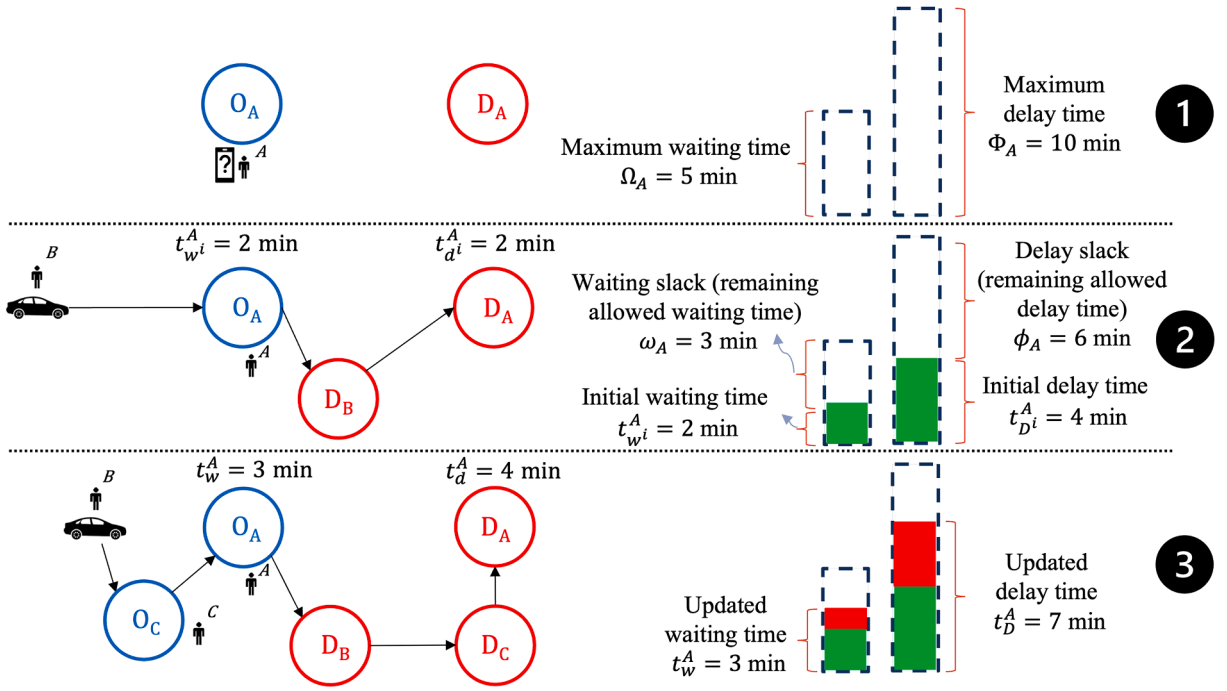
2.1.4. Residual slack after initial assignment

We have defined the type of unreliability this study aims to address, and we now identify another trip characteristic that enables such unreliability to occur. It is quite likely that, at the time of the assignment of a passenger to a vehicle, the initial values of travel-time components do not fully utilise the maximum waiting time and maximum delay time. Therefore, the remaining capacity of maximum waiting/delay time can be used for future matching with new passengers. We refer to this remaining capacity as **residual slack after initial assignment**, defined as the difference between the maximum waiting/delay time and the initial value of waiting/delay time. For simplicity, we refer to this quantity as **slack** in the remainder of the paper. Slack is the source that allows future changes, i.e., unreliability in travel time, and is defined by Eq. (2).

$$\text{Waiting Slack} = \omega_r = \Omega_r - t^r_{w^i}, \qquad \text{Delay Slack} = \phi_r = \Phi_r - t^r_{D^i} \tag{2}$$

The concept of slack is illustrated in Fig. 4. In this example, passenger A requests a ride and, after the initial assignment, is expected to experience 2 min of waiting time and 2 min of detour time due to serving the onboard passenger B. This results in an initial delay of 4 min, leaving out 6 min of delay slack. If another passenger, such as passenger C, is to be matched with passenger A, the feasibility of this match depends on whether the extra time the vehicle needs to pick up and drop off passenger C is less than 6 min, i.e., can be accommodated within this available delay slack. It should be noted that similar reasoning applies to the waiting slack and the maximum waiting time, where we only consider the first detour to pick up the new passenger, as the second detour to drop off the new passenger is after the waiting phase.

[3] If there are two or more shortest paths between i and j, we select one arbitrarily



	Stage 1: Request	Stage 2: Assignment	Stage 3: Unreliability
Maximum waiting time	5	5	5
Maximum delay time	10	10	10
Waiting time	N/A	2	3
Waiting slack	N/A	**3**	2
Delay time	N/A	4	7
Delay slack	N/A	**6**	3

Fig. 4. Residual slack after initial assignment. The waiting and delay slacks are the remaining capacity of the maximum waiting and delay times, immediately after the initial assignment (stage 2). Future matching with a new passenger would consume some of the slack (red portion). Therefore, the slack is the source that allows unreliability to occur (stage 3).

It should also be noted that slack can be generally defined as the remaining capacity at any specific moment of time (not only the moment of assignment). This is also shown in Fig. 4 where we can see new values of slack in stage 3. Nevertheless, since we are interested in preventing the very first update in the travel time (as this is the most annoying one according to the literature review), we only consider, study, and manage the slack immediately after the moment of assignment.

2.2. Assignment and routing algorithm

In order to operate a ridepooling system, we need a method to efficiently assign passengers to vehicles and find suitable routes for vehicles. This study uses an assignment and routing algorithm[4] that is a slight modification of the method by Simonetto et al. (2019), tailored to suit the specific needs of this study. This is a rolling-horizon algorithm, and it is executed iteratively at the end of a pre-defined time interval in which passengers are batched over time to be assigned collectively. At each iteration, three steps are followed:

1. *Finding all feasible matches between requests and vehicles*
   For each new request, we identify which vehicles can potentially serve it given the current system state. Possible vehicle schedules to serve the new request are generated using an insertion heuristic, which inserts the new passenger's pickup and

[4] For a detailed methodology on the algorithm, refer to Appendix A.1.

drop-off points into the existing route while preserving the order of previously assigned passengers. Each generated route is then checked for feasibility with respect to vehicle capacity and the waiting and delay constraints of all affected passengers. For each feasible request-vehicle pair, a service cost is computed based on passenger travel times and additional distance.

2. *Finding the optimal match between requests and vehicles*

   All feasible request-vehicle pairs are then considered jointly in a batch assignment problem. A centralised integer linear program assigns each request to at most one vehicle or rejects it. A request may be rejected either because no feasible match with any vehicle exists, or because the cost of serving it is too high relative to the system objective. The optimisation minimises the total system cost of serving and rejecting requests, thereby determining vehicle assignments while avoiding greedy or local matching rules.

3. *Rebalancing idle vehicles based on rejected requests*

   After assignment, some vehicles may remain idle and some requests may be rejected. Rejected requests are interpreted as indicators of higher-demand areas. Idle vehicles are repositioned toward nearby rejected requests by solving a secondary optimization problem. This rebalancing step prepares the fleet for future demand.

This is a one-to-one match, meaning that at each iteration, maximum one passenger can be assigned to a vehicle. It should be noted that pooling still happens, as the vehicle might have some passengers from previous iterations. To find a feasible match, the route by which the vehicle would serve the request should be planned and checked. This is done through an insertion heuristic, i.e., the order of pick-up and drop-off nodes of previously assigned passengers (if any) is preserved. The feasibility is checked according to the maximum waiting and delay time of all users included in the vehicle (including the passenger being assigned), as well as the vehicle's capacity. Out of all feasible routes, the one yielding the lowest cost is selected.

While the methodology of this paper can be modified for other assignment and routing algorithms, two conditions are needed: 1) At each iteration, maximum one passenger is assigned to a vehicle[5], and 2) The order in which the vehicle visits the pickup and drop-off nodes of already assigned passengers is maintained through future assignments [6].

2.3. Shareability shadow model

Now that we know the base method to assign the passengers and route the vehicles, we need an additional tool to control the unreliability. As described above, unreliability is caused by future demand that will change the schedule of a vehicle. In this section, a predictive model that provides useful indices about possible future changes is introduced. This predictive model is based on the notion of **shareability shadow** introduced by Tachet et al. (2017) and Bilali et al. (2019). The shareability shadow is defined as the region around the current itinerary of the vehicle that confines the possible pick-up and drop-off nodes of future shareable requests. This means that if both the pick-up and drop-off points of a new request fall under the shareability shadow, the new request is feasible to be served by the vehicle, i.e., is shareable.

The shareability shadow is constructed for each accepted passenger at the moment of assignment, in order to assess the probability that a future request could be assigned to the same vehicle and thus cause a change in the accepted passenger's travel time. We refer to the passenger for whom we construct the shareability shadow as the **target passenger**, and the possible shareable future request is called the **potential passenger**.

Although the shareability shadow is constructed at the moment of assignment, it is not limited to the instantaneous state of the system at that time. Once a passenger is assigned, the full planned route of the vehicle is known. Based on this route, we construct instantaneous shareability shadows (Section 2.3.1) at all positions along the trajectory, and their integration yields the time-integrated shareability shadow (Section 2.3.3). As a result, the shareability score captures the likelihood of future matching over the entire trip of the target passenger, rather than only at the moment of assignment.

2.3.1. Instantaneous shareability shadow

Fig. 5 represents a simple schema of a shareability shadow. Since this shadow was constructed at a specific timestamp t, we call it **instantaneous shareability shadow**. The vehicle position at time t is $p_v(t)$, and its current itinerary is the solid purple line. The red nodes represent the main nodes, namely the current position of the vehicle and the pick-up and drop-off locations of passengers. The grey nodes correspond to intermediate nodes that the vehicle moves through when following the shortest path between consecutive main nodes. A **section** is defined as a part of the vehicle's current itinerary that starts at a main node and ends at the next main node. The first section always starts with the current position of the vehicle and ends at the next main node.

The reason behind defining sections is that it will provide us with an efficient way to determine the feasible regions. When a vehicle is instructed to deviate from its path to pick up or drop off a new passenger, it is allowed to ignore intermediate nodes; however, main nodes must still be visited. In other words, a deviation from the current path always starts at a main node and ends at the next main node, while the intermediate nodes might be ignored. It should be noted that this is only true under the assumption

[5] This is precisely the difference between the method by Simonetto et al. (2019) and the well-known algorithm by Alonso-Mora et al. (2017).

[6] While this insertion heuristic does not provide the optimal solution, we adopt it for two reasons. First, allowing reordering introduces an additional source of unreliability, as already assigned passengers may experience increased travel time due to changes in service order rather than new requests. Second, optimising the visiting order can yield moderate improvements but at a computational cost up to 10,000 times higher (Fiedler et al., 2022), making insertion heuristics more practical. While this assumption is used in constructing the shareability shadow (Section 2.3.1), the concept can be extended to settings with reordering.

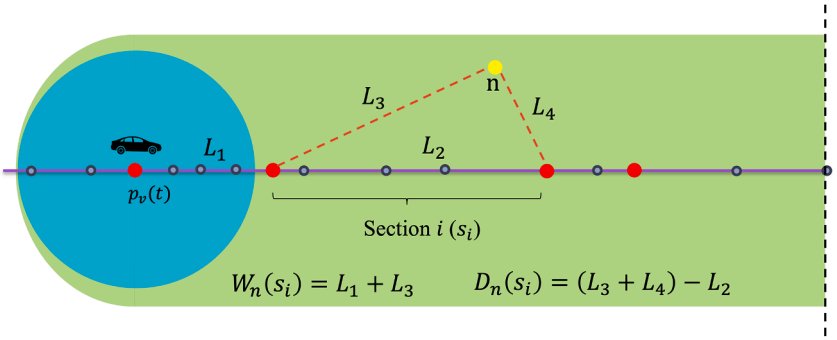


Fig. 5. The instantaneous shareability shadow. The blue region represents the area where a new request can emerge and be picked up, and the green region is where the new request can be dropped off, both without violating any constraints. The instantaneous shareability shadow provides insights about the shareability of the potential passenger at time t.

considered in the insertion heuristic that we used in the assignment algorithm, i.e., the order of the previously scheduled main nodes remains the same. Given that, we can analyse two consecutive main nodes and all intermediate nodes between them collectively, i.e., a section, to identify feasible regions. In this way, we do not need to examine every individual node in the itinerary, whether main or intermediate, to find the feasible regions.

Two subregions are defined in each instantaneous shareability shadow:

1. **Feasible pick-up region**. This is the blue region in Fig. 5. If the potential passenger is to be feasible for a shared ride, then their pick-up node must be feasible as well. The feasibility of the pick-up node is determined by the constraints of all passengers involved in the journey. Let us denote the set of sections of vehicle v at time t by $\mathcal{S}_v(t)$. Then, the blue area represents the region in which there exists a section $s_i \in \mathcal{S}_v(t)$ such that two conditions are satisfied (as detailed in Fig. 5): i) the travel time from the vehicle's position to the beginning of the section s_i and then to the (pick-up) node n, denoted $W_n(s_i)$, does not violate the maximum waiting time of the potential passenger Ω_p; and ii) the extra travel time caused by the deviation to visit the pick-up node n in section s_i, denoted by $D_n(s_i)$, does not violate any of the waiting and delay slacks of the passengers that remain in the vehicle or are waiting for it when the vehicle is in s_i, including the target passenger. That is, the feasible pick-up region encompasses the set of nodes shown by Eq. (3):

$$\mathcal{N}_{PU}(t) = \left\{ n \in \mathcal{N} \mid \exists s_i \in \mathcal{S}_v(t) \text{ such that } W_n(s_i) \leq \Omega_p \ \wedge \ D_n(s_i) \leq \min(C_v(s_i)) \right\} \tag{3}$$

   where $\mathcal{N}_{PU}(t)$ is the set of all feasible pick-up nodes in the instantaneous shareability shadow constructed at time t, and the set $C_v(s_i)$ represents the slack constraints of all passengers (either waiting or onboard) already assigned to vehicle v, when the vehicle is in section s_i of its route. These slack values determine whether a detour required to serve an additional node is feasible, as the extra travel time induced by the detour must not exceed the available slack of any existing passenger[7], as discussed in Fig. 4. This set is defined over each section because once a pick-up or drop-off node is visited, the corresponding waiting or delay slack constraints are relaxed, i.e., it does not exist in the following sections.
2. **Feasible drop-off region**. This is the green region in Fig. 5. The feasibility of the potential passenger also depends on the location of their drop-off node. The feasible drop-off region represents the area inside which there exists a section $s_i \in \mathcal{S}_v(t)$ such that one condition is satisfied: i) the deviation of the vehicle from its current itinerary to visit the (drop-off) node n in section s_i, denoted by $D_n(s_i)$, does not violate the slacks of waiting or travelling passengers when the vehicle is in section s_i, including the target passenger. We note that the potential passenger's maximum delay time is not considered, as the real pick-up node of the potential passenger is unknown and the objective of the instantaneous shareability shadow is to depend solely on the current state of the vehicle and its existing requests, i.e., information that is already available[8]. The feasible drop-off region includes the set of nodes shown by Eq. (4):

$$\mathcal{N}_{DO}(t) = \left\{ n \in \mathcal{N} \mid \exists s_i \in \mathcal{S}_v(t) \text{ such that } D_n(s_i) \leq \min(C_v(s_i)) \right\} \tag{4}$$

   where $\mathcal{N}_{DO}(t)$ is the set of all feasible drop-off nodes calculated in the instantaneous shareability shadow constructed at time t.

