Transportation Research Part C 136 (2022) 103520

Contents lists available at ScienceDirect

Transportation Research Part C

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

A business class for autonomous mobility-on-demand: Modeling service quality contracts in dynamic ridesharing systems

Breno A. Beirigo [*], Rudy R. Negenborn, Javier Alonso-Mora, Frederik Schulte

Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Autonomous vehicle
Mobility-on-demand
Ridesharing
Service quality
On-demand hiring
Multi-objective optimization

ABSTRACT

With the popularization of *transportation network companies* (TNCs) (e.g., Uber, Lyft) and the rise of *autonomous vehicles* (AVs), even major car manufacturers are increasingly considering themselves as *autonomous mobility-on-demand* (AMoD) providers rather than individual vehicle sellers. However, matching the convenience of owning a vehicle requires providing consistent service quality, taking into account individual expectations. Typically, different classes of users have different *service quality* (SQ) expectations in terms of responsiveness, reliability, and privacy. Nonetheless, AMoD systems presented in the literature do not enable active control of service quality in the short term, especially in light of unusual demand patterns, sometimes allowing extensive delays and user rejections. This study proposes a method to control the daily operations of an AMoD system that uses the SQ expectations of heterogeneous user classes to dynamically distribute service quality among riders. Additionally, we consider an elastic vehicle supply, that is, privately-owned freelance AVs (FAVs) can be hired on short notice to help providers meeting user service-level expectations. We formalize the problem as the *dial-a-ride problem with service quality contracts* (DARP-SQC) and propose a multi-objective matheuristic to address real-world requests from Manhattan, New York City. Applying the proposed service-level constraints, we improve user satisfaction (in terms of reached service-level expectations) by 53% on average compared to conventional ridesharing systems, even without hiring additional vehicles. By deploying service-quality-oriented on-demand hiring, our hierarchical optimization approach allows providers to adequately cater to each segment of the customer base without necessarily owning large fleets.

1. Introduction

The upcoming *autonomous vehicle* (AV) transition is expected to re-shape urban transportation in the next decades. The current personal mobility paradigm, mainly based on vehicle ownership, is likely to be phased out as *autonomous mobility-on-demand* (AMoD) systems develop (Litman, 2017). All major car manufacturers have already anticipated this shift, announcing their plans to use *shared autonomous vehicle* (SAV) fleets to provide seamless transportation solutions to travelers rather than selling individual vehicles (Hyland and Mahmassani, 2017).

However, most AMoD systems are unable to actively control service quality in the short-term, at the operational level. Once a particular service level is defined (e.g., in terms of maximum waiting times), they determine a fleet size that maintains such level at a reasonable rate (see, e.g., Alonso-Mora et al., 2017; Fagnant et al., 2015; Boesch et al., 2016; Vazifeh et al., 2018). As high

* Corresponding author.
E-mail addresses: b.alvesbeirigo@tudelft.nl (B.A. Beirigo), r.r.negenborn@tudelft.nl (R.R. Negenborn), j.alonsomora@tudelft.nl (J. Alonso-Mora), f.schulte@tudelft.nl (F. Schulte).

https://doi.org/10.1016/j.trc.2021.103520
Received 23 January 2021; Received in revised form 7 December 2021; Accepted 7 December 2021
Available online 20 January 2022

this rate might be, even a small lack of reliability can undermine the acceptance of mobility-on-demand services, especially for customers who cannot afford service rejection or excess delays. User acceptance, however, is a crucial factor in ridesharing systems: peak performance can only be achieved when a sufficiently large number of riders are willing to participate (Krueger et al., 2016).

Moreover, ridesharing research generally does not consider that different customer groups widely vary in their behavior and desires (Zeithaml et al., 2001). Companies from various sectors acknowledge that profits and perceived *service quality* (SQ) can be improved by directly catering to segments of their customer base. For example, airline passengers can generally choose from first, business, or economy class, whereas current on-demand urban *transportation network companies* (TNCs), such as Uber and Lyft, offer a range of service tiers, spanning from pooled riders up to executive limousine services. Similarly, future AMoD systems are also likely to consider customer preferences when matching overlapping itineraries to construct viable routes.

This study proposes a hierarchical multi-objective approach to dynamically control service quality — defined in terms of responsiveness, reliability, and privacy — to meet user needs. To this end, we take the perspective of a ridesharing AMoD company servicing a diversified user base where riders have both different sharing preferences and *service level* (SL) requirements, which are expressed in terms of maximum tolerance delays. By varying such parameters, we define a strict *service quality contract* (SQC) that specifies the minimum SQ-settings for three user classes, namely, business (B), standard (S), and low-cost (L). Whenever the operating fleet cannot sustain the service quality demanded by incoming requests, third-party vehicles are hired online to guarantee the expected user experience required by each SQ class.

Although fleet size elasticity is already a core element of current TNCs, drivers' availability remains a bottleneck for reliable operations. An AV-based solution tends to be more robust: autonomous ridesharing platforms do not need to overcome the opportunity cost of drivers but the opportunity cost of idle AVs. These could include, for instance, staying parked (possibly incurring parking costs) or fulfilling other tasks to their owners. Similar to Campbell (2018), in this study, we consider that many AV owners would find value in sharing their vehicles when not in use. Once vehicle availability is disassociated from driver availability, independent AV owners could profit from otherwise unproductive idle times (e.g., parked at work), making their vehicles available to AMoD systems in exchange for compensation on a contract basis. On the other hand, by leveraging the underutilized capacity of a *freelance autonomous vehicles* (FAV) fleet, providers can consider a substantially larger pool of hireable vehicles to resolve supply and demand mismatches, adjusting the fleet size in the short term to meet users' service level expectations.

For example, if 1% of the cars daily entering Manhattan were autonomous, a pool of about 7000 vehicles could be activated (Beaton, 2019). The proposed model excels to the extent that it can take advantage of unutilized, abundant vehicle supply to react in the short term to demand fluctuations. Although this scenario is more likely to unfold in an AV-predominant world, it is worth noting that our method is agnostic to vehicle automation. The freelance fleet could comprise both AV and non-AV owners (a mix prone to appear in a transition phase to widespread automation), and operations could also rely entirely on a dynamically hired fleet like current TNCs.

Fig. 1 illustrates how a batch of requests from different SQ classes are serviced under this scenario, using a set of working and hireable vehicles. It can be seen from the output sub- Fig. 1(b) that the high-SQ class request B5 prompts a hiring operation ($H3 \rightarrow W3$) and enjoys a private ride. In contrast, the low-SQ class request L4 waits longer to be picked up and does not travel directly to its destination (the ride is shared with user S2 momentarily). The working vehicle 2 stops rebalancing to service ODs 1, 2, and 4, whereas candidate vehicle 3 is hired to service ODs 5 and 3. Since working vehicle 1 cannot access any users in time, it repositions to the origin of the hireable vehicle 3 (presumably, an undersupplied region), whereas the idle FAV 4 is dismissed. Therefore, hireable FAVs join the working fleet to help providers honoring service quality contracts by preventing delays and rejections from happening.

1.1. Contributions

Our contribution is fourfold. First, we propose a new approach to dynamically control service quality in AMoD systems, allowing providers to significantly better meet user service-level expectations. Second, we acknowledge the inherent differences between user profiles by segmenting the demand into service quality classes and formalize the requirements of these profiles using detailed SQCs, which are upheld fully on an operational level using online hiring (i.e., short-term fleet size elasticity). This setup allows providers greater leeway since they can trade off class-specific user delay tolerance and on-demand hiring. Third, we contribute to the DARP literature by modeling such SQCs through service quality constraints and show how a typical AMoD provider can benefit from enforcing service quality in a real-world scenario. Finally, to deal with large-scale real-world instances, we propose a matheuristic that creates a graph of feasible visiting plans and optimally assigns these plans to active vehicles, hierarchically minimizing vehicle hiring and user dissatisfaction.

1.2. Organization of the paper

The subsequent sections are divided as follows. Section 2 reviews the concepts encompassed by our problem formulation. Section 3 presents a multi-objective *mixed integer linear programming* (MILP) formulation for the *dial-a-ride problem with service quality contracts* (DARP-SQC) as well as a dynamic formulation for it, and describe how objectives optimized hierarchically in order of importance. Section 4 introduces a multi-objective matheuristic for the dynamic version of the problem. Then, Section 5 describes the experimental settings we use to create a series of case studies. These case studies comprise different user base compositions, service-level enforcement rates, and SQC settings for each SQ class. Next, Section 6 reports the results for both static and dynamic formulations, highlighting the tradeoffs between fulfilling user expectations and hiring new vehicles. Finally, Section 7 concludes our work and presents an outlook for future research on service quality and fleet size elasticity.

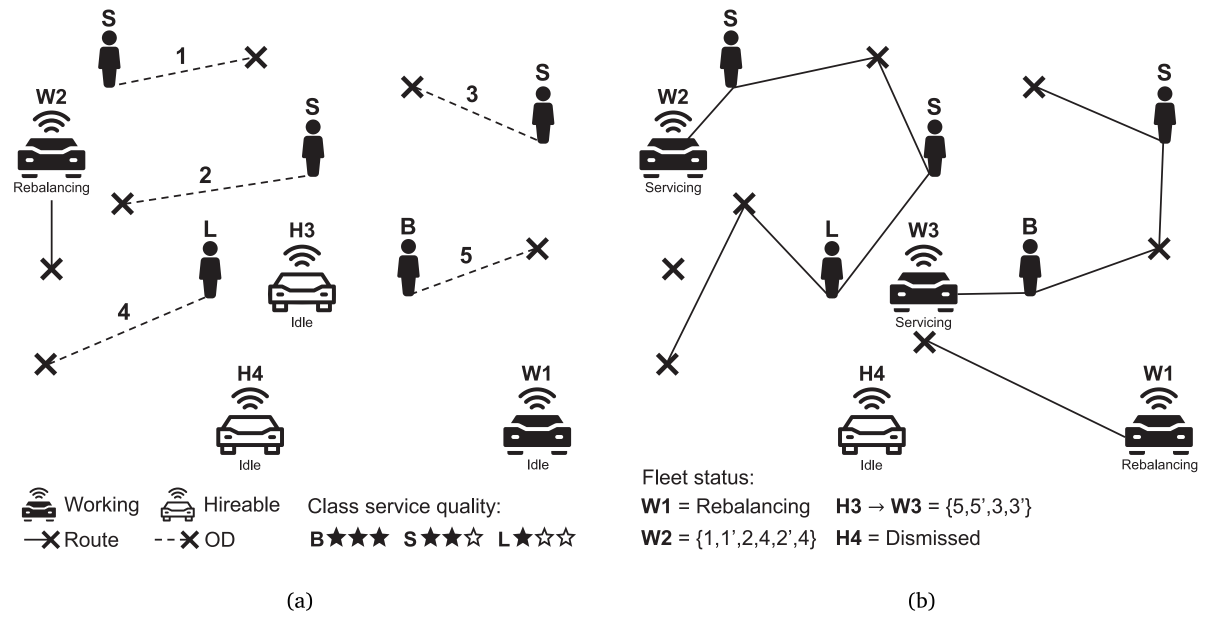

Fig. 1. (a) The user base is segmented into three classes, namely, business (B), standard (S), and low-cost (L), which have different expectations towards service quality. The fleet comprises working vehicles (third-party- or company-owned) and hireable vehicles (third-party-owned). (b) To adequately fulfill user expectations across classes, we rely on on-demand vehicle hiring, inflating the fleet size to avoid service-level violations.

2. Literature review

The ridesharing problem of pooling multiple users in a single ride has its roots in the dynamic variant of the classic *dial-a-ride problem* (DARP), introduced by Cordeau (2006). DARPs relate to the transportation of passengers or goods between specific origins and destinations at the request of users (Cordeau et al., 2007), explicitly controlling user inconvenience through service quality constraints (Berbeglia et al., 2010).

Subsequently, we review the concept of service quality across several vehicle routing studies, highlighting the characteristics that enable us to develop the AMoD system we propose.

2.1. Service levels, service rate, and fleet size

Service level, *quality of service*, and *service quality* are terms used interchangeably in DARP formulations to denote the degree to which a system can avoid excessive delays or comply with prespecified time window constraints (see Paquette et al., 2009). However, compliance with these service levels may differ. Studies occasionally allow for violations from the preferred service levels and consider that providers can make selective visits, deciding which requests to accommodate (Ho et al., 2018). This is the case for most recent AMoD studies, in which the quality of proposed matching and schedule operations is also measured regarding the *service rate* (i.e., the percentage of requests serviced).

Service levels, service rates, and fleet sizes have been consistently shown to be inextricably linked in both static- and dynamic-demand contexts. Under a static demand assumption, where users do not react to the service quality offered, studies focus on determining fleet sizes required to achieve acceptable service rates assuming target service levels. For example, Molenbruch et al. (2017) describe an approximately linear relationship between service level settings and various performance indicators and demonstrate that operational efficiency can be improved even further if heterogeneous user expectations regarding service levels are considered. Accordingly, in an analysis of the minimum fleet sizing problem, Vazifeh et al. (2018) have shown that the shorter the maximum delays between two consecutive trips, the bigger is the fleet required to keep service at a reasonable rate.

On the other hand, under a dynamic-demand assumption, fleet sizes ultimately affect the demand model once subpar service quality outcomes may cause users to opt for competing transportation modes in the long run. For example, Liu et al. (2019) show that over consecutive interactions with a multimodal system, users eventually learn to make consistent choices based on the experienced historical service quality of travel modes (viz., mobility-on-demand and public transit). Similarly, Hörl et al. (2021) computes for different fleet sizes, states where demand, cost, and waiting times are in equilibrium. Since the authors assume a fleet cost-covering model, increasing fleet size may eventually decrease the demand since higher operational costs lead to higher fares, ultimately pricing out users. In such models, therefore, the concept of service rate is more nuanced since customers ultimately refrain from using the service upon learning that their expected service levels cannot be fulfilled.

Ultimately, regardless of the type of demand context assumed, upon determining a compromise between service quality and fleet productivity, an initial fleet size is chosen to launch operations. Typically, fleet sizing is regarded as a tactical decision once adding extra vehicles is a long-term investment (Hörl et al., 2019). Consequently, as the operational environment changes, operators must periodically re-evaluate the system performance to adjust the fleet size accordingly. Failing to (promptly) do so, may ultimately inconvenience users (due to vehicle under-supply) or increase operational costs (due to vehicle over-supply).

In contrast, this study considers AMoD operators can increase and decrease fleet size at the operational level to quickly react to supply and demand imbalances. This way, as long as there are sufficient hireable vehicles, our method enables operators to consistently meet target heterogeneous user service quality expectations.

2.2. Heterogeneous users

The literature on classic routing problems provides many variants regarding user heterogeneity formulations. On *heterogeneous dial-a-ride* (HDARP) formulations, for example, a heterogeneous fleet has to service multiple user categories with different transportation requirements (Parragh, 2011). However, most categories focus on vehicle layout or equipment availability (Ho et al., 2018); service levels generally do not vary among users.

As argued by Molenbruch et al. (2017), it is plausible to assume service level requirements depend on why a trip is undertaken (e.g., scheduled appointment, leisure activity, commuting). Moreover, as shared autonomous transportation start to compete with current modes, users may adopt AMoD services that best reflect or improve their current ride experience.

By explicitly addressing heterogeneous user preferences, transportation providers can segment their user base and improve their operational efficiency and profitability by catering to each class accordingly. For instance, Wong et al. (2008) extend a model of urban taxi services in congested networks to the case of multiple user classes (low- and high-income), multiple taxi modes (normal and luxury) with distinct combinations of service area restrictions (urban or rural), and fare levels (mileage- and congestion-based). Chen et al. (2016) show that AMoD providers can find a compromise between profitability and service coverage by offering high *value of travel time* (VOTT) users, refined, work-enhancing vehicle environments at higher fares. Lokhandwala and Cai (2018) allow users to choose either private or shared rides and the extent to which they can tolerate deviations from the original trip distance. Zhang et al. (2015) also model the user's willingness to share rides and adapt their waiting tolerances from random hourly incomes. Beirigo et al. (2020), on the other hand, consider equity aspects of service quality for AMoD in areas with limited public transit access.

Addressing heterogeneous user requirements is also common outside transportation-on-demand formulations. For example, Smith et al. (2010) investigate a generalization of the *dynamic traveling repairperson problem* (DTRP) that heavily penalizes service delays of high-priority demand classes. Bulhões et al. (2018) consider a *vehicle routing problem with service levels* (VRP-SL), where a logistics provider has to balance service quality and operational costs, prioritizing different groups of requests according to their relative profitability.

Still, throughout all these studies, users either leave the system unserved or endure longer delays when these requirements cannot be fulfilled. In contrast, we consider rejections as an unacceptable breach of service quality contract, such that we seek to inflate fleet size to at least guarantee full coverage.

