Ye, Jinwen; Pantuso, Giovanni; Pisinger, David

Article

Order dispatching and vacant vehicles rebalancing for the first-mile ride-sharing problem

EURO Journal on Transportation and Logistics (EJTL)

Provided in Cooperation with:
Association of European Operational Research Societies (EURO), Fribourg

Suggested Citation: Ye, Jinwen; Pantuso, Giovanni; Pisinger, David (2024) : Order dispatching and vacant vehicles rebalancing for the first-mile ride-sharing problem, EURO Journal on Transportation and Logistics (EJTL), ISSN 2192-4384, Elsevier, Amsterdam, Vol. 13, Iss. 1, pp. 1-19, https://doi.org/10.1016/j.ejtl.2024.100132

EURO Journal on Transportation and Logistics 13 (2024) 100132

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

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

Order dispatching and vacant vehicles rebalancing for the first-mile ride-sharing problem

Jinwen Ye [a,*], Giovanni Pantuso [a], David Pisinger [b]

[a] University of Copenhagen, Copenhagen, Denmark
[b] Technical University of Denmark, Copenhagen, Denmark

ARTICLE INFO

Keywords:
First-mile
Ride-sharing
Order dispatching
Rebalancing
Rolling horizon

ABSTRACT

Given a set of transport requests to a transit station and a set of homogeneous vehicle, both geographically dispersed in a business area, the First-Mile Ride-Sharing Problem (FMRSP) consists of finding least cost vehicle routes to transport passengers to the station by shared rides. In this paper we formulate the problem as a mathematical optimization problem and study the effectiveness of preventive movements of idle vehicles (i.e., rebalancing) in order to anticipate future demand. That is, we identify promising rebalancing locations based on historical data and give the model incentives to assign vehicles to such location. We then assess the effectiveness of such movements by simulating online usage of the mathematical model in a rolling-horizon framework. The results show that rebalancing is consistently preferable both in terms of profits and service rate. Particularly, in operating contexts where the station is not centrally located, rebalancing movements increase both profits and service rates by around 30% on average.

1. Introduction

Ride-sharing services, which are linked to a reduction of the number of private cars on the road, emissions and congestion (Al-Abbasi et al., 2019), have emerged as a potential solution to the increase in road congestion and air pollution generated by growing urban areas and population (Taniguchi et al., 2014). Such services have yet significant potential for development. As an example, according to the NYC taxicab data (Commission and Limousine, 2023), during January 2020 there were 363,874 taxi trips to the Pennsylvania Station, a fairly busy transit station in New York City see Fig. 1(a), that is, on average 12,129 taxis trips daily to the station. Of these, only 6% were shared by multiple passengers, see Fig. 1(b), which leaves significant margins for more efficient connections to the station.

An effective implementation of ride-sharing services requires adequate responses to potentially frequent changes in demand patterns during the day that may determine geographical mismatches between demand as supply. Fig. 2 illustrates the location of the requests of transportation to Pennsylvania Station during January 2014, showing that the majority of the requests arrive from the North-East area, whereas much fewer requests arrive from the remaining zones of the city. This suggests implementing mechanisms that prepare the geographical distribution of the fleet in such a way to anticipate demand and perhaps reduce waiting and response times as well as service rate.

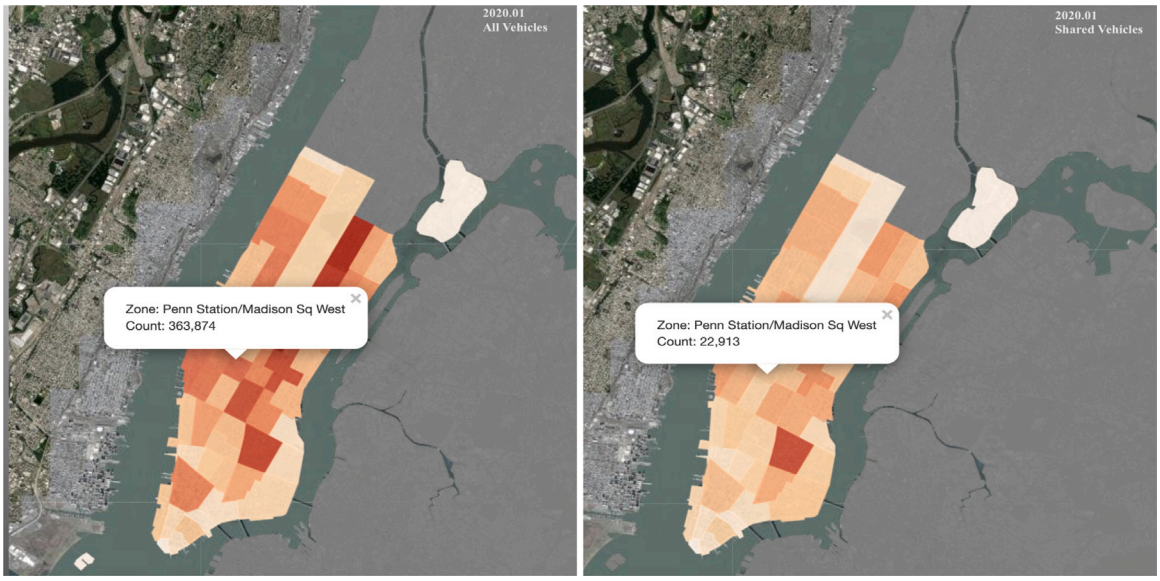
The existing literature study various aspects of ride-sharing services, including pricing mechanisms (Bian and Liu, 2019b,a; Bian et al., 2020; Chen and Wang, 2018), integration with public transport (Shen et al., 2018), order dispatching and vehicle routes (Wang, 2019; Chen et al., 2020). Conversely, strategies for anticipating demand through, e.g. preventive or rebalancing movements (Wen et al., 2018), remain, to a large extent, an open research question. Particularly, efficient ways to simultaneously determine both dispatching and rebalancing movements have, to the best of our knowledge, been neglected.

We contribute to filling this gap by providing a mathematical programming model for joint order dispatch and rebalancing decisions in a first-mile ride-sharing service which transports passengers from their initial location to a common destination (e.g., a transit station). We will refer to this decision problem as the First-Mile Ride-Sharing Problem (FMRSP). In addition, we propose a strategy for identifying promising locations where to rebalance empty vehicles. The model and rebalancing strategies are tested in a rolling-horizon framework which simulates on-line usage.

The rest of this paper is structured as follows. In Section 2 we review the related literature and underline the contribution of this article. In Section 3 we formally introduce the problem and the corresponding mathematical programming model. In Section 4 we describe two methods for deciding where to relocate vehicles in anticipation of future

* Corresponding author.
E-mail addresses: j.y@math.ku.dk (J. Ye), gp@math.ku.dk (G. Pantuso), pisinger@man.dtu.dk (D. Pisinger).

https://doi.org/10.1016/j.ejtl.2024.100132
Received 9 March 2023; Received in revised form 13 March 2024; Accepted 26 March 2024
Available online 28 March 2024



a *All Trips* b *Shared Trips*

Fig. 1. Taxi trips to Penn Station. Data from Commission and Limousine (2023).

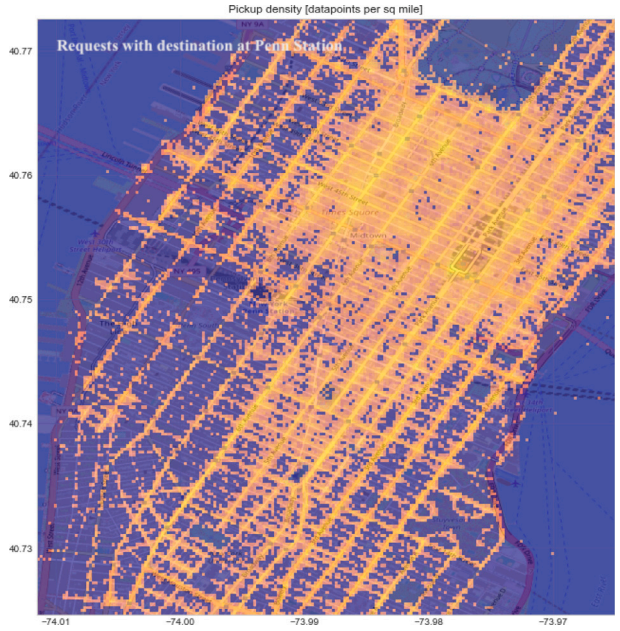


Fig. 2. Distribution of trips to Penn Station.

demand. In Section 5 we describe the simulation framework and the numerical experiments we performed with the model and illustrate the results. Finally, we draw conclusions in Section 6.

2. Literature review

The routing decisions considered in the FMRSP share similarities with those involved in well studied routing problems. Among these we find the Vehicle Routing Problem (VRP). Starting from the seminal paper of Dantzig and Ramser (1959), several exact and heuristics algorithms were proposed to solve VRPs (Bräysy and Gendreau, 2005; Bertsimas et al., 2019; Toth and Vigo, 2002) and several flavors of the problem have been studied, see e.g., the surveys (Kumar and Panneerselvam, 2012; Pillac et al., 2013; Lin et al., 2014; Ritzinger et al., 2016; Braekers et al., 2016). One of the major differences between the FMRSP and the different variants of the VRP is that VRPs typically consist of designing tours returning to the depot, while the FMRSP designs open paths from the vehicle's origins to a common destination. Arguably, a (variant of the) VRP would resemble more closely a last-mile ride-sharing problem where a vehicle departs and returns to the station visiting the destinations of a number of customers. Furthermore, FMRSPs focus on transporting customers from multiple locations to the destination (station) while VRPs are typically concerned with the delivery of goods to customers. This impacts the types of restrictions imposed on the routes.