[7] This is particularly important because even in high-demand areas with abundant pooling opportunities, if the slack of already assigned passengers is limited, the vehicle lacks sufficient flexibility to accommodate detours. As a result, the probability of a future pooling remains low despite the high demand.

[8] To also account for the potential passenger's maximum delay time, one could consider each node in the blue region together with its corresponding green region, and then aggregate all such blue-point-green-region pairs to obtain a shareability shadow. However, this approach is computationally expensive and time-consuming, and would make the algorithm impractical for real-time use.

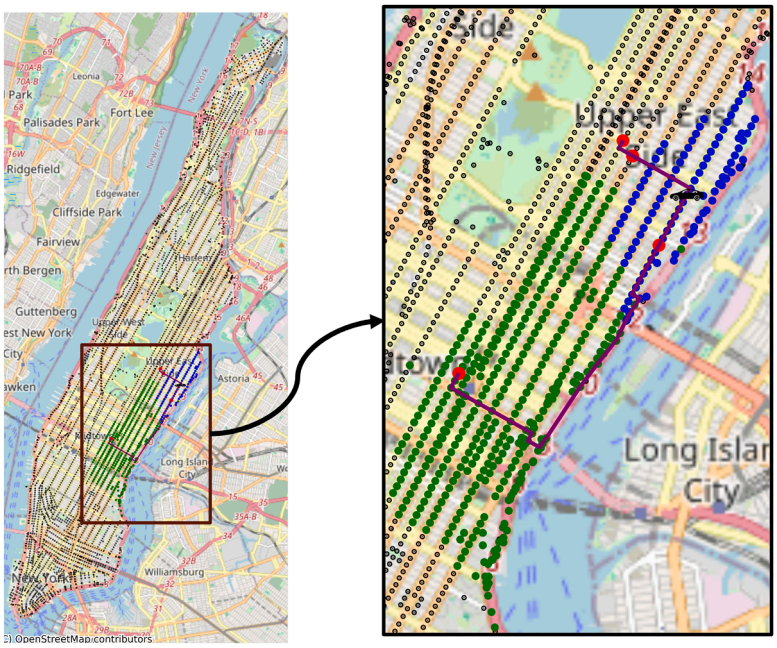


Fig. 6. Real-world representation of the instantaneous shareability shadow constructed over the Manhattan network, considering different arc travel times.

Since the feasible pick-up region has one additional constraint, it is always a subset of the feasible drop-off region[9]. The potential passenger's trip is shareable if their pick-up point is in the blue region and their drop-off point is in the green region. Both conditions must be satisfied together. If a potential passenger does appear in this region and shares their trip with the target passenger, they cause a change and update in the travel time of the target passenger.

Fig. 5 is a simple representation of the shadow for the purpose of introduction. Considering different travel times of different arcs in the real network, Fig. 6 displays the actual representation of an instantaneous shareability shadow over the network of Manhattan, a network that will be explained and used later.

As such, the instantaneous shareability shadow can be used to study how likely it is for a potential passenger to emerge at time t, who could make a change in the target passenger's trip. To do this, we need to combine this shadow with some information about the demand rate at every node, as we now explain.

2.3.2. *Instantaneous shareability score*

In order to extract useful indices from the shareability shadow that inform about the possible future changes in the target passenger's trip, we calculate a new variable called **shareability score**. The shareability score is an index computed for each target passenger that is positively correlated with the chance that a new shareable request (potential passenger) will appear and negatively impact the travel time of the target passenger.

We first define the **instantaneous shareability score** $S_r(t)$, representing the shareability score of request r (target passenger) at time t, derived from the corresponding instantaneous shareability shadow. The instantaneous score will later be integrated to build the actual score. In order to explain the definition of $S_r(t)$ step by step, let us start with a simpler version where we assume that the demand is uniformly distributed. This simplified version is denoted $\widetilde{S_r(t)}$ and is defined by Eq. (5). In this context, λ is the trip demand rate of the entire network. We define $c_v(t)$ as the binary capacity variable of vehicle v assigned to the target passenger at time t; if the vehicle is full, then $c_v(t) = 0$, otherwise $c_v(t) = 1$. This is needed because when the vehicle is full, it cannot accommodate the potential passenger (the blue region does not exist), which means that the potential passenger never imposes a change on the target passenger.

$$\widetilde{S_r(t)} = c_v(t) \times \lambda \times \frac{|\mathcal{N}_{PU}(t)|}{|\mathcal{N}|} \times \frac{\left|\mathcal{N}_{DO}^{\text{tot}}(t)\right|}{|\mathcal{N}|} \tag{5}$$

Therefore, the instantaneous shareability score derived from the instantaneous shareability shadow can be interpreted as the shareable demand rate at time t that appears under the shadow, contingent on the vehicle availability. This is because the last two

[9] This additional constraint also makes the feasible pick-up region significantly depend on the current position of the vehicle, as depicted in Fig. 5 (the potential passenger would not wait for a vehicle that is too far away). Therefore, to favour the computation time, we only consider the first two sections when computing the set of all feasible pick-up nodes.

terms are dimensionless ratios that convert the total demand rate into shareable demand rate. The greater the shareability score, the higher the chance of emergence of the potential passenger that will impose unexpected travel time changes on the target passenger, i.e., unreliability.

As discussed above, Eq. (5) gives an idea of the instantaneous shareability score, but that is only valid if the demand was uniformly distributed. In other words, we want to replace the factor λ from Eq. (5), by a factor $\lambda_{z_O,z_D}(t)$, that depends on an origin z_O, a destination z_D, and the time t. The literature offers multiple ways to estimate these numbers, using historical demand or recently observed one, doing it by node or by clusters, and with different techniques to synthesise the past information into a single value (Fielbaum et al., 2022; Zwick and Axhausen, 2022). In this study, we account for the spatial and temporal components as follows:

1. **Spatial Distribution of Demand.** The origin-destination pairs of trips are not uniformly distributed across the entire network, and different zones carry different potential to generate or attract a trip. As a result, to have a more accurate study, this paper considers many different zones instead of the whole network, each of which has a unique potential to generate or attract trips. The splitting of the network into many different zones is done by the well-known K-means clustering algorithm. Therefore, after the zoning, λ becomes λ_{z_O,z_D}, which is the demand rate that originates in zone z_O and terminates in zone z_D.
2. **Temporal Distribution of Demand.** It is a well-known fact that the distribution of demand is not uniform over time. There are times, like morning and evening peaks, when the trip demand rises significantly, as well as times when the trip demand drops. Therefore, one must take this fluctuation into account to reach a more realistic study. In this paper, a demand predictive model called exponential smoothing is employed to consider the fluctuation (Hyndman et al., 2008). This model predicts the future demand based on the recent observed demands. This is done by the following equation:

$$\hat{\lambda}^{i}_{z_O,z_D} = \alpha\lambda^{i-1}_{z_O,z_D} + \alpha(1-\alpha)\lambda^{i-2}_{z_O,z_D} + \alpha(1-\alpha)^2\lambda^{i-3}_{z_O,z_D} + \cdots + \alpha(1-\alpha)^{n-1}\lambda^{i-n}_{z_O,z_D}. \tag{6}$$

where $\lambda^{i}_{z_O,z_D}$ and $\hat{\lambda}^{i}_{z_O,z_D}$ stand for observed demand and predicted demand in time interval i. Furthermore, $\lambda^{i-1}_{z_O,z_D}, \lambda^{i-2}_{z_O,z_D}, \cdots$ is the observed demand in one time interval before i, in two time intervals before i, etc. For example, if we want to predict the demand between 13:10-13:20, then we record and use the demand occurring in 13:00-13:10 as one time interval before, 12:50-13:00 as two time intervals before, etc. It should be noted that the demands in previous intervals are obtained through the online record of the demand as the system operates. α is the smoothing constant governing how slowly or fast older demand observations are smoothed. The higher the α, the more weight is allocated to the more recent observed demands to predict the future demand.

Applying Eq. (6) to $\hat{\lambda}^{i-1}_{z_O,z_D}$, a simpler version is obtained as shown in the equation below.

$$\hat{\lambda}^{i}_{z_O,z_D} = \alpha\lambda^{i-1}_{z_O,z_D} + (1-\alpha)\hat{\lambda}^{i-1}_{z_O,z_D} \tag{7}$$

Eq. (7) states that the predicted demand is a combination of the observed demand and the predicted demand in the last time interval, and the weight of each of these two terms is determined by the smoothing constant α. Having the predicted demand for each time interval, λ_{z_O,z_D} becomes $\lambda_{z_O,z_D}(t)$, which is the demand rate that originates in z_O and terminates in z_D at time t, considering the corresponding time interval that t belongs to.

It is worth mentioning that while we used these two particular methods for the temporal and spatial distribution of demand, the shareability shadow model is compatible with any other methods for this purpose. Applying these two modifications to Eq. (5), we reach Eq. (8).

$$\widehat{S_r(t)} = \sum_{z_O\in\mathcal{Z}_{PU}(t)}\sum_{z_D\in\mathcal{Z}_{DO}(t)} c_v(t)\cdot\lambda_{z_O,z_D}(t)\cdot\frac{\left|\mathcal{N}_{PU}(z_O,t)\right|}{\left|\mathcal{N}_{z_O}\right|}\cdot\frac{\left|\mathcal{N}^{\text{tot}}_{DO}(z_D,t)\right|}{\left|\mathcal{N}_{z_D}\right|} \tag{8}$$

Here, $\mathcal{Z}_{PU}(t)$ and $\mathcal{Z}_{DO}(t)$ are the sets of all origin and destination zones, each containing at least one feasible pick-up or drop-off node in the shareability shadow constructed at time t. $\lambda_{z_O,z_D}(t)$ is the demand rate of the trips starting in the origin zone z_O and ending in the destination zone z_D at time t. $\mathcal{N}_{z_O}$ and $\mathcal{N}_{z_D}$ are the set of all nodes that belong to the origin zone z_O and destination zone z_D, respectively. Finally, $\mathcal{N}_{PU}(z_O,t)$ and $\mathcal{N}^{\text{tot}}_{DO}(z_D,t)$ are the sets of all feasible pick-up and drop-off points belonging to the origin zone z_O and destination zone z_D at time t.

Note that in Eq. (8) we still denote the variable by $\widehat{S_r(t)}$ instead of $S_r(t)$. The last change has to do with computational efficiency. In fact, the double sum in Eq. (8) is $O(n^2)$, where n is the number of nodes in the network. Typical cities have a few thousand nodes, so this can scale badly, and our experiments show that the computational times can increase by about eight times. In practice, this would compromise a very relevant characteristic of the method, namely its ability to be applied in real-time[10]. To have a more efficient and applicable model, we decouple the origin-destination pair (OD) in Eq. (8) into origin plus destination (O + D) as shown in Eq. (9). In this way, we drop the order of the instantaneous shareability score at time t from $O(n^2)$ to $O(n)$. Eq. (9) is the complete definition of the instantaneous shareability score. In Eq. (9), $\lambda_{z_O}(t_n)$-or-$\lambda_{z_D}(t_n)$ represent, respectively, the demand rate of the trip starting-or-ending at the origin-or-destination zone z_O-or-z_D.

$$S_r(t) = c_v(t)\cdot\left(\sum_{z_O\in\mathcal{Z}_{PU}(t)}\lambda_{z_O}(t)\cdot\frac{\left|\mathcal{N}_{PU}(z_O,t)\right|}{\left|\mathcal{N}_{z_O}\right|} + \sum_{z_D\in\mathcal{Z}_{DO}(t)}\lambda_{z_D}(t)\cdot\frac{\left|\mathcal{N}^{\text{tot}}_{DO}(z_D,t)\right|}{\left|\mathcal{N}_{z_D}\right|}\right) \tag{9}$$

[10] Note that, similar to many methods in this field (e.g., Simonetto et al. 2019, Alonso-Mora et al. 2017), our method can be fully distributed so that each vehicle computes its own feasible routes and the shareability score for its new passenger. The only non-distributed step is the solution to the LP proposed by Simonetto et al. (2019), which can be solved very quickly.

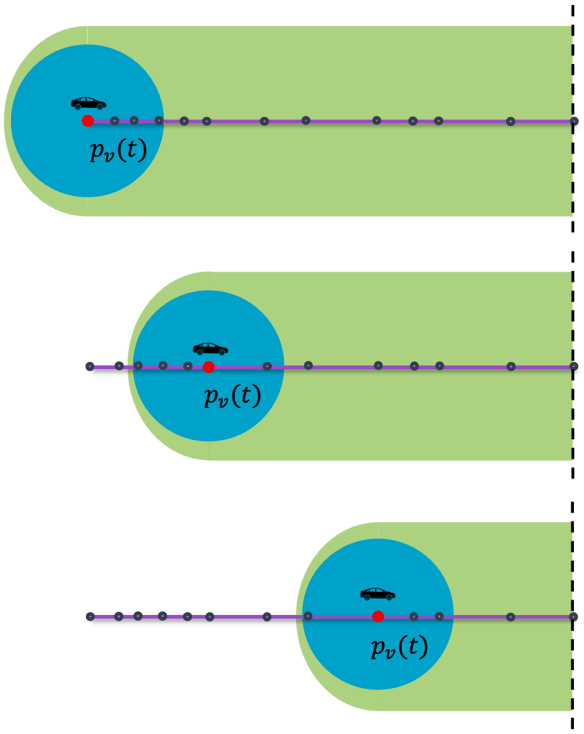


Fig. 7. Time-integrated shareability shadow. Since the possibility of facing a change exists from the moment the target passenger is assigned until the moment they are dropped off, the shareability shadow must also be considered throughout this time range.

2.3.3. *Time-integrated shareability shadow and score*

The shareability shadow given in Section 2.3.1, and illustrated in Fig. 5, is a snapshot in time when the vehicle is at position $p_v(t)$ at time t. It represents the likelihood of a shareable passenger appearing just at t. However, for the target passenger to face a change, the potential request could appear later. Therefore, temporal shareability should also be considered, i.e., time-varying shareability shadows for when the vehicle is at different positions at different times. We are now to construct the **time-integrated shareability shadow**, which is used to predict the shareability of the target passenger along their whole path. To study the trip unreliability of each target passenger, the time-integrated shareability shadow is formed by constructing the shareability shadow from the moment the target passenger is assigned to the moment they are dropped off. Fig. 7 illustrates a general representation of the time-integrated shareability shadow.