2.3. Heterogeneous vehicles & fleet size elasticity

Although most studies on urban mobility assume that a single service provider takes care of all requests, this centralized setting is unlikely to happen in reality since multiple providers can exist in one operation area (Ho et al., 2018). Despite optimal from a fleet operational perspective, centralized operations entail a transition from a diversified mobility market to a monopolistic market, possibly leading to less competition and, consequently, higher prices for passengers (Vazifeh et al., 2018), and different services in different zones (Beirigo et al., 2018).

Vehicle automation further diversifies the mobility market since idle privately-owned AVs may also become micro-operators. Once vehicles remain parked about 95% of the time (Shoup, 2017), incentives for individuals to simultaneously own and share their AVs when they are not in use will likely be high (Campbell, 2018). These idle hireable vehicles, which we refer to as FAVs, may be readily available during predefined time windows to join larger, centrally controlled fleets in exchange for compensation. As suggested by Hyland and Mahmassani (2017) and shown by Beirigo et al. (2021), AV fleet managers can significantly benefit from such short-term fleet size elasticity, increasing or decreasing the fleet to adequately meet the demand, by either hiring privately-owned AVs or drivers with non-AVs.

2.4. Shared autonomous vehicles systems

Following the taxonomy of Narayanan et al. (2020) concerning the operation of SAV systems, we model an on-demand booking service with dynamic vehicle assignment and mixed sharing system (both ridesharing and carsharing setups are possible).

Table 1 shows how our work fits into related literature concerning service quality, fleet characteristics, and objective function. First, the "Service quality" header comprises three columns where we identify whether users (i) can demand shared and/or private rides, (ii) can choose from heterogeneous service levels, and (iii) can have priority over others. The "Fleet" header groups four columns indicating the SAV fleet characteristics, namely vehicle homogeneity (concerning capacity), vehicle ownership (company or private), availability (permanent or time windowed), and fleet size inflation enabled by dynamic hiring. Finally, the "Objective" column summarizes the goal of each approach. Typically, works that do not aim to minimize rejections assume travelers are willing to wait indefinitely but penalize this waiting accordingly. For instance, Hyland and Mahmassani (2018) prevent travelers from going from assigned to unassigned as well as being reassigned more than once, besides penalizing the assignment to busy vehicles picking

Table 1
On-demand booking and dynamic vehicle assignment ridesharing systems classified according to service quality, fleet characteristics, and objective function. Objectives separated by "–" are linearly combined and objectives separated by ">" are optimized hierarchically.

Reference	Service quality			Fleet				Objective
	(S)hared (P)rivate rides	Het. service levels	User class priority	Het. vehicle capacity	(C)ompany (P)rivate vehicles	(P)ermanent (W)indowed availability	Dynamic hiring	
Fagnant and Kockelman (2018)	S	–	–	–	C	P	–	W
Alonso-Mora et al. (2017)	S	–	–	–	C	P	–	R - W
Fiedler et al. (2018)	S	–	–	–	C	P	–	D
Gueriau and Dusparic (2018)	S	–	–	–	C	P	–	P
Simonetto et al. (2019)	S	–	–	–	C	P	–	W
Santos and Xavier (2015)	S	–	–	–	C	P	–	R - C
Wallar et al. (2019)	S	–	–	✓	C	P	–	H - U
Hyland and Mahmassani (2018)	P	–	–	–	C	P	–	W - D - B
Gurumurthy and Kockelman (2018)	P	–	–	–	C	P	–	W
Fagnant and Kockelman (2014)	P	–	–	–	C	P	–	W
Chen et al. (2016)	P	✓	–	–	C	P	–	C
Beirigo et al. (2021)	P	✓	–	–	C, P	P, W	✓	F
Lokhandwala and Cai (2018)	S, P	✓	–	–	C	P	–	D - S
Zhang et al. (2015)	S, P	✓	–	–	C	P	–	C
This study	S, P	✓	✓	✓	C, P	P, W	✓	R > H > S > W

Objectives: W: Min. waiting, **R**: Min. rejections, **H**: Min. fleet size, **S**: Min. service level violation, **D**: Min. travel distance, **C**: Min. cost, **F**: Max. profit, **P**: Max. vehicle pickups, **U**: Max. utilization vehicle capacity, **B**: Min. assignment to busy.

up or dropping off. We refer to these penalties in the objective function as "Min. assignment to busy" since they aim to improve the service quality of users previously assigned. In contrast, when waiting is bounded, a sufficiently large fleet size is considered. In Fagnant and Kockelman (2018), for example, the minimum fleet size is defined using a "seed" day simulation in order to guarantee travelers are never rejected and do not wait for more than ten minutes. Regarding the objective function labels, when a method aims to minimize costs that depend solely on the distance traveled, we label it with "D" (i.e., min. travel distance). Conversely, if other elements are used to calculate the cost (e.g., hourly salaries, VOTT, travel times), we use the label "C".

In contrast to typical formulations, we analyze a highly diversified market scenario in which private AV owners provide freelance rides to a mobility firm in exchange for compensation. The firm is assumed to own a homogeneous fleet but occasionally hires third-party vehicles to avoid user service-level violations. To make it more realistic, we assume such a backup fleet comprises heterogeneous vehicles with different capacities. However, opposed to most HDARP formulations in which vehicle/user compatibility depends on physical characteristics, we treat seats as commodities. Any customer can be transported by any vehicle as long as service quality can be maintained.

3. Problem formulations

In this section, we introduce the mathematical model for the problem at hand considering a static context (Section 3.1) and present the changes necessary to implement the dynamic version (Section 3.2). The static formulation allows us to assess the impact of organizing operations (i.e., fleet size and mix) to meet the service-level expectations prescribed in SQCs. On the other hand, the dynamic formulation can be seen as an operational tool that allows providers to control service quality on short notice, constantly routing, re-routing, rebalancing, and hiring AVs to keep a consistent user experience even in the light of unexpected demand patterns. Since both formulations aim to minimize a multi-objective function hierarchically, Section 3.3 further explains how this process is carried out using a lexicographic method.

3.1. Static DARP-SQC

The DARP-SQC is modeled on a digraph $G = (N, E)$. The node set N is partitioned into $\{P, D, O\}$ where $P = \{1, \dots, n\}$ is both the set of pickup nodes and request indices, $D = \{n+1, \dots, 2n\}$ is the set of destination nodes, and O is the set of origins o^k of vehicles $k \in K$. We consider a free-floating fleet where vehicles do not need to depart from or come back to a central station. They can start from different origins and park nearby the delivery location of the last dropped request upon finishing the service. We let Z be the set of discretized locations on a street network, such that all nodes in N map to a single location in Z, using the function $g : N \to Z$. The shortest path between nodes i and j in N is given by $Z_{g(i),g(j)} \subseteq Z$ which we denote as $Z_{i,j}$ for simplicity. In turn, the minimum travel time to traverse path $Z_{i,j}$ is $t_{i,j}$.

Each vehicle k has capacity Q^k and is available to work on a contract during a time window $[e^k, l^k]$. For company vehicles, this window spans the entire planning horizon, whereas, for third-party vehicles, various time windows may be set. Users are segmented into service quality classes $c \in C$ such that the set of request indices P can be further partitioned as $P = \bigcup_{c \in C} P^c$. The arc set E is defined as $E = \{(i,j) \mid i \in O,\ j \in P \text{ or } i,j \in P \cup D, i \neq j \text{ and } i \neq n+j\}$. To each node $i \in N$ is associated a load q_i, corresponding to the number of passengers, such that $q_i = 0\ \forall i \in O$. Regarding the pickup and destination nodes, $q_i \geq 0\ \forall i \in P$ and $q_i = -q_{i-n}$ $\forall i \in D$, that is, the load acquired upon picking up a request has to be equally consumed in the request's destination. Nodes $i \in N$

Table 2
Sets, parameters, and variables of the DARP-SQC.

Sets	
K	Vehicles
C	Sorted SQ classes (from highest to lowest priority)
P	Pickup nodes and request indices
P^c	Pickup nodes and request indices of SQ class $c \in C$
D	Delivery nodes
O	Origin nodes o^k of vehicles $k \in K$
N	$= P \cup D \cup O$
Z	Discrete street network locations
$Z_{i,j}$	Shortest path between locations i and j
Parameters	
o^k	Origin node of vehicle $k \in K$
c_i	SQ class of request $i \in P$
Q^k	Capacity of vehicle $k \in K$
$t_{i,j}$	Travel time from node i to node j
d_i	Service duration at node $i \in N$
q_i	Number of passengers of request i
e_i	Earliest pickup time of request i
e^k	Contract start time of vehicle k
l^k	Contract end time of vehicle k
σ_c	Service-level enforcement rate of SQ class $c \in C$
α_c	Expected max. pickup delay of users in SQ class $c \in C$
β_c	Total delay tolerance of users in SQ class $c \in C$
ρ_c	(Binary) 1 if SQ class $c \in C$ does not allow ridesharing, 0 otherwise
Variables	
$x^k_{i,j}$	(Binary) 1 if vehicle k traverses arc (i, j), 0 otherwise
y_i	(Binary) 1 if user i service level is achieved, 0 otherwise
δ_i	Pickup delay of user i
Δ^k_i	In-vehicle delay of user i in vehicle k
τ^k_i	Arrival time of vehicle k at node i
ω^k_i	Load of vehicle k after visiting node i

are also associated with a service duration d_i, which may correspond to, for instance, embarking and disembarking delays, for $i \in P$ and $i \in D$, respectively. Service levels for SQ classes $c \in C$ are represented by constants α_c and β_c. The former corresponds to the expected maximum pickup delay over the revealing time e_i of request $i \in P^c$. The latter is the maximum total delay tolerated by user i. Hence, users from class c are satisfied if picked up within α_c but can tolerate up to a β_c delay, which comprises both pickup and in-vehicle delays. The binary parameter ρ_c is 1, if users from SQ class $c \in C$ demand private rides (i.e., ridesharing is disabled).

The decision variable $x^k_{i,j}$ is 1 when vehicle $k \in K$ traverses arc $(i, j) \in E$ and the load of a vehicle k upon leaving node $i \in N$ is ω^k_i. On the other hand, the decision variable y_i determines whether the service level of user $i \in P$ is achieved, setting a viable range for i's pickup delay variable δ_i. If we can meet user i service levels, that is, variable y_i is 1, then $0 \le \delta_i \le \alpha_{c_i}$. On the contrary, if y_i is 0, the minimum service-level rate σ_{c_i} of SQ class $c_i \in C$ does not cover user i. In this case, user i needs to wait longer, that is, $\alpha_{c_i} < \delta_i \le \beta_c$. Regardless of the case, the variable Δ^k_i, representing the delay of request $i \in P$ inside vehicle k, can take up all the remaining tolerated delay budget not spent while picking up i, that is, $\Delta^k_i \le \beta_{c_i} - \delta_i$. Our service quality formulation associates user satisfaction with complying with expected maximum pickup delays α_c because users are typically more inconvenienced while waiting for a vehicle (e.g., due to weather conditions, safety concerns) than while waiting inside it. Still, worse than waiting for a vehicle is having the request rejected. Hence, adopting more flexible pickup delays, controlled by service-level rates, gives the provider more leeway to fulfill upcoming requests. Finally, variable τ^k_i corresponds to the time at which vehicle k arrives at node $i \in N$. All elements of the DARP-SQC problem are summarized in Table 2.

3.1.1. Multi-objective function

Let $\mathcal{X}$ be the solution space, such that $s \in \mathcal{X}$ is given by $s = \{x^k_{i,j}, y_i, \delta_i, \Delta^k_i, \tau^k_i, \omega^k_i\}$. We consider a multi-objective function with non-commensurate objectives, ranked according to their respective priority. First, since using smaller vehicles may help to decrease external costs (e.g., by occupying less parking and road space, consuming less fuel/power, and lowering emissions), we prioritize solutions that minimize the total number of seats across all vehicles with different capacities

$$f^{\text{capacity}}(s) = \sum_{k \in K} \sum_{j \in P} Q^k x^k_{o^k,j}. \tag{1}$$

Second, to prioritize solutions featuring fewer vehicles and, ultimately, increased sharing, we aim to minimize the fleet size

$$f^{\text{fleet}}(s) = \sum_{k \in K} \sum_{j \in P} x^k_{o^k,j}. \tag{2}$$

Next, assuming classes in C are sorted in decreasing priority order (e.g., business, standard, low-cost), we aim to minimize, the sum of service level violations

$$\{f_c^{\text{violate}}(s) = \sum_{i \in P^c} (y_i - 1) \mid \forall c \in C\}, \tag{3}$$

and then the total waiting time for each class

$$\{f_c^{\text{wait}}(s) = \sum_{i \in P^c} \delta_i + \sum_{i \in P^c} \sum_{k \in K} \Delta_i^k \mid \forall c \in C\}. \tag{4}$$

Finally, the multi-objective function is given by

$$f(s) = \{f^{\text{capacity}(s)}, f^{\text{fleet}}(s),\ f_c^{\text{violate}}(s)\ \forall c \in C,\ f_c^{\text{wait}}(s)\ \forall c \in C\}. \tag{5}$$

3.1.2. MILP model

The formulation of the DARP-SQC is as follows:

Minimize:

$$f(x_{i,j}^k, y_i, \delta_i, \Delta_i^k, \tau_i^k, \omega_i^k) \tag{6}$$

Subject to:

$$\sum_{k \in K} \sum_{j \in N} x_{i,j}^k = 1 \qquad \forall i \in P \tag{7}$$

$$\sum_{i \in N} x_{i,j}^k - \sum_{i \in N} x_{i,n+j}^k = 0 \qquad \forall j \in P,\ k \in K \tag{8}$$

$$\sum_{i \in P} x_{o^k,i}^k \leq 1 \qquad \forall k \in K \tag{9}$$

$$\sum_{i \in N} x_{i,j}^k - \sum_{i \in N} x_{j,i}^k \geq 0 \qquad \forall k \in K,\ j \in P \cup D \tag{10}$$

$$e^k \leq \tau_i^k \leq l^k \qquad k \in K,\ \forall i \in N \tag{11}$$

$$\tau_j^k \geq (t_{i,j} + d_i + \tau_i^k) x_{i,j}^k \qquad \forall i, j \in N,\ k \in K \tag{12}$$

$$\Delta_i^k = \tau_{n+i}^k - (\tau_i^k + t_{i,n+i} + d_i) \qquad \forall i \in P,\ k \in K \tag{13}$$

$$\omega_j^k \geq (\omega_i^k + q_j) x_{i,j}^k \qquad \forall i, j \in N,\ k \in K \tag{14}$$

$$\max\{0, q_i\} \leq \omega_i^k \leq \min\{Q^k, Q^k + q_i\} \qquad \forall i \in N,\ k \in K \tag{15}$$

$$\sum_{i \in P^c} y_i \geq \left\lceil \sigma_c \cdot |P^c| \right\rceil \qquad \forall c \in C \tag{16}$$

$$\alpha_{c_i}(1 - y_i) \leq \delta_i \leq \alpha_{c_i} y_i + \beta_{c_i}(1 - y_i) \qquad \forall i \in P \tag{17}$$

$$e_i \leq \tau_i^k \leq e_i + \delta_i \qquad \forall i \in P,\ k \in K \tag{18}$$

$$\Delta_i^k + \delta_i \leq \beta_{c_i} \qquad \forall i \in P,\ k \in K \tag{19}$$

$$\sum_{k \in K} \omega_i^k = q_i \qquad \forall i \in P : \rho_{c_i} = 1 \tag{20}$$

$$\sum_{k \in K} x_{i,n+i}^k = 1 \qquad \forall i \in P : \rho_{c_i} = 1 \tag{21}$$

$$x_{i,j}^k \in \{0, 1\} \qquad \forall k \in K,\ i, j \in N \tag{22}$$

$$y_i \in \{0, 1\},\ \delta_i \in \mathbb{N} \qquad \forall i \in P \tag{23}$$

$$\Delta_i^k \in \mathbb{N} \qquad \forall k \in K, \forall i \in P \tag{24}$$

$$\tau_i^k,\ \omega_i^k \in \mathbb{N} \qquad \forall k \in K,\ \forall i \in N \tag{25}$$

The aim of the hierarchical multi-objective function (6) is first to determine the minimum fleet size and vehicle mix that satisfies the demand and then to improve class service levels. Constraints (7) and (8) ensure that all users are serviced exactly once and that the same vehicle visits their origin and destination nodes. Constraints (9) and (10) guarantee that every scheduled vehicle k departs from its origin o^k and stops at the delivery node of its last request. Constraints (11) impose that vehicles can only pick up and deliver users within their working time window. The consistency of arrival and ride times is ensured by constraints (12) and (13), whereas the consistency of load variables is ensured by inequalities (14) and (15). Constraints (16)–(21) implement the user SQCs. First, constraints (16) enforce that the service-level expectations of a minimum share of users from each class are met. Next, constraints (17), (18), and (19) ensure that service levels are consistent with maximum pickup and in-vehicle delays. Then, equalities (20) and

Table 3
Sets, parameters, and variables of the dynamic formulation.