Particularly, the FMRSP shares features with the Dial-a-ride Problem (DARP) and the Pick-up-and-delivery Problem (PDP) (Cordeau and Laporte, 2003; Ropke and Cordeau, 2009; Berbeglia et al., 2010), which are generalizations of the VRP. A comprehensive review of DARP and PDP can be found in Ho et al. (2018). The goal of the DARP is to minimize the cost/time to transport a set of passenger by means of a fixed fleet of vehicles. Requests have different pickup and delivery locations, and the vehicles can pick up more than one passengers at a time. Also for the DARP different variants can be found, such as where the objective is to minimize the detour for the customers on board the vehicles (Pfeiffer and Schulz, 2022). The DARP can be considered as a variant of the PDP. The PDP typically deals with the transportation of goods while the DARP deals with passenger transportation (Parragh et al., 2008). Thus, the difference between DARP and PDP is usually expressed in terms of additional constraints or objectives that explicitly take user (in)convenience into account (e.g., time window and vehicle capacity constraints). The FMRSP can be seen as a special case of DARP where passengers travel to a common depot (station) and with additional service-specific constraints. In particular, the FMRSP takes the desired arrival time of accepted customers as constraints. This, in turn, implicitly shapes feasible time window for the other customers on board the same vehicle and for the newly arrived customers in a rolling horizon optimization framework. Particularly, in this paper, we study on-line dispatch and rebalancing decisions. That is, we consider the allocation of customers requests to vehicles as they arrive and while vehicles are busy with other transportation requests. This entails dealing two types of customers. First, we find customers whose request has been accepted in previous decision epochs and have not yet picked up. These customers requests must be satisfied. Second, we find new customers whose request may or may not be accepted, similarly to a

Table 1
Summary of the available literature. Under "*Decisions*" we report the main decisions addressed by the article. Matching refers to the assignment of customers to vehicles. Rebalancing refers to the assignment of vehicles to zones. Under "*Service*" we use RS for a general ride-sharing service and FM for a first-mile ride-sharing service. Under "*Model*" a Yes or a No indicate whether the study provides an optimization model or not.

Study	Service	Decisions	Model	Method
Shen et al. (2018)	FM	Fleet size, matching, routing	No	Simulation
Zhao et al. (2018)	RS	Matching	Yes	MILP, Heuristics
Bertsimas et al. (2019)	RS	Matching	Yes	MILP, Heuristics
Wang et al. (2018)	RS	Matching	Yes	MILP, Heuristics
Chen et al. (2020)	FM	Matching	Yes	MILP
Lotfi and Abdelghany (2022)	RS	Matching	No	Heuristics
Santos and Xavier (2015)	RS	Matching	Yes	MILP
Elting and Ehmke (2021)	RS	Matching	Yes	Constraint Satisfaction Problem
Zheng and Pantuso (2023)	FM	Matching	Yes	MILP, Heuristics
Fagnant and Kockelman (2018)	RS	Fleet size, matching	No	Simulation
Lokhandwala and Cai (2018)	RS	Matching	No	Simulation
Mao et al. (2020)	RS	Rebalancing	Yes	MILP, Reinforcement Learning
Noruzoliaee and Zou (2022)	RS	Matching	Yes	MILP
Beirigo et al. (2022)	RS	Matching	Yes	MILP
Bongiovanni et al. (2022)	RS	Matching	Yes	MILP
Wallar et al. (2018)	RS	Zone partition, rebalancing of idle vehicles	Yes	MILP, Poisson process
Wen et al. (2018)	RS	Rebalancing of idle vehicles	No	Reinforcement learning
Alonso-mora et al. (2018)	RS	Matching, rebalancing of idle vehicles	No[a]	MILP
Sayarshad and Chow (2017)	RS	Matching, rebalancing of idle vehicles	Yes	MINLP
Ma et al. (2019a)	RS	Matching, rebalancing of idle vehicles	No	Queue theory

[a] The authors provide a verbal description of the mathematical model.

price-collecting TSP (Balas, 1989) where visiting customers is optional and provides a reward. Finally, empty vehicles may be moved to rebalancing centers in order to position for future demand.

Due to the fast development of GPS technology and widespread use of smart-phones, ride-sharing services enabled by mobile applications have attracted broad attentions. Commercial companies such as Uber and Didi have implemented versions of the service (Xu et al., 2018; Lin et al., 2018). The attention of the research community has grown providing both optimization methods (Stiglic et al., 2015; Masoud et al., 2017; Masoud and Jayakrishnan, 2017; Alonso-mora et al., 2018; Stiglic et al., 2018; Huang et al., 2014; Wang et al., 2018; Mourad et al., 2019) and reinforcement learning methods (Xu et al., 2018; Lin et al., 2018; Li et al., 2019; Tang et al., 2019; Qin et al., 2019).

The FMRSP and, in general, ride-sharing problems are, however, a relatively new family of problems and the corresponding literature is somewhat sparse. In what follows we review the available literature before highlighting how our work extends the state-of-the-art. The literature is also summarized in Table 1 for the reader's convenience.

Shen et al. (2018) study the integration of a FMRS service based on autonomous vehicles (AVs) with public transportation. The idea is to preserve high demand bus routes while using shared AVs as an alternative for low demand routes. In a simulation framework they use simple heuristics to match passengers to vehicles and define routes. Chen et al. (2020) provide a mixed-integer linear programming (MILP) model to decide the assignment of request groups to AV in FMRS service. The objective is that of minimizing operational costs. The authors devise a cluster-based solution method to deal with large-scale instances. Zhao et al. (2018) address the joint problem of optimally matching passengers and vehicles and that of routing each vehicle. The problem is formulated as a PDP with the addition of space–time windows. Wang et al. (2018) consider a ride-share setting in which a ride-share provider receives trip requests over time from potential participants. A trip can be either a driver or a rider. This process generates two disjoint sets of trip requests. The authors focus on finding a stable match between the two sets. Bertsimas et al. (2019) study the problem of assigning customers to vehicles. The authors include service-specific constraints, including time windows and latest time for accepting or rejecting a transportation request. The authors address the problem via periodic re-optimization. Lotfi and Abdelghany (2022) consider a set of passengers requiring a ride and study the problem of assigning passengers to vehicles. This set of passengers is known at the time the problem is solved. Passengers are characterized by origin, destination, earliest pick-up time, latest drop-off time and willingness to share the ride. The authors consider two objectives, namely maximizing profit and maximizing passengers' travel experience, measured in terms of transfers and travel times. The problem is solved by means of a heuristic. In the study of Santos and Xavier (2015) users decide whether to share either their own car or a taxi. They specify pickup and drop-off location, earliest pick-up time, latest drop-off time and the maximum cost tolerated. In addition, car owners also specify the departure time and the maximum accepted delay. The authors address the problem of matching users to vehicles and of determining the routes. Elting and Ehmke (2021) assess the economic potential of shared taxi services. Here a service operator collects requests from individual travelers. Based on every request's origin and destination as well as the desired pick-up time, the service provider matches vehicles and travelers and builds a route plan for each vehicle. Zheng and Pantuso (2023) consider a FMRS service where passengers have to be transported to a common destination. The problem consists of assigning customers to vehicles and deciding vehicle routes. Transport requests may be rejected. The problem is formulated as a bi-objective MILP where the two conflicting objectives are travel costs and service rates. Fagnant and Kockelman (2018) consider an AV-based ride-sharing service. They use simulation to find the best fleet size. In their simulation they match passengers to AVs based on specific rules, such as assigning passengers to the nearest vehicle. Their study shares similarities with (Lokhandwala and Cai, 2018) who also propose a simulation study for quantifying the environmental impact of ride-sharing with AVs over traditional taxis. In their simulation framework, passengers are assigned to vehicles based on a detailed algorithm which takes into account the preferences of the customer and looks for suitable AV routes. Noruzoliaee and Zou (2022) study the problem of assigning the multiple requests (from different origins and destinations) to shared AVs with the scope of avoiding undesired rider en-route transfers. Beirigo et al. (2022) also study the assignment of requests to vehicles in an AV-based ride-sharing system where users differ according to expectations in terms of responsiveness, reliability, and privacy. The authors assume possible that privately owned freelance AVs can be hired on short notice. They propose a multi-objectives MILP which optimizes vehicle occupancy, number of AVs used, service level violation, and the waiting times. Bongiovanni et al. (2022) also study the problem of assigning passengers to AVs. They propose a two-phase heuristic which assigns new requests AVs and subsequently re-optimizes such assignments through intra- and inter-vehicle route moves.

A number of studies address the problem of rebalancing vehicles. Wen et al. (2018) propose a reinforcement learning method to move idle vehicles in a shared mobility-on-demand systems. They test their solution method on a first-mile ride-sharing service in the city of London. Mao et al. (2020) consider a taxi sharing systems with AVs. They study the problem of determining the number of AVs to send from a zone of the city to another in order to minimize the expected cost of repositioning AVs. They compare a reinforcement learning algorithm with an integer programming model that assumes full knowledge of future demand. Wallar et al. (2018) propose algorithms for partitioning the operating area into zones, estimating the real-time demand and rebalancing idle vehicles. Sayarshad and Chow (2017) propose a queue-based model for matching and rebalancing decisions. They assume that the number of idle vehicles is known in advance. Alonso-mora et al. (2018) design a matching algorithm for on-demand ride-sharing. The method incorporates rebalancing decisions for idle vehicles. The authors describe a MILP formulation for the problem and solve the problem via a specialized procedure that begins by assigning passengers to vehicles and finding feasible trips, and terminates by rebalancing idle vehicles. Ma et al. (2019a) study a more involved system in which a ride-sharing fleet is operated jointly with public transport services in order to arrange complete on-demand journeys for their customers. The authors consider also the rebalancing of idle vehicles. They propose a queueing-theoretic model for the problem.