The instantaneous shareability score calculated in Eq. (9) is also for a timestamp when the vehicle is at time t. Analogously, the calculation of the shareability score must be carried out from the moment the target passenger is assigned to a vehicle until the moment they are dropped off at their destination. Therefore, using the time-integrated shareability shadow, Eq. (10) gives us the shareability score for the entire duration of the trip.

$$S'_r = \int_{t^r_a}^{t^r_{DO}} S_r(t)\, dt \tag{10}$$

where t^r_a is the time of assignment, t^r_{DO} is the drop-off time, and S'_r is the integrated shareability score from the time of assignment to the drop-off time, all belonging to the target passenger r. Furthermore, the interpretation of the integrated shareability score derived from the time-integrated shareability shadow is the total number of shareable requests that appear under the shadow from the assignment moment of the target passenger until their drop-off moment, contingent on the vehicle availability. As the time-integrated shareability score is the one that represents the full likelihood to face a change, from now on we refer to this quantity simply as the shareability score of the passenger.

However, since there is no closed-form expression of the instantaneous shareability score function, it is not possible to directly calculate the score using integration. Therefore, the time-integrated shareability shadow is approximated by a discrete reconstruction of the instantaneous shareability shadow at various times. Here, we use the visiting times of the nodes, either main or intermediate, along the vehicle's current itinerary, from the vehicle's current node (the time of assignment) to the destination node of the target passenger. The time-integrated shareability shadow is approximated in this way because the shadow is not changing significantly between two consecutive nodes. This is because when the vehicle is moving between two nodes, it can only acquire a new path once it reaches the next node, and it cannot do so while in the middle of the arc between the two nodes. Given that, Eq. (11) is the final

equation to compute the time-integrated shareability score.

$$S_r = \sum_{t_n \in \mathcal{T}(I_v^r)} S_r(t_n) \cdot \Delta t_n \tag{11}$$

Here, $\mathcal{T}(I_v^r)$ is the set of times at which the vehicle is scheduled to visit nodes in the current itinerary I_v^r of vehicle v to serve target passenger r. The nodes in the itinerary start from the current position of the vehicle at the time of assignment, include all main or intermediate nodes between, and end at the drop-off node of the target passenger. Considering t_n as the visit time of the node n, Δt_n is the time it takes for a vehicle to travel from the previous node to that node. This is the time interval during which the appearing potential passengers are predicted in the instantaneous shareability shadow constructed when visiting node n.

Based on the shareability score equation, we calculate two finer-grained scores:

- **Waiting Shareability Score** S_r^w. This score is computed using Eq. (11) when the set $\mathcal{T}(p_v(t_a), D_r)$ is constrained to include only the zones (either origin or destination) that would be visited *before* the origin node of the target passenger. This score positively correlates with the chance of emergence of the potential passenger such that the waiting time of the target passenger is impacted.
- **Detour Shareability Score** S_r^d. This score is computed using Eq. (11) when the set $\mathcal{T}(p_v(t_a), D_r)$ is constrained to include only the zones (either origin or destination) that would be visited *after* the origin node and *before* the destination node of the target passenger. This score positively correlates with the chance of emergence of the potential passenger such that the detour time of the target passenger is impacted.

It is worth mentioning that if some nodes of a zone are visited before the origin/destination node of the target passenger and some are visited after, then the zone is temporarily split into two subzones for the purpose of the calculation.

2.4. Utilisation of the shareability score

We now know how to compute shareability score S_r for every passenger at the moment they are assigned. In what follows, we propose two methods to utilise the shareability score in order to favour the reliability of the system. If we are to reduce unreliability using shareability scores, we first need to define a metric by which we study the unreliability. The primary metric to study unreliability is whether each passenger did or did not face any changes at all, and calculate the percentage of passengers who faced a change throughout their trip. This is because users primarily value trips without any unexpected changes, as this ensures a reliable estimated time of arrival. Therefore, the first change is a key aspect in the study of unreliability (DfT, 2021; Concas and Kolpakov, 2009). Accordingly, throughout this paper, we focus on the first change, as it enables a clear distinction between reliable and unreliable trips.

2.4.1. Slack management

The primary usage of the shareability score is to directly reduce the unreliability of the system, without significantly affecting its efficiency. As discussed in Section 2.1.4, the main source of unreliability is the slack that exists for each passenger after the initial assignment. Therefore, slack management is the key to controlling unreliability. Having passengers with a zero value of slack would ensure that those passengers would not face a change (the primary metric of unreliability in this paper) because it will not allow additional travel time, but this would be at the cost of increased rejection rate (the primary metric of inefficiency in this paper), as explained and shown in Fig. 3 and Fielbaum and Alonso-Mora (2020). This means that there is a trade-off between the unreliability and the inefficiency of ridepooling systems. Therefore, we must attempt to reduce unreliability while also not increasing inefficiency. In other words, reliability and efficiency are two conflicting objectives in ridepooling systems, and we should aim to take both of them into account when we try to improve one of them.

To study the trade-off between unreliability and inefficiency, we construct policy curves using two different slack management policies. Each curve is generated by varying the level of unreliability management within a given policy, resulting in different operational rules and different points along the curve. In the first policy, which acts as our benchmark, the trade-off between unreliability and rejection rate is explored by directly manipulating the slack uniformly across passengers; in the second, we use the shareability score to manage slack in a more targeted way. Specifically:

- **Ordinary Control.** In this model, we do not use the shareability scores. The ordinary control curve in this scenario is created by allocating the same percentage of the slack value to all passengers, and changing the percentage from 0% to 100% to induce different conditions, i.e., different points on the curve. The smaller the slack, the greater the reliability but the lower the efficiency. This will be used as a benchmark to study the benefits of our model. It is worth mentioning that this benchmark is also related to how previous papers studied unreliability. In particular, previous studies have addressed unreliability by applying uniform controls across all passengers. For example, Fielbaum and Alonso-Mora (2020) directly prohibit further changes for all passengers to eliminate unreliability, while analytical approaches such as Daganzo et al. (2020) and Wang et al. (2021) adjust system-wide constraints (e.g., maximum delay or detour) to study their impact on matching rates.
- **Shadow Control.** This policy leverages the shareability scores. In this scenario, passengers with low shareability scores are given 0% of their slack values, and passengers with high shareability scores are allowed to use 100% of their slack values. The different points on the shadow control curve are obtained via the threshold that distinguishes between high and low shareability scores. This is done by sorting the passengers in ascending order based on their shareability scores, and marking the first x% of passengers as low probability, with x varying from 0% to 100%. In this context, the threshold is defined in a percentile-based (relative) manner.

It is worth mentioning that drastic slack management in the shadow policy (allocating either 0% or 100% of the slack) is due to the fact that, from the user's perspective, the occurrence of the first increase in the travel time is more irritating and frustrating than the value of that increase, and also than the occurrence or value of subsequent increases (DfT, 2021; Concas and Kolpakov, 2009). Therefore, by setting the slack to zero, we prevent low-score passengers from facing any changes. We remark that this method provides an additional benefit not captured in the shadow control curve: passengers deemed as low probability ones are given a guarantee of a fully reliable trip at the time of their request, because of their zero slack.

In an alternative approach, instead of computing the time-integrated shareability score only at the moment of assignment for the entire trip, one could update the score by recomputing it at intermediate points during the trip, considering only the remaining trajectory from that point until the drop-off. This would allow for a more refined and continuous slack management, where decisions are updated based on the evolving system state. However, such an approach would not align with the objective of this work, which is to determine the passenger's level of unreliability at the moment of assignment. In particular, updating the score during the trip would imply that the passenger may be exposed to unreliability before the adjustment takes place, and would also make the assigned level of reliability subject to change over time[11]. For these reasons, we compute the score once at the moment of assignment and use it for the entire trip.

Furthermore, the choice of assigning zero slack to low-shareability passengers and full slack to high-shareability passengers is driven by the asymmetric impact on system efficiency. Setting the slack to zero removes future pooling opportunities for a passenger, and the resulting efficiency loss depends on how likely that passenger is to be matched with future requests. For low-shareability passengers, this loss is limited, as their pooling potential is already low. In contrast, high-shareability passengers have a much greater potential for future matches, and restricting their slack would eliminate a significant number of feasible pooling opportunities, leading to a substantial reduction in efficiency. Therefore, the proposed policy targets slack reduction where it has minimal impact on efficiency, while preserving flexibility where it is most valuable.

The proposed approach can be naturally extended to subsequent changes along a passenger's trip. Specifically, while in this work we adjust the slack immediately after the initial assignment to control the occurrence of the first change, a similar mechanism could be applied after the first change to control the second change, and so on. However, we focus on the first change as it allows us to clearly distinguish between fully reliable and unreliable trips. By preventing the first increase in travel time for low-score passengers, we are able to provide a guarantee of a fixed travel time at the time of assignment. Such a guarantee would not be possible if even a single change were allowed, as passengers would still face uncertainty regarding whether their travel time may increase.

2.4.2. Informing passengers

This approach considers a setting in which no operational constraints, such as slack, are modified. Its purpose is to explore an alternative way of addressing unreliability, where passengers are informed about the uncertainty of their travel time rather than controlling it through system decisions. In this setting, all passengers remain eligible for future insertions and may experience updates to their travel time. As a result, system efficiency is fully preserved while reliability is addressed through information provision. We assume that providing this information does not induce behavioural responses, such as passengers reconsidering their trips.

To support this objective, the shareability score is used in a different way compared to the slack management approach. Instead of restricting passengers, the score is used to distinguish between those with higher and lower likelihood of experiencing a change in travel time, and this information is communicated to them. The passengers classification is similar to Section 2.4.1 but requires a different implementation: while slack management relies on a static, exogenous threshold to separate low- and high-score passengers-where the threshold is varied across simulations to generate the policy curves-the informing passengers strategy uses a dynamic, endogenous threshold that is optimised and updated during the simulation based on recent prediction performance. This ensures that the classification remains meaningful for informing passengers, as the goal is not to control system behaviour but to provide accurate and relevant information. In this case, the threshold is defined as an absolute (value-based) cutoff on the shareability score, as opposed to the percentile-based (relative) threshold used in Section 2.4.1.

In particular, after the calculation of the shareability score at the time of assignment, we categorise the passenger into one of the following two groups based on their shareability score: *low probability* or *high probability*. This is done by comparing the shareability score with an endogenous dynamic threshold explained below. If the score is lower than the threshold, the passenger is labelled as low probability; if it is higher than the threshold, the passenger has high probability.

However, since demand is inherently stochastic, a fixed threshold is not sufficient to maintain reliable predictions under varying demand patterns. Therefore, the threshold is optimised and updated over time to adapt to these changes. Once passengers are categorised, the system is executed-without modifying the assignment algorithm (i.e., without adjusting the slack)- to reveal their actual trip experiences. Then, the predicted information given to users is compared against their actual experience to assess the accuracy of the prediction. Based on the predicted probability of facing a change and the actual experience of the trip, each passenger can have four different prediction outcomes:

- **True Positive** is when the passenger is predicted as high probability, and they actually faced a change in their trip.
- **True Negative** is when the passenger is predicted as low probability, and they actually did not face a change in their trip.

[11] For example, if a passenger is initially classified as high-score and later reclassified as low-score, then the portion of the trip before the adjustment remains exposed to unreliability. Conversely, if a passenger is initially classified as low-score and later reclassified as high-score, not only does the remaining portion of the trip become unreliable, but the initial segment also loses its perceived reliability, as the passenger may anticipate further adjustments to their slack.

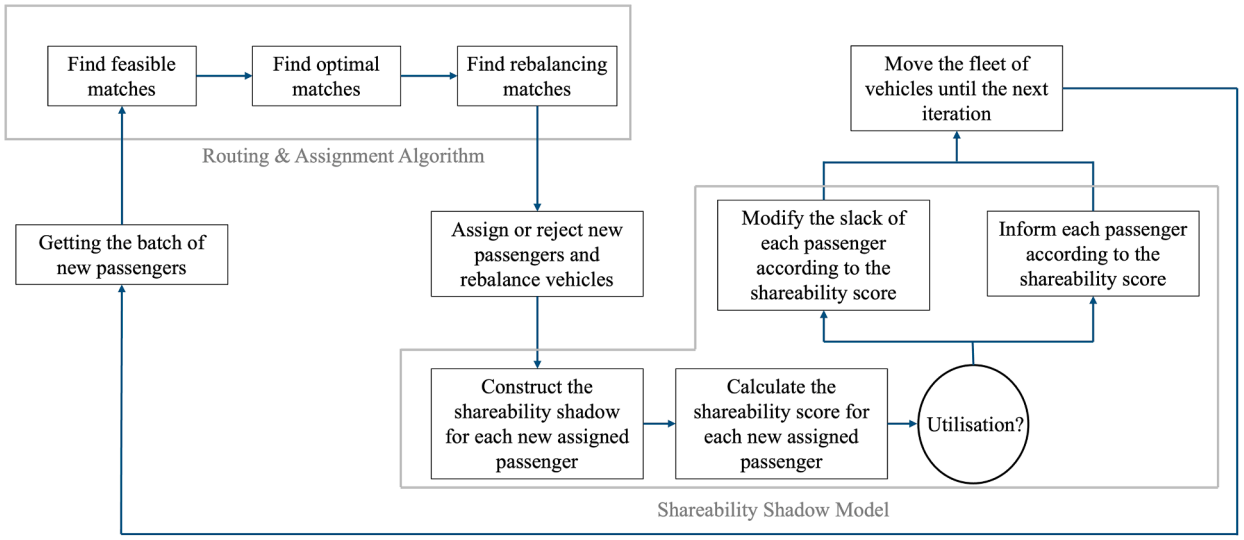


Fig. 8. Flowchart of the paper's methodology. Independent and modular behaviour of the shareability shadow model makes it compatible with any other routing and assignment algorithm, given the proper adjustments.

- **False Positive** is when the passenger is predicted as high probability, but they actually did not face a change in their trip.
- **False Negative** is when the passenger is predicted as low probability, but they actually faced a change in their trip.

To ensure a suitable value is maintained for the threshold over time, the threshold value T is optimised and updated every t minutes (10, in this study) based on the passengers served in the last t minutes. In particular, the objective function maximises the prediction accuracy over this past data, and the resulting threshold is then used for future classifications.

$$\max \quad \sum_{r \in \mathcal{R}^t} \left(p_h^r \cdot u^r + p_l^r (1 - u^r) - p_h^r (1 - u^r) - p_l^r \cdot u^r \right) \tag{12}$$

$$\text{s.t.} \quad p_l^r + p_h^r = 1, \qquad \forall r \in \mathcal{R}^t \tag{13}$$

$$s^r - T \leq M(1 - p_l^r), \qquad \forall r \in \mathcal{R}^t \tag{14}$$

$$s^r - T \geq -M \cdot p_l^r, \qquad \forall r \in \mathcal{R}^t \tag{15}$$

$$p_l^r,\ p_h^r \in \{0, 1\}, \qquad \forall r \in \mathcal{R}^t \tag{16}$$

$$T \in \mathbb{R}. \tag{17}$$

Here, $\mathcal{R}^t$ denotes the set of passengers served in the last t minutes. For each passenger $r \in \mathcal{R}^t$, s^r is the shareability score and u^r is a binary parameter indicating whether that passenger experienced an update in travel time ($u^r = 1$) or not ($u^r = 0$). The decision variable T represents the threshold used to separate low- and high-probability passengers. The binary variables p_l^r and p_h^r indicate whether passenger r is classified as low probability or high probability, respectively. Constraint (13) ensures that each passenger is assigned to exactly one category. Constraints (14)–(15) link the classification variables to the threshold through a standard big-M formulation, so that passengers with scores below the threshold are classified as low probability and the others as high probability. The objective in (12) maximises the number of correct classifications by rewarding correct predictions and penalising incorrect ones. In particular, the objective function consists of four components corresponding to True Positives, True Negatives, False Positives, and False Negatives, respectively, as described above. This optimisation is performed separately for the threshold of waiting shareability score and the threshold of detour shareability score, using their corresponding scores and update indicators.