Sets	
Z^*	Regional centers of street network locations Z
K^{PAV}	Company vehicles
K_t^{FAV}	Hireable vehicles at time t
K_t^{H}	Vehicles hired at time t
K_t^{P}	Parked vehicles at time t
$\mathcal{P}^k$	Passengers (picked up requests) of vehicle k
$\mathcal{R}^k$	Assigned requests (non picked up) of vehicle k
$\mathcal{S}^k$	Node itinerary of vehicle k
v^k	Visiting plan $\{\mathcal{P}^k, \mathcal{R}^k, \mathcal{S}^k\}$ of vehicle k
$\mathcal{V}_{\text{R}}^k$	Candidate visiting plans where k is in the *rebalancing* state
$\mathcal{V}_{\text{S}}^k$	Candidate visiting plans where k is in the *servicing* state
v_{idle}^k	Candidate visiting plan where k is in the *idle* state
$\mathcal{V}^k$	Feasible visiting plans $\mathcal{V}_{\text{S}}^k \cup \mathcal{V}_{\text{R}}^k \cup \{v_{\text{idle}}^k\}$ for vehicle k
$\mathcal{V}$	Visiting plans $\bigcup_{k\in K} \mathcal{V}^k$ in ERTV graph
$\mathcal{V}_i$	Visiting plans in $\mathcal{V}$ including request i
$\mathcal{V}_i^{\text{SL}}$	Visiting plans in $\mathcal{V}$ that meet request i target SL
P^A	Requests previously assigned to vehicles
B_t	Request batch placed in period t
P_t	$= B_t \cup P^A$
P_t^c	Requests in P_t of class $c \in C$
O_t^*	Origins $o^k \in Z^*$ of hired vehicles K_t^{H}
P_t^{U}	Pickup nodes of service-level violated requests at time t
J_t	Rebalancing targets $O_t^* \cup P_t^{\text{U}}$
Parameters	
γ_{hire}	Hiring penalty
γ_{sl}	Service-level violation penalty
γ_{reject}	Rejection penalty
κ	Round duration
l^k	Contract deadline of vehicle k
T	Total time horizon
δ_{max}	Maximal hiring delay
Variables	
h^k	(Binary) 1 if vehicle $k \in K_t^{\text{FAV}}$ is hired, 0 otherwise
x_v	(Binary) 1 if visiting plan v is chosen, 0 otherwise
y_i	(Binary) 1 if service level of request i is achieved, 0 otherwise
z_i	(Binary) 1 if request i is rejected, 0 otherwise
δ_{iv}	Pickup delay of user i in visiting plan v
Δ_v	Delay sum of all requests in visiting plan v
Δ_v^c	Delay sum of class c requests in visiting plan v
a_c	Number of service-level violations in class c

(21) ensure that users whose class entails a private ride are picked up by empty vehicles and travel directly to their destination. Finally, constraints (22)–(25) declare the variables.

The quadratic constraints (12) and (14) are linearized as in Cordeau et al. (2007):

$$\tau_j^k - \tau_i^k \geq t_{i,j} + d_i - M_{i,j}^k(1 - x_{i,j}^k) \qquad \forall i, j \in N,\ k \in K \tag{26}$$

$$\omega_j^k - \omega_i^k \geq q_j - W_{i,j}^k(1 - x_{i,j}^k) \qquad \forall i, j \in N,\ k \in K \tag{27}$$

The validity of the Big-M constants W and M in (26) and (27) is ensured by setting $W_{i,j}^k \geq \min\{2Q^k, 2Q^k + q_i\}$ and $M_{i,j}^k \geq \max\{0, l_i + t_{i,j} + d_i - e_j\}$ $\forall k \in K$ and $i, j \in N$.

3.2. Dynamic DARP-SQC

In order to deal with real-world instances, we break apart the time horizon into κ-second rounds $t \in \{1, 2, \ldots, T\}$. At each round t, we process a request batch indexed by P_t, which is comprised of requests accumulated throughout the time slot $[\kappa \cdot (t-1),\ \kappa \cdot t)$. Hence, passengers wait at most κ seconds to have their requests handled by the system. If rejected, we assume they walk away or may opt to re-enter the system in the next round, possibly choosing a different service quality class. Additionally, we search for available freelance vehicles parked throughout locations in Z to construct the set of backup FAVs $K_t^{\text{FAV}} \subseteq K$, which can privately service all requests in P_t.

In the following, we further detail the elements we use to expand the static DARP-SQC problem definition presented in Section 3.1 to accommodate dynamic execution. These elements are summarized in Table 3.

Requests. A request r_i is defined by a tuple $\{i, i', \tau_i^k, \tau_{i'}^k, e_i, e_{i'}, q_i, c_i\}$, which consists of origin $i \in P$, destination $i' \in D$, with respective expected arrival times τ_i^k and $\tau_{i'}^k$, placement time e_i, minimum delivery time $e_{i'} = e_i + t_{i,i'}$, load q_i, and service level class c_i.

Vehicles. We assume all vehicles k are associated to visiting plans $v^k = \{\mathcal{P}^k, \mathcal{R}^k, \mathcal{S}^k\}$. Sets $\mathcal{P}^k$ and $\mathcal{R}^k$ are comprised of passengers (i.e., picked up requests) and assigned requests, respectively. In turn, $\mathcal{S}^k = \{\mathcal{S}_0^k, \mathcal{S}_1^k, \ldots, \mathcal{S}_m^k\}$ is a sequence of nodes representing the vehicle's itinerary, such that $\mathcal{S}^k \subseteq N$. While $\mathcal{S}_0^k$ is vehicle k's last visited node, $\mathcal{S}_1^k, \ldots, \mathcal{S}_m^k$ are the subsequent m nodes to visit. Hence, vehicle k's current load after visiting $\mathcal{S}_0^k$ in the interval $[\tau^k_{\mathcal{S}_0^k}, \tau^k_{\mathcal{S}_0^k} + d_{\mathcal{S}_0^k})$ is given by

$$\omega^k_{\mathcal{S}_0^k} = \sum_{r_i \in \mathcal{P}^k} q_i,$$

that is, the sum of the loads q_i of each request r_i in the passenger set $\mathcal{P}^k$.

Visiting plan feasibility. A valid visiting plan v^k from vehicle k consists of a sequence of trip pairs $(i, j) \in \{(\mathcal{S}_0^k, \mathcal{S}_1^k), (\mathcal{S}_1^k, \mathcal{S}_2^k), \ldots, (\mathcal{S}_{m-1}^k, \mathcal{S}_m^k)\}$ that represent a feasible solution with respect to the following constraints:

(C1) $e_j \le \tau_i^k + d_i + t_{i,j} \le \tau_j^k$ (arrival consistency)
(C2) $\omega_i^k + q_j \le Q^k$ (load consistency)
(C3) $i \in P \wedge \rho_{c_i} = 1 \rightarrow j = i'$ and $j \in P \wedge \rho_{c_j} = 1 \rightarrow \omega_i^k = 0$ (privacy requirement)
C4) $\tau_i^k \le l^k$ (vehicle contract deadline)

It is worth noting that when a request is assigned to a vehicle, we assume that it can still be further delayed or even picked up by different vehicles but never rejected. Such flexibility allows a higher number of feasible rides to arise at each period, which ultimately favors the inclusion of high-priority users by disrupting previous visiting plans.

Vehicle states and visiting plan types. At each round, a vehicle k can execute three actions, namely, (i) move to pick up or deliver requests, (ii) stay idle at its current position, and (iii) move empty to another location. Accordingly, we consider that vehicles k executing (i), (ii), or (iii), are in states *servicing*, *idle*, or *rebalancing*, respectively. The best plan is chosen from a set of feasible candidates $\mathcal{V}^k$, which is partitioned based on the possible vehicle states using

$\mathcal{V}_\mathrm{S}^k$ Set of candidate visiting plans where vehicle is servicing users, that is, $|\mathcal{P}^k| > 0$ or $|\mathcal{R}^k| > 0$,

v_idle^k Candidate visiting plan where vehicle is idle, that is, $\mathcal{R}^k = \mathcal{P}^k = \emptyset$ and $|\mathcal{S}^k| = 1$, such that $\mathcal{S}_0^k$ is a parking place,

$\mathcal{V}_\mathrm{R}^k$ Set of candidate visiting plans where vehicle is rebalancing, that is, $\mathcal{R}^k = \mathcal{P}^k = \emptyset$ and $|\mathcal{S}^k| = 2$, such that $\mathcal{S}_0^k$ is a parking place and $\mathcal{S}_1^k$ is a rebalance target.

Updating vehicle progress. Visiting plans are updated using the function `updateProgress`k, t, where the time period t is used to compute the progress of vehicle k throughout the itinerary $\mathcal{S}^k$ and update sets $\mathcal{R}^k$ and $\mathcal{P}^k$. First, regardless of node type, $\forall i \in \mathcal{S}^k$, if $\tau_i^k \ge e_i$, $\mathcal{S}^k = \mathcal{S}^k \setminus \{i\}$. If i is a pickup node, then $\mathcal{R}^k = \mathcal{R}^k \setminus \{r_i\}$ and $\mathcal{P}^k = \mathcal{P}^k \cup \{r_i\}$, whereas if i is a delivery node, then $\mathcal{P}^k = \mathcal{P}^k \setminus \{r_i\}$.

Optimal itineraries. Based on the passengers $\mathcal{P}^k$ of vehicle k and a candidate request set $\mathcal{R}$, there can be several feasible $\mathcal{S}$ itineraries created using nodes from $\mathcal{P}^k \cup \mathcal{R}$ that fulfill constraints C1, C2, C3, and C4. These itineraries compose candidate visiting plans $v = \{\mathcal{P}^k, \mathcal{R}, \mathcal{S}\}$ in the set $\mathcal{V}_\mathrm{S}^k$. The total delay associated with an itinerary $\mathcal{S}$ of visiting plan v consists of the sum of the arrival delays at destination nodes, which is given by

$$\Delta_v = \sum_{i' \in \mathcal{S} \cap D} \tau_{i'}^k - e_{i'}.$$

Using delays Δ_v, we can determine the visiting plan featuring the global minimum total delay

$$v_g^k = \arg\min_{v \in \mathcal{V}_\mathrm{S}^k} \Delta_v. \tag{28}$$

From a SQ class perspective, the total delay of requests from class c consists of the sum of delays to reach these requests' destinations, which is given by

$$\Delta_v^c = \sum_{\substack{i' \in \mathcal{S} \cap D \\ c_i = c}} \tau_{i'}^k - e_{i'}.$$

Likewise, using delays Δ_v^c, we can determine the visiting plan featuring the minimum total delay for class c

$$v_c^k = \arg\min_{v \in \mathcal{V}_\mathrm{S}^k} \Delta_v^c. \tag{29}$$

Hiring. We outsource requests to privately-owned vehicles (FAVs) whenever the current fleet can no longer meet the minimum service level requirements of SQ classes. For each period t, we search regional centers for available FAVs that can back up the PAV fleet if it cannot service a request batch P_t adequately. We assume that such freelance vehicles are readily available to join the fleet in exchange for compensation, and their supply is sufficient to match the overall excess demand. We collect these vehicles in set $K_t^{\text{FAV}} = \{\texttt{closestFAVToRequest}i, Z* \mid \forall i \in P_t\}$, where $\texttt{closestFAVToRequest}i, Z*$ is a function that returns the vehicle k parked at regional center location in $Z^* \subseteq Z$ that can pick up request i.

To determine regional centers in Z^*, we implement a variant of the facility-location problem proposed by Toregas et al. (1971). Our formulation aims to determine the minimum set of facilities that together can cover all other locations in Z within δ_{max} time units. For instance, a long δ_{max} results in fewer regional centers, increasing the pickup time of requests and, consequently, service-level violations. Conversely, a short δ_{max} leads to many regional centers, allowing the hireable fleet to access users faster. In this study, we refer to δ_{max} as the ***maximal hiring delay*** and adjust it such that hired vehicles can pick requests within their expected service level, regardless of class.

All vehicles $k \in K_t^{\text{FAV}}$ have contract deadlines $min\{l^k, T\}$, such that k is dismissed at a later time t' as long as $t' >= l^k$. Finally, to guarantee PAV availability, we set $l^k = T$ for all vehicles $k \in K^{\text{PAV}}$. Additionally, we consider hired vehicles to have precisely the capacity of the number of passengers determined by the request they are supposed to service. Therefore, throughout the simulation, the working fleet dynamically changes from homogeneous to heterogeneous, especially at high-demand times.

Decision variables. Decision variables are grouped in solution tuples $s = \{x_v, y_i, z_i, h^k\}$ with $x_v, y_i, z_i, h^k \in \{0, 1\}$ such that the solution space is defined using

$$\mathcal{X} = \{x_v, y_i, z_i, h^k \mid \forall v \in \mathcal{V}^k,\ \forall k \in K \ \text{ and } \forall i \in P_t\}.$$

Variable $x_v = 1$ when the visiting plan $v \in \mathcal{V}^k$ from vehicle $k \in K$ is chosen, and variables y_i and z_i help to distinguish how user i is serviced. If $y_i = 1$, the system could meet request i class service levels, whereas if $z_i = 1$ request i is rejected. Finally, for vehicles $k \in K_t^{\text{FAV}}$, $h^k = 1$ signals that k was hired to carry out a visiting plan at the current period t.

3.2.1. *Multi-objective function*

For the dynamic version of our problem, we also propose a multi-objective function. First, we aim to minimize, in turn, the sum of rejection penalties γ_{reject} given by the ordered list of objectives

$$\{f_c^{\text{reject}}(s) = \sum_{i \in P^c} \gamma_{\text{reject}} \cdot z_i \mid \forall c \in C\}. \tag{30}$$

After preventing rejections, we aim to hire the least number of vehicles necessary to meet user service levels, such that we aim to minimize

$$f^{\text{hire}}(s) = \sum_{k \in K_t^{\text{FAV}}} Q^k h^k. \tag{31}$$

Next, we seek to minimize the sum of violation penalties γ_{sl} in the ordered list of objectives

$$\{f_c^{\text{violate}}(s) = \sum_{i \in P^c} \gamma_{\text{sl}} \cdot (y_i - 1) \mid \forall c \in C\}. \tag{32}$$

Then, we consider the minimization of waiting times across classes:

$$\{f_c^{\text{wait}}(s) = \sum_{v \in \mathcal{V}^k} \Delta_v^c \cdot x_v \mid \forall c \in C\}. \tag{33}$$

Finally, we define the multi-objective function of our dynamic formulation using

$$f(s) = \{f_c^{\text{reject}}(s)\ \forall c \in C,\ f^{\text{hire}}(s),\ f_c^{\text{violate}}(s)\ \forall c \in C,\ f_c^{\text{wait}}(s)\ \forall c \in C\}. \tag{34}$$

According to each class priority, the order of the objectives guarantees that visiting plans are set up first to avoid rejections. In the next steps, vehicles are hired only to prevent service level violations, which are minimized subsequently also following the order of C. Once optimal service level distribution is guaranteed, the solution featuring the minimum waiting time is chosen.

3.3. *Lexicographic method for multi-objective optimization*

We use the lexicographic method to determine the optimal solution for the multi-objective function $f(s)$ (for both static and dynamic models). This method consists of solving, in decreasing priority order, a sequence of single-objective problems

$$\text{Minimize} \quad \{f_r(s) \mid f_l(s) \leq \psi_l,\ l = 1, \ldots, r-1,\ s \in \mathcal{X}\},$$

where $r \in \{1, 2, \ldots, |f|\}$ is the index of the current objective. A solution s is feasible only if it does not degrade the optimal values ψ_l found for previous objectives f_l with higher priority.

Although we adopt a hierarchical optimization approach, it is worth noting that service quality objectives such as f^{violate} and f^{wait} are good candidates for linear combination. By deliberately weighing the tradeoffs, providers could establish a utility function that best represents their efforts to cater to users across classes.

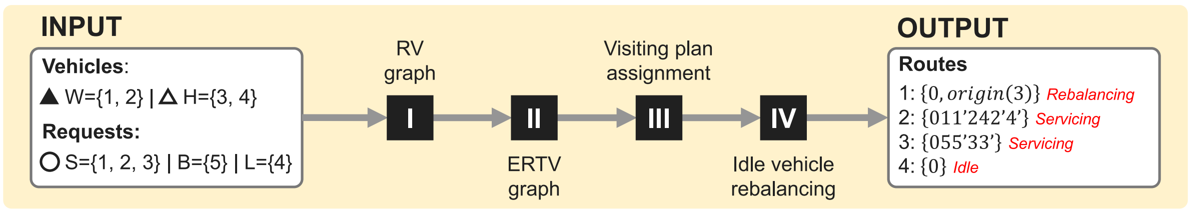

Fig. 2. Method overview for the input example presented in Fig. 1(a). Input is processed into a RV graph (Fig. 3) where initial ridesharing opportunities are identified. Then, the RV graph is used to build an ERTV -graph (Fig. 4), which connects all vehicles and requests through feasible visiting plans. Next, these plans are optimally assigned to available vehicles (Fig. 5). Finally, remaining idle vehicles are optimally rebalanced to underserved areas (Fig. 6). The output consists of updated vehicles plans, which are illustrated in Fig. 1(b).