As it is evident in Table 1, the available literature has typically addressed matching decisions (i.e., the assignment of passengers to vehicles) and rebalancing decisions (i.e., the assignment of vehicles to zones) separately. Furthermore, when rebalancing decisions are addressed, they concern mainly idle vehicles, that is vehicles which have not been dispatched to customer requests. Thus, matching and rebalancing decisions have been understood as sequential decisions. First, vehicles match current requests, then the remaining ones may be rebalanced. Finally, it is possible to notice that not all articles that study rebalancing decisions provide an optimization model for that. We extend the state-of-the-art in the following ways:

1. We address matching, routing and rebalancing decisions simultaneously. This entails that we do not necessarily rebalance only idle vehicles. In our approach, vehicles may move to promising demand areas in advance, even if this entails giving up the profit of a current request.
2. We consider online optimization with binding acceptance of transportation requests. This entails that a subset of the customers (those whose request has been accepted in previous decision problems) must be serviced, while the remaining customers (those newly arrived) may be picked up if feasible and profitable. A side effect of this is that previously and newly accepted customers have an impact on the time window of the vehicle.
3. For this problem we provide an explicit mathematical model. The model includes service-specific constraints such as maximum waiting time, latest arrival time, and the necessity of fulfilling binding acceptance of transport requests.
4. We propose simple techniques to identify promising locations where to rebalance.
5. We test our model in a rolling-horizon simulation framework with periodic re-optimization based on randomly generated instances to assess the solutions delivered by the model and, particularly, the advantage provided by rebalancing activities.

It must be noted that rebalancing decisions have been extensively studied in other emerging problems in shared mobility. These include carsharing (Illgen and Höck, 2019; Folkestad et al., 2020; Pantuso, 2022), bike sharing (Faghih-Imani et al., 2017; Liu et al., 2016; Chemla et al., 2013), scooter sharing (Osorio et al., 2021). Nevertheless, the relocation problem involved is significantly different. In the ride-sharing problem, a vehicle has to drive (with its own driver) to a more promising location. In the other vehicle-sharing problems, vehicles have to be picked up by drivers or service vehicles to be moved to more promising locations. The amount of work in the latter is typically much higher and the relocation problem alone may involve complex optimization problems. Similarities may emerge in the methods used to predict demand occurrence. However, we believe the methods proposed are not immediately applicable due to the inherent differences in the systems and types of demand.

3. The first-mile ride-sharing problem

In this section, we formally introduce the First-Mile Ride-Sharing Problem. We start, in Section 3.1, by providing a general introduction to the problem. Following, in Section 3.2, we introduce a mathematical model for the FMRSP. In addition, in Appendix A we provide a table that summarizes the notation and in Appendix B we provide a simple example that illustrates possible feasible solutions to the problem.

3.1. Problem statement

We consider the operator of a fleet of vehicles $\mathcal{K} := \{1, \dots, K\}$ concerned with dispatch and relocation decisions in order to ensure a first-mile ride-sharing service. The fleet is homogeneous with capacity Q. We assume the operator makes dispatch and relocations decisions periodically, e.g., every 5 or 10 min, as a result of the arrival of new transportation requests. We refer to these decision times as "(re)-optimization phases". At each re-optimization phase, the available customers can be partitioned in two sets, namely $\mathcal{N}_P := \{1, \dots, N_P\}$ which contains the customers whose transportation request had already been accepted during a previous optimization phase, and $\mathcal{N}_C := \{1, \dots, N_C\}$ which contains newly arrived customer requests which have not been considered in previous optimization phases. We assume that the customers in $\mathcal{N}_C$ may be either accepted (and thus assigned to a vehicles) or rejected, while the customers in $\mathcal{N}_P$ must be picked up (thus we assume acceptance decisions are binding). For convenience we set $\mathcal{N}_U := \mathcal{N}_C \cup \mathcal{N}_P$.

All customers travel to a common destination d located in position $o(d)$ (e.g., a transit station) and for each customer i, the operator knows the requested pick-up time T_i^P, the requested arrival time T_i^A and the origin $o(i)$. We let Δ be the maximum waiting time (i.e., difference between actual pick-up time and requested pick-up time). Similarly, at the beginning of the re-optimization phase, denoted T, each vehicle k is located at $o(k)$ as a result of previous deployment or relocation decisions. The vehicle is either idle in its location, or traveling between customers or to the station. In addition, vehicles might initially have customers on board. We denote V_k the number of customers on board of vehicle k at the beginning of the re-optimization phase and T_k the earliest arrival time of the passengers already on board vehicle k. The operator needs to ensure that vehicles with customers on board terminate their journey to the station. Conversely, vehicles with no customers on board may be sent to a rebalancing point or stay at their origin location $o(k)$. A set $\mathcal{R} := \{1, \dots, R\}$ of potential rebalancing points in the operating area is available. For each rebalancing point r we let B_r denote an upper bound on the number of vehicles that can be dispatched to the rebalancing point.

The operator bears transportation costs generated by vehicle movements. Particularly, we assume travel times are known, with T_{ij} being the travel time between locations $o(i)$ and $o(j)$ with $i \in \mathcal{K} \cup \mathcal{N}_U$, $j \in \mathcal{N}_U \cup \mathcal{R} \cup \{d\}$ and cost C is born for each unit of travel time. The operator collects a revenue P_i when picking up customer i, for $i \in \mathcal{N}_C$. Note that the revenue is collected only when picking up new customers as we assume the revenue for the customers in $\mathcal{N}_P$ has been collected during previous optimization phases. Furthermore, E_r denotes the expected revenue collected for each vehicle relocated to rebalancing center $i \in \mathcal{R}$. Parameter E_i is calculated as $\bar{P}_i - CT_{id}$, where $\bar{P}_i$ is the

expected revenue obtained from dispatching a vehicle to rebalancing center $i \in \mathcal{R}$. Expected future revenues from rebalancing activities are discounted using a parameter β that denotes the weight of the rebalancing reward.

The decisions made by the operator can be formalized as follows. We let x_{ij}^k take value 1 if vehicle k moves directly between $o(i)$ and $o(j)$, 0 otherwise, for all $i \in \{k\} \cup \mathcal{N}_U, j \in \mathcal{N}_U \cup \mathcal{R} \cup \{d\}, k \in \mathcal{K}$. Furthermore, we let t_k^A denote the actual arrival time of vehicle k to the station, for $k \in \mathcal{K}$ and t_i^P denote the actual pick-up time of customer i, for $i \in \mathcal{N}_U$.

Thus, we use a 3-index formulation of size $O(|\mathcal{N}_C||\mathcal{K}||\mathcal{R}|)$. The FMRSP is NP-hard, as it contains the prize-collecting TSP (Balas, 1989) as a special case.

3.2. Mathematical model

Having defined all decision variables and parameters, we may formulate the problem as follows.

$$\max \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}_C} \sum_{j\in\mathcal{N}_U\cup\{d\}} P_i x_{ij}^k - \sum_{i\in\{k\}\cup\mathcal{N}_U} \sum_{j\in\mathcal{N}_U\cup\mathcal{R}\cup\{d\}} \sum_{k\in\mathcal{K}} CT_{ij} x_{ij}^k + \beta \sum_{i\in\mathcal{R}} \sum_{k\in\mathcal{K}} x_{ki}^k E_i \tag{1a}$$

$$\text{s.t.} \sum_{j\in\mathcal{N}_U\cup\{d\}} \sum_{k\in\mathcal{K}} x_{ij}^k \le 1 \qquad \forall i \in \mathcal{N}_C \tag{1b}$$

$$\sum_{j\in\mathcal{N}_U\cup\{d\}} \sum_{k\in\mathcal{K}} x_{ij}^k = 1 \qquad \forall i \in \mathcal{N}_P \tag{1c}$$

$$\sum_{i\in\mathcal{N}_U\cup\mathcal{K}} x_{id}^k \le 1 \qquad \forall k \in \mathcal{K} \tag{1d}$$

$$\sum_{i\in\mathcal{N}_U\cup\{k\}} x_{ij}^k = \sum_{i\in\mathcal{N}_U\cup\{d\}} x_{ji}^k \qquad \forall j \in \mathcal{N}_U, k \in \mathcal{K} \tag{1e}$$

$$\sum_{j\in\mathcal{R}\cup\mathcal{N}_U\cup\{d\}} x_{kj}^k = \sum_{j\in\mathcal{N}_U\cup\{k\}} \sum_{i\in\mathcal{R}\cup\{d\}} x_{ji}^k \qquad \forall k \in \mathcal{K} \tag{1f}$$

$$\sum_{i\in\{k\}\cup\mathcal{N}_U} \sum_{j\in\mathcal{N}_U} x_{ij}^k + V_k \le Q \qquad \forall k \in \mathcal{K} \tag{1g}$$

$$\sum_{k\in\mathcal{K}} x_{kj}^k \le B_j \qquad \forall j \in \mathcal{R} \tag{1h}$$

$$V_k \le Q(1 - \sum_{j\in\mathcal{R}} x_{kj}^k) \qquad \forall k \in \mathcal{K} \tag{1i}$$

$$V_k \le Q \sum_{j\in\mathcal{N}_U\cup\{d\}} x_{kj}^k \qquad \forall k \in \mathcal{K} \tag{1j}$$

$$t_i^P + T_{ij} \le t_j^P + T^L(1 - \sum_{k\in\mathcal{K}} x_{ij}^k) \qquad \forall i, j \in \mathcal{N}_U \tag{1k}$$

$$T + T_{kj} \le t_j^P + T^L(1 - x_{kj}^k) \qquad \forall j \in \mathcal{N}_U, k \in \mathcal{K} \tag{1l}$$

$$t_i^P - T_i^P \le \Delta \qquad \forall i, j \in \mathcal{N}_U \tag{1m}$$

$$t_k^A \le T_i^A + T^L(1 - \sum_{j\in\mathcal{N}_U\cup\{k\}} x_{ji}^k) \qquad \forall i \in \mathcal{N}_U, k \in \mathcal{K} \tag{1n}$$

$$t_k^A \le T_k \qquad \forall k \in \mathcal{K} \tag{1o}$$

$$t_j^P + T_{jd} x_{jd}^k \le t_k^A + T^L(1 - x_{jd}^k) \qquad \forall j \in \mathcal{N}_U, k \in \mathcal{K} \tag{1p}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall i \in \{k\} \cup \mathcal{N}_U, j \in \mathcal{N}_U \cup \mathcal{R} \cup \{d\}, k \in \mathcal{K} \tag{1q}$$

$$t_k^A \in \mathbb{R}^+ \qquad \forall k \in \mathcal{K} \tag{1r}$$

$$t_i^P \in \mathbb{R}^+ \qquad \forall i \in \mathcal{N}_U \tag{1s}$$

Objective function (1a) represents the profit for the operator. The first term represents the revenue generated by picking up customers, the second term the total cost born of the vehicles movements and, finally, the third term is the discounted expected profit obtained in the rebalancing centers.