2.5. Summary of the methodology

Fig. 8 recaps the methodology defined in this section. Fig. 8 shows that the shareability shadow model is a modular attachment to the routing and assignment algorithm. This means that with appropriate adjustments, the shareability shadow is compatible with any other routing and assignment algorithm. It should be noted that once the utilisation of the score is decided, it is fixed in all iterations of the simulation.

3. Results

3.1. Setup

We analyse a ridepooling system using the dataset of the 2013[12] NYC Taxi & Limousine Commission in Manhattan, which serves as our primary network. The Manhattan network consists of 4091 nodes and 9452 arcs. The dataset includes ten days of yellow-taxi requests, starting from 14/01/2013 from 1 pm to 3 pm, totalling 226,676 trip requests. To assess the robustness of our findings, we also consider a secondary network, used in Section 3.5 only, based on the city of Utrecht, the Netherlands. The Utrecht network is constructed from OpenStreetMap and consists of 9616 nodes and 22,735 arcs. The demand data is based on real public transport usage provided by QBuzz (Durán-Micco et al., 2022) and is distributed from stops to nearby nodes following the procedure in Fielbaum and Alonso-Mora (2024). We focus on one morning peak period (6–9 AM), resulting in a dataset of 26,897 trip requests.

Each passenger has a 5-minute maximum waiting time and a 10-minute maximum delay time. To serve the requests, a fleet of 2000 vehicles, each offering 4 available seats, is employed. At each pick-up or drop-off node, the vehicle is assumed to spend a time δ due to braking, re-accelerating, and waiting for the passenger to board (we use $\delta = 24$ seconds following Harmann et al. (2025) for the Manhattan network, and $\delta = 0$ for the Utrecht network[13]). This setting will be referred to as the Base scenario[14]. Finally, to reach a steady state of the system, a warm-up phase of duration t_{warm} is considered at the beginning of the simulation (we use $t_{\text{warm}} = 20$ min).

Out of the total requests, some are not considered in the simulation after applying two eliminating criteria: a) trip requests with direct travel time less than 3 min are omitted under the assumption that these requests are not willing to use a ridepooling system to reach their nearby destination (Liu et al., 2019a). b) any request involving more than 4 people is removed, following the vehicle capacity. When a simulation is finished, some requests that were not served in a steady state are not considered in the results. These requests are either: a) requests served in the warm-up phase, or b) requests that are still being served (waiting or travelling) when the duration of the simulations is finished. Excluding the requests that were not considered in either the simulation or the results, about 62% of the total requests remained.

3.2. Assessing the relationship between the shareability score and unreliability

As discussed before, the shareability score is an index defined to study the probability of the emergence of a potential passenger that might affect the travel time of the target passenger. The higher the score, the greater the chance that the target passenger faces a change in their travel time caused by the vehicle serving the potential passenger. The validity of this statement is shown by Fig. 9. In the plots, the shareability scores are grouped into bins of width 10, and the metric of the vertical axes is calculated for each bin. As shown in Fig. 9a, as the score increases, the average number of changes in the delay time (and the probability of facing at least one change) rises. There is significant variation towards the right extreme of the plot, but that is because of the very few passengers — either a single passenger or fewer than 10 passengers, over the ten days of the dataset — having that value of score. Fig. 9b also shows the distribution of passengers across different values of the score. Furthermore, the plot in Fig. 9a displays a decreasing slope as the score increases. This is because the shareability shadow, as well as the shareability score, studies the shareability regarding the first potential passenger. However, when the potential passenger is assigned to the target passenger's vehicle, the shareability shadow and score of a subsequent potential passenger become smaller as the new assignment consumes part of the previous users' slack. In other words, the shareability shadow and score calculated at the time of assignment study the probability of the first change, and subsequent changes could have their unique updated shadows and scores. Nevertheless, as we are mostly interested in the very fact of facing a change, Fig. 9a confirms that the shareability score is a powerful tool to distinguish the unreliable trips from the reliable ones.

3.3. Results related to slack management

The results of the proposed methodology to reduce unreliability in ridepooling systems are shown in Fig. 10 for the Base scenario. Recall that the metric to study unreliability is the percentage of passengers who faced additional delay in their travel time, and the rejection rate is chosen to be the representative of the inefficiency of the system. The plot is generated for the delay time, as it is the indicator of both waiting and detour times[15] ($t_D = t_w + t_d$). Each point in vivid colour is the average of 10 simulations depicted in pale colour, which are done with the same settings and conditions but over 10 different days (10 different sets of passengers). Therefore, each plot is composed of 36 vivid points that are derived from 360 different simulations shown in pale colours. Circled points are selected points for a deeper study that will be elaborated later in Table 2.

Different points are obtained by applying a different slack management policy. The red curve belongs to the ordinary control in which the slacks of all passengers are adjusted in the same way. The green curve is for the shadow control, in which the slacks of

[12] We used the 2013 dataset, as the currently available datasets of NYC TLC are all aggregated over zones; however, our dataset is disaggregated and the actual location of pick-up and drop-off nodes is critical to assess unreliability.

[13] Since the service time δ also affects the updated travel time, we consider both its presence and absence to have a more comprehensive analysis.

[14] You can see the parameter table in Table A.4.

[15] You can see the plots specific to waiting and detour times in Fig. B.15

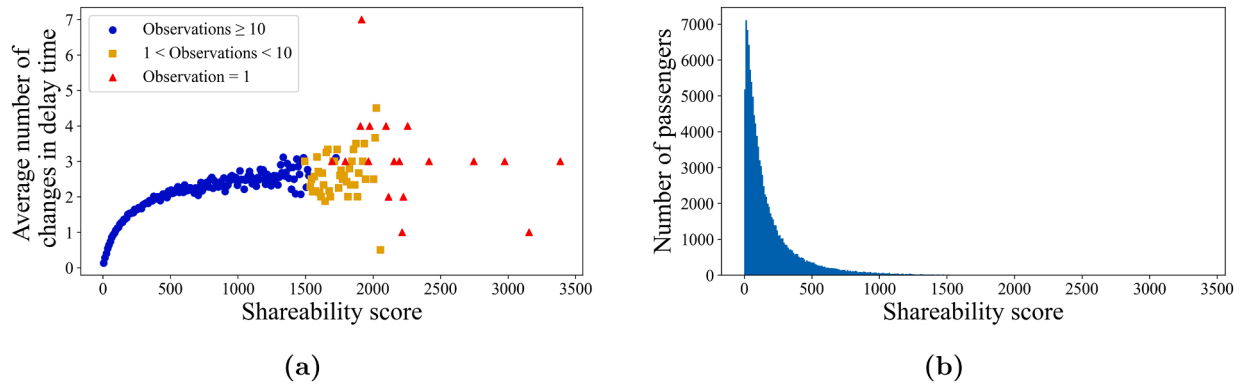


Fig. 9. Shareability scores against the number of changes observed in the delay time (a), and the distribution of the shareability score (b). The shareability score has a positive correlation with the chance of facing a change in the travel time.

passengers are manipulated in a smart way according to their shareability score.[16] The endpoints of the two policy curves always coincide and overlap exactly, shown in blue or orange. This is because at these points, both policies control the slack in the same way. Specifically, at the blue point, all passengers are allocated 100% of the slack in the ordinary control, and in the shadow control, all passengers are marked as having a high shareability score, resulting in being allocated 100% of their slack. Therefore, this point represents a **Normal Ridepooling** system in which no measure is used to address the unreliability. By similar reasoning, at the orange point, all passengers in both policies have 0% of the slack, demonstrating an identical result. This point indicates a **Private Ridehailing** system, as passengers cannot tolerate any additional delay caused by sharing the requests. This is a private ridehailing because at each iteration of our simulation, a maximum of one passenger is assigned to a vehicle, and once a passenger is assigned, adding another passenger in subsequent iterations to the vehicle before the drop-off point of the assigned passenger would violate the zero slack constraint of that passenger.

As shown in Fig. 10, the expected trade-off exists between unreliability and inefficiency in the policy curves, i.e., a reduction in one induces a rise in the other (except for a few points in the lower-right region of the ordinary control curve, arising from stochastic variability in the simulation results). Moreover, unreliability is positively correlated with the system's sharing ratio. This means that the more we transform the private ridehailing system into the normal ridepooling one, the more unreliable the system becomes.

Generally, since we try to minimise both objectives, the closer we get to the origin, the more favourable conditions the ridepooling system is running in. However, in the ridepooling system with the ordinary control policy, this movement is limited by the red policy curve. Nevertheless, if we change the policy of the system, we might be able to move the policy curve closer to the origin. This is the desired outcome that the shadow control policy was significantly successful at achieving, as shown in Fig. 10, by introducing a well-separated, closer-to-origin curve that is always below[17] the one that belongs to the ordinary control policy. In simple words, **the Shadow Control can simultaneously increase the efficiency and the reliability of the system**, meaning that, except for the two endpoints where the curves overlap, for every point on the ordinary control curve, there exists at least one point on the shadow control curve with both higher efficiency and higher reliability. The area between the two curves is the improved operational region unlocked by the shareability shadow model. This accomplishment is due to the smart slack management in the shadow control. In fact, in the ordinary control, the slacks of all passengers were adjusted in the same way regardless of their probability of facing a change. In the shadow control, on the other hand, the slack of each passenger is modified according to the probability of facing a change derived from the shareability score.

The primary result of this paper is presented in Fig. 10, as the shareability shadow model is designed to predict the probability of facing a change for each user, rather than the magnitude of that change. Nevertheless, when reducing the probability aspect of unreliability, it is important to verify that this does not come at the cost of increased delay magnitude. This is examined in Fig. 11, where unreliability is measured by the average magnitude of additional delay over all passengers.

As observed in Fig. 11, there is a close overlap between ordinary control and shadow control, indicating that the magnitude of additional delay remains largely unchanged. Taken together, Figs. 10 and 11 show that the shadow control policy significantly reduces the probability of facing a change (as evidenced by the vertical gap in Fig. 10) without causing a meaningful increase in the magnitude of that change. In fact, if one aims to keep the rejection rate from increasing significantly-i.e., to select a point toward the upper-left region of the policy curve-then the shadow control also helps reduce the average magnitude of the additional delay.

The policy curves in Fig. 10 also allow for different types of comparisons depending on the objective of the analysis. In this paper, rejection rate is used as the primary measure of inefficiency, while the percentage of passengers experiencing a change is used as

[16] Refer to Section 2.4.1 for the detailed slack management policies

[17] The comparison holds not only for the average across days, but also consistently on a same-day basis: for each individual day of demand, the policy curve under shadow control lies below that of the ordinary control. This behaviour is also observed across the different scenarios defined in Section 3.4, demonstrating the generality of the observed improvement.

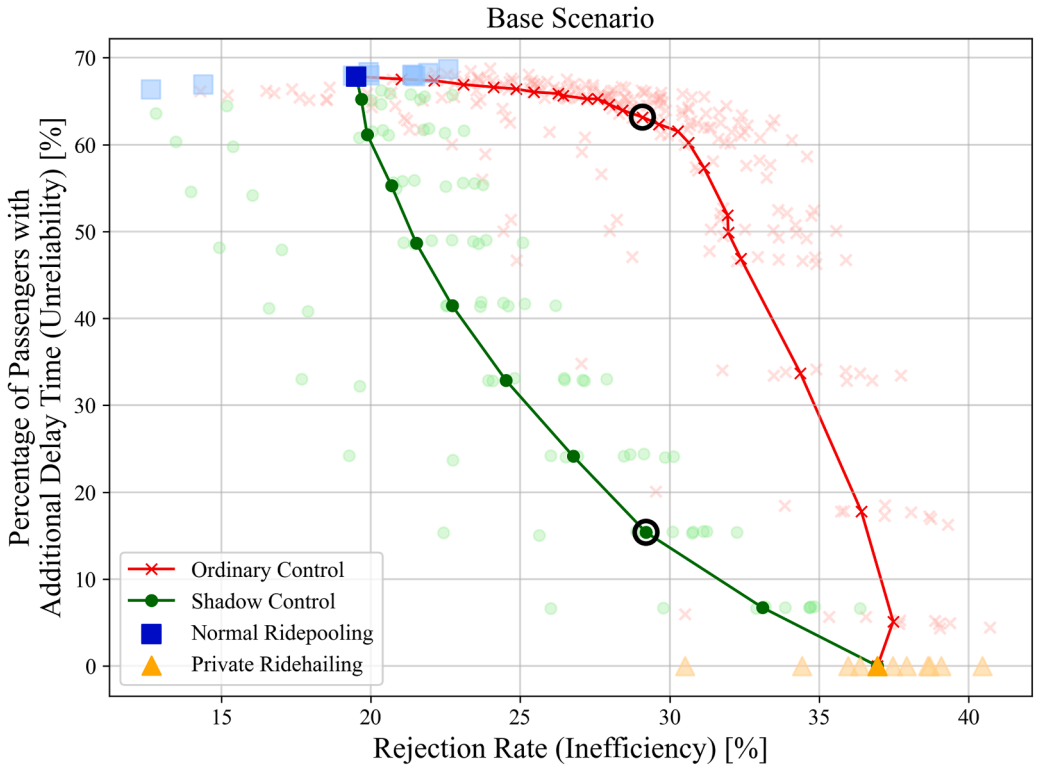


Fig. 10. Policy curves constructed for ordinary and shadow control policies. Each pale point is the result of a simulation of a day, and the vivid point is the average of the simulations with the same setup but over 10 different days. The shadow curve dominates the ordinary curve under all conditions, allowing for a less unreliable ridepooling system. The two circled points are selected for an in-depth study of these two policies.

Table 1
Definition of indices used to evaluate unreliability.

Index	Definition
Rejection Rate	Percentage of passengers whose trip was rejected by the system
Avg. VMT	Average vehicle minutes travelled
Avg. (h)	Average of travel time component h over all passengers
$P(\Delta h)$	Percentage of passengers who faced a change in h
$P^{\text{allowed}}(\Delta h)$	Percentage of passengers who are allowed to face a change in h
$E(\Delta h)$	Magnitude of the change in h, averaged over $P(\Delta h)$
$E^{\text{allowed}}(\Delta h)$	Magnitude of the allowable change in h, averaged over $P^{\text{allowed}}(\Delta h)$
Avg. $N(\Delta h)$	Number of changes in h throughout the trip, averaged over $P(\Delta h)$
Max $N(\Delta h)$	Maximum number of changes in h observed in a single request

the primary measure of unreliability. The selected circled points are chosen for a specific comparison: they are horizontally aligned, meaning that they have similar inefficiency. This choice is intentional, as it isolates the reliability improvement achieved by shadow control at approximately the same primary inefficiency level. Other comparisons are also possible. For example, vertically aligned points could be used to compare efficiency improvements at a similar level of unreliability, while diagonally aligned points could be selected to show moderate improvements in both objectives. The points corresponding to normal ridepooling and private ridehailing, as well as the circled-in-black points on each curve that have similar rejection rates (between 29% and 30%), are chosen for a more in-depth study, as shown in Table 2. The definition of all indices used here is provided in Table 1. In these tables, variable h is used to determine the travel time component, where $h \in \{t_w, t_d, t_D\}$. The point inside the black circle on the ordinary control curve is for the conditions in which all passengers are allocated 35% of their slack, whereas its counterpart point on the shadow control curve corresponds to the conditions that mark 20% of passengers with the highest shareability scores as high probability, which will give them full slack.