4. A matheuristic for the dynamic DARP-SQC

The following sections describe how we build on the state-of-the-art method proposed by Alonso-Mora et al. (2017) to solve the problem's dynamic version. The method was chosen because (i) its conventional formulation offers optimality guarantees and (ii) it is flexible to accommodate the contributions put forward in this study, namely, service level constraints and hiring capabilities.

First, request demand and vehicle supply information are combined into a pairwise-shareability graph (Section 4.1), which is subsequently used to iteratively compute a graph of feasible visiting plans (Section 4.2). Next, visiting plans are optimally assigned to vehicles using a multi-objective MILP formulation whose goal is to minimize user dissatisfaction hierarchically (according to the importance of each class) while hiring the least number of vehicles (Section 4.3). Then, a MILP formulation is used to optimally rebalance idle vehicles to underserved areas, where either service level violations (i.e., extra delays or rejections) or vehicle hirings occurred (Section 4.4).

The necessary steps to calculate a complete solution are described in Algorithm 1 and summarized in Fig. 2, which provides an overview of our method using as an example the input presented in Fig. 1(a). Since the construction steps I and II (where feasible visiting plans are determined) are followed by assignment steps III and IV (where vehicles are optimally assigned to visiting plans), the complete strategy consists of a matheuristic, that is, a heuristic that incorporates phases where mathematical programming models are solved (Archetti and Speranza, 2014).

Algorithm 1: Dynamic DARP-SQC matheuristic

```
Input: Request index batches $P = \{B_1, B_2, B_3, \dots, B_T\}$ for total horizon $T$, service level classes $C$, initial fleet $K^{\mathrm{PAV}}$ randomly distributed throughout
       street network locations in $Z$, and regional centers $Z^*$.
1  $K = K^{\mathrm{PAV}}$
2  for $t = 1, 2, \dots, T$ do
3      updateProgress($k$,$t$), $\forall k \in K$
4      $K = K \setminus \{k\}$ if $t \geq l^k$ $\forall k \in K$                    //Remove vehicles whose contracts expired
5      $P^A = \{i \mid \forall r_i \in \bigcup_{k \in K} \mathcal{R}^k\}$                    //Build set of assigned requests
6      $P_t = B_t \cup P^A$                                        //Build set of requests in progress
7      $K_t^{\mathrm{FAV}} = \{$closestFAVToRequest($i$, $Z^*$) $\mid \forall i \in P_t\}$
8      $K = K \cup K_t^{\mathrm{FAV}}$
9      Build RV graph using vehicles $K$ and requests $P_t$ (Section 4.1)
10     From RV graph, build the ERTV graph with feasible visiting plans $\mathcal{V}$ (Section 4.2)
11     (MILP) Match a visiting plan in $\mathcal{V}$ to each vehicle in $\mathcal{K}$ (Section 4.3)
12     $K = K \setminus \{k\}$ if $h^k = 0$ $\forall k \in K_t^{\mathrm{FAV}}$                          //Remove unhired FAVs
13     $O_t^* = \{o^k \mid h^k = 1$ $\forall k \in K_t^{\mathrm{FAV}}\}$                          //Origins of hired vehicles
14     $P_t^{\mathrm{U}} = \{i \mid y_i = 0,\ r_i \in P_t\}$                          //Origins of SL violated requests
15     $J_t = \{i \mid \forall\ i \in O_t^* \cup P_t^{\mathrm{U}}\}$                                  //Rebalancing targets
16     $K_t^{\mathrm{P}} = \{k \mid$ if $v^k = v_{\mathrm{idle}}^k \forall k \in K\}$                                //Idle vehicles
17     Create rebalancing visiting plans $\mathcal{V}_{\mathrm{R}}^k$ using $K_t^{\mathrm{P}}$ and $J_t$
18     (MILP) Match rebalancing visiting plans to vehicles (Section 4.4)
19     Implement all visiting plans $v \in \mathcal{V}^k$ if $x_v = 1$ $\forall k \in K$
```

4.1. Pairwise request–vehicle graph

The pairwise *request–vehicle* (RV) graph is based on the idea of shareability graphs proposed by Santi et al. (2014). In the RV graph, two requests are connected only if they can be serviced by an empty virtual vehicle starting at the origin of one of them. Likewise, a vehicle is connected to a request if the request can be serviced by the vehicle.

Since we consider previously assigned requests can still be picked up by different vehicles, these requests also integrate the RV graph construction. At each time period t, after updating all vehicle tuples according to the current period t as well as dismissing hired vehicles with expired contract deadlines, we define the set of assigned requests $P^A = \{i \mid \forall r_i \bigcup_{k \in K} \mathcal{R}^k\}$, and the set of all requests $P_t = B_t \cup P^A$. Then, we use the total fleet K and requests P_t to build the RV graph.

Fig. 3 presents the RV graph created from the problem input described in Fig. 1(a). The graph features six request–request (RR) pairs and seven request–vehicle (RV) pairs.

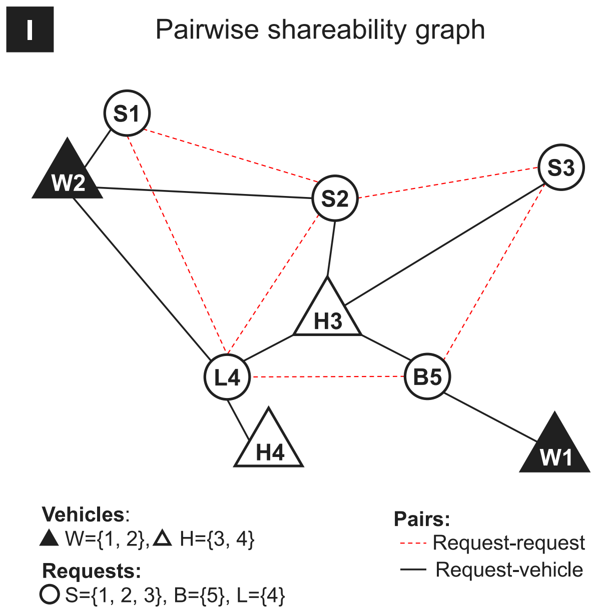

Fig. 3. RV graph resulting from the problem input presented in Fig. 1(a).

4.2. Extended request–trip–vehicle graph

Similarly to Alonso-Mora et al. (2017), we use the RV graph to compute the *request–trip–vehicle* (RTV) graph, where request and vehicle nodes are connected to trip nodes, which are comprised of request sets. Feasible trips featuring $q \in \{1,2\}$ request derive directly from the RV graph, whereas trips involving $q \geq 3$ requests are feasible only if all possible subtrips with $q-1$ requests are also feasible. By carrying out this sub-feasibility assessment process iteratively, increasingly higher q-trip combinations can be evaluated up to the capacity of vehicle k, that is, $q \in \{1,2,\dots,Q^k\}$.

In contrast to the original RTV graph formulation, where trip nodes consist of a request set that refers to a unique minimum waiting visiting plan, we propose visiting plan nodes (v-nodes). We consider a single request set $\mathcal{R}$ can lead to up to $|C|+1$ visiting plans. Besides the minimum total delay plan v_g^k, we also consider the minimum total delay plans v_c^k for each class $c \in \{c_i \mid r_i \in \mathcal{R}\}$. Adding extra visiting plans focusing on SQ classes' total delay minimization adds flexibility to the assignment algorithm once it can consider a larger pool of itineraries to balance service level provision between classes.

We refer to this expanded version, where request sets generate several visiting plans, as the *extended request–trip–vehicle* (ERTV) graph. Hence, whenever a vehicle k can successfully service a request set $\mathcal{R}$, instead of adding edges $e(r_i,\mathcal{R})\ \forall r_i \in \mathcal{R}$ and $e(\mathcal{R},k)$, we add edges where $\mathcal{R}$ is replaced by visiting plans v, which are guaranteed to be valid with respect to feasibility constraints C1, C2, C3, and C4.

Once these visiting plans entail a one-to-one relationship to both vehicles and requests, we can attribute weights to edges. As a result, a feasible visiting plan v leads to edges $e(r_i,v,\delta_{iv})\ \forall r_i \in \mathcal{R}$ and $e(v,k,\Delta_v)$, where δ_{iv} is the delay to pick up request r_i through sequence $\mathcal{S}$ and Δ_v is the total waiting time to realize $\mathcal{S}$.

We employ an exhaustive search over all feasible itineraries created from $\mathcal{R}$ to find the visiting plans v_g^k and v_c^k. However, it is worth noting that we are able to do so because we consider low-capacity vehicles with at most four seats. In order to consider high-capacitated vehicles, local improvement methods or exploration heuristics might be necessary to curb computation time.

Additionally, to ensure every vehicle will be assigned to a visiting plan, we add two extra edges. For parked vehicles, we add edge $e(k,v_{\text{idle}}^k,0)$ to guarantee they can stay parked. Consequently, if v_{idle}^k is assigned to $k \in K_t^{\text{FAV}}$, k did not need to be hired. For unloaded vehicles cruising to pick up users (i.e., $\mathcal{P}^k = \emptyset$ and $|\mathcal{R}^k| \geq 1$), we add edge $e(k,v_{\text{stop}}^k,0)$, where $v_{\text{stop}}^k \in \mathcal{V}_{\text{R}}^k$ is a special rebalance case, where $\mathcal{R}^k$ is emptied and the vehicle parks at a nearby location in Z. Both v_{idle}^k and v_{stop}^k have zero-valued weights because we aim to avoid unnecessary trips, such that vehicles remain still whenever possible. Vehicles can also continue to carry out their incumbent visiting plan, which is naturally added to the ERTV graph during the construction phase. Algorithm 2 presents all the construction steps of the ERTV graph.

Finally, upon adding edges that link requests i to visiting plans v, we populate the set $\mathcal{V}_i$ of plans where i can be serviced. Then, we define set $\mathcal{V}_i^{\text{SL}} \subseteq \mathcal{V}_i$ where $\delta_{iv} \leq \alpha_{c_i}$, which consists of the set of plans where request i desired service levels are satisfied.

Fig. 4 displays a portion of the ERTV graph based on the pairwise RV graph. It shows how requests $P_t = \{1,2,4,5\}$ from period t are associated to vehicles 1, 2, and 4 through visiting plans $\mathcal{V}^1 = \{v_{17}, v_{18}\}$, $\mathcal{V}^2 = \{v_0, v_1, \dots, v_{14}\}$, and $\mathcal{V}^4 = \{v_{15}, v_{16}\}$, respectively. It is worth noting that vehicles 1, 2, and 4 include their incumbent visiting plans, namely, v_{18}, v_7, and v_{16}. This way, whenever appropriate, vehicles can continue carrying out these plans. Moreover, for vehicle 2, the plans v_{10}, v_{11}, and v_{12} stem from the same

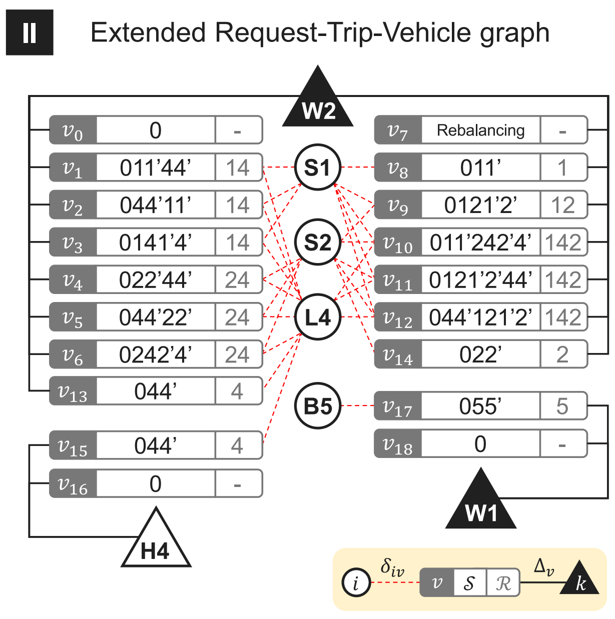

Fig. 4. ERTV graph created from the RV graph showed in Fig. 3. For brevity, we omit the elements concerning vehicle H3. Edges connecting requests i to visiting plans v weight requests' pickup delays δ_{iv}, whereas edges connecting vehicles to visiting plans weight plans' total delays Δ_v (i.e., sum of all users' pickup and in-vehicle delays).

Algorithm 2: ERTV graph construction

Input: RV graph from period t with request–request (RR) edges $e(r_i, r_j)$ with $i, j \in P_t$ and request–vehicle (RV) edges $e(r_i, k)$ with $i \in P_t$ and $k \in K$.
Output: ERTV graph with request and v-node edges $e(r_i, v, \delta_{iv})$ and v-node and vehicle edges $e(v, k, \Delta_v)$.

```
1  Function getVisitingPlans(k, $\mathcal{R}$)
2      $V = \emptyset$
3      $V \leftarrow v^k_g$                                          //Min. total delay (28)
4      for $c \in \{c_i \mid r_i \in \mathcal{R}\}$ do
5          $V \leftarrow v^k_c$                                      //Min. total delay for class $c$ (29)
6      return $V$
7  Function addVisitingPlansToERTV($V$)
8      Add request and v-node edge $e(r_i, v, \delta_{iv})\ \forall r_i \in \mathcal{R},\ \forall v \in V$
9      Add v-node and vehicle edge $e(v, k, \Delta_v)\ \forall v \in V$
10 Function addFeasiblePlansFromRequests($R^k_q$, $k$, $R'$)
11     for $\mathcal{R} \in R'$ do
12         $V$ = getVisitingPlans($k$, $\mathcal{R}$)
13         if $V \neq \emptyset$ then
14             $R^k_q \leftarrow \mathcal{R}$
15             addVisitingPlansToERTV($V$)
16 begin
17     for $k \in K$ do
18         addVisitingPlansToERTV($\{v^k_{\text{stop}}, v^k_{\text{idle}}\}$)
19         $R^k_q = \emptyset\ \forall q \in \{1, 2, \dots, Q^k\}$
20         $R' = \{\{r_i\} \mid \forall e(r_i, k) \in \text{RV graph}\}$              //Candidate one-request trips
21         addFeasiblePlansFromRequests($R^k_1$, $k$, $R'$)
22         $R' = \{\{r_i, r_j\} \mid \forall r_i, r_j \in R^k_1\}$                  //Candidate two-request trips
23         $R' = \{\mathcal{R} \mid \forall \mathcal{R} \in R' \wedge e(\mathcal{R}_1, \mathcal{R}_2) \in \text{RV graph}\}$    //Filter unfeasible
24         addFeasiblePlansFromRequests($R^k_2$, $k$, $R'$)
25         for $q \in \{3, \dots, Q^k\}$ do
26             $R' = \{\mathcal{R}_i \cup \mathcal{R}_j \mid \forall \mathcal{R}_i, \mathcal{R}_j \in R^k_{q-1}\}$
27             $R' = \{\mathcal{R} \mid \forall \mathcal{R} \in R' \wedge |\mathcal{R}| = q\}$       //Select candidate q-request trips
28             $R' = \{\mathcal{R} \mid \forall \mathcal{R} \in R' \wedge \forall r_i \in \mathcal{R},\ \mathcal{R} \setminus r_i \in R^k_{q-1}\}$   //Filter candidate unfeasible trips
29             addFeasiblePlansFromRequests($R^k_q$, $k$, $R'$)
```

Table 4
Feasible request combinations for each vehicle throughout the iterations of Algorithm 2. Valid combinations are created based on the RV graph showed in Fig. 3 considering four-seat vehicles, totaling 135 pickup and delivery combinations.

	N. of requests (N. of possible visiting plans)				
Vehicle	1(1)	2(6)	3(90)	4(2,520)	Total
W1	$B5$		–	–	1
W2	$S1$	S1,S2	–	–	7
	$S2$	S2,L4	–	–	7
	$L4$	S1,L4	S1,S2,L4	–	97
H3	$S2$	S2,S3	–	–	7
	$S3$	S3,B5	–	–	4[a]
	$L4$	S2,L4	–	–	7
	$B5$	L4,B5	–	–	4[a]
H4	$L4$	L4	–	–	1

[a]No ridesharing in itineraries featuring business users.

request set $\mathcal{R} = \{r_1, r_4, r_2\}$. Since $\mathcal{R}$ features two different SQ classes, v_{10} is the visiting plan achieved seeking to minimize the total delay, whereas v_{11} and v_{12} are the visiting plans achieved when the goal is to minimize the delays of standard and low-cost classes. A similar outcome can be found for plans v_4, v_5, and v_6, which all stem from request set $\mathcal{R} = \{r_2, r_4\}$. Table 4 presents the number of pickup and delivery permutations evaluated at each iteration of the Algorithm 2, considering vehicles can carry up to four passengers. Since some permutations might violate requests' service-level requirements, only a subset of them ends up becoming valid visiting plans.