Constraints (1b) and (1c) state that new customers may be picked up at most once and customers already accepted must be picked up exactly once, respectively. Observe, in (1b) and (1c), that after visiting a customer $i \in \mathcal{N}_C$ or $i \in \mathcal{N}_P$, the vehicle can only move to another customer $i \in \mathcal{N}_P \cup \mathcal{N}_C$ or to the station d. Constraints (1d) ensure that vehicles travel to the station at most once. Constraints (1e) state that whenever a vehicle arrives at a customer location, it must then move to another customer or to the station. Notice that a vehicle can arrive at a customer location j either from another customer or from the vehicle's original location $o(k)$. We remind the reader that variable x_{kj}^k must be understood as vehicle k moving from its original location $o(k)$ to the location $o(j)$ of customer j. Constraints (1f) state that, if a vehicle departs from its original location $o(k)$, i.e., $x_{kj}^k = 1$ for some j, it must terminate its journey either at the station or at a rebalancing point. Also in this case, variables x_{kj}^k must be understood as the vehicle moving from its origin, $o(k)$, see Section 3.1.

Constraints (1g) ensure that the capacity of the vehicles is not exceeded, while constraints (1h) ensure that the total number of vehicles dispatched to a rebalancing center will not exceed the upper bound on the vehicles dispatchable at the rebalancing center. Constraint (1i) state that only empty vehicles may be dispatched to rebalancing centers. For instance, if vehicle k is dispatched to one of the rebalancing center, the right-hand-side becomes 0, and the constraints can only be satisfied when V_k is equal to 0. If vehicle k is not dispatched to any rebalancing center, the right-hand-side reduces to the capacity of the vehicle, and the constraint holds with any value of V_k. Notice that the movements between customer points $\mathcal{N}_U$ and rebalancing points are automatically forbidden by the absence of the corresponding x_{ij}^k variables. Constraints (1j) state that the vehicles that already have customers on board at the beginning of the period must be dispatched (i.e., cannot stay idle). If V_k is strictly positive, the constraint forces the right-hand-side to be strictly positive as well, and thus to dispatch the vehicle.

Constraints (1k) state that if customer j is picked up by vehicle k immediately after picking up customer i, then the actual picking up time of customer i plus the travel time between customer i and j must be less or equal to customer j's actual pick up time. Here $T^L := max_i\{T_i^A\}$ for $i \in \mathcal{N}_U$ is an upper bound on the requested arrival time. Similarly, constraints (1l) denote the pick-up time for the first customers in the route. Constraints (1m) ensure that the difference between the actual pick up time and the requested pick up time of the customer does not exceed the maximum waiting time Δ. Constraints (1n) ensure that the actual arrival time of vehicle k must be earlier than the requested arrival time of any of the customers on board of it. For instance, if customer i is picked up by vehicle k, the right-hand-side becomes T_i^A enforcing that the actual arrival time of vehicle k is before T_i^A. Constraints (1o) ensure that the actual arrival time of vehicle k is earlier than the earliest requested arrival time T_k of the passengers

on board at the beginning of the re-optimization phase. Constraints (1p) state the relationship between pick-up time and arrival time. For example, if j is the last customer picked up by vehicle k before the station x^k_{jd} takes value 1, the left-hand-side becomes $t^P_j + T_{jd}$, and the right-hand-side becomes t^A_k, enforcing that the actual pick-up time of customer j plus the travel time between customer j and station be less than or equal to the actual arrival time of vehicle k. If j is not the last customer picked up by the vehicle k before arrive at the station, then the left-hand-side becomes t^P_j, the right-hand-side becomes $t^A_k + T^L$, which always holds. Finally, constraints (1q)–(1s) define the domain of the decision variables.

An illustrative example of possible solutions to the problem is provided in Appendix B.

4. Finding rebalancing centers

Identifying where to rebalance in order to anticipate demand, and how many vehicles to send to each rebalancing point is currently an open research question. Any such prediction model could be used to feed rebalancing centers to model (1). In this section we introduce a clustering-based methods for identifying rebalancing centers. We refer to the method as the *K-means Clustering* (KC) method. The method identifies both the location and demand of the rebalancing centers, and this in turn allows us to determine the upper bound on the number of vehicles dispatchable to the different rebalancing centers.

Given a number k of rebalancing centers to find, the KC method finds rebalancing centers by partitioning all requests received in the current re-optimization phase into $k = |\mathscr{R}|$ clusters. Clusters are created in such a way as to minimize the total distance between the points allocated to the cluster and the centroid of the cluster. The centroids of the clusters will then be used as rebalancing centers. The expected demand (number of requests) of the rebalancing centers will be set equal to the number of requests in the corresponding cluster. We let D_r be the demand of rebalancing point $r \in \mathscr{R}$.

The rational behind the clustering method is the following. Assume that the decision maker performs frequent re-optimizations and that the demand distribution changes slow enough. Then the geographical distribution of demand in the near future is approximately the same as the current demand. Current requests represent a sample from this (unknown) distribution. Thus, over many repetitions one expects to put rebalancing centers where there is actually more demand. We believe our assumption of frequent re-optimizations and demand changing slower than the re-optimizations is reasonable in real-life. However, clearly, the prediction accuracy of the KC method is expected to fall when either (i) there is no underlying pattern in the demand (we argue that any prediction method would probably fail in this case) and (ii) when the demand changes rapidly or re-optimizations are not performed sufficiently frequently. In the computational study we assess both cases.

In Section 5 we compare the KC method against two benchmarks, namely random selection and no rebalancing. The random selection method (hereafter named RS method) consists of randomly selecting $|\mathscr{R}|$ points in the operating area as rebalancing centers. To each rebalancing center is assigned a demand equal to the number of customer requests of the current re-optimization phase within a certain distance (e.g., 1 km) of the rebalancing center. No rebalancing entails $\mathscr{R} = \emptyset$.

For both the KC and RS methods, the upper bound B_r of the number vehicles that can be dispatched to each rebalancing center r is defined as $B_r = D_r/\bar{Q}$, where $\bar{Q}$ denotes the average number of customer on board a vehicle during one trip.

5. Numerical experiments

In this section we report the results of our numerical experiments. The scope of the experiments is to assess, in terms of profits and service rates, two different configurations of the service which we refer to as *without rebalancing* (`woR`) and *with rebalancing* (`wR`). The configuration `woR` refers to the situation where the service provider dispatches the vehicles only based on customer requests of the current re-optimization phase. For the model without rebalancing, we simplify our model in Section 3 by having an empty set of rebalancing centers. In the configuration `wR`, the service provider makes the dispatching decision based both on current customer requests and on predicted demand using rebalancing centers. In this case, we test the model with two different method for finding rebalancing centers. In the first case, which we refer to as `wRKC`, the company use KC to obtain the location and demand of the rebalancing centers. In the second case, which we refer to as `wRRS`, the company uses the RS method to find the locations and demands of the rebalancing centers.

Observe that the potential value of relocation activities in on-demand mobility has been the focus of other studies, such as Ma et al. (2019b), Jamshidi et al. (2021), Sayarshad and Chow (2017), Kash et al. (2022) and Danassis et al. (2022) for different configurations of the service. Particularly, our work shares similarities with Sayarshad and Chow (2017) who also explicitly model the decision of the service provider as an optimization model. However, with respect to Sayarshad and Chow (2017) we consider (i) binding previously accepted requests, (ii) customers desired arrival times and (iii) an upper bound on the maximum waiting time. We believe our study can provides evidences based on a more involved setup of the service.

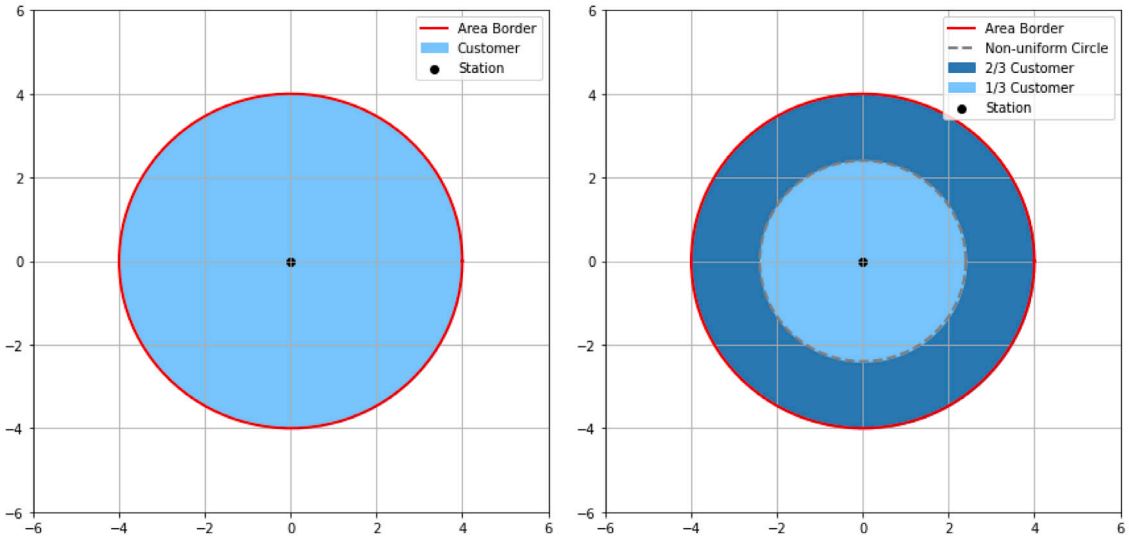
The different configurations are tested on a set of random instances introduced in Section 5.2. All problems are solved with the Python libraries of GUROBI 9.5.0 and a server equipped with Intel Core i5 and 16 GB of RAM.