Considering the results related to normal ridepooling, we can see that unreliability is quite a prevalent phenomenon in ridepooling systems, with 67.84% of passengers facing at least one change in their delay time. However, waiting time is much less prone to undergo a change (3.38%), and the characteristics of unreliability in delay time are mostly inherited from unreliability in detour time (as also found by Fielbaum and Alonso-Mora, 2020). To completely eliminate unreliability, we can set the slack to zero for all passengers, which would get us to private ride-hailing, but this will be at the cost of roughly doubling the rejection rate (19.51% to 36.95%);

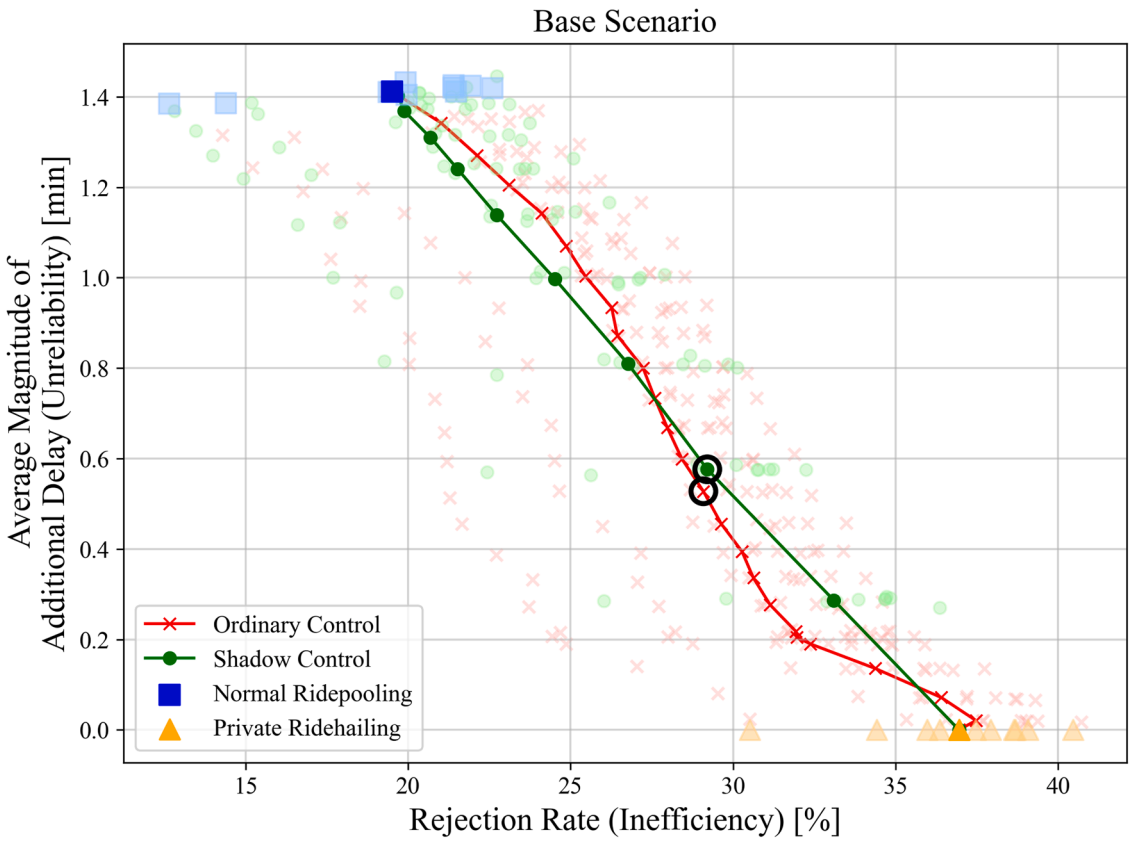


Fig. 11. Policy curves constructed for ordinary and shadow control policies, with the unreliability metric being the average magnitude of additional delay. There is a close tie between the two policies. The two circled points are selected for an in-depth study of these two policies.

removing the sharing feature from the system would increase the average minutes travelled per vehicle (60.24 to 76.41). Aiming for a balance between the unreliability and the inefficiency of the system, one must select points in the interior of the policy curve, and that is why the points related to ordinary control and shadow control are selected to compare two middle points with different slack management policies.

As mentioned earlier, the goal of this paper is to reduce unreliability without significantly increasing the inefficiency (rejection rate) of the system. Therefore, the points related to the two policies are selected in a way to represent two systems with similar inefficiency (approximately the same horizontal distance on the rejection-rate axis), and then assess the unreliability. Before analysing the unreliability indices in these two policies, it is worth comparing the total value of each travel time component. While the average total waiting time increased by 13.68%, the average total detour time and the average total delay time decreased by 33.17% and 5.88%, respectively.

Comparing the percentage of passengers who faced a change in delay time, we can see that 63.17% of passengers in ordinary control faced such a change, while this number is only 15.39% in shadow control, which indicates an absolute reduction of 47.78% when using the shadow control policy. More importantly, almost all passengers (99.95%) in ordinary control are allowed to face some change, while shadow control only permits 16.15% of passengers to experience unreliability. This is because in ordinary control, although the slack is adjusted, almost all passengers still have some slack that allows for future changes (the exception are the 0.05% whose maximum waiting or delay time is completely consumed at the time of assignment, leaving no corresponding slack behind). In shadow control, on the other hand, only 16.15% of passengers are left with (full) slack. Therefore, the shadow control guarantees 83.85% of passengers to face no change at all, as they have zero slack, while such a guarantee cannot be provided to anybody in the ordinary control. As a result, **using shareability shadow, not only are fewer passengers facing changes, but also the vast majority are having a fully reliable trip, receiving the exact travelling time at the moment of being assigned.**

Nonetheless, this achievement comes at an increase in additional delay for the passengers who actually faced a change, as shown by $E(\Delta t_D)$ index. It should be noted that this metric is different from the one used in the policy curve, because this one averages only over the passengers that actually experienced a change. Hence, in the ordinary control, 63.17% of passengers experience an average of additional delay equal to 0.84 min, while this number is 3.74 min for 15.39% of passengers in the shadow control. This is because the passengers with full slack compensate the passengers with zero slack.

Table 2
Indices to study unreliability across the selected points in the Base scenario.

Index	h	Normal Ridepooling	Private Ridehailing	Ordinary Control	Shadow Control
Inefficiency Indices					
Rejection Rate [%]	–	**19.51**	**36.95**	**29.09**	**29.21**
Avg. VMT [min]	–	60.24	76.41	**67.02**	**70.26**
Avg. (h) [min]	t_w	2.64	3.46	2.85	3.24
	t_d	3.13	0.60[a]	2.08	1.39
	t_D	5.77	4.07	**4.93**	**4.64**
Unreliability Indices					
$P(\Delta h)$ [%]	t_w	**3.38**	0	1.45	0.19
	t_d	**66.88**	0	62.30	15.37
	t_D	**67.84**	0	**63.17**	**15.39**
$P^{\text{allowed}}(\Delta h)$ [%]	t_w	99.66[b]	0	99.54[b]	18.50
	t_d	99.92[b]	0	99.95[b]	16.15
	t_D	99.92[b]	0	**99.95**[b]	**16.15**
$E(\Delta h)$ [min]	t_w	0.51	N/A	0.41	0.46
	t_d	2.09	N/A	0.84	3.74
	t_D	2.08	N/A	**0.84**	**3.74**
$E^{\text{allowed}}(\Delta h)$ [min]	t_w	2.39	N/A	0.75	0.80
	t_d	5.65	N/A	1.13	6.16
	t_D	5.65	N/A	1.13	6.16
Avg. $N(\Delta h)$	t_w	1.02	N/A	1.00	1.00
	t_d	1.67	N/A	1.05	1.80
	t_D	1.67	N/A	1.05	1.80
Max $N(\Delta h)$	t_w	3	N/A	1	2
	t_d	7	N/A	2	5
	t_D	7	N/A	2	5

[a] This value is 0 in pure private ridehailing. For the reason behind the current value, refer to Appendix B.2

[b] These values could be expected to be 100%. The reason behind this difference is that the initial travel times of a few passengers consume their maximum waiting/delay time entirely, leaving no slack for future changes.

Table 3
Scenario comparison across operational parameters.

Scenario	Fleet Size	Vehicle Capacity	Maximum Waiting Time	Maximum Delay Time
Base	2000	4	5	10
Large Fleet	3000	4	5	10
High Capacity	1000	8	5	10
More Tolerance	2000	4	7.5	15

3.4. Robustness across different scenarios

To study the robustness of the shareability shadow model under different circumstances, we have defined three more scenarios as shown in Table 3. These scenarios vary the parameters defining the fleet size, vehicles' capacity, and the maximum waiting and delay times. The Large Fleet scenario is aimed to study a system in which pooling happens less, while High Capacity and More Tolerance scenarios are designed to analyse systems with higher sharing ratios, all compared to the Base scenario.

The results of the policy curves associated with the three new scenarios are demonstrated in Fig. 12. The first and most notable conclusion is that all the main trends discussed for the Base scenario remain valid for each of the other scenarios. On top of that, these scenarios provide additional insights on the relationships between sharing, efficiency, and reliability. As discussed in Fig. 10, unreliability is positively correlated with the sharing ratio. Therefore, the level of unreliability for each scenario changes according to how we defined it, i.e., the Large Fleet scenario is less unreliable due to less sharing, while the High Capacity and More Tolerance scenarios are more unreliable because of the higher sharing ratios. This can be verified by considering the values of the peak points in the vertical axes and comparing those values with the corresponding ones in the Base scenario. Specifically, while 67.84% of passengers face a change in their travel time in the Base scenario, this number decreases to 61.81% in the Large Fleet scenario, and increases to 73.97% and 69.79% in the High Capacity and More Tolerance scenarios, respectively. As shown in the left plots of Fig. 12, regardless of the level of unreliability, the shareability shadow model successfully shifts the policy curve closer to the origin,

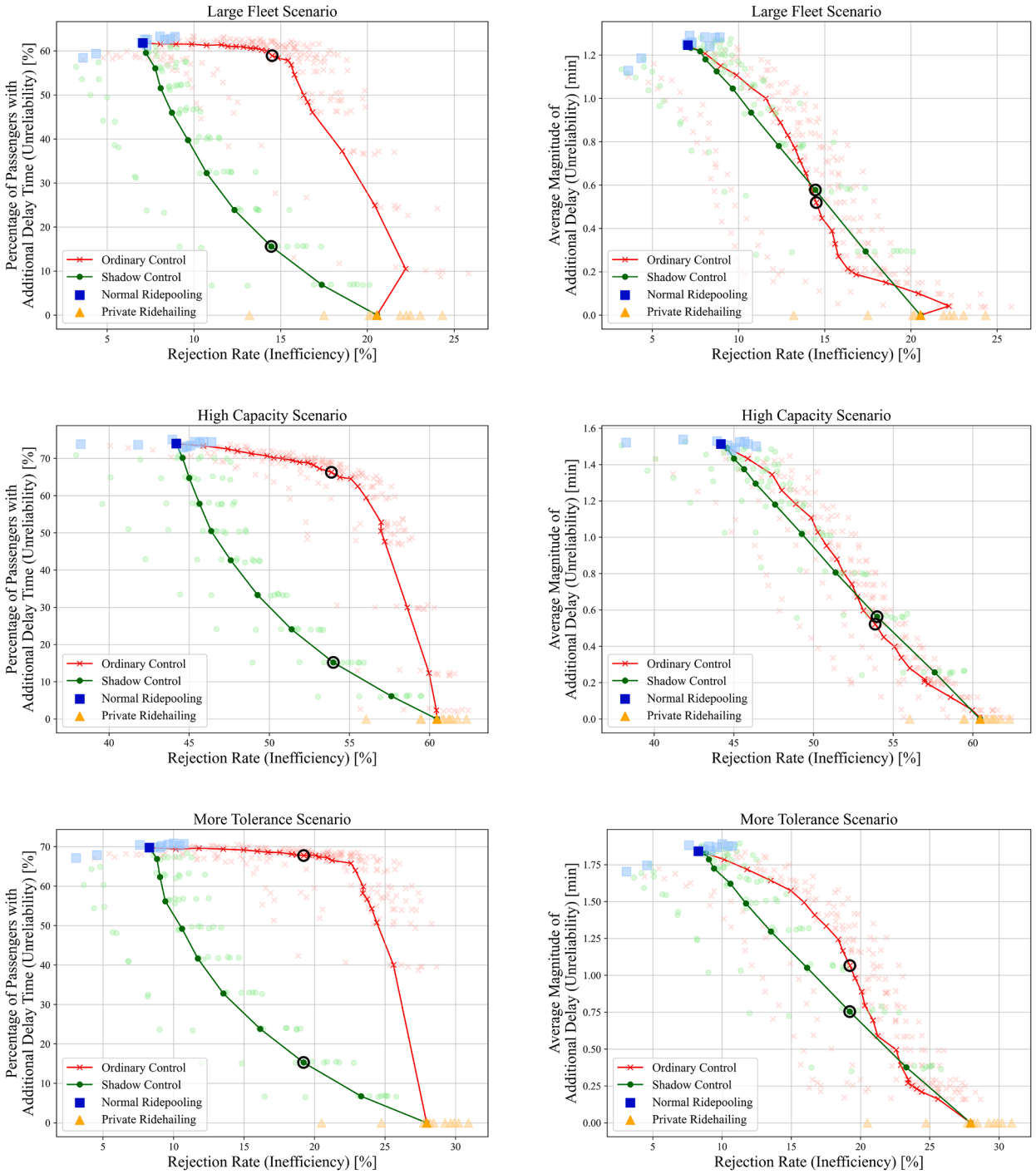


Fig. 12. Policy curves constructed for the two metrics of unreliability across the new three scenarios. The shadow control policy is capable of reducing the number of passengers who faced additional delay without significantly worsening the average magnitude of that delay, in all scenarios.

making new feasible conditions available. In addition, this attainment is done without significantly affecting the average of additional delay over all passengers, as demonstrated in the right plots of Fig. 12. All the inferences and interpretations discussed in the Base scenario are also valid for the three new scenarios, only to a different extent or value. The reader is referred to Tables B.6–B.8 for the corresponding tables of indices for these scenarios (like the one presented in Table 2 for the Base scenario).