4.3. Visiting plan assignment formulation

Upon creating period t's ERTV graph embedding feasible visiting plans $\mathcal{V} = \bigcup_{k\in K} \mathcal{V}^k$, we formulate an assignment problem, formalized using the following MILP model:

Minimize:

$$f(x_v, y_i, z_i, h^k)$$

Subject to:

$$\sum_{v\in\mathcal{V}^k} x_v = 1 \qquad \forall k \in K \tag{35}$$

$$z_i = 1 - \sum_{v\in\mathcal{V}_i} x_v \qquad \forall i \in B_t \tag{36}$$

$$\sum_{v\in\mathcal{V}_i} x_v = 1 \qquad \forall i \in P^A \tag{37}$$

$$h^k = \sum_{v\in\mathcal{V}_{\mathrm{S}}^k} x_v \qquad \forall k \in K_t^{\mathrm{FAV}} \tag{38}$$

$$y_i = \sum_{v\in\mathcal{V}_i^{\mathrm{SL}}} x_v \qquad \forall i \in P_t \tag{39}$$

$$\sum_{i\in P_t^c} y_i \geq \left\lceil \sigma_c \cdot |P_t^c| \right\rceil \qquad \forall c \in C \tag{40}$$

The optimization problem aims at minimizing the multi-objective function (34) in order to find the least number of hirings necessary to meet user service level expectations throughout SQ classes. Constraints (35) guarantee that each vehicle realizes a single visiting plan. In turn, constraints (36) ensure that every unassigned user is either assigned or rejected, whereas (37) guarantee that previously assigned requests continue so but allows the original plan to be changed (other vehicles can carry out the service). Constraints (38), guarantee the consistency of the hiring process, such that $h^k = 1$ whenever a visiting plan from $\mathcal{V}_{\mathrm{S}}^k$ of vehicle $k \in K_t^{\mathrm{FAV}}$ is chosen. Constraints (39) guarantee the consistency of each variable y_i, which is 1 only when the expected service level of request i is fulfilled. Finally, analogously to the static formulation, constraints (40) enforce that the service-level expectations of a minimum share of users from each class are met.

Fig. 5 illustrates the assignment step for the input example presented in Fig. 1(a). The feasible visiting plans featured in Fig. 4 are optimally assigned to associated vehicles according with the multi-objective function (34). From the output, it can be seen that all user expectations were met (i.e., no delays or rejections). Additionally, previously rebalancing vehicle 2 was assigned to pick up users 2 and 4, whereas vehicle 3 was hired to pick up requests 3 and 5.

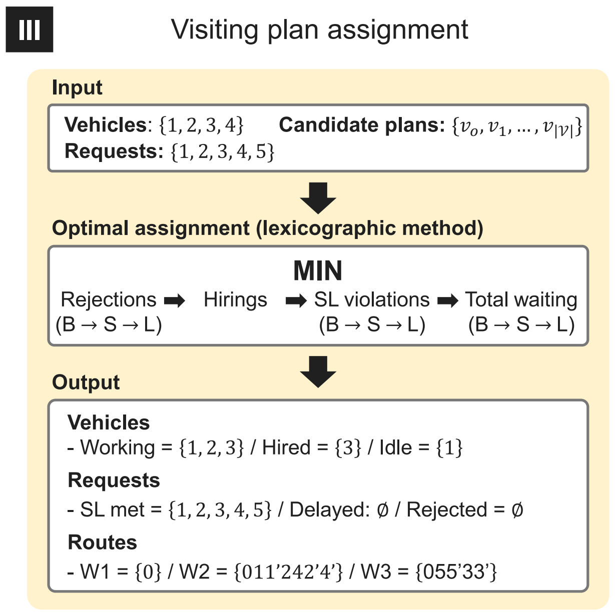
III
Visiting plan assignment
Input
Vehicles: {1, 2, 3, 4}
Candidate plans: {v_o, v_1, ..., v_|V|}
Requests: {1, 2, 3, 4, 5}
Optimal assignment (lexicographic method)
MIN
Rejections (B → S → L)
Hirings
SL violations (B → S → L)
Total waiting (B → S → L)
Output
Vehicles
- Working = {1, 2, 3} / Hired = {3} / Idle = {1}
Requests
- SL met = {1, 2, 3, 4, 5} / Delayed: ∅ / Rejected = ∅
Routes
- W1 = {0} / W2 = {011'242'4'} / W3 = {055'33'}

Fig. 5. At each period, feasible visiting plans are optimally assigned to available vehicles such that rejections, hirings, service level violations, and waiting times are hierarchically minimized. The order objectives are addressed following the lexicographic method is indicated by the arrows.

Feasibility. Due to constraints (40), the MILP model is infeasible whenever the number of vehicles is insufficient to honor SQCs. Still, we can determine the best possible outcome by minimizing violations to a relaxed version:

$$\sum_{i \in P_t^c} y_i + a_c \geq \left\lceil \sigma_c \cdot |P_t^c| \right\rceil \qquad \forall c \in C, \tag{41}$$

where a_c represents the number of service level violations in class c, such that $0 \leq a_c \leq \left\lceil \sigma_c \cdot |P_t^c| \right\rceil$, $\forall c \in C$. Then, we can minimize a_c across classes using the ordered list of objectives

$$\{f_c^{\text{feasible}}(s) = a_c \mid \forall c \in C\}. \tag{42}$$

4.4. Idle vehicle rebalancing formulation

Due to the spatiotemporal characteristics of transportation requests, vehicles can end up stranded in low-demand areas of a city, operating at subpar productivity rates. Hence, to address ongoing supply–demand imbalances, relocation strategies are commonly employed to route idle vehicles to regions where requests are more prone to appear. Following Alonso-Mora et al. (2017), we fix supply and demand imbalances by optimally rebalancing parked vehicles $K_t^{\text{P}} = \{k \mid \text{if } v^k = v_{\text{idle}}^k \forall k \in K\}$ to locations where the system failed to meet user needs. In the original formulation, vehicle rebalancing targets were the pickup points of rejected users. Since our hiring setup can avoid rejections altogether, we adopt two new rebalancing stimuli to move vehicles throughout the street network. First, since hiring indicates a lack of proper vehicle supply, we consider that the origins of hired vehicles $K_t^{\text{H}} = \{k \mid h^k = 1 \; \forall k \in K_t^{\text{FAV}}\}$ are also rebalancing targets. We refer to these origins as $O_t^* = \{o^k \mid \forall k \in K_t^{\text{H}}\}$, since they are a subset of regional centers Z^*. Second, we consider that service-level violations in a particular area also indicate insufficient vehicle supply. This way, besides the origins of rejected users, the origins of assigned but dissatisfied users also integrate the list of rebalancing targets. We refer to the origins of service-level violated requests as $P_t^{\text{U}} = \{i \mid y_i = 0, \; r_i \in P_t\}$, such that the list of rebalancing targets is $J_t = \{i \mid \forall \; i \in O_t^* \cup P_t^{\text{U}}\}$. Finally, we minimize the total travel times to reach the rebalancing targets J_t.

For each vehicle $k \in K_t^{\text{P}}$ we define a set of candidate visits $\mathcal{V}^k = \{v_{\text{idle}}^k\} \cup \mathcal{V}_{\text{R}}^k$, where visiting plans $v^k \in \mathcal{V}_{\text{R}}^k$ are based on sequences $\{S^k \mid S^k = \{S_0^k, i\} \; \forall i \in J_t\}$. Then, we use variables x_v to optimally assign vehicles to targets as follows:

Minimize:

$$\sum_{k \in K_t^{\text{P}}} \sum_{v \in \mathcal{V}_{\text{R}}^k} \Delta_v x_v \tag{43}$$

Subject to:

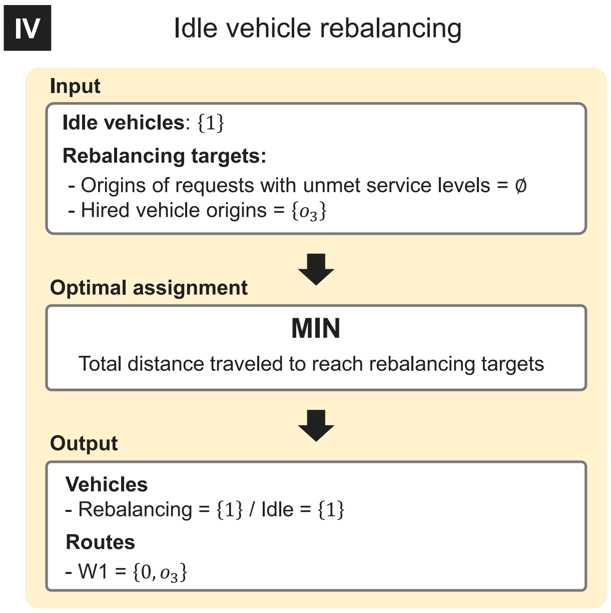
IV
Idle vehicle rebalancing
Input
Idle vehicles: {1}
Rebalancing targets:
- Origins of requests with unmet service levels = ∅
- Hired vehicle origins = {o3}
Optimal assignment
MIN
Total distance traveled to reach rebalancing targets
Output
Vehicles
- Rebalancing = {1} / Idle = {1}
Routes
- W1 = {0, o3}

Fig. 6. Idle working vehicles are optimally rebalanced to underserved areas.

$$\sum_{v \in \mathcal{V}^k} x_v = 1 \qquad \forall k \in K_t^{\mathrm{P}} \tag{44}$$

$$\sum_{k \in K_t^{\mathrm{P}}} \sum_{v \in \mathcal{V}_{\mathrm{R}}^k} x_v = \min\left(|J_t|, |K_t^{\mathrm{P}}|\right) \tag{45}$$

The objective function (43) aims at minimizing the total rebalancing delay. Eqs. (44) guarantee that all available idle vehicles K_t^{P} are used to reach targets in J_t (if any), and Eqs. (45) ensure that each idle vehicle ends up either rebalancing or staying parked. Fig. 6 illustrates how the rebalancing process takes place for the idle vehicles resulting from the assignment step illustrated in Fig. 5.

5. Experimental study

In the following sections, we present the SQ parameters (Section 5.1), describe how simulation settings such as trip and travel data are set up (Section 5.2), and design twelve operational scenarios (Section 5.3) to study the influence of different user-base and FAV-supply configurations.

5.1. Service quality settings

The primary objective of our experiments is to assess the operational impact of actively controlling service quality for a heterogeneous user base. Since the interplay between service levels, fleet size, and service denial has already been thoroughly examined in the literature (see, e.g., Molenbruch et al., 2017; Hörl et al., 2021), we focus our analysis on the operational adjustments required to service the request demand entirely, such that formerly inconvenienced users can also be successfully serviced under strict SQCs.

In order to design a heterogeneous user base, we first define in Section 5.1.1 the class SQC configurations adopted in this study. Next, to evaluate the outcome of meeting SQCs, in 5.1.2, we present three service-level enforcement rate scenarios, where we vary provider's leeway to violate the SQCs. Finally, to evaluate how class SQC configurations can influence fleet operations, in Section 5.1.3, we propose three user segmentation scenarios, where we vary the predominance of a certain class within the transportation demand.

5.1.1. Service quality contract s

Table 5 summarizes the SQC settings we adopt for each user class. Besides priority order and sharing preferences, it shows the expected maximum pickup and total delays of each SQ class. The class-specific settings can represent, for instance, high-, intermediate-, and low-SQ requirements, possibly arising from high-, middle-, and low-income users, respectively. Besides income, the criteria to choose a suitable class can also be related to the nature of the appointment of a user. For example, trips to scheduled events (e.g., meetings, concerts, doctor appointments) are usually stricter, requiring higher service rates and levels. Alternatively, to make a parallel with current urban mobility options (regarding the expected quality of service), the low-cost class best represents former transit passengers, whereas standard and business classes best represent former ridesharing and carsharing/taxi passengers, respectively.

Table 5
Service quality contract (SQC) for each SQ class.

SQ class	Priority	Ride preference	Service level (min)	
			Expected max. pickup (α)	Max. total delay (β)
Business	1st	Private	3	7
Standard	2nd	Shared	5	7
Low-cost	3rd	Shared	7	7

Table 6
User base segmentation scenarios define different distributions of SQ classes across the same transportation demand by varying, in turn, the proportion of requests in each class.

User base	SQ class		
	Business	Standard	Low-cost
B+	68%	16%	16%
S+	16%	68%	16%
L+	16%	16%	68%

5.1.2. *Service-level enforcement rate*

We assess the outcome of different service-level enforcement rates σ_c through four scenarios where we assume the system addresses all SQ classes $c \in C$ using a common rate $\sigma \in \{0, 0.8, 0.9, 1\}$. The scenario where $\sigma = 0$ simulates the typical AMoD system in which service-level preferences are not enforced. The remainder scenarios allow us to investigate how increasingly enforcing the fulfillment of service-level expectations in 10% steps across user classes affects hiring, until the point at which service-level expectations are fully upheld, for $\sigma = 1$. Hence, scenarios $\sigma = 0$ and $\sigma = 1$ represent lower and upper service-level bounds, whereas $\sigma = 0.8$ and $\sigma = 0.9$ allow the system to violate service-level expectations of 20% and 10% of the requests, respectively. Although we do not consider costs in our model, the service-level enforcement rates are tunable parameters that allow providers to trade off vehicle hiring and user dissatisfaction costs.

5.1.3. *User base segmentation*

To investigate the interplay between requests of distinct SQ classes, we create three user base segmentation scenarios, namely, B+, S+, and L+, in which we vary the proportion of users belonging to each SQ class in the transportation demand (Table 6). We assume that the service quality demanded by each user base follows a normal distribution in which the predominant class of such base covers the average SQ requirements of most users (68%) whereas the other two classes can adequately service the rest.

The proposed scenarios aim to represent service quality requirements arising in different regions, on different occasions, or at different times. For example, user base L+ may better represent the vicinity of a campus area, where most users (presumably students) are willing to wait longer in exchange for cheaper rides. Conversely, the user base B+ fits affluent areas where privacy and high responsiveness are prone to play a more significant role.

Throughout the simulation, we use the user base segmentation scenarios to determine the share of requests from the Manhattan, New York City taxi dataset that belongs to each SQ class.

5.2. *Simulation settings*

Street network. We use the Osmnx package (see Boeing, 2017) to construct a multidigraph from the street network of Manhattan comprised of 4,548 locations and 9,701 links. Travel times are drawn from the shortest distances between the street network locations, considering an average travel speed of 30km/h. For simplicity, we assume that congestion is an exogenous effect; neither the time of the day nor the number of vehicles traveling throughout the street network affects travel times.

Region center distribution. Fig. 7 shows the distribution of 68 regional centers, optimally determined considering a maximal hiring delay $s = 150$s. This value allows that available hireable vehicles can fulfill the expectations of even business users who have placed their request at the beginning of a thirty-second round.

Transportation demand settings. We run our simulation on 42,702 real-world taxi requests from Manhattan, New York City, occurring in the evening peak, from 18 h to 19 h, on the first day of February 2011. The raw dataset, containing detailed taxi information for the whole city, is processed in three phases. First, we filter out the demands occurring outside Manhattan. To do so, we remove all requests whose origin/destination GPS coordinates cannot be matched (within a 50 meters range) to an intersection of such graph or whose travel distances are small (below 50 m). After the matching process, we replace origin and destination GPS locations with their correspondent node IDs in the graph. Second, we select only the relevant trip data fields required to carry out our simulation, namely, pickup times (which are adapted to request times), origin/destination IDs, and number of passengers. Additionally, we filter out records whose number of passengers is higher than four (i.e., the maximum vehicle capacity we consider). After this step, the final shares of requests requiring one, two, three, and four seats are 77.00%, 16.54%, 4.53%, and 1.93%, respectively. Finally, we use the roulette wheel selection to randomly attribute SQ classes to users, following the proportions specified in the user base segmentation scenarios.

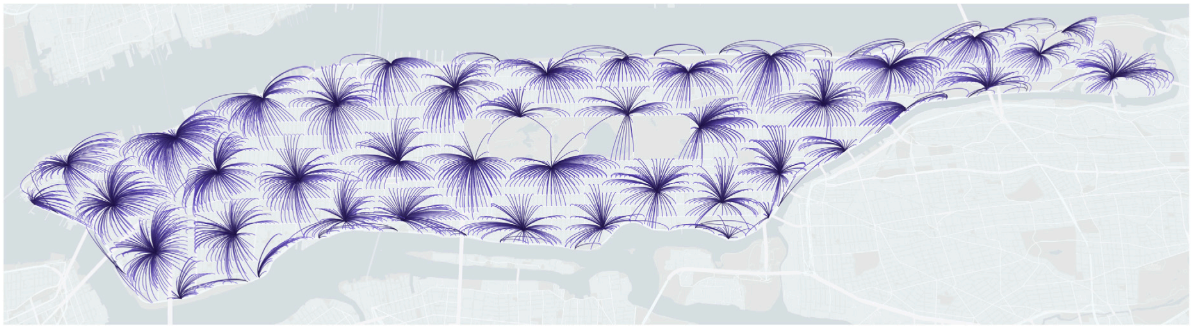

Fig. 7. Optimal distribution of regional centers with corresponding reachable locations throughout the street map of Manhattan, New York City, considering a maximal hiring delay of 150 s at 30 km/h.