5.1. Simulation framework

We test our model in a simulation framework, based on rolling-horizon optimization, with a planning horizon of one hour. We assume online re-optimization happens every 5 minutes. This means that every simulation requires the solution of 12 optimization problems (1). At each re-optimization we update the status of the system, and randomly generate (as explained in the next section) a number of new customers. Particularly,

- At the initial optimization phase, say $T = 0$, we assume that the $\mathscr{N}_P$ is empty. This means that there is no customer whose request had already been accepted in a previous optimization phase. We generate a number of new customers $\mathscr{N}_C$, rebalancing centers $\mathscr{R}$, and initial vehicle positions (as explained in the next section) and solve the resulting model (1).
- We then step five minutes forward in time, say optimization phase $T = 1$, and assume the solution to the model for $T = 0$, has been implemented. This entails the vehicle followed the routes determined by the previous optimization model for five minutes, moving either to customers locations or to rebalancing centers. This provides their updated location for the new re-optimization phase. We then partition the customers of the previous optimization phase into three groups:

    1. The first group contains those customers that had been assigned to a vehicle but have not yet been picked up in the five minutes interval between the two re-optimizations (i.e., the route of the vehicle assigned to the customer did not stop by the customer within the five-minute interval between re-optimizations). These customers form the set of mandatory customers $\mathscr{N}_P$ in the new re-optimization phase and must be picked up by some vehicle (possibly different from the one assigned in the previous re-optimization phase).



a *Uniform* b *Non-uniform*

Fig. 3. Uniform and non-uniform distribution of customers with the station located in the center.

2. The second group contains those customers that had been assigned to a vehicle in the previous re-optimization phase and the route of the assigned vehicle stopped by the customer within the five-minute interval between re-optimizations. In the new re-optimization phase these customers represents occupied seats (V_k) in the vehicles to which they were assigned. Therefore, these customers represent fulfilled requests and do not appear in $\mathcal{N}_P$ in the new re-optimization phase.
3. The third group contains those customers that had not been assigned to a vehicle in the previous re-optimization phase. These represent customers whose request has been rejected and will not show up in the new re-optimization phase.

Observe, that in the new re-optimization phase, vehicles may find themselves into one of the following situations. (A) The vehicle is empty at a given position and was on the way to pick-up customers or to a rebalancing centers. (B) The vehicle has passengers on board at a given position and was on the way to pick-up additional customers or to the station. In case (A), in the new re-optimization phase the vehicles may be assigned to new customers or to a rebalancing center, independently of the decision made in the previous re-optimization phase. That is, it is possible that the vehicle is assigned to a set of customers different from the ones previously assigned to the vehicle. In case (B) the vehicle cannot be sent to a rebalancing center but may be assigned to new customers or sent directly to the station. Following, we generate new customers $\mathcal{N}_C$ for the new re-optimization phase and resolve a problem (1).

The procedure continues stepping five minutes forward in time until the end of the one-hour planning horizon. Thus, we are able to collect statistics on the performance of the service. Particularly, the profit is computed as follows. At the end of each re-optimization phase, we collect the fee for all the customers that have been accepted (i.e., a vehicle has been assigned to them) and picked up (i.e., the vehicle has arrived at their location during the five-minute interval between re-optimizations) and subtract the cost of the movements the vehicles have done during the five-minute interval between re-optimizations. The final total profit is then the sum of the individual profits made during the one-hour planning horizon. The procedure is explained by the following example.

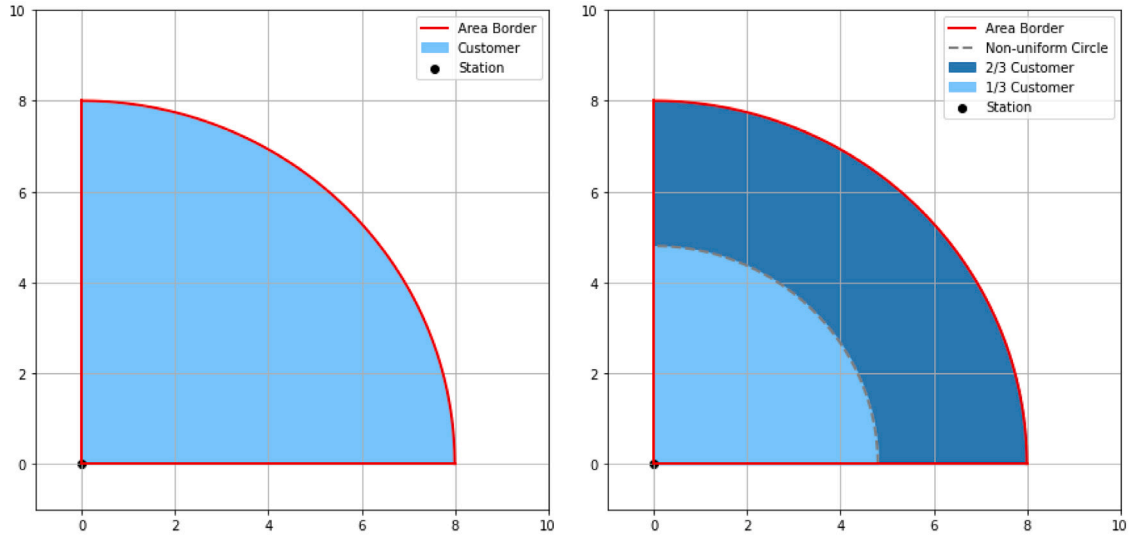
- Assume that at $T = 0$, $\mathcal{N}_P$ is empty and $V_k = 0$ for all vehicles $k \in \{D_1, D_2, D_3\}$. That is, there is no customer whose request had already been accepted in a previous optimization phase. We generate a number (say four) of new customers $\mathcal{N}_C := \{C_1^0, C_2^0, C_3^0, C_4^0\}$, rebalancing centers $\mathcal{R} := R_1^0$, and initial vehicle positions $o(D_1), o(D_2), o(D_3)$ and solve the resulting model (1). Assume that the solution determines the following routes for the three vehicles: $\texttt{Route1}^0 := \{D_1, C_1^0, C_3^0, d\}$, $\texttt{Route2}^0 =: \{D_2.C_2^0, d\}$, $\texttt{Route3}^0 := \{D_3, R_1^0\}$ and that customer C_4^0 is rejected.
- The solution computed at $T = 0$ is implemented and the vehicle follow the respective routes for five minutes. This provides updated system information for optimization phase $T = 1$. That is, we updated the location of the vehicles $\{o(D_1), o(D_2), o(D_3)\}$. We observe that,

  1. For $\texttt{Route1}^0$, the vehicle D_1 already picked up customer C_1^0, and is still on the way to pickup C_3^0, so we can delete C_1^0 from the system, update $V_{D_1} = 1$ and move C_3^0 to mandatory customer set $\mathcal{N}_P := \{C_3^0\}$.
  2. For $\texttt{Route2}^0$, vehicle D_2 already picked up customer C_2^0 and is still on the way to station, so we can delete C_2^0 from the system and update $V_{D_2} = 1$.
  3. For $\texttt{Route3}^0$, vehicle D_3 arrived at the rebalancing center R_1^0, thus $V_{D_3} = 0$.

  In the new optimization phase $T = 1$, since the vehicles D_1, D_2 already have customers on board, they cannot be sent to a rebalancing center but may be assigned to new customers or sent directly to the station. However, vehicle D_3 may be assigned to new customers or to a rebalancing center as it is still empty. Following, we generate new customers $\mathcal{N}_C := C_1^1, C_2^1, C_3^1, C_4^1$ for the new re-optimization phase $T = 1$ and resolve the problem (1).

5.2. Instance generation

We generate a number of artificial and randomly generated instances that mimic real-life operating scenarios for the service. Particularly, we assume a fleet of homogeneous vehicles of capacity $Q = 4$. The position of the vehicles for the first re-optimization phase is generated randomly in the business area (defined below) and initially vehicles are assumed to have no passenger on board. For the re-optimization phases other than the first, the position of the vehicles, and the number of



a *Uniform* b *Non-uniform*

Fig. 4. Uniform and non-uniform distribution of customers with the station located in a corner.

passengers on board is computed as the result of previous optimization phases.

We consider two different geographies of the business area. In the first geography, depicted in Fig. 3, the station is located at the center of the business area, and the business area itself is represented by a circle of radius $R = 4$ km. In the second geography, see Fig. 4, the station is located in a corner and the business area is a quarter of a circle of radius of $R = 8$ km. The second geography is meant to represent urban contexts where the demand is concentrated only on one side of the station due to, e.g., physical barriers such as rivers or harbors.

For each geography, and for each re-optimization phase, customer requests are generated in the following two different scenarios of demand distribution. In the first scenario, referred to as the *uniform demand scenario*, pickup locations are randomly scattered in the whole business area, see Figs. 3(a) and 4(a). In the second scenario, referred to as the *non-uniform demand scenario*, one third of the requests arrive, randomly, from inside the inner circle of radius $R^I = 0.6R$ (where R is the radius of the outer circle), while the remaining requests arrive, randomly, from the outer portion of the circle, see Figs. 3(b) and 4(b). We obtain, in total, four configurations namely

1. station in the center and uniform demand (`UCt`),
2. station in the center and non-uniform demand (`NUCt`),
3. station in the corner and uniform demand (`UCn`),
4. station in the corner and non-uniform demand (`NUCn`).

For each request, the requested pickup time (T_i^P) is randomly generated uniformly between 0 and 3 minutes after the beginning of the planning horizon, and the requested arrival time (T_i^A) is set as the sum of requested pickup time, T_i^P, travel time between customer i and the station d, and a buffer time randomly generated between 5 and 8 minutes. Travel time T_{ij} are calculated using Euclidean distances and assuming an average speed of 36 Km/h (Commission and Limousine, 2023). The unit transportation cost C is set to \$11.25/h (English, 2008) and trip revenues P_i are computed using a fare of \$2.59 per Km traveled plus \$0.74 per minute traveled, with a minimum fare of \$8 following the setting in INSHUR (2022).