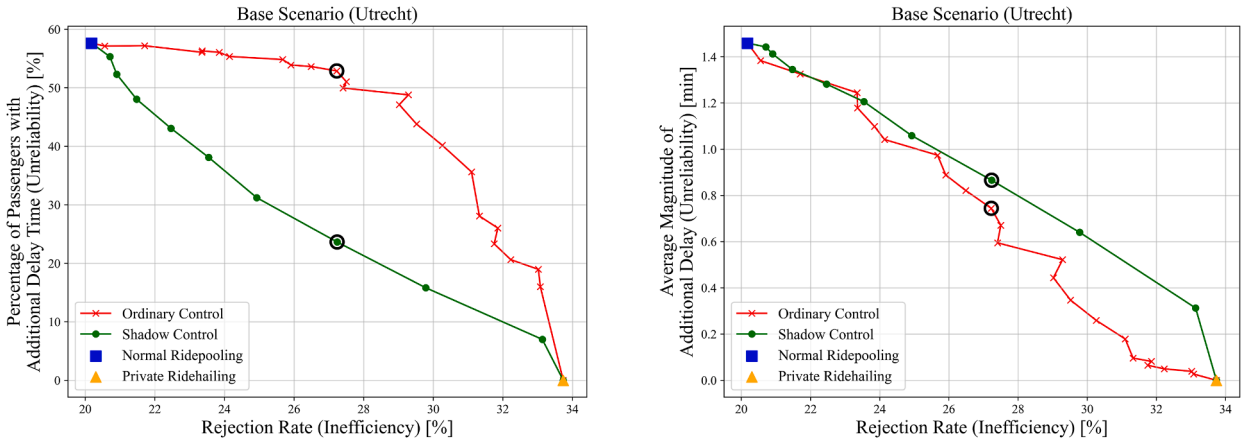


Fig. 13. Policy curves constructed for the two metrics of unreliability in the Utrecht network. The shadow control policy reduces the number of passengers experiencing additional delay without significantly increasing the average magnitude of that delay.

3.5. Robustness across different networks

To further assess the robustness of the proposed framework, we conduct additional experiments on the Utrecht network under the Base scenario. This allows us to evaluate the performance of the method in a different network setting while keeping the operational parameters consistent (except for the service time as explained in Section 3.1). The corresponding results are presented in Fig. 13, where analogous trends to those observed in the Manhattan network can be seen.

As shown in the left plot of Fig. 13, the shadow control policy shifts the policy curve closer to the origin, unlocking improved operational regions and improving both reliability and efficiency. The right plot further shows that this improvement is achieved with only a modest increase in the average magnitude of delay. In particular, for the circled points with similar rejection rates (approximately 27%), the shadow control policy reduces the percentage of passengers experiencing a change in delay time from 52.88% to 23.64%, while the average magnitude increases from 0.74 to 0.87 min. Since the Utrecht experiments are conducted for a single day, the distinction between pale and vivid points is not present. The circled points are selected for the corresponding table of indices reported in Table B.9.

3.6. Results related to informing passengers

Although the shadow control policy greatly reduces the chance of facing a change, it is inevitably accompanied by a higher rejection rate compared to the normal ridepooling. If the rejection of the system is to remain intact, then we can tackle the unreliability by informing passengers about their chances of facing a change, without modifying their slacks.

Fig. 14 demonstrates the accuracy of the predicted information provided to the users about the probability of facing a change, computed across all four different scenarios.

As shown in Fig. 14 (left), the prediction accuracy regarding waiting time is very high (94% - 97%). This is due to the fact that the waiting phase of the trip is very unlikely to undergo a change, as mentioned in Table 2. Therefore, the model can reach high accuracy by predicting that almost all passengers have a low probability of facing a change in waiting time. Detour time, on the other hand, is the most unreliable phase of the trip, and it is much harder to predict. The result of the prediction accuracy of a change in detour time is displayed in Fig. 14 (right).

With a prediction accuracy between 72% - 79% in all four scenarios, the shareability shadow model provides a fairly acceptable prediction regarding unreliability in detour time. Therefore, we can use this information to notify passengers in this way: passengers predicted as low probability are informed that the chance of facing a change throughout their trip is relatively small, while the high-probability passengers are notified about this chance being significant. Furthermore, all passengers are informed that if they do face a change, the magnitude of the change is bounded by the value of the (unchanged) slack.

Comparing each scenario individually in Fig. 14 (right), we can see variation in the prediction accuracy. This is primarily explained by *competitor vehicles*, i.e., other vehicles that can also serve the potential passenger. These vehicles are not considered in the shareability shadow. In other words, when a potential passenger appears under the shadow and is shareable, it can impose additional travel time on the target passenger. However, there might also be other vehicles in the proximity of the potential passenger that could serve them, and if that happens, the travel time of the target passenger would remain unchanged. While the accuracy of prediction is 75.87% in the Base scenario, it decreases to 72.23% in the Large Fleet scenario because the number of competitor vehicles is higher due to the larger fleet. Analogously, the accuracy rises to 79.28% in the High Capacity scenario as we have fewer competitor vehicles. In the More Tolerance scenario, we have more competitor vehicles because the shareability shadow is bigger due to the higher maximum waiting and delay time, and it will encompass more vehicles within. However, the prediction accuracy in this scenario is 76.31%, which is about the same as the Base scenario. This is because the newly-added competitor vehicles are far away from the

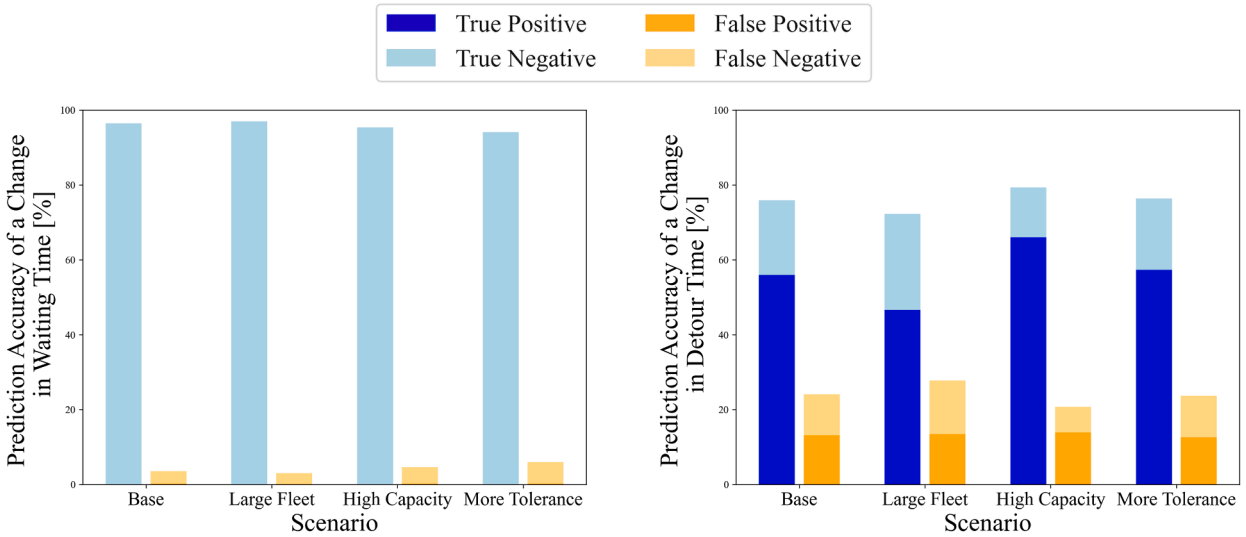


Fig. 14. Prediction accuracy of the shareability shadow model across different scenarios. The shareability shadow has shown an acceptable performance in all scenarios, making it a strong tool to inform passengers about the unreliability of their trip.

potential passenger, making them less likely to serve the passenger. Moreover, since rejecting passengers imposes a high penalty on the objective function, the vehicles are more inclined towards serving some passengers that would have been rejected in the lower tolerance scenarios, rather than competing with each other. Therefore, this is why the impact of competitor vehicles in the More Tolerance scenario is mild.

4. Conclusions and future research

This paper tackles the one-time unreliability of travel time. That is, the unreliability specific to dynamic ridepooling systems, defined as the unpleasant experience of an already-assigned user (target passenger) when they face a change in their travel time caused by the detour the vehicles must take to serve a new passenger (potential passenger). A necessary condition for this change to occur is to have some *slack*, i.e., not having consumed the maximum waiting or delay time after the initial assignment of the target passenger. However, eliminating the slack after the initial assignment is not a smart way to solve unreliability, as it significantly increases the inefficiency of the system. This means that there is a trade-off between the unreliability and inefficiency of the system, and addressing unreliability should be done without a significant increase in inefficiency.

Building upon the idea that different passengers have different chances of facing a change throughout their trip, we used the so-called shareability shadow notion to study and predict this chance for each individual passenger at their time of assignment. The shareability shadow represents a surrounding region around the vehicle's route that determines the feasibility of sharing for a potential passenger. Considering the shareability shadow and the spatial-temporal distribution of the trip demand rate, we introduced an index called the *shareability score*. This score represents the expected number of shareable potential passengers throughout the trip of the target passenger. Therefore, it has a positive correlation with the probability of facing a change in the travel time, which is the unreliability we are addressing. We then propose two ways to use this score, namely to manage the slack in a smart way, or to provide better information to the passengers.

We run extensive simulations over a real-life dataset in Manhattan and Utrecht, and find that:

- **Slack Management:** Shareability score can be leveraged to directly tackle unreliability by adjusting the slack of each passenger according to their score. To study the mentioned trade-off, we built policy curves, one for a benchmark ridepooling and one for a ridepooling in which we used the shareability shadow model. The model was able to generate a policy curve entirely separated from the benchmark one, meaning that by using the shareability score, we are able to reduce both unreliability and inefficiency of the system jointly. Considering two points on the two curves with similar rejection rates, the shareability shadow was able to reduce the percentage of passengers who faced a change by 47.78%, and give a guarantee of a fully reliable trip with accurate arrival time to 83.85% of passengers at their time of assignment. Such a guarantee cannot be offered to any passenger in the benchmark. Furthermore, the total delay time is also slightly reduced.
- **Informing Passengers:** With the system operating as a normal ridepooling with the benchmark efficiency (i.e., no increase in the rejection rate at all), the shareability score is effective at communicating the probability of sharing and therefore reducing uncertainty. In this analysis, the unreliability was indirectly managed by informing passengers about their probability of facing a change throughout their trip, using the shareability score. Considering different scenarios, the shareability score was able to inform passengers with a prediction accuracy between 72%-79%.

Overall, these methods provide a strong foundation to tackle one of the most relevant problems that prevent ridepooling from becoming more widespread. Our results demonstrate that the operators do not need to choose between reliability and efficiency, but can leverage previous data and smart models to improve both.

Future research could extend this work in several directions. First, incorporating the prediction of the magnitude of the change alongside the probability estimation could allow for more nuanced slack adjustments. Second, integrating competitor vehicle dynamics into the shareability shadow may improve predictive accuracy, especially with dense fleets. Next, there are emerging equity issues: in our model, some users face zero changes, but the rest face an increased magnitude of the change. Including equity as a third dimension in the analysis is a promising avenue for further investigations. Furthermore, future research should investigate passengers' sensitivity to different changes in travel time, particularly the relative importance of the first, second, and subsequent updates. Such insights would enable a more balanced slack management, where the level of protection against changes is adjusted according to passenger preferences. Finally, future work could extend the framework beyond a single shortest path by evaluating the shareability score over multiple alternative paths, including slightly longer ones, to explicitly trade off efficiency and reliability.

CRediT authorship contribution statement

Amir Elmi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Emily Moylan:** Writing – review & editing, Supervision, Methodology, Investigation, Formal analysis, Conceptualization; **Javier Alonso-Mora:** Supervision, Methodology, Conceptualization; **Andres Fielbaum:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Acknowledgments

This paper was built upon preliminary ideas developed in the MSc thesis by Mees Matthieu at TU Delft. Dr Andres Fielbaum has been partially funded by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) DE250100417. This publication received support from the project FlexMobility with file number 22150 of the research programme OTP which is financed by the Dutch Research Council (NWO) under the grant ID https://doi.org/10.61686/MJKFO05020.

Data availability

Data will be made available on request.

Appendix A.

A.1. Assignment and routing algorithm

This paper employs the algorithm proposed by Simonetto et al. (2019), which is the relaxed version of the algorithm introduced by Alonso-Mora et al. (2017) for computational efficiency. This is a rolling-horizon algorithm, meaning that each iteration of the algorithm executes at the end of a predefined time interval. At each iteration, the assignment algorithm proceeds through the following three steps:

1. **Identifying feasible matches**. Given the state of the vehicles and passengers, the first step is to determine which requests are feasible to be served by at least one vehicle. This is done by constructing a bipartite graph called RV (request-vehicle) graph. The nodes on one side of this bipartite graph represent requests and the nodes on the other side represent vehicles, i.e., the nodes are $\mathcal{R} \cup \mathcal{V}$. An rv edge is added if request r can be matched to vehicle v to be served through a feasible route. To find an efficient feasible route, the insertion heuristic is employed as follows: when a new passenger is to be added to the vehicle, the order of the pick-up and drop-off nodes of the previously assigned passengers is maintained, and the pick-up and drop-off nodes of the new passenger are inserted in between, provided that the drop-off node is always inserted after the pick-up point. The insertion of the nodes in different spaces can generate many different routes. For a route to be feasible, it should not violate the following constraints: maximum waiting time of passenger $i \in \mathcal{P}_v$ (Ω_i), maximum delay time of passenger $i \in \mathcal{P}_v$ (Φ_i), and the capacity of vehicle v (c_v). $\mathcal{P}_v$ denotes the set of all passengers assigned to vehicle v, including the to-be-assigned request r. Among all feasible routes, the one that minimises Eq. (A.1), which is the cost of serving request r with vehicle v, is chosen as the efficient feasible route.

$$c_{r,v} = \left[p_w t^r_w(v) + p_v t^r_v(v)\right] + \sum_{i \in \mathcal{P}_v \setminus \{r\}} \left[p_w \Delta t^i_w(r,v) + p_v \Delta t^i_v(r,v)\right] + c_0 \Delta L(r,v). \tag{A.1}$$

where $t^r_w(v)/t^r_v(v)$ is the waiting/in-vehicle time of request r when served by vehicle v, $\Delta t^i_w(r,v)/\Delta t^i_v(r,v)$ is the additional waiting/in-vehicle time of the already-assigned passenger i of vehicle v after including request r, $\Delta L(r,v)$ is the extra distance vehicle v has to travel to serve request r, and p_w, p_v, c_0 are the weight costs associated with each term. In other words, the first term is the cost related to the waiting time and the in-vehicle time of serving the new request. The second term considers the extra waiting and in-vehicle times of previously assigned passengers. The third term demonstrates the cost associated with the distance travelled by the vehicle.