Table 7
Instance settings for both static and dynamic problem formulations.

Characteristic	Problem formulation	
	Static	Dynamic
#Requests	{15, 20}	42,702
#Instances	5 request samples	–
Round duration (κ)	30 s	30 s
#Rounds	1	120
Company fleet size	–	1,000
Capacity of company vehicle	–	4 seats
Hireable fleet size	Equal to n. of requests	Unlimited
Capacity of hireable vehicle	4 seats	1, 2, 3, or 4 seats

5.3. Case study configuration

We combine both service-level enforcement rates {0, 0.8, 0.9, 1} (Section 5.1.2) and user base segmentation scenarios {B+, S+, L+} (Section 5.1.3) to create a series of case studies for the static and dynamic formulations. Regardless of the case study considered, users expect to be serviced according to the SQC configuration presented in Table 5. Table 7 summarizes the settings used to create the instances for both formulations.

5.3.1. Static DARP-SQC

Each instance is a combination of a request batch of size $n \in \{15, 20\}$ and a user base segmentation $u \in \{\text{B+}, \text{S+}, \text{L+}\}$. To create such instances, we collect n requests from thirty-second batches of our trip data and randomly distribute SQ classes according to the proportions defined in the user base u. We maintain requests' original placement times as well as pickup and delivery locations but set all passenger counts to one to enable more ridesharing opportunities.

Since we assume a no-rejection policy, we adjust the fleet size and the disposition of the vehicles according to the settings of each instance. First, to guarantee all requests can be reached by a vehicle on time, we determine the minimum set of regional centers from which vehicles can access all nodes in less than three minutes (i.e., the shortest expected maximum pickup delay of all SQ classes). Then, for each request, we assume there will be a four-seat vehicle, available throughout the entire planning horizon, stationed at a regional center that can adequately reach it, such that $|K| = n$. Although our model allows for vehicle heterogeneity, we do not investigate such a feature in this formulation to curb computational complexity. Creating a viable instance for a heterogeneous fleet would require stationing at least $Q_{\max} \times n$ vehicles at each regional center, assuming vehicle capacities can range from one to $Q_{\max} = \max\{Q^k \mid \forall k \in K\}$.

5.3.2. Dynamic DARP-SQC

We assume that the provider relies on an initial working fleet of 1,000 four-seat vehicles to service the transportation demand described in 5.2. If the policy allows, the provider can inflate the fleet size by occasionally hiring privately-owned vehicles (on a single-ride basis) to meet minimum class service level requirements. By default, we assume the capacity of the hired vehicles is equal to the number of passengers of the request that first prompted the hiring.

5.4. Dynamic formulation benchmarking

We implement three policies by varying the objective function of our matching algorithm:

Min. waiting (MW): Standard formulation, with no service level constraints and no hiring, where the goal is to hierarchically minimize the rejection penalties and waiting times across classes $f^{MW} = \{f_c^{\text{reject}}\ \forall c \in C,\ f_c^{\text{wait}}\ \forall c \in C\}$:

Min. $f^{MW}(x_v, z_i)$ s.t. (35), (36), (37).

SL constraints (SL): No hiring formulation, enforcing service-level constraints, and aiming to hierarchically minimize $f^{SL} = \{f_c^{\text{feasible}}\ \forall c \in C,\ f_c^{\text{reject}}\ \forall c \in C, f_c^{\text{violate}}\ \forall c \in C,\ f_c^{\text{wait}}\ \forall c \in C\}$:

Min. $f^{SL}(x_v, y_i, z_i, a_c)$ s.t. (35), (36), (37), (39), (40), (41).

SL constraints + Hire (SLH): Proposed formulation, enforcing service-level constraints, allowing hiring, and aiming to hierarchically minimize $f^{SLH} = \{f_c^{\text{feasible}}\ \forall c \in C, f_c^{\text{reject}}\ \forall c \in C, f^{\text{hire}}, f_c^{\text{violate}}\ \forall c \in C,\ f_c^{\text{wait}}\ \forall c \in C\}$:

Min. $f^{SLH}(x_v, y_i, z_i, h^k, a_c)$ s.t. (35), (36), (37), (38), (39), (40), (41).

Although we assume providers are allowed to break service level expectations of up to $(1-\sigma_c)$ percent of users for each class $c \in C$, we consider rejections a more critical service-level violation than delays. Following this assumption, throughout all policies, we first seek to minimize rejections. Subsequently, policy MW focuses on minimizing waiting times, whereas policies SL and SLH focus on minimizing service-level violations, which cover both rejections and pickup delays. Particularly in SLH, we use the hiring capabilities to prevent rejections from happening even further, once avoiding user rejection is more important than fleet size minimization. Ultimately, by contrasting MW, SL, and SLH, we can assess the effect of enforcing service-level constraints alone as well as how they influence vehicle hiring.

6. Results

The static and dynamic formulations were developed in Python and Java, respectively, and all MILPs were implemented using Gurobi 8.1.0. The experiments were performed on an AMD Opteron central processing unit (CPU) running at 2.10 GHz and 128 GB of random access memory (RAM).

The static formulation results (Section 6.1) help us to understand how fleet size varies to completely accommodate the service quality requirements of different user bases throughout the whole optimization horizon. In contrast, the dynamic formulation results (Section 6.2) highlight the tradeoffs involved in controlling service quality online by periodically organizing operations to account for the needs of incoming request batches.

6.1. Static DARP-SQC

Table 8 presents the results for the static DARP-SQC model. Each instance is run for a maximum of five hours. The figures for each number of requests, user base, and service rate combination are obtained by averaging the results of the five different request distributions. In column "N. of hired", we present the average number of hired vehicles, considering that the fleet size equals the number of requests. Next, for each SQ class, we show the pickup and ride delays. Finally, column "N. of solved inst." presents the number of instances that could be solved optimally. We use these instances to calculate the averages.

The fleet sizes achieved under the no service level enforcement scenario (i.e., $\sigma = 0$) represent the minimum number of vehicles required to pick up all users. In contrast, the slightly higher fleet sizes achieved under the 80%, 90%, and 100% service rates highlight the compromise between the number of vehicles and the class delay tolerances enforced by the SQC constraints. From an AMoD platform's perspective, these results illustrate how exploring users' tolerance to extra delays effectively decreases fleet size while maintaining the strict service quality imposed by previously laid out agreements. By setting up balanced SQCs, platforms can avoid excessive hiring by exploiting the delay tolerance of specific classes while guaranteeing maximum performance for high-demanding user classes.

From Table 8, we can also see that, even for a small twenty-request batch, optimal solutions could be found only for a few instances from user bases S+ and L+. Conversely, the predominance of business-class requests in the user base B+ allows reaching optimal solutions in every case. Having more standard and low-cost class users, who are willing to share a ride and wait longer, leads to a larger number of possible routes, which translates to additional complexity. Ultimately, the results indicate that addressing realistic instances for the DARP-SQC requires more computationally efficient algorithms.

6.2. Dynamic DARP-SQC

Throughout the following sections, we provide an exploratory analysis of the aggregate results regarding all user bases and service-level enforcement rates for all policies. We use the case study consisting of user base S+ and service-level enforcement rate $\sigma = 0.9$ to illustrate our results further and refer to it as the reference case study. In this case study, the most frequent SQ class (i.e., the standard class) imposes a balanced SQC when compared to its counterparts, requiring reasonably fast service levels and also allowing for ridesharing. On the other hand, the reference service-level enforcement rate allows us to thoroughly assess the flexibility achieved by violating the expectations of 10% of the users.

Table 8
Number of vehicles hired, pickup delay, and ride delay for the static DARP-SQC instances across all case studies. Figures correspond to the average of the results achieved for instances that could be solved to optimality.

N. of req.	User base	Service rate	N. of hired	Pickup delay (s)			Ride delay (s)			N. of solved inst.
				B	S	L	B	S	L	
15	B+	0%	10.4	138.3	175.6	203.5	0.0	22.9	12.9	5
		80%	11.8	90.9	134.2	221.6	0.0	48.4	12.9	5
		90%	12.2	90.5	109.1	220.5	0.0	48.4	12.9	5
		100%	12.2	90.5	109.1	220.5	0.0	48.4	12.9	5
	S+	0%	9.0	57.6	143.3	176.3	0.0	46.8	57.0	4
		80%	9.2	57.6	126.0	175.0	0.0	60.7	56.6	4
		90%	9.2	57.6	126.0	175.0	0.0	60.7	56.6	4
		100%	9.8	57.6	115.0	175.0	0.0	49.8	56.6	4
	L+	0%	9.0	68.8	152.3	165.1	0.0	0.0	67.3	4
		80%	9.2	59.2	147.9	163.6	0.0	0.0	67.3	4
		90%	9.5	59.2	124.6	163.6	0.0	0.0	60.8	4
		100%	9.5	59.2	124.6	162.4	0.0	0.0	61.9	4
20	B+	0%	14.0	124.2	227.8	205.6	0.0	28.5	8.1	5
		80%	16.8	73.1	162.0	223.5	0.0	28.5	12.3	5
		90%	17.2	73.1	134.4	199.2	0.0	6.0	11.5	5
		100%	17.2	73.1	134.4	199.2	0.0	6.0	11.5	5
	S+	0%	11.0	170.0	199.7	159.1	0.0	25.9	0.2	2
		80%	12.5	94.6	193.0	184.7	0.0	23.9	30.8	2
		90%	13.5	76.0	166.7	174.8	0.0	14.5	7.2	2
		100%	14.5	74.8	141.0	173.8	0.0	14.5	7.2	2
	L+	0%	10.0	116.0	201.7	181.7	0.0	42.3	49.7	1
		80%	10.0	116.0	103.3	197.7	0.0	151.3	39.4	1
		90%	10.0	116.0	103.3	197.7	0.0	151.3	39.4	1
		100%	11.0	116.0	69.2	197.7	0.0	133.2	35.2	1

6.2.1. Service quality distribution

Table 9 summarizes the average service quality outcome across classes achieved by each policy for all case studies. The column "Serviced" shows the percentage of users picked up by the AMoD system. In turn, the column "Met SL" presents the percentage of satisfied users, serviced according to their expected service levels. Subsumed under the header "Violated SL", we present the two service level violations we consider in our approach. The column "Delayed" indicates the percentage of users whose desired service levels could not be fulfilled, whereas the column "Rejected" indicates the percentage of requests that could not be serviced. It is worth noting that each percentage under the "Serviced" column is the sum of corresponding percentages under "Met SL" and "Delayed" columns.

The user base segmentation scenarios play a significant role in the number of users serviced when vehicle hire is not enabled. Once user base scenario B+ comprises more business users, which expect shorter delays and private rides, less sharing occurs, leading to about 10% fewer pickups than the other bases. However, by exploiting the user delay tolerances, our SL policy can achieve higher service rates than the MW policy across case studies.

By delaying users who can wait longer, the SL-policy matching process can enable additional possibilities to combine overlapping routes. Although policy MW minimizes user class rejections and waiting times hierarchically, the lack of service-level enforcement constraints prevents the optimization process from adequately harnessing class delay tolerances. For example, for the reference case study, enforcing service-level constraints leads to a 29.81% percentage point increase over the 55.92% number of satisfied users (i.e., met service levels) from policy MW. At the same time, from Table 9, we can see that the number of serviced users increases moderately (from 96.14% to 98.16%). Ultimately, these results suggest that our service-level enforcement constraints were useful to reorganize pickups, resulting in a 53.3% increase in service quality across classes in the reference case study.

Figs. 8 and 9, detail the service-level distribution for all requests, considering user base S+ and service-level enforcement rate $\sigma = 0.9$. In Fig. 8, we can see that because policy MW does not enforce users' class service-level expectations, pickup waiting times of business users end up being the highest among the classes. Once we aim to first minimize rejections across classes in MW, this result suggests that by delaying the business users, the optimization process could pick up more users from the remaining classes. Fig. 8 also shows that policy SL rejects thirty four business users (0.4% of the 6,831 business requests) but is able to achieve the expected three-minute maximum waiting time for around 75% of them.

Service-level enforcement rate versus vehicle hiring. Fig. 9 offers an alternative perspective on how each policy makes use of the 10% service level violation allowed by adopting the service-level enforcement rate $\sigma = 0.9$. For SL, which features the relaxed version of the service-level enforcement constraints (41), we can verify how this relaxation took place across classes. For business and standard users, the share of SL violated requests are 11.24% and 17.73%, which correspond to 1.24 and 7.73 percentage points higher than the 10% violation limit. In contrast, for the SLH policy, the violations for business and standard users total 3.73% and 5.60%, figures significantly lower than the SL violation limit. This result could indicate that the optimization process is overhiring

Table 9
Service quality achieved using each policy for all case studies.

User base	Service rate	Policy	Serviced	Met SL	Violated SL	
					Delayed	Rejected
B+	0%	Min. waiting	84.99%	37.35%	47.64%	15.01%
	80%	SL constrs.	85.09%	55.32%	29.77%	14.91%
		SL constrs. + Hire	100.00%	91.98%	8.02%	–
	90%	SL constrs.	85.18%	56.01%	29.17%	14.82%
		SL constrs. + Hire	100.00%	95.91%	4.09%	–
	100%	SL constrs.	84.92%	56.84%	28.08%	15.08%
		SL constrs. + Hire	100.00%	100.00%	–	–
S+	0%	Min. waiting	96.14%	55.92%	40.22%	3.86%
	80%	SL constrs.	98.19%	85.58%	12.61%	1.81%
		SL constrs. + Hire	100.00%	91.05%	8.95%	–
	90%	SL constrs.	98.16%	85.73%	12.43%	1.84%
		SL constrs. + Hire	100.00%	95.58%	4.42%	–
	100%	SL constrs.	98.37%	86.69%	11.68%	1.63%
		SL constrs. + Hire	100.00%	100.00%	–	–
L+	0%	Min. waiting	96.53%	78.53%	18.00%	3.47%
	80%	SL constrs.	98.00%	93.52%	4.48%	2.00%
		SL constrs. + Hire	99.34%	96.14%	3.20%	0.66%
	90%	SL constrs.	97.79%	94.07%	3.72%	2.21%
		SL constrs. + Hire	99.63%	98.05%	1.59%	0.37%
	100%	SL constrs.	97.76%	94.41%	3.35%	2.24%
		SL constrs. + Hire	100.00%	100.00%	–	–

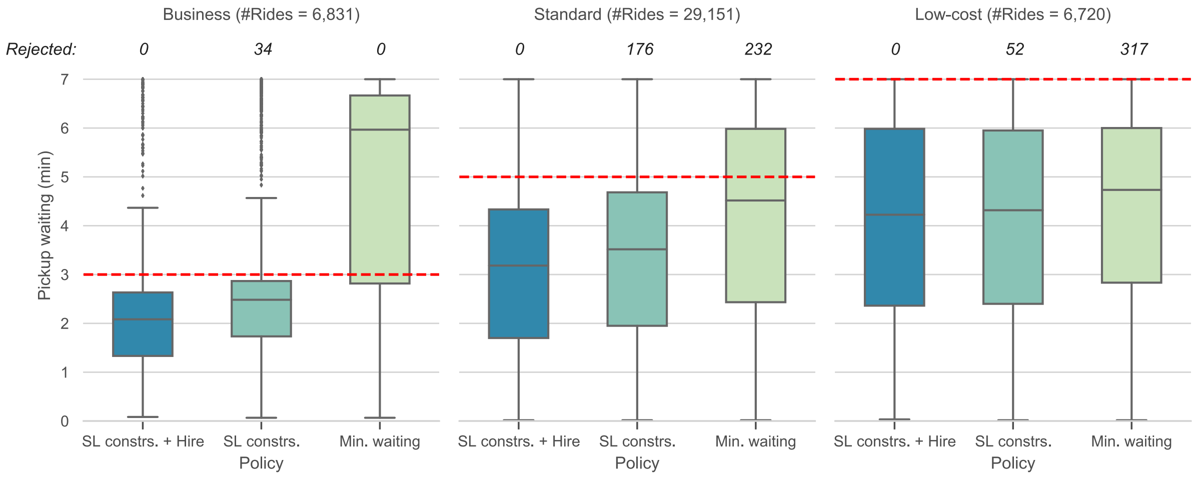

Fig. 8. Pickup waiting time distribution and number of rejected users for each SQ class across all policies for user base S+ and service-level enforcement rate $\sigma = 0.9$ for policies SL and SLH. The dashed lines mark the expected max. pickup delays of user classes.

vehicles since there was still extra room for violating user expectations (up to 10%). However, by analyzing which user class hired vehicles are addressing the most, we can further understand why the service violation limit was underutilized.

Table 10 expands the summary statistics presented in Table 9. For each user base segmentation scenario, it shows for every user SQ class the percentage of users picked up by company or freelance vehicles as well as the percentage of users whose service-level expectations are violated. From Table 10, we can see that vehicle hiring occurs predominantly to fulfill business user requests. Therefore, these hirings end up freeing the four-seat company vehicles that would have been used to transport these requests privately. As a result, the ridesharing requests from standard and low-cost classes can enjoy a larger vehicle supply, which leads to fewer service-level violations. This extra vehicle supply, in turn, is reflected in the underutilization of the service-level violation limits.