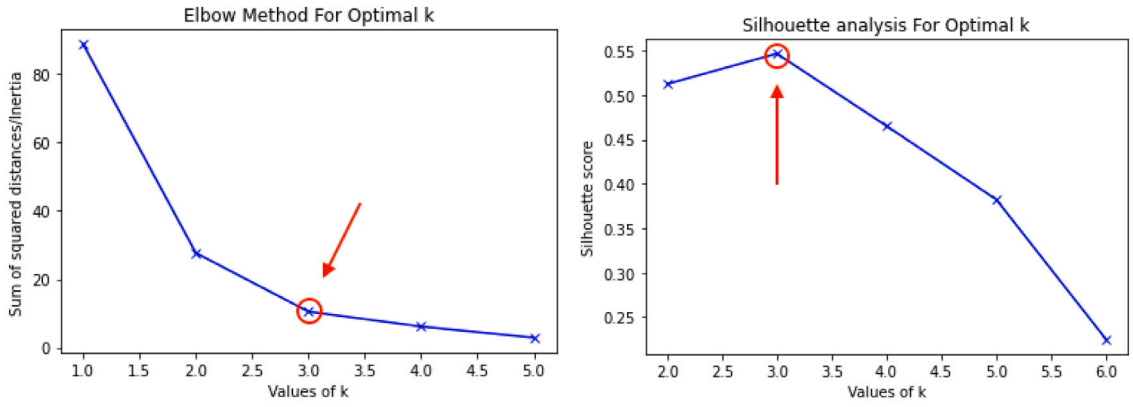
We set the value of β to 0.1 in the objective function, unless otherwise specified. Observe that β determines the impact of rebalancing movements. High values will increase the potential benefit of rebalancing and might lead to rejecting current customers while low values might lead the model to provide myopic decisions. The impact of different values of β will be assessed through sensitivity analysis in Section 5.5. The expected revenue for rebalancing a vehicle to center $i \in \mathscr{R}$, E_i, is calculated as $\bar{P}_i - CT_{id}$, where $\bar{P}_i$ is the revenue of dispatching a vehicle to rebalancing center $i \in \mathscr{R}$, and is calculated as $\bar{P}_i = Q\ P_i^A$, where P_i^A the average revenue for the requests in the cluster where i is the centroid and Q is the capacity of the vehicles. C is the unit transportation cost, set as above, and T_{id} is the distance between rebalancing point i and the station. To obtain the upper bound on the number of vehicles dispatchable to a rebalancing center (B_r, see Section 4) we set the average number of customers on board during one trip of a vehicle $\bar{Q}$ equal to half of the capacity $Q = 4$.

For each configuration, we generate different instances varying in the number of vehicles and customers that appear at each new re-optimization phase. Particularly, we create instance classes named $C|\mathscr{N}_C|V|\mathscr{K}|$ with number of customers $|\mathscr{N}_C| \in \{6, 7, 8\}$, and number of vehicles, $|\mathscr{K}| \in \{10, 12, 14\}$. As an example, $C8V10$ indicates a class of instances with 8 new customers in each re-optimization phase and 10 vehicles available for dispatching for the whole planning period. For each instance class we randomly generate 3 different instances. Observe, however, that for each instance we solve 12 different optimization problems in our simulation framework.

We set the number of rebalancing centers $|\mathscr{R}|$ to 3 in all instances (later we perform sensitivity analysis with respect to this parameter.) . This number is found using the *Elbow Method* (EM) and the *Silhouette Analysis Method* (SAM) (Mahendru, 2019). Particularly, we use the instances with $|\mathscr{N}_C| = 8$ as a reference case to find a suitable value for $k = |\mathscr{R}|$. First we randomly generate 8 customers in the operating area, then we use the EM approach to show the performance of the KC method for different values of k. For each k we consider the *Within-Cluster-Sum of Squared Errors* (WSS). We then plot the WSS versus k, and choose the value of k for which the WSS flattens. This point is referred to as an "Elbow", see Fig. 5(a).

Nevertheless, in some cases, the EM does not give precise answers as it is not always clear for which value of k the change of slope is significant. In such cases, we will use the SAM to make a decision. The silhouette score is a measure of how similar a customer request is to its own rebalancing cluster compared to other rebalancing clusters. To be more precise, the silhouette score of one customer request i can be calculate as below:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2}$$



a *Elbow Method* b *Silhouette Analysis*

Fig. 5. Identifying a suitable number of clusters.

Table 2
Profit [$] in the simulations with $|\mathscr{R}| = 3$.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	57.52	59.06	58.20	68.15	68.31	71.14	79.16	81.46	85.17
	wRRS	55.62	58.04	55.54	64.59	67.58	70.51	75.36	80.47	77.63
	woR	56.37	56.61	56.42	63.96	67.08	65.72	74.71	77.52	78.71
UCt	wRKC	54.35	57.53	54.90	60.64	64.79	65.54	72.78	74.55	76.47
	wRRS	51.41	53.99	53.45	60.97	62.20	60.98	70.65	70.52	74.08
	woR	50.88	55.03	54.35	61.46	61.68	65.22	66.79	71.78	68.77
NUCn	wRKC	98.00	105.55	108.57	114.57	121.98	133.76	125.51	148.86	154.95
	wRRS	70.39	74.43	80.23	80.20	95.47	101.69	93.04	97.07	101.03
	woR	71.16	72.65	85.50	64.70	82.34	88.40	73.28	76.91	94.33
UCn	wRKC	95.42	100.10	103.81	108.41	119.32	118.73	114.99	128.97	146.30
	wRRS	69.91	77.88	88.76	86.11	86.95	93.30	76.58	88.92	104.48
	woR	70.33	69.94	89.82	77.23	94.43	86.60	90.63	106.04	91.04

where $b(i)$ is the average of the minimum euclidean distance between customer i and the customers in clusters other than the one customer i belongs to. Parameter $a(i)$ is the average euclidean distance from customer i to other customer requests inside customer i's own rebalancing cluster. For each k, we sum $s(i)$ for all customer requests, and we plot it against k, see a qualitative description in Fig. 8. We then choose the value of k for which the sum is the highest (the higher $s(i)$ the higher is the difference between a point i and the clusters other than the one it belongs to). In our case, the best value of k was found to be $k = 3$ and therefore we use this value in the computational study.

5.3. Managerial insights

In this subsection, we assess the solutions provided by the model in terms of service rates and profits. The service rate is computed as the ratio between the total number of customers transported during the whole planning period over the total number of requests received in the same period. The profit consists of the cumulative profit over the entire simulated period (thus one hours with re-optimization every five minutes). Particularly, we compare three different strategies, namely no rebalancing (woR), rebalancing to random rebalancing centers (wRRS), and rebalancing to rebalancing centers found with the KC method (wRKC), see Section 4. These three strategies are assessed on different configurations of the service, namely UCt, NUCt, UCn, NUCn, see Section 5.2.

Tables 2 and 3 report the profit and service rate for the different strategies and configurations of the service. When the station is in the center (UCtand NUCt) we notice that all rebalancing strategies yield a 90% or higher service rate. The profits are likewise relatively similar across rebalancing strategies. This is mainly due to the geography of the instance, with the station located in the center. In this case, no rebalancing corresponds to leaving the vehicle close to the station, which is on average halfway to the next request in the worst case. Nevertheless, we observe that both the service rate and profits in the wRKC case are systematically higher than in the case woR, illustrating that rebalancing activities still pay off. Rebalancing according to the wRKC method is consistently preferable to rebalancing to random locations (wRRS) and rebalancing to random locations is not necessarily better than not rebalancing (woR). This is somewhat expected: When there is a clear pattern in the demand, sending vehicles to locations that are not carefully chosen represents, in general, a waste of resources. In general, when the station is in the center of the business area, we do not observe significant changes between the case with uniform demand (UCt) and non-uniform demand (NUCt).

When the station is located in a corner (configurations UCn and NUCn) we observe a much more marked difference in performance between the setting with and without rebalancing. Particularly, we notice that the setting wRKC improves both profits and service rates by approximately 30%, compared to woR in the case UCn. In the configuration NUCn the improvement is even more marked with approximately 50% higher in service rates and 60% higher profits. Random rebalancing (wRRS) is still slightly preferable to no rebalancing (woR) but lags significantly behind compared to the strategy of using clustering-based rebalancing centers (wRKC).

In Tables 2 and 3 it can be further observed that the gap between wRKC and woR tends to increase with the number of customers, keeping the fleet size fixed. In the NUCncase we observe that, in the wRKC case, both profits and service rates improve as the number of vehicles increases, keeping fixed the number of customers. Compared to the UCncase we notice that for the woR strategy, the highest service rate for the *C8V** instances is lower than 60%, while the lowest service rate for the same instances in the UCn scenario is above 60%.