2. **Finding the optimal match**. Given all possible matches at hand, now is the time to find the optimal matches between requests and vehicles. This is done by solving an LP in which we try to optimise the following objective function:

$$\min \sum_{r,v\in\mathcal{E}_{RV}} c_{r,v}\epsilon_{r,v} + \sum_{r\in\mathcal{R}} c_{ko}\chi_r \tag{A.2}$$

where $\epsilon_{r,v}$ and χ_r are binary variables. If vehicle v serves request r then $\epsilon_{r,v} = 1$, otherwise 0. If request r is rejected $\chi_r = 1$, otherwise 0. $c_{r,v}$ and c_{ko} are costs of either serving or rejecting a request. $\mathcal{E}_{RV}$ represents the set of all possible matches between requests and vehicles in the RV graph.

Eq. (A.2) represents the total cost of serving or rejecting requests. The optimisation is done under two main constraints: *i*) each vehicle should at most be assigned to one request, and *ii*) each request should either be assigned or rejected.

3. **Rebalancing idle vehicles**. After each assignment, some vehicles may remain idle, and some requests may be rejected. Assuming that rejected requests are signs of the whereabouts of zones with higher demand, this paper utilises a method to rebalance the fleet of vehicles by moving the idle vehicles towards busier zones. This is done by optimising another LP in which we try to send idle vehicles to the nearest rejected request. Each rejected request can be the destination of one vehicle only, and each vehicle can move towards only one rejected request. If rejected requests outnumber idle vehicles, some rejected requests will remain unattended; otherwise, some vehicles will remain idle.

There are two common practices in the assignment algorithms of previous papers. First, rejected requests of one iteration can be carried over to the next iteration, in the hope that they will be accepted in the next round. Second, assigned but waiting passengers of one iteration can be reassigned in the next iterations to reach better efficiency. Neither of these two strategies is used in this paper, as the predictive model described in the next sections does not take these behaviours into account. This means that the incorporation of these two strategies will reduce the effectiveness of the proposed method to reduce unreliability. Furthermore, previous studies have shown that the efficiency gained by these two practices is not significant, so removing them from the assignment algorithm would not harm the efficiency of the system very much. Finally, the second practice would also induce another source of unreliability, i.e., the change of the assigned vehicle, which is against the goal of this paper.

A.2. Table of parameter values in the Base scenario

Table A.4
Parameter values in the Base scenario.

Parameter	Value
c_v	4
Ω_r	5 min
Φ_r	10 min
δ	24 s
t_{warm}	20 min
c_{ko}	40 min
p_w	2
p_v	1
c_o	1

A.3. Table of notations

Table A.5
Notations and their definitions.

Symbol	Definition
$\mathcal{N}$	Set of all nodes in the network
$\mathcal{A}$	Set of all arcs in the network
$\mathcal{V}$	Set of all vehicles
$t(i,j)$	Travel time from node i to node j following the shortest path
t_r	Request time of request r
O_r	Origin of request r
D_r	Destination of request r
κ_r	Number of people involved in request r
T_r	Direct origin-to-destination travel time of request r
c_v	Capacity (number of seats) of vehicle v
t^r_a	Assignment time of request r
t^r_{PU}	Pick-up time of request r
t^r_{DO}	Drop-off time of request r
t^r_w	Waiting time of request r
t^r_v	In-vehicle time of request r
t^r_d	Detour time of request r
t^r_D	Delay time of request r
Ω_r	Maximum tolerable waiting time of request r
Φ_r	Maximum tolerable delay time of request r
$t^r_{h'}$	Initial value of travel time component h of request r, where $h \in \{w,d,D\}$
Δt^r_h	Increase (change) in travel time component h of request r, where $h \in \{w,d,D\}$
ω_r	Waiting slack of request r
ϕ_r	Delay slack of request r
$p_v(t)$	Position of vehicle v at time t
$W_n(s_i)$	Waiting time to visit node n if the vehicle deviates in section s_i
$D_n(s_i)$	Deviation time occurred in section s_i to visit node n
$S_v(t)$	Set of all sections in the itinerary of vehicle v at time t
$C_v(s_i)$	Set of all waiting and delay slacks of passengers of vehicle v, existing in section s_i
$\mathcal{N}_{PU}(t)$	Set of all feasible pick-up nodes at time t
$\mathcal{N}_{DO}(t)$	Set of all feasible drop-off nodes at time t
$c_v(t)$	Binary variable of capacity of vehicle v at time t
λ	Trip demand of the entire network
$S_r(t)$	Instantaneous shareability score of request r at time t
α	Smoothing constant used in exponential smoothing
λ_{z_O,z_D}	Trip demand starting in origin zone z_O and ending in destination zone z_D
$\lambda_{z_O,z_D}(t)$	Trip demand starting in origin zone z_O and ending in destination zone z_D at time t
$\lambda_{z_O}(t)$	Trip demand generated in origin zone z_O at time t
$\lambda_{z_D}(t)$	Trip demand attracted in destination zone z_D at time t
$\mathcal{Z}_{PU}(t)$	Set of origin zones, each containing at least one feasible pick-up node at time t
$\mathcal{Z}_{DO}(t)$	Set of destination zones, each containing at least one feasible drop-off node at time t
$\mathcal{N}_{PU}(z_O,t)$	Set of all feasible pick-up nodes in origin zone z_O at time t
$\mathcal{N}^{\text{tot}}_{DO}(z_D,t)$	Set of all feasible drop-off nodes across all sections in destination zone z_D at time t
$\mathcal{N}_{z_O}$	Set of all nodes in origin zone z_O
$\mathcal{N}_{z_D}$	Set of all nodes in destination zone z_D
S_r	Time-integrated shareability score of request r
t_n	Time at which the vehicle is scheduled to visit node n
$\mathcal{T}(\mathcal{I}^r_v)$	Set of times at which the vehicle is scheduled to visit nodes in the current itinerary $\mathcal{I}^r_v$ of vehicle v to serve target passenger r
δ	Service time at each pick-up or drop-off node
t_{warm}	The duration of the warm-up phase at the beginning of the simulation
$\mathcal{P}_v$	Set of all passengers assigned to vehicle v
$c_{r,v}$	Cost of serving request r with vehicle v
c_{ko}	Cost of rejecting a request
$\epsilon_{r,v}$	Binary variable of assigning request r to vehicle v. 1 if assigned; 0 otherwise.
χ_r	Binary variable of rejecting request r. 1 if rejected; 0 otherwise.
$\mathcal{E}_{RV}$	Set of all edges between requests $\mathcal{R}$ and vehicles $\mathcal{V}$ in the RV graph.
p_w [p_v]	Waiting [in-vehicle] time cost weight
$t^r_w(v)$ [$t^r_v(v)$]	Waiting [in-vehicle] time of new request r when served by vehicle v
$\Delta t^i_w(r,v)$ [$\Delta t^i_v(r,v)$]	Extra waiting [in-vehicle] time of previously-assigned request i when new request r is served by vehicle v
c_0	operational cost weight associated with the distance travelled by a vehicle
$\Delta L(r,v)$	Extra distance vehicle v has to travel to serve request r,

Appendix B.

B.1. Policy curves for waiting and detour times

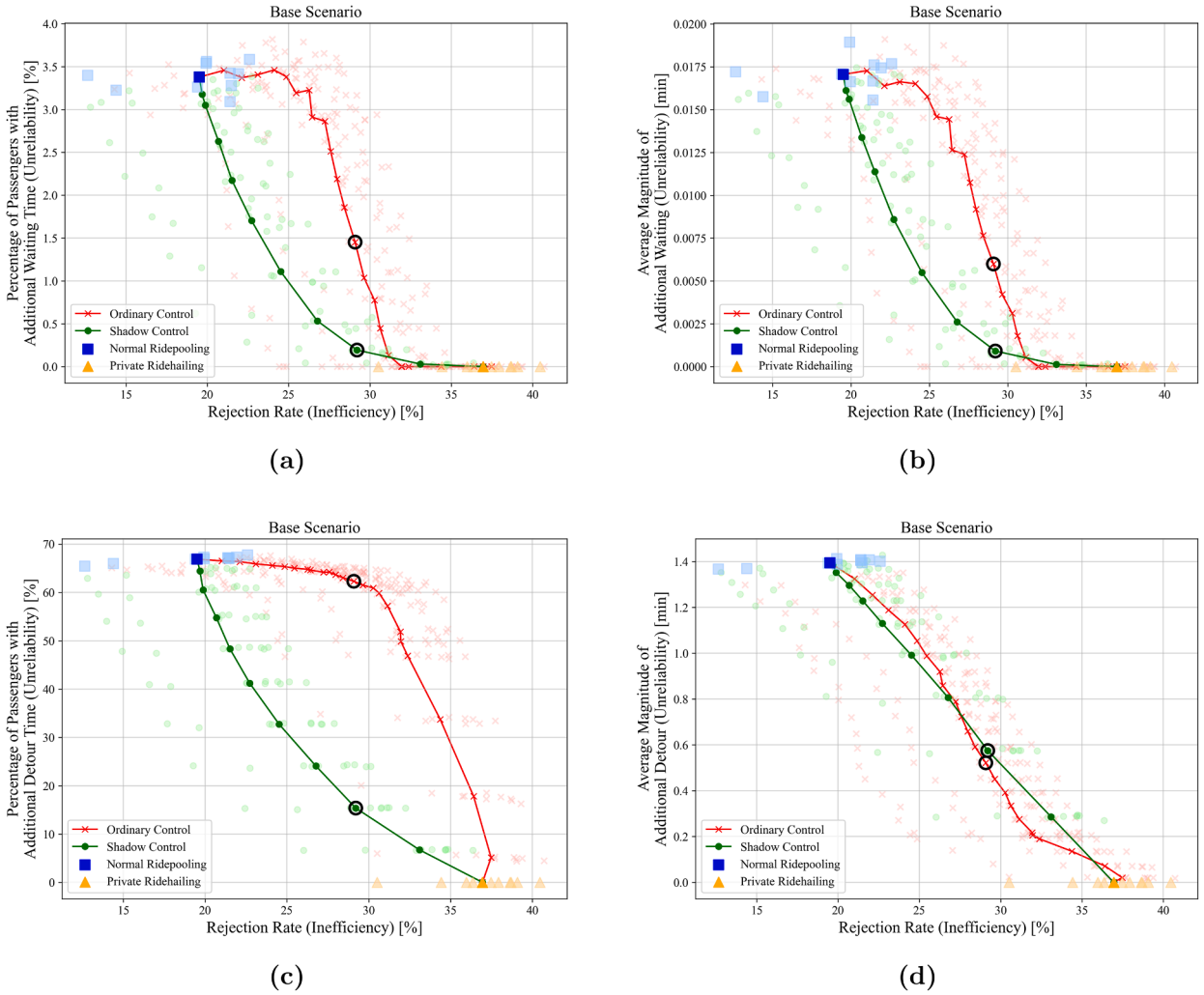


Fig. B.15. Policy curves for waiting and detour times in the Base scenario in Manhattan

B.2. Detour time in private ridehailing

In private ridehailing systems, passengers only expect a waiting time until the vehicle arrives at their pick-up point, and they will be directly transferred to their drop-off point, which means no detour time for passengers. In this paper, private ridehailing is indirectly created by setting the slack of all passengers to zero, even though the initial waiting and detour times are allowed. This is because of the following: In the assignment algorithm of this paper, at most one passenger is assigned to each vehicle at each iteration. When an idle empty vehicle starts working in this system and is assigned to serve a passenger, some waiting time and zero detour time (because there is no other passenger) are defined for the passenger, and then the slack of this passenger is set to zero. If another passenger in subsequent iterations is to be assigned to this vehicle, they cannot be picked up before the drop-off of the first passenger, as this imposes additional travel time and violates the zero slack constraint. This means that the next passenger should be served when the current passenger is dropped off, and the vehicle is empty again, which means no detour time for subsequent passengers either, even though the initial detour time is allowed and only additional detour time is prohibited by the zero slack.

However, there is one particular condition in which some passengers experience some detour time, and that is demonstrated in Fig. B.16. If a new passenger appears at the same pick-up point of the previously-assigned passenger while the vehicle is moving towards that pick-up point, then there is no constraint preventing the pooling of these two passengers. This is because only the additional travel time is prevented by the zero slack constraint, but the initial detour time is allowed, and in this particular case, the first passenger (passenger A) shares their trip with the second passenger (passenger B) without facing any additional travel time, whereas the second passenger will be served by some initial detour time. This means that although the system performs a door-to-door private service for most passengers, a few passengers might experience some detour time. It should be noted that this detour time is compensated to some extent, as the vehicle is moving towards the origin of the passenger even before they request a ride.

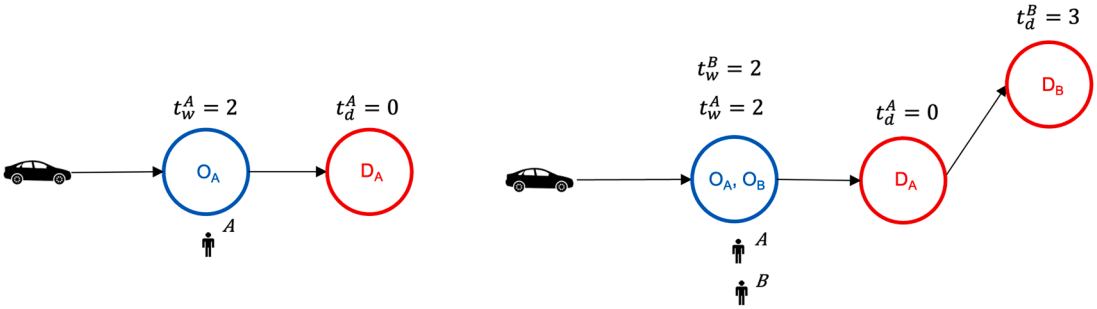


Fig. B.16. Detour time in private ridehailing systems. If the origin of two passengers are the same, the second-assigned passenger might experience some detour time, as the initial detour time is allowed and only additional detour time is banned by the zero slack constraint. This particular condition induces detour time for some passengers even in private ridehailing.

B.3. Tables of indices for other scenarios

Table B.6
Indices to study unreliability across the selected points in the Large Fleet scenario.

Index	h	Normal Ridepooling	Private Ridehailing	Ordinary Control	Shadow Control
Inefficiency Indices					
Rejection Rate [%]	–	**7.06**	**20.55**	**14.51**	**14.46**
Avg. VMT [min]	–	44.99	61.52	**50.65**	**54.10**
Avg. (h) [min]	t_w	2.18	3.11	2.45	2.82
	t_d	2.51	0.50	1.75	1.25
	t_D	4.69	3.61	**4.20**	**4.07**
Unreliability Indices					
$P(\Delta h)$ [%]	t_w	**2.90**	0	1.41	0.26
	t_d	**60.91**	0	58.10	15.58
	t_D	**61.81**	0	**58.93**	**15.61**
$P^{\text{allowed}}(\Delta h)$ [%]	t_w	99.81	0	99.72	18.69
	t_d	99.96	0	99.96	16.43
	t_D	99.96	0	**99.96**	**16.43**
$E(\Delta h)$ [min]	t_w	0.51	N/A	0.41	0.51
	t_d	2.02	N/A	0.88	3.70
	t_D	2.01	N/A	**0.88**	**3.70**
$E^{\text{allowed}}(\Delta h)$ [min]	t_w	2.84	N/A	0.89	1.02
	t_d	6.56	N/A	1.34	6.71
	t_D	6.56	N/A	1.34	6.71
Avg. $N(\Delta h)$	t_w	1.02	N/A	1.00	1.00
	t_d	1.64	N/A	1.07	1.78
	t_D	1.64	N/A	1.07	1.78
Max $N(\Delta h)$	t_w	2	N/A	1	2
	t_d	7	N/A	2	5
	t_D	7	N/A	2	5

Table B.7
Indices to study unreliability across the selected points in the High Capacity scenario.