Long-term fairness. Ultimately, it is worth noting that, although we enforce class service levels, individual users may experience service-level violations repeatedly, over a longer time horizon. To further minimize user dissatisfaction, a real-world service provider could strive to make up for repeating users who have had their service levels violated in previous rides. Provided that users $i \in P$

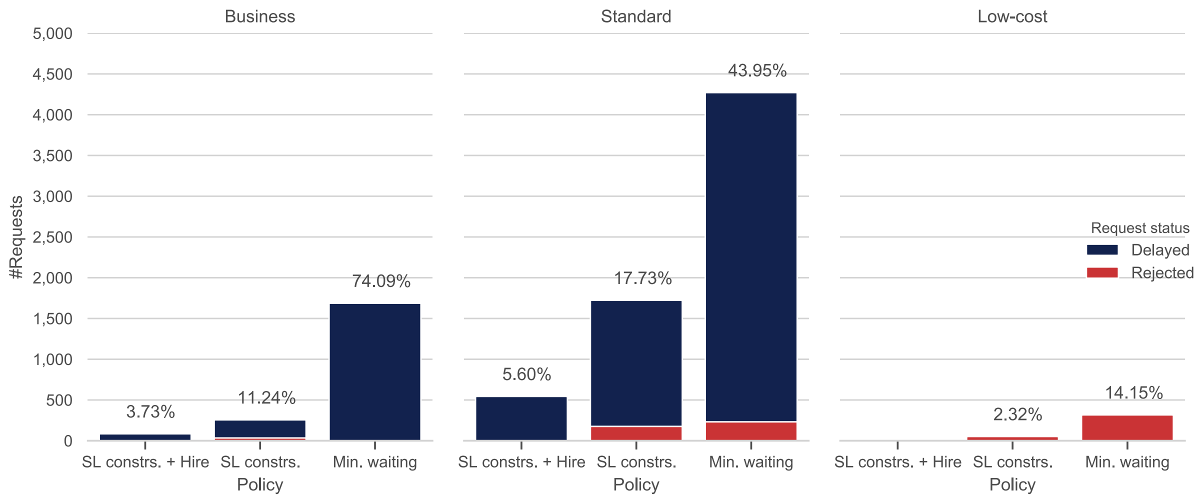

Fig. 9. Service-level violation breakdown into percentages of rejected and delayed users for each SQ class across all policies for user base S+ and service-level enforcement rate $\sigma = 0.9$ for policies SL and SLH.

Table 10
Share of requests whose service-level expectations were met (through company or freelance vehicles) or violated, according to SQ class for each user base distribution, service-level enforcement rate, and policy.

User base	Service rate	Policy	Business			Standard			Low-cost		
			Company	Freelance	Violated	Company	Freelance	Violated	Company	Freelance	Violated
B+	0%	Min. waiting	14.6%	–	85.4%	73.3%	–	26.7%	98.3%	–	1.7%
	80%	SL constrs.	38.2%	–	61.8%	87.1%	–	12.9%	96.0%	–	4.0%
		SL constrs. + Hire	60.4%	29.7%	9.9%	92.1%	–	7.9%	100.0%	–	–
	90%	SL constrs.	38.7%	–	61.3%	87.5%	–	12.5%	98.2%	–	1.8%
		SL constrs. + Hire	64.1%	31.2%	4.8%	94.8%	*	5.2%	100.0%	–	–
	100%	SL constrs.	39.3%	–	60.7%	89.4%	–	10.6%	98.7%	–	1.3%
		SL constrs. + Hire	66.8%	33.2%	–	99.4%	*	–	100.0%	–	–
S+	0%	Min. waiting	25.9%	–	74.1%	56.0%	–	44.0%	85.8%	–	14.2%
	80%	SL constrs.	86.0%	–	14.0%	82.9%	–	17.1%	96.7%	–	3.3%
		SL constrs. + Hire	75.4%	16.8%	7.8%	88.4%	*	11.3%	100.0%	–	–
	90%	SL constrs.	88.8%	–	11.2%	82.3%	–	17.7%	97.7%	–	2.3%
		SL constrs. + Hire	78.2%	18.1%	3.7%	92.8%	1.6%	5.6%	100.0%	–	–
	100%	SL constrs.	90.3%	–	9.7%	83.1%	–	16.9%	98.4%	–	1.6%
		SL constrs. + Hire	76.6%	23.4%	–	97.9%	2.1%	–	99.9%	*	–
L+	0%	Min. waiting	25.4%	–	74.6%	61.8%	–	38.2%	94.9%	–	5.1%
	80%	SL constrs.	83.8%	–	16.2%	87.3%	–	12.7%	97.2%	–	2.8%
		SL constrs. + Hire	82.6%	9.0%	8.4%	88.4%	–	11.6%	99.0%	–	*
	90%	SL constrs.	84.4%	–	15.6%	89.0%	–	11.0%	97.5%	–	2.5%
		SL constrs. + Hire	80.3%	15.9%	3.7%	92.6%	1.2%	6.2%	99.5%	–	*
	100%	SL constrs.	85.5%	–	14.5%	90.1%	–	9.9%	97.5%	–	2.5%
		SL constrs. + Hire	77.2%	22.8%	–	95.8%	4.2%	–	99.3%	*	–

Values lower than 0.01% are marked with "*" and absent data are marked with "–"

are associated with unique ids in the AMoD system, the variable of y_i can be set up ad-hoc to grant users the best service level. This way, σ_c rates could also be applied individually by keeping track of each traveler's service level over multiple rides.

The effect of rebalancing. Regarding the rebalancing approach, the inclusion of service-level violations as an additional stimulus to prompt rebalancing (besides service rejection) has been shown to increase fleet productivity. It can be seen from Fig. 10 that in MW, idle vehicles can still be found until about 18:30, whereas in SL there are no idle vehicles from about 18:10 and onward. The number of idle vehicles drops sharply as soon as the system fails to fulfill user expectations across classes. As a result, by rebalancing vehicles earlier, policy SL contributes to achieving a two percentage point increase in the number of users picked up compared to MW. Moreover, when users cannot be rejected (i.e., policy SLH), the hiring operations have successfully replaced the user rejection stimuli to indicate where vehicles are needed.

Table 11
Average number of hired vehicles per capacity and summary statistics for the hired fleet in each round.

User base	Service rate	Avg. hired/Vehicle capacity				Hired vehicles		
		1	2	3	4	Avg.	Median	Max.
B+	80%	76.39%	19.41%	2.74%	1.46%	223.88	309	493
	90%	79.67%	15.79%	3.21%	1.32%	243.75	336	497
	100%	79.92%	14.75%	4.16%	1.17%	254.74	355	520
S+	80%	76.41%	15.48%	4.82%	3.29%	32.79	22	102
	90%	73.45%	17.40%	5.45%	3.70%	47.41	43	168
	100%	74.29%	18.57%	4.94%	2.20%	60.32	65	167
L+	80%	66.49%	26.89%	3.36%	3.26%	15.64	8	72
	90%	77.33%	17.55%	3.50%	1.62%	30.85	22	104
	100%	76.71%	18.01%	2.61%	2.67%	56.07	51	191

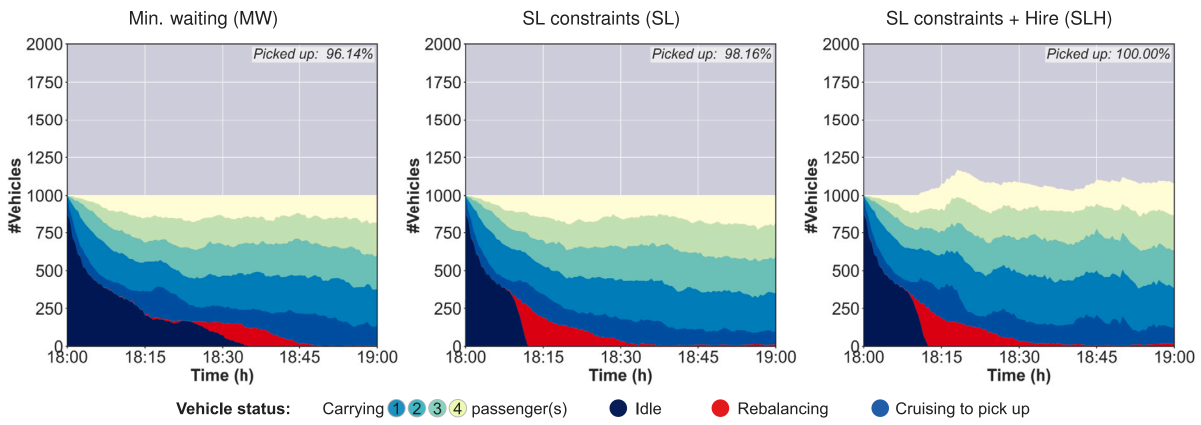

Fig. 10. Size of the working fleet every thirty seconds over the course of one hour for each policy, considering user base S+ and a 90% SL enforcement for SL and SLH. Colors help to identify the part-to-whole ratio of vehicles per status at each period.

6.2.2. *Fleet size inflation*

From the overall fleet statistics presented in Table 11, we can see that the average, maximum, and median numbers of hired seats throughout all rounds are proportional to the service quality of each user base and service-level enforcement rate. For instance, we can see that the overrepresentation of business-class users is translated accordingly in the fleet of vehicles hired to service formerly dissatisfied users in policies MW and SL. Moreover, most hired vehicles end up being of low capacity. Once we assume that a hired vehicle's capacity equals the number of seats required by the request that prompted the hiring, this outcome is congruent with the characteristics of the user demand (77% of the requests demand only one seat). For the reference case study, Table 11 shows that at the demand peak, the total fleet capacity grows by 168 vehicles (see the hiring peak in Fig. 10 between 18:15 and 18:30).

6.2.3. *Whole day operation*

In Fig. 11, we show the fleet status for a full-day instance considering 210,269 requests configured according to the reference case study. Hirings occur predominantly in the evening, and the surplus number of vehicles over the 1000-vehicle fleet is lower than 200. From the figure, we can conclude that the fixed working fleet can meet the entire demand most of the time and still be sub-utilized at some periods (e.g., before 6:00). Our results stress how much vehicle downtime a provider can avoid by partially relying on the FAV fleet to completely fulfill the demand.

6.2.4. *Computational complexity*

The same shortcomings found for the static version regarding the more flexible user-segmentation scenarios are also present in the dynamic version. In these scenarios, the lexicographical method cannot be completed on some rounds due to the maximum computation time we have imposed on the optimization process. Consequently, some assignments are sub-optimal, covering a subset of the objectives (high-priority first). This outcome is particularly present for scenario L+. From Table 11, we can see that the maximum number of hired vehicles for this user segmentation for a 100% service rate is higher than the S+, where users have a lower waiting tolerance. Additionally, Table 9 shows that 0.37% of low-cost users are rejected while using the hiring policy SLH, where hired vehicles are made available to pick up all users. This indicates that, for some rounds, the hierarchical optimization stops prematurely, failing to minimize class rejections and vehicle hire.

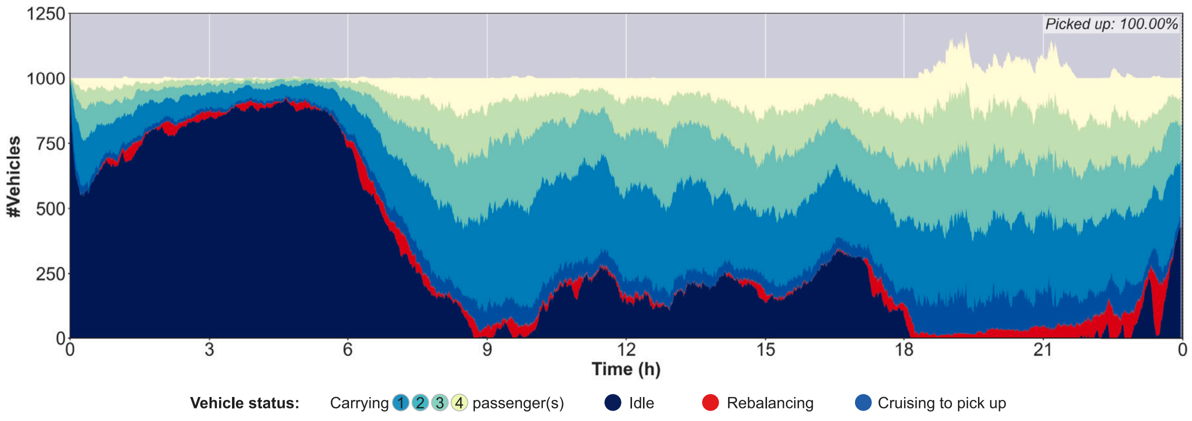

Fig. 11. Size of the working fleet every thirty seconds over the course of an entire day, considering user base S+ and a 90% SL enforcement for SLH, policy. Colors help to identify the part-to-whole ratio of vehicles per status at each period.

6.3. Managerial insights

With widespread AV adoption, cities may implement policies that prioritize high occupancy vehicles or even charge empty-seat taxis. Such policies serve two main purposes. First, they discourage excessive low-occupancy AV rides, counterbalancing potential rises in overall *vehicle kilometers traveled* (VKT). Second, they contribute as alternative sources of income for cities, especially in the shadow of an expected revenue drop due to less parking space requirements (Nourinejad et al., 2018). Hence, keeping high occupancy rates and reducing total VKT are critical factors for future AMoD providers. Our multi-objective function covers both goals once it determines the minimum fleet size and mix to meet the demand expectations.

However, despite advantageous for the operator, single-ride contracts are likely to be unpopular among owners. Under such an agreement, there might be occasions where vehicles are made available for a whole day but make only a couple of trips. In this low-profit scenario, owners could, for instance, link their participation to a minimum compensation, such that their vehicles are used more frequently.

Alternatively, to guarantee frequent use, providers could establish long-duration contracts to actively manage and rebalance vehicles as if they belonged to the fixed working fleet. Under such contracts, however, operators may be required to bear all costs within the leasing period. These may include not only the costs to rebalance and service users but also the opportunity costs of hireable vehicles working for other competing operators. Therefore, once costs are considered, the rider's satisfaction must be adequately translated into user class fares to cover the provider's outlay concerning vehicle hire.

Moreover, when hiring, we assume there is an infinite number of FAVs readily available in each request's surroundings. This assumption entails that there exists an efficient hiring strategy in place, which hires vehicles of adequate capacities in advance and rebalances them to locations where the own fleet historically cannot fulfill the demand. Such an anticipatory hiring strategy could be the key to reaching a more reasonable compromise between FAV owners and providers' objectives in a real-world scenario. In this case, independent owners have an extra incentive to make their vehicles available to join the provider's fleet once they are compensated even before they are assigned to a request.

7. Conclusions

In this study, we have introduced a new operational approach to actively control service quality in AMoD systems, increasing and decreasing the number of used vehicles in the short term to meet diversified user expectations. We have used these expectations to establish service quality contracts, allowing heterogeneous users to choose ride experiences that best match their preferences, especially regarding maximum waiting times and willingness to share a ride. Based on an experimental study using New York City taxi data, we have found that the developed approach allows us to significantly improve the service quality of all considered user categories. Enforcing the proposed service-level constraints, we can meet the expectations of 85.7% of the users across classes, a 53% average increase in comparison to conventional ridesharing systems. When hiring is enabled, we can meet the expectations of 95.6% of the user requests, at the expense of a mild fleet inflation (a maximum surplus of 168 FAVs is observed at the evening demand peak).

The proposed method allows providers to make a compromise between avoiding service-level violations and hiring extra vehicles, steering the vehicle supply to service the highest priority or most profitable customer segments, according to the particular conditions of an operational environment (e.g., availability of idle vehicles, hiring costs, user dissatisfaction costs). Nevertheless, hiring occurs only to the extent that it preserves minimum service level requirements. Hence, we add to recent literature by providing an active means to control service quality on an operational level and distinguishing services classes in terms of user expectations.

Moreover, we have formalized the problem using a MILP formulation and proposed a matheuristic to deal with large-scale real-world instances. This matheuristic encompasses the construction of feasible ridesharing visiting plans and the optimal assignment

Table 12
Total execution time and average round time (in seconds) for the one-hour experiments (120 thirty-second rounds) considering the cuts described on configurations C60_4_240 (adopted in this study) and C30_2_30 (stricter timeouts and smaller solution graph).