Table 3
Service rate [%] in the simulations with $|\mathscr{R}| = 3$.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	97%	97%	96%	94%	96%	98%	92%	96%	98%
	wRRS	96%	95%	93%	90%	91%	96%	88%	93%	91%
	woR	94%	94%	94%	89%	94%	91%	90%	90%	92%
UCt	wRKC	97%	99%	98%	95%	97%	97%	96%	98%	98%
	wRRS	94%	94%	95%	93%	93%	95%	94%	92%	95%
	woR	91%	95%	94%	92%	95%	95%	91%	96%	92%
NUCn	wRKC	83%	88%	91%	81%	83%	90%	70%	85%	89%
	wRRS	62%	67%	70%	59%	65%	71%	58%	59%	62%
	woR	63%	66%	72%	48%	60%	62%	47%	48%	57%
UCn	wRKC	88%	89%	91%	86%	88%	88%	77%	86%	91%
	wRRS	64%	70%	80%	70%	65%	73%	57%	63%	71%
	woR	66%	69%	78%	63%	76%	68%	66%	72%	65%

Table 4
Profit in the simulations with $|\mathscr{R}| = 3$ rebalancing centers and fast-changing demand pattern.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	65.57	67.23	66.30	70.11	75.07	77.70	80.14	88.19	88.76
	wRRS	62.04	66.91	64.73	68.57	69.69	75.67	78.83	83.10	83.80
	woR	58.82	65.61	67.61	67.93	73.54	73.50	81.01	84.25	84.64
NUCn	wRKC	114.36	125.30	125.72	119.47	131.25	138.95	131.91	156.95	163.13
	wRRS	63.26	75.78	99.18	77.84	98.63	105.68	83.16	106.60	113.75
	woR	66.49	80.39	93.81	75.75	75.82	98.48	77.79	87.51	99.30

In conclusion, wRKC outperform both woR and wRRS in all scenarios. When the station is in the center (UCt and NUCt), the model reaches a service rate higher than 90% regardless of whether and how rebalancing is done. Nevertheless, profits and service rates are consistently higher when rebalancing centers are chosen with the clustering method (wRKC). When the station is in a corner (UCn and NUCn), rebalancing to centers chosen with the clustering method (wRKC) yields a much more marked improvement both in profits and service rate, especially with the demand is not evenly spread in the service area (NUCn). Furthermore, as we observe in this case, keeping fixed the number of customers, the profit increases with the size of the fleet. As more vehicles provide better opportunities for rebalancing and anticipating future requests. This also shows that rebalancing to appropriately chosen locations (e.g., wRKC) can improve the utilization of empty vehicles.

As explained in Section 4, the strategy of finding rebalancing centers by clustering existing requests is expected to work well provided that the demand changes slower than the re-optimization frequency. For this reason, we assessed the performance of the method assuming the demand pattern changes faster than our re-optimization frequency. Particularly, we studied the geographies NUCt and NUCn, where the demand is not uniformly distributed, and assumed the distribution chances between re-optimizations. In both cases, we start with having 10% of the demand in the inner zone at the first optimization phase, and we increase the demand in the inner zone with an additional 5% at every re-optimization. In this way, the distribution for last (the 12th) re-optimization sees 65% of the demand coming from the inner zone. Tables 4 and 5 report profits and service rates, respectively. Compared with the default setup where one third of the demand comes from the inner region (see Tables 2 and 3) we notice that the profit increases substantially. This is due to the fact that, with a changing demand pattern, there will be more requests coming from the inner zone, which in general means short travel distances to the station. Despite this, the service rate for the wRKC strategy is slightly lower, due to the reduced prediction ability. Nevertheless, the difference between the wRKC, the random and the no-rebalancing strategies is still visible and in some cases more marked.

5.4. Experiments on a real case

We performed additional tests on a real-life data set. The data obtained from the New York City Yellow Taxi data set (Commission and Limousine, 2023). These additional tests are meant to validate the results obtained on the randomly generated instances.

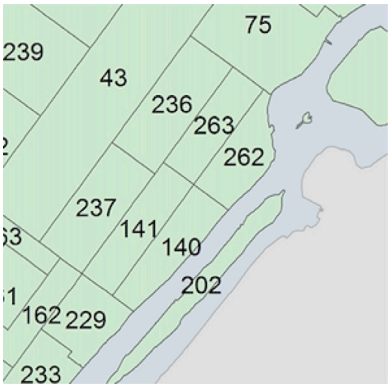


Fig. 6. Taxi Zones.

Particularly, we generate instances by selecting from the records of taxi rides occurred on February 1st 2023 between 16:00 and 17:00. We focus in particular on zones 140, 141 and 237 and set zone 236 as the common destination zone, see Fig. 6. Thus, the geography of the instances is similar to the one of the randomly generated instances with the station in a corner. We set the re-optimization frequency to 15 minutes and consider a planning horizon made of 12 re-optimizations, hence 180 minutes. We assume that the distribution of the demand is as described in Table 6.

The speed is calculated as the average speed of all requests in the focal data set. The required pickup time is generated randomly between 3 and 5 minutes. The requested arrival time is generated as the travel time from the customer pick-up location to the station, plus a buffer time randomly generated between 3 and 5 minutes. The fare for picking up new customers is obtained from the original data by subtracting the tip. The original locations of the vehicles are also generated randomly in the selected zones.

We compare the rebalancing strategy wRKC against no rebalancing. In addition, we test a rebalancing strategy based on the aggregation of historical data. Particularly, in a new rebalancing strategy named wHwe generate rebalancing points by clustering all requests obtained during

Table 5
Service rates in the simulations with $|\mathcal{R}| = 3$ rebalancing centers and fast-changing demand pattern.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	93%	96%	93%	91%	96%	98%	88%	96%	96%
	wRRS	88%	94%	91%	90%	88%	94%	87%	91%	91%
	woR	83%	91%	94%	87%	94%	93%	89%	92%	93%
NUCn	wRKC	81%	87%	87%	77%	83%	88%	71%	85%	88%
	wRRS	46%	54%	69%	51%	64%	68%	49%	60%	63%
	woR	48%	57%	65%	50%	51%	65%	45%	51%	56%

Table 6
Demand distribution in the Yellow Taxi instances.

Zone	140	141	237
Percentage of demand	70%	20%	10%

Table 7
Tests based on NYC real life data.

	Service Rate			Profit		
Instance	wRKC	woR	wH	wRKC	woR	wH
C10V6	97%	33%	82%	2222.07	750.28	1828.03
C10V7	97%	38%	76%	2221.14	883.36	1694.45
C10V8	100%	86%	70%	2275.13	2025.83	1533.21
C12V6	92%	43%	71%	2567.93	1261.87	1956.97
C12V7	95%	60%	69%	2623.73	1741.86	1857.48
C12V8	98%	42%	80%	2662.10	1202.44	2119.25
C14V6	89%	58%	74%	2852.79	1964.34	2396.37
C14V7	99%	52%	83%	3164.63	1707.04	2712.59
C14V8	96%	67%	73%	3091.74	2237.09	2266.38

Table 8
Tests based large instances generated from NYC real life data with 180 s time limit.

	Service Rate			Profit		
Instance	wRKC	woR	wH	wRKC	woR	wH
C30V40	98%	59%	76%	6822.86	4218.23	5123.71
C30V50	93%	65%	60%	6475.25	4573.18	3985.30
C30V60	94%	86%	66%	6499.27	6003.25	4388.29

the previous two weeks (i.e., the last two weeks of January 2023). The rationale behind this is that by considering more historical data, we obtain more precise estimates of where requests may occur. The results are presented in Table 7. Also in this case we observe a marked superiority of the proposed rebalancing strategy. The service rate is almost always above 90%. The difference with the no-rebalancing strategy is particularly evident when the ratio between customers and vehicles is high. Furthermore, we observe that generating rebalancing points based on historical data (wH) does not improve the wRKCstrategy. Apparently, if the frequency of re-optimization is high enough, following the occurrence of recent requests is a good enough strategy.

Finally, in order to mimic scenarios that are somewhat closer to real life contexts, we performed tests on larger instances of the problem. The instances contain 30 new customers appearing at each re-optimization phase and 40 to 60 vehicles. At each re-optimization phase, we give the solver a 180-second time limit, after which we get the best available solution. The results are summarized in Table 8. Essentially, the results confirm our observations from the tests on small instances.

5.5. Sensitivity analysis

In this section we analyze the effect of various parameters on the results obtained. We start by analyzing how the efficiency of rebalancing is affected by the β parameter, see (1). With the other parameters unchanged, for each instance we generate nine variants with different values of β, namely 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. We conduct experiments on all instances assuming an non-uniform configurations (NUCn and NUCt) to assess the impact on profits of the different rebalancing strategies under different weights β.

Fig. 7 illustrates the results for the NUCn configuration. We observe that the performance of woR, wRRSand wRKC are stable under different setting of β. Similarly, Fig. 8 reports the results under the NUCt configuration. Here we observe that when the ratio of customers to vehicles is relatively high, i.e. *C6V10*, *C7V10*, *C8V10* and *C8V12*, the performance of the wRKC is sensitive to β. Particularly, for instances *C8V10* and *C8V12*, when β increases, the performance of wRKC can drop below that of wRRS or even below woR while the woR and wRRS are relatively stable to the change of β. When the customers to vehicles ratio is relatively low, the results appear insensitive to β.

Fig. 9 reports the proportion of vehicles going to rebalancing centers under the configuration NUCt for instances *C8V10* and *C8V14*. The proportion is computed by the ratio between the average number of vehicles going to rebalancing centers of 3 randomly generated instances in the 12 re-optimization phases and the total number of vehicles. When β increases, the proportion of dispatched vehicles increases significantly for instance *C8V10*. The proportion of vehicles going to rebalancing centers is relatively stable for *C8V14*, which is consistent with the results in Fig. 8.

Finally, in the instances considered in Section 5.3 we used three rebalancing centers. We assessed how a higher number of rebalancing centers affects the results. Particularly, we run the tests with four and five rebalancing centers. We found no statistically significant differences which allow us to draw conclusions. In other words, our results suggest that adding potential rebalancing centers does not necessarily yield better results. The results are provided in Appendix C.

5.6. Complexity of the models

For the numerical experiments presented above, we choose to use relatively small instances in order to obtain results that are not affected by optimality gaps. All the instances presented could be solved, in all re-optimization phases, to provable optimality within less than a second in average (45 seconds in the worst case).