Index	h	Normal Ridepooling	Private Ridehailing	Ordinary Control	Shadow Control
Inefficiency Indices					
Rejection Rate [%]	–	**44.18**	**60.45**	**53.87**	**53.99**
Avg. VMT [min]	–	87.44	101.72	**95.31**	**97.87**
Avg. (h) [min]	t_w	2.91	3.65	3.13	3.49
	t_d	3.81	0.72	2.35	1.55
	t_D	6.72	4.37	**5.48**	**5.04**
Unreliability Indices					
$P(\Delta h)$ [%]	t_w	**4.44**	0	1.34	0.14
	t_d	**72.87**	0	65.45	15.18
	t_D	**73.97**	0	**66.27**	**15.19**
$P^{\text{allowed}}(\Delta h)$ [%]	t_w	99.51	0	99.38	17.82
	t_d	99.87	0	99.91	15.50
	t_D	99.87	0	**99.91**	**15.50**
$E(\Delta h)$ [min]	t_w	0.49	N/A	0.41	0.47
	t_d	2.05	N/A	0.79	3.70
	t_D	2.05	N/A	**0.79**	**3.71**
$E^{\text{allowed}}(\Delta h)$ [min]	t_w	2.13	N/A	0.66	0.70
	t_d	4.80	N/A	0.97	5.71
	t_D	4.80	N/A	0.97	5.71
Avg. $N(\Delta h)$	t_w	1.03	N/A	1.00	1.00
	t_d	1.84	N/A	1.04	1.81
	t_D	1.85	N/A	1.04	1.81
Max $N(\Delta h)$	t_w	3	N/A	1	1
	t_d	8	N/A	2	6
	t_D	8	N/A	2	6

Table B.8
Indices to study unreliability across the selected points in the More Tolerance scenario.

Index	h	Normal Ridepooling	Private Ridehailing	Ordinary Control	Shadow Control
Inefficiency Indices					
Rejection Rate [%]	–	**8.29**	**27.95**	**19.23**	**19.22**
Avg. VMT [min]	–	60.03	80.24	**67.17**	**73.02**
Avg. (h) [min]	t_w	3.55	5.05	4.04	4.77
	t_d	3.83	1.46	3.15	2.21
	t_D	7.38	6.50	**7.18**	**6.98**
Unreliability Indices					
$P(\Delta h)$ [%]	t_w	**5.79**	0	6.62	0.33
	t_d	**68.30**	0	65.82	15.26
	t_D	**69.79**	0	**67.77**	**15.30**
$P^{\text{allowed}}(\Delta h)$ [%]	t_w	99.86	0	99.74	18.74
	t_d	99.98	0	99.97	16.08
	t_D	99.98	0	**99.97**	**16.08**
$E(\Delta h)$ [min]	t_w	0.63	N/A	0.50	0.63
	t_d	2.64	N/A	1.57	4.93
	t_D	2.64	N/A	**1.57**	**4.93**
$E^{\text{allowed}}(\Delta h)$ [min]	t_w	3.99	N/A	1.71	1.18
	t_d	9.47	N/A	2.73	9.06
	t_D	9.47	N/A	2.73	9.06
Avg. $N(\Delta h)$	t_w	1.05	N/A	1.01	1.01
	t_d	1.89	N/A	1.30	2.01
	t_D	1.90	N/A	1.33	2.01
Max $N(\Delta h)$	t_w	3	N/A	2	2
	t_d	8	N/A	3	6
	t_D	8	N/A	3	6

B.4. Tables of indices for the Utrecht network

Table B.9
Indices to study unreliability across the selected points in the Utrecht network.

Index	h	Normal Ridepooling	Private Ridehailing	Ordinary Control	Shadow Control
Inefficiency Indices					
Rejection Rate [%]	–	**20.18**	**33.73**	**27.23**	**27.24**
Avg. VMT [min]	–	101.15	113.22	**105.09**	**104.55**
Avg. (h) [min]	t_w	2.55	2.63	2.58	2.67
	t_d	3.24	1.30	2.47	2.22
	t_D	5.79	3.93	**5.05**	**4.89**
Unreliability Indices					
$P(\Delta h)$ [%]	t_w	**2.19**	0.0	1.45	0.47
	t_d	**56.77**	0.0	52.10	23.56
	t_D	**57.64**	0.0	**52.88**	**23.64**
$P^{\text{allowed}}(\Delta h)$ [%]	t_w	99.62	0.0	99.66	28.22
	t_d	99.95	0.0	99.92	26.76
	t_D	99.95	0.0	**99.92**	**26.76**
$E(\Delta h)$ [min]	t_w	0.60	0.0	0.40	0.58
	t_d	2.55	0.0	1.42	3.66
	t_D	2.53	0.0	**1.41**	**3.66**
$E^{\text{allowed}}(\Delta h)$ [min]	t_w	2.48	0.0	1.21	1.28
	t_d	5.68	0.0	2.04	6.54
	t_D	5.68	0.0	2.04	6.54
Avg. $N(\Delta h)$	t_w	1.01	0.0	1.00	1.00
	t_d	1.41	0.0	1.07	1.49
	t_D	1.42	0.0	1.08	1.49
Max $N(\Delta h)$	t_w	2.0	0.0	1.0	1.0
	t_d	5.0	0.0	2.0	4.0
	t_D	5.0	0.0	2.0	4.0

References

Alonso-González, M.J., van Oort, N., Cats, O., Hoogendoorn-Lanser, S., Hoogendoorn, S., 2020. Value of time and reliability for urban pooled on-demand services. Transp. Res. C: Emerg. Technol. 115, 102621.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. 114 (3), 462–467.

Bailey, N.K., 2022. Dynamic Ridesharing under Travel Time Uncertainty: Passenger Preference and Optimal Assignment Methods. Ph.D. thesis. Massachusetts Institute of Technology.

Bilali, A., Dandl, F., Fastenrath, U., Bogenberger, K., 2019. An analytical model for on-demand ride sharing to evaluate the impact of reservation, detour and maximum waiting time. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 1715–1720.

Concas, S., Kolpakov, A., 2009. Synthesis of Research on Value of Time and Value of Reliability. Technical Report. Center for Urban Transportation Research, University of South Florida.

Daganzo, C.F., Ouyang, Y., Yang, H., 2020. Analysis of ride-sharing with service time and detour guarantees. Transp. Res. B: Methodol. 140, 130–150.

, 2021. Rail Passenger Experience of Disruption Handling. Technical Report. Department for Transport. London, UK.

Dieter, P., Stumpe, M., Ulmer, M.W., Schryen, G., 2023. Anticipatory assignment of passengers to meeting points for taxi-ridesharing. Transp. Res. D: Transp. Environ. 121, 103832.

Durán-Micco, J., van Kooten Niekerk, M., Vansteenwegen, P., 2022. Designing bus line plans for realistic cases-the utrecht case study. Expert Syst. Appl. 187, 115918.

Engelhardt, R., Dandl, F., Bogenberger, K., 2022. Simulating ride-pooling services with pre-booking and on-demand customers. arXiv preprint arXiv:2210.06972

Fayed, L., Nilsson, G., Geroliminis, N., 2023. On the utilization of dedicated bus lanes for pooled ride-hailing services. Transp. Res. B: Methodol. 169, 29–52.

Fayed, L., Nilsson, G., Geroliminis, N., 2024. A dynamic macroscopic framework for pricing of ride-hailing services with an optional bus lane access for pool vehicles. Transp. Res. C: Emerg. Technol. 169, 104854.

Fiedler, D., Čertickỳ, M., Alonso-Mora, J., Pěchouček, M., Čáp, M., 2022. Large-scale online ridesharing: the effect of assignment optimality on system performance. J. Intell. Transp. Syst. 28 (2), 189–210.

Fielbaum, A., 2025. Coordination costs in spatial matching: assigned versus onsite transport modes. Transp. Res. A: Policy Pract. 199, 104556.

Fielbaum, A., Alonso-Mora, J., 2020. Unreliability in ridesharing systems: measuring changes in users' times due to new requests. Transp. Res. C: Emerg. Technol. 121, 102831.

Fielbaum, A., Alonso-Mora, J., 2024. Design of mixed fixed-flexible bus public transport networks by tracking the paths of on-demand vehicles. Transp. Res. C: Emerg. Technol. 168, 104580.

Fielbaum, A., Bai, X., Alonso-Mora, J., 2021. On-demand ridesharing with optimized pick-up and drop-off walking locations. Transp. Res. Part C: Emerg. Technol. 126, 103061.

Fielbaum, A., Kronmueller, M., Alonso-Mora, J., 2022. Anticipatory routing methods for an on-demand ridepooling mobility system. Transportation 49 (6), 1921–1962.

Hansen, T., Sener, I.N., 2023. Strangers on this road we are on: a literature review of pooling in on-demand mobility services. Transp. Res. Rec. 2677 (3), 1368–1381.

Harmann, D., Trick, J.-S., Bienzeisler, L., Friedrich, B., 2025. Moving beyond assumptions: stop times and key determinants for pick-ups and drop-offs in ridepooling systems. Transp. A: Transp. Sci. 1–20.

Hörl, S., Zwick, F., 2022. Traffic uncertainty in on-demand high-capacity ride-pooling. In: 101st Annual Meeting of the Transportation Research Board (TRB).
Hyland, M., Mahmassani, H.S., 2020. Operational benefits and challenges of shared-ride automated mobility-on-demand services. Transp. Res. A: Policy Pract. 134, 251–270.
Hyndman, R., Koehler, A., Ord, K., Snyder, R., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer.
Kucharski, R., Fielbaum, A., Alonso-Mora, J., Cats, O., 2021. If you are late, everyone is late: late passenger arrival and ride-pooling systems' performance. Transp. A: Transp. Sci. 17 (4), 1077–1100.
Li, C., Parker, D., Hao, Q., 2021. Vehicle dispatch in on-demand ride-sharing with stochastic travel times. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5966–5972.
Li, W., Pu, Z., Li, Y., Ban, X.J., 2019. Characterization of ridesplitting based on observed data: a case study of Chengdu, China. Transp. Res. C: Emerg. Technol. 100, 330–353.
Li, Y., Chung, S.H., 2020. Ride-sharing under travel time uncertainty: robust optimization and clustering approaches. Comput. Ind. Eng. 149, 106601.
Li, Y., Sun, H., Zhang, W., Lv, Y., Chang, X., 2025. On-demand ride-sharing with the walking incentive for meeting points. Transp. Res. C: Emerg. Technol. 174, 105098.
Liu, T. L.K., Krishnakumari, P., Cats, O., 2019a. Exploring demand patterns of a ride-sourcing service using spatial and temporal clustering. In: 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, pp. 1–9.
Liu, X., Li, W., Li, Y., Fan, J., Shen, Z., 2021. Quantifying environmental benefits of ridesplitting based on observed data from ridesourcing services. Transp. Res. Rec. 2675 (8), 355–368.
Liu, Z., Miwa, T., Zeng, W., Bell, M. G.H., Morikawa, T., 2019b. Dynamic shared autonomous taxi system considering on-time arrival reliability. Transp. Res. C: Emerg. Technol. 103, 281–297.
Pellegrini, A., Fielbaum, A., 2025. Are users ready to accept fully flexible walking in on-demand mobility? Transp. Res. C: Emerg. Technol. 178, 105210.
Repoux, M., Geroliminis, N., Kaspi, M., 2021. Operational analysis of an innovative semi-autonomous on-demand transportation system. Transp. Res. C: Emerg. Technol. 132, 103373.
Sajid, M., Mishra, S., Prajith, P.K., 2025. Why not a ride together? A sequential exploratory study on the barriers and backlashes to ridesharing among urban commuters. Res. Transp. Bus. Manag. 60, 101366.
Simonetto, A., Monteil, J., Gambella, C., 2019. Real-time city-scale ridesharing via linear assignment problems. Transp. Res. C: Emerg. Technol. 101, 208–232.
Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2015. The benefits of meeting points in ride-sharing systems. Transp. Res. B: Methodol. 82, 36–53.
Su, Q., Luo, X., Wang, H., Huo, Z., Yamamoto, T., 2024. Dispatching prebooking-based shared autonomous vehicles with hybrid ownership: an approach of vehicle trip planning with uncertain travel times. Transp. A: Transp. Sci. 1–40.
Tachet, R., Sagarra, O., Santi, P., Resta, G., Szell, M., Strogatz, S.H., Ratti, C., 2017. Scaling law of urban ride sharing. Sci. Rep. 7 (1), 42868.
Theodoridis, N., Fielbaum, A., Alonso-Mora, J., Atasoy, B., 2023. Quantifying the benefits of short-term reservations in on-demand ridepooling. In: Australasian Transport Research Forum (ATRF), 44th, 2023, Perth, Western Australia, Australia.
Tirachini, A., Gomez-Lobo, A., 2020. Does ride-hailing increase or decrease vehicle kilometers traveled (VKT)? A simulation approach for Santiago de Chile. Int. J. Sustain. Transp. 14 (3), 187–204.
Wallar, A., Van Der Zee, M., Alonso-Mora, J., Rus, D., 2018. Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 4539–4546.
Wang, J., Wang, X., Yang, S., Yang, H., Zhang, X., Gao, Z., 2021. Predicting the matching probability and the expected ride/shared distance for each dynamic ridepooling order: a mathematical modeling approach. Transp. Res. B: Methodol. 154, 125–146.
Wolf, F., Engelhardt, R., Zhang, Y., Dandl, F., Bogenberger, K., 2023. Effects of dynamic and stochastic travel times on the operation of mobility-on-demand services. In: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 5476–5481.
Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. Eur. J. Oper. Res. 185 (2), 534–551.
Yan, J., Martin, S., Taylor, S.J., 2025. Trading flexibility for adoption: from dynamic to static walking in ride-sharing. Manag. Sci. 71 (7), 5875–5892.
Zhang, X., Zhong, S., Jia, N., Ling, S., Yao, W., Ma, S., 2024. A barrier to the promotion of app-based ridesplitting: travelers' ambiguity aversion in mode choice. Transp. Res. A: Policy Pract. 181, 103971.
Zhou, Z., Roncoli, C., 2022. A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion. Transp. Res. C: Emerg. Technol. 139, 103658.
Zhu, S., Levinson, D.M., 2011. Disruptions to transportation networks: a review. In: Network Reliability in Practice: Selected Papers from the Fourth International Symposium on Transportation Network Reliability. Springer, pp. 5–20.
Zwick, F., Axhausen, K.W., 2022. Ride-pooling demand prediction: a spatiotemporal assessment in Germany. J. Transp. Geogr. 100, 103307.