User base	Service rate	Policy	Configuration C60_4_240		Configuration C30_2_30	
			Total	Round	Total	Round
B+	–	Min. waiting	21,289.24	108.62	1,694.11	8.69
	80%	Enforce SL	11,992.77	64.48	1,637.73	8.44
		Enforce SL + Hire	6,993.08	35.68	1,853.63	9.7
	90%	Enforce SL	11,291.58	58.81	1,524.22	7.82
		Enforce SL + Hire	5,851.98	31.46	1,704.90	9.12
	100%	Enforce SL	10,729.68	56.18	1,557.43	8.28
		Enforce SL + Hire	3,153.85	16.78	836.98	4.52
S+	–	Min. waiting	24,505.63	126.32	2,026.37	10.34
	80%	Enforce SL	19,293.13	98.43	2,101.49	11.3
		Enforce SL + Hire	23,943.19	119.72	2,439.43	12.57
	90%	Enforce SL	18,452.69	95.61	1,986.42	10.19
		Enforce SL + Hire	22,525.65	119.82	2,255.83	12.19
	100%	Enforce SL	16,928.43	87.71	2,074.08	10.58
		Enforce SL + Hire	6,047.36	30.85	1,414.21	7.52
L+	–	Min. waiting	25,109.09	132.85	1,699.82	8.85
	80%	Enforce SL	19,271.08	98.83	1,918.99	9.84
		Enforce SL + Hire	23,657.90	116.54	2,371.18	12.29
	90%	Enforce SL	18,681.01	97.81	1,886.39	9.62
		Enforce SL + Hire	23,333.19	119.05	2,432.48	12.29
	100%	Enforce SL	16,926.03	88.16	1,897.20	9.68
		Enforce SL + Hire	14,348.55	74.73	1,813.61	9.75

of plans to available vehicles. Besides ride-matching, the assignment phase also includes a reactive rebalancing strategy that uses both service level violations and vehicle hire as stimuli to reposition vehicles. Using a lexicographic method, we have solved a multi-objective function where the primary goal is to minimize rejections (i.e., the most significant source of inconvenience) to subsequently minimize fleet size and service-level violations. We have also evaluated to what extent hiring extra vehicles affects overall service quality and fleet usage, by designing twelve scenarios where we vary the rate at which providers commit to fulfilling user service level expectations. In this way, our results help understand the operational impact of meeting user service quality expectations and the tradeoffs entailed by occasionally violating service quality expectations.

Although we have restricted the analysis to three user segments, the service quality classes we propose are general enough to serve as the basis for more diversified customer-centric services. For the classes considered, our results show that, regardless of the user base, most hired vehicles join the initial fleet to service one-passenger requests from the high-quality user segment (i.e., the business class). These findings indicate that providers may benefit from leveraging the strengths of two AMoD paradigms, namely, ridesharing and single-passenger personal mobility, on a common platform, without the necessity of owning the entire fleet.

Future work will focus on setting the relation between service quality and total costs. This way, operators can weigh the advantage of (1) investing in PAVs or hiring FAVs online to avoid service quality contract breaches and (2) compensating users in the event of poor service. Ultimately, service fares must reflect PAV operational costs (e.g., maintenance, management, cleaning, investment, parking, insurance, city fees), FAV owner's custom preferences and hiring costs, and the platforms' reputation as a function of the consistent fulfillment of service quality expectations. Additionally, a promising future direction is considering AMoD providers integrate a broader transportation ecosystem. In this setting, different modes are orchestrated to achieve overarching mobility goals, such as mitigating congestion and distributing accessibility over different regions and demographics. From the providers' perspective, such an integration would require weighing a series of alternative factors (e.g., congestion pricing, empty-vehicle fees, parking costs, ride subsidization schemes) into the fleet management strategy. Finally, one could also focus on addressing the inherent uncertainty of the proposed scenario, in which both vehicle supply and demand unfold throughout time.

CRediT authorship contribution statement

Breno A. Beirigo: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Rudy R. Negenborn:** Conceptualization, Resources, Supervision, Project administration, Funding acquisition. **Javier Alonso-Mora:** Conceptualization, Methodology, Writing – original draft. **Frederik Schulte:** Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Acknowledgments

This research is supported by the project "Dynamic Fleet Management (P14-18 – project 3)" (project 14894) of the research programme i-CAVE, partly financed by the Netherlands Organization for Scientific Research (NWO), domain Applied and Engineering Sciences (TTW).

Appendix. Execution time vs. solution quality

We decrease execution time by parallelizing the construction of both RV and ERTV graphs. As in Alonso-Mora et al. (2017), to build the RV graph, we determine suitable pairs for requests in parallel, whereas to build ERTV, we create vehicle visiting plans in parallel, exploring increasingly longer visiting sequences for each vehicle separately. Also, we tradeoff solution quality with faster computation by limiting the sizes of RV and ERTV graphs and imposing maximum execution timeouts.

In this study, for the RV graph, we assume requests can be connected to up to sixty vehicles and as many requests as possible. If hiring is enabled, we guarantee that the request–vehicle edge set will always include the edge featuring the closest backup hireable vehicle, even if better options (i.e., closer vehicles) in the working fleet are available. Doing so allows us to achieve our objective of minimizing rejections once the hireable edge will certainly integrate the pool of visiting plans. As for the ERTV graph, we consider the vehicle visiting plan's exploration timeout to 0.4 s and let the optimal assignment method execute for at most 240 s in the Gurobi solver. We refer to this configuration as C60_4_240.

To show how limiting the execution times and graph size affects solution quality, we rerun our experiments considering that for the RV graph, each request is connected to the closest fifteen vehicles (including the hireable, if applicable) and fifteen requests, therefore totaling at most thirty pairs. Regarding the ERTV graph, we determine a timeout of 0.2 s per vehicle to interrupt the

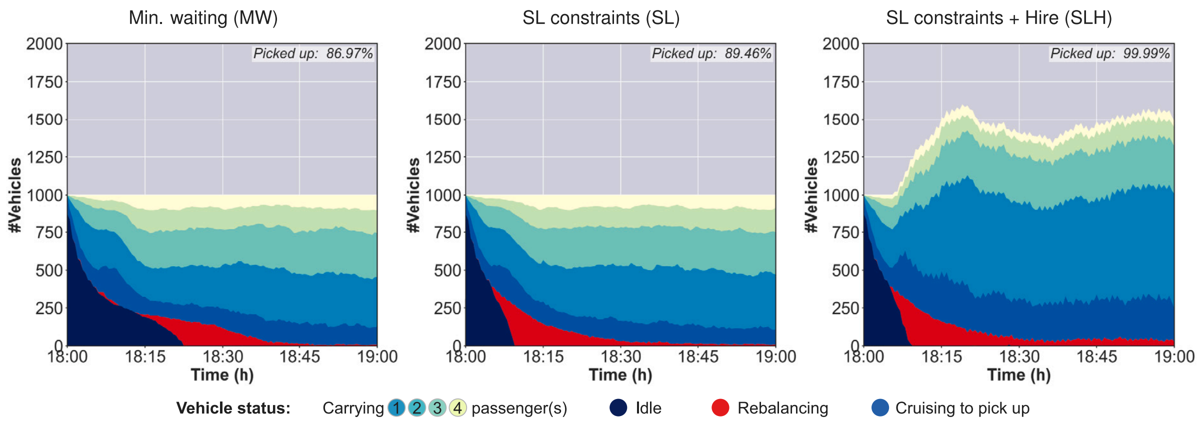

Fig. 12. Test case depicted in Fig. 10 considering less computation time and a smaller graph of feasible solutions. The number of users picked up drops about 10% for policies MW and SL. Due to the premature termination of the lexicographic method, policy SLH is unable to realize all the hires necessary to pick up all requests.

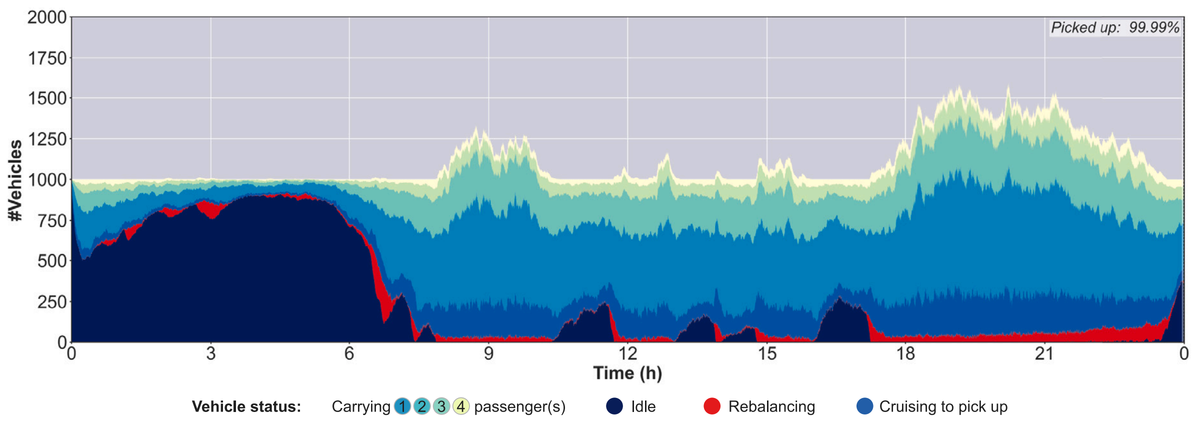

Fig. 13. Test case depicted in Fig. 11 considering less computation time and a smaller graph of feasible solutions. Vehicle hires at the evening demand peak are three times bigger and the previously low number of hires in the morning are now substantially higher.

exploration of candidate visiting plans and set up a thirty-second maximum execution time for the matching formulation. We refer to this configuration as C30_2_30.

Table 12 presents the execution times for all test cases assuming the limitations described for both configurations. Similar to the static formulation, the higher the flexibility of the predominant user class, the more complex the problem, resulting, therefore, in longer execution times. In Figs. 12 and 13, we replicate Figs. 10 and 11 considering configuration C30_2_30. Under stricter timeouts, the optimization process has fewer opportunities to find high-quality solutions, leading to decreased sharing (see the increase in vehicles carrying one passenger) and substantially more hirings at the demand peaks. Additionally, less computational time may harm user experience even when hiring is enabled. In Fig. 10, for example, 0.01% of users are rejected because the maximum execution time expires before a no-rejection solution could be found.

References

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.

Archetti, C., Speranza, M.G., 2014. A survey on matheuristics for routing problems. EURO J. Comput. Optim. 2 (4), 223–246.

Beaton, E., 2019. New York City Mobility Report. Technical Report, NYC Department of Transportation, URL https://www1.nyc.gov/html/dot/downloads/pdf/mobility-report-singlepage-2019.pdf.

Beirigo, B.A., Schulte, F., Negenborn, R.R., 2018. Dual-mode vehicle routing in mixed autonomous and non-autonomous zone networks. In: Proceedings of the 21st International Conference on Intelligent Transportation Systems, ITSC, Maui, HI, United States, pp. 1325–1330.

Beirigo, B.A., Schulte, F., Negenborn, R.R., 2020. Overcoming mobility poverty with shared autonomous vehicles: A learning-based optimization approach for Rotterdam Zuid. In: Proceedings of the 11th International Conference on Computational Logistics ICCL2020, Enschede, the Netherlands, pp. 492–506.

Beirigo, B.A., Schulte, F., Negenborn, R.R., 2021. A learning-based optimization approach for autonomous ridesharing platforms with service-level contracts and on-demand hiring of idle vehicles. Transp. Sci..

Berbeglia, G., Cordeau, J.F., Laporte, G., 2010. Dynamic pickup and delivery problems. European J. Oper. Res. 202 (1), 8–15.

Boeing, G., 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Comput. Environ. Urban Syst. 65, 126–139.

Boesch, P.M., Ciari, F., Axhausen, K.W., 2016. Autonomous vehicle fleet sizes required to serve different levels of demand. Transp. Res. Rec. J. Transp. Res. Board 2542, 111–119.

Bulhões, T., Hà, M.H., Martinelli, R., Vidal, T., 2018. The vehicle routing problem with service level constraints. European J. Oper. Res. 265 (2), 544–558.

Campbell, H., 2018. Who will own and have propriety over our automated future? considering governance of ownership to maximize access, efficiency, and equity in cities. Transp. Res. Rec. 2672 (7), 14–23.

Chen, T.D., Kockelman, K.M., Hanna, J.P., 2016. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. Transp. Res. A 94, 243–254.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54 (3), 573–586.

Cordeau, J.-F., Laporte, G., Savelsbergh, M.W., Vigo, D., 2007. Handbooks in Operations Research and Management Science, Vol. 14. Elsevier, pp. 367–428.

Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transp. Res. C 40, 1–13.

Fagnant, D.J., Kockelman, K.M., 2018. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. Transportation 45 (1), 143–158.

Fagnant, D.J., Kockelman, K.M., Bansal, P., 2015. Operations of shared autonomous vehicle fleet for austin, texas market. Transp. Res. Rec. J. Transp. Res. Board 2536, 98–106.

Fiedler, D., Čertický, M., Alonso-Mora, J., Čáp, M., 2018. the impact of ridesharing in mobility-on-demand systems: Simulation case study in prague. In: Proceedings of the 21st International Conference on Intelligent Transportation Systems, ITSC, Maui, HI, United States, pp. 1173–1178.

Gueriau, M., Dusparic, I., 2018. SAMoD: Shared autonomous mobility-on-demand using decentralized reinforcement learning. In: Proceedings of the 21st International Conference on Intelligent Transportation Systems, ITSC, Maui, HI, United States, pp. 1558–1563.

Gurumurthy, K.M., Kockelman, K.M., 2018. Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from Orlando, Florida. Comput. Environ. Urban Syst. 71, 177–185.

Ho, S.C., Szeto, W.Y., Kuo, Y.H., Leung, J.M., Petering, M., Tou, T.W., 2018. A survey of dial-a-ride problems: Literature review and recent developments. Transp. Res. B 111, 1–27.

Hörl, S., Becker, F., Axhausen, K.W., 2021. Simulation of price, customer behaviour and system impact for a cost-covering automated taxi system in Zurich. Transp. Res. C 123, 102974.

Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K., 2019. Fleet operational policies for automated mobility: A simulation assessment for Zurich. Transp. Res. C 102, 20–31.

Hyland, M.F., Mahmassani, H.S., 2017. Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility. Transp. Res. Rec. J. Transp. Res. Board 2653 (1), 26–34.

Hyland, M., Mahmassani, H.S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transp. Res. C 92, 278–297.

Krueger, R., Rashidi, T.H., Rose, J.M., 2016. Preferences for shared autonomous vehicles. Transp. Res. C 69, 343–355.

Litman, T., 2017. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning. Technical Report, Victoria Transport Policy Institute.

Liu, Y., Bansal, P., Daziano, R., Samaranayake, S., 2019. A framework to integrate mode choice in the design of mobility-on-demand systems. Transp. Res. C 105, 648–665.

Lokhandwala, M., Cai, H., 2018. Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC. Transp. Res. C 97, 45–60.

Molenbruch, Y., Braekers, K., Caris, A., 2017. Operational effects of service level variations for the dial-a-ride problem. CEJOR Cent. Eur. J. Oper. Res. 25 (1), 71–90.

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transp. Res. C 111, 255–293.

Nourinejad, M., Bahrami, S., Roorda, M.J., 2018. Designing parking facilities for autonomous vehicles. Transp. Res. B 109, 110–127.

Paquette, J., Cordeau, J.F., Laporte, G., 2009. Quality of service in dial-a-ride operations. Comput. Ind. Eng. 56 (4), 1721–1734.

Parragh, S.N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transp. Res. C 19 (5), 912–930.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S.H., Ratti, C., 2014. Quantifying the benefits of vehicle pooling with shareability networks. Proc. Natl. Acad. Sci. 111 (37), 13290–13294.

Santos, D.O., Xavier, E.C., 2015. Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. Expert Syst. Appl. 42 (19), 6728–6737.

Shoup, D., 2017. The High Cost of Free Parking: Updated edition. Routledge.

Simonetto, A., Monteil, J., Gambella, C., 2019. Real-time city-scale ridesharing via linear assignment problems. Transp. Res. C 101, 208–232.

Smith, S.L., Pavone, M., Bullo, F., Frazzoli, E., 2010. Dynamic vehicle routing with priority classes of stochastic demands. SIAM J. Control Optim. 48 (5), 3224–3245.

Toregas, C., Swain, R., ReVelle, C., Bergman, L., 1971. The location of emergency service facilities. Oper. Res. 19 (6), 1363–1373.

Vazifeh, M.M., Santi, P., Resta, G., Strogatz, S.H., Ratti, C., 2018. Addressing the minimum fleet problem in on-demand urban mobility. Nature 557 (7706), 534–538.

Wallar, A., Schwarting, W., Alonso-Mora, J., Rus, D., 2019. Optimizing multi-class fleet compositions for shared mobility-as-a-service. In: Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference, ITSC, Auckland, New Zealand, pp. 2998–3005.

Wong, K.I., Wong, S.C., Yang, H., Wu, J.H., 2008. Modeling urban taxi services with multiple user classes and vehicle modes. Transp. Res. B 42 (10), 985–1007.

Zeithaml, V.A., Rust, R.T., Lemon, K.N., 2001. The customer pyramid: Creating and serving profitable customers. Calif. Manage. Rev. 43 (4), 118–142.

Zhang, W., Guhathakurta, S., Fang, J., Zhang, G., 2015. Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. Sustainable Cities Soc. 19, 34–45.