Nevertheless, in practical situations one may encounter instances significantly larger than the ones we used. In this section, we assess how solution times and optimality gaps scale with the size of the instance. Particularly, we create instance classes named $C|\mathcal{N}_C|V|\mathcal{K}|$ for $|\mathcal{N}_C| \in \{60\}$, and $|\mathcal{K}| \in \{40, 50, 60\}$. As an example, *C60V40* indicates a class of instances with 60 new customers in each re-optimization phase and 40 vehicles available for dispatch for the whole planning period. Fig. 10 reports the progression of upper and lower bounds for instances *C60V40*, *C60V50* and *C60V60*. The optimality gaps of the three instances are 21.1%, 71.7% and 26.0%, respectively, after 120 seconds (an amount of time which is sufficiently large if the operator re-optimizes, e.g., every 5 minutes). We observe significant optimality gaps and, particularly, that the primal bound improves slowly while the dual bound remains steady. We believe that finding optimal or close-to-optimal solutions is valuable in this context. In case of lack of performance (in some dimension), the decision maker would be able to rule out the possibility that this is caused by highly sub-optimal solutions, and understand that the reason should rather be found in choices in terms of system design.

Fig. 11 provides the same information for three smaller instances, namely *C25V15*, *C25V20* and *C25V25*. In this case, the instances are small enough to observe an improvement of the upper bound. Nevertheless, while a good primal solution is found rather quickly, the upper

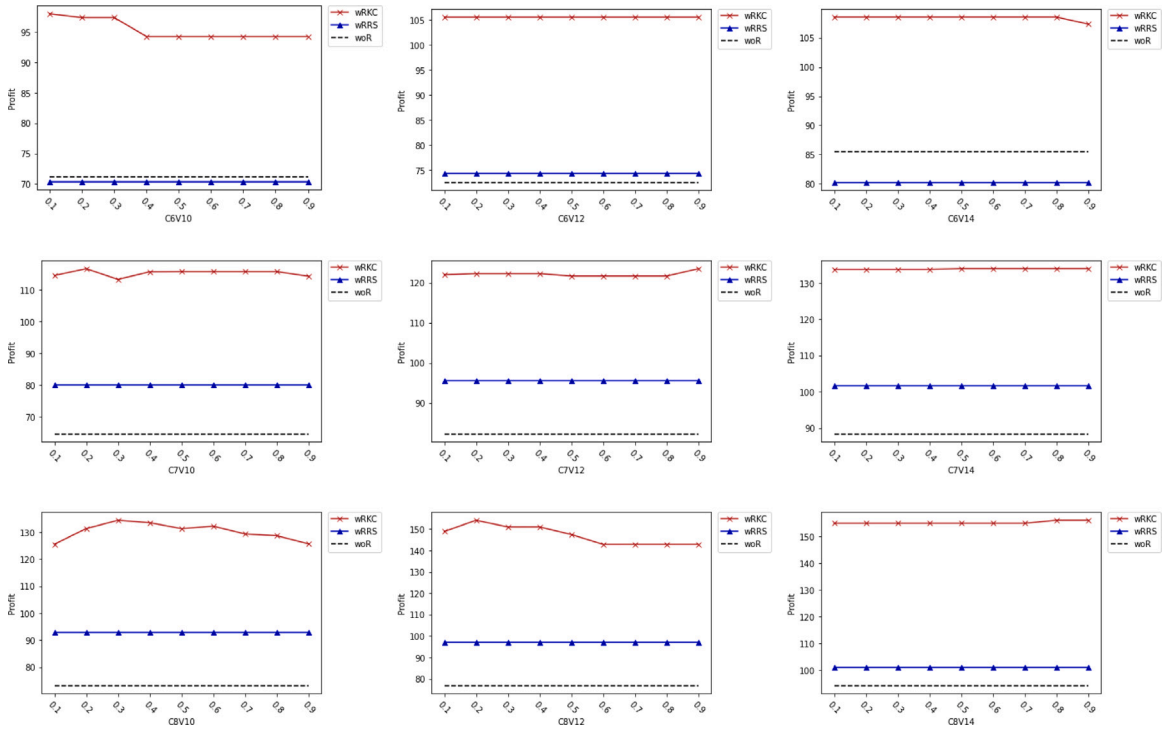


Fig. 7. Sensitivity analysis for the NUCn configuration.

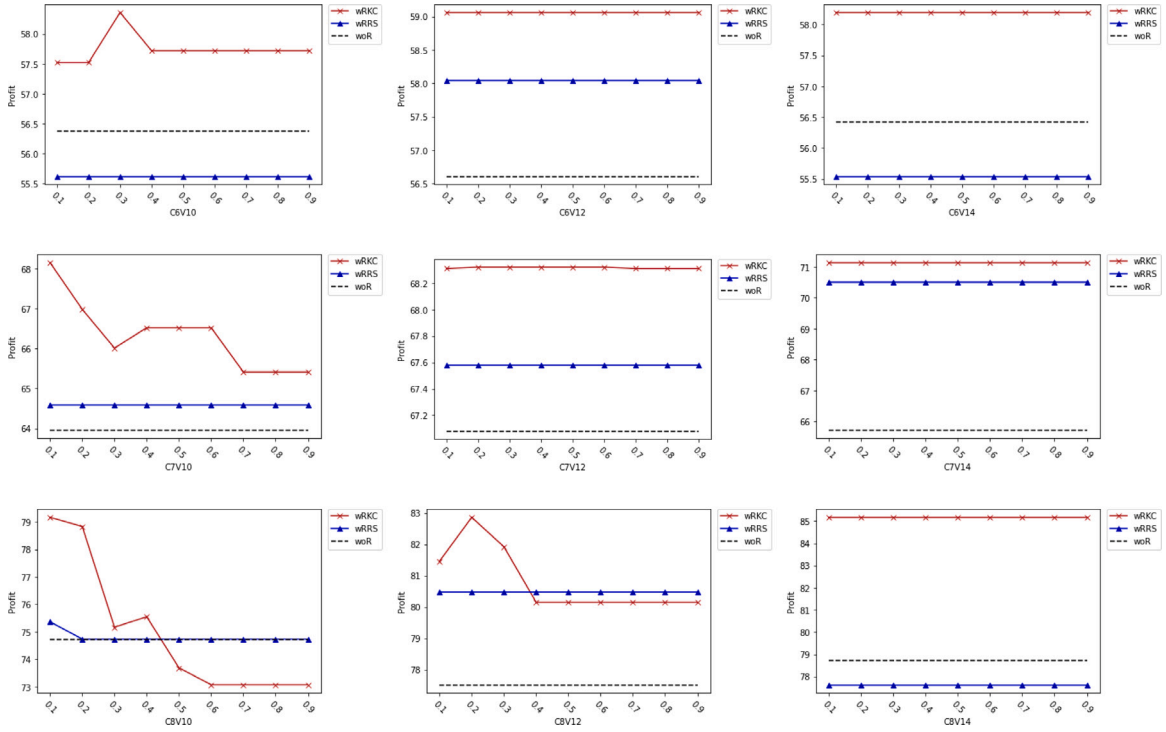


Fig. 8. Sensitivity analysis for the NUCt configuration.

bound improves slowly. This suggests new lines of research that provide both tighter formulations and methods, perhaps heuristic, to quickly find primal solutions in large scale instances.

6. Conclusions

In this paper we developed a MILP model for online order dispatching and vehicle rebalancing in a first-mile ride-sharing service. The model was used in a rolling-horizon simulation framework based on constructed instances. We assessed whether rebalancing is advantageous over leaving idle vehicles in their positions. We have done this using a clustering method to identify promising rebalancing locations. The results show that rebalancing using a clustering-based strategy consistently outperforms strategies based on rebalancing to random locations or not rebalancing at all. Particularly, rebalancing strategies perform dramatically better in a context where the station is not

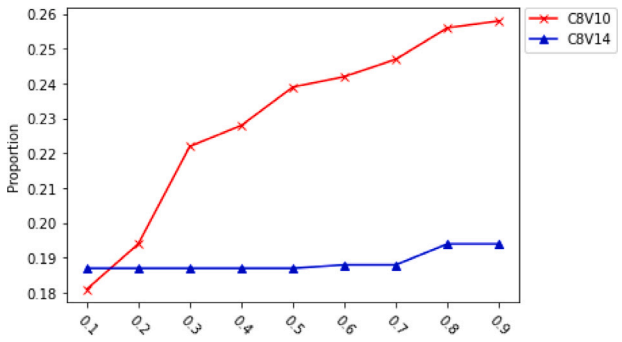


Fig. 9. Proportion of vehicles dispatched to rebalancing centers.

centrally located (e.g., the business area is entirely on one side of the station).

The proposed method to find promising rebalancing locations based on current requests has some inherent limits. Particularly, it is founded on the assumption that current demand represents future demand. This is true in contexts where re-optimization is performed frequently and/or demand distribution changes slowly enough to make the difference between to re-optimizations negligible. Performing re-optimization frequently requires the ability of solving the proposed model efficiently also for large-scale instance. However, our tests show that finding high quality solutions and bounds to the problem presented becomes problematic as the size of the problem increases. This suggests future research on both efficient solution methods and tighter formulations. Furthermore, additional research is needed to develop methods that can accurately predict future occurrence of demand and promising rebalancing locations in more general conditions than the ones we assume in this article. Finally, the assumptions made in the optimization model could be relaxed to improve the model in a number of ways. As an example, we assume that a vehicle with passengers onboard cannot drive to a rebalancing center. However, it is reasonable to imagine a scenario where a vehicle drops off the passengers on board and then moves to a rebalancing center rather than waiting at the station. These solutions are currently infeasible in our model.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jinwen Ye reports financial support was provided by European Union.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199.

Appendix A. Notation

Sets:

$\mathcal{N}_P := \{1, \dots, N_P\}$ is the set of customers that have been assigned to vehicles but have not been picked up yet in previous re-optimization phase, in which customers can be reassigned but cannot be rejected in current re-optimization phase.

$\mathcal{N}_C := \{1, \dots, N_C\}$ is the set of new customers for current re-optimization phase, in which customers can be accepted or rejected

$\mathcal{N}_U := \mathcal{N}_C \cup \mathcal{N}_P$ is the combined set of assigned and new customers.

$\mathcal{R} := \{1, \dots, R\}$ is the set of rebalancing centers

$\mathcal{K} := \{1, \dots, K\}$ is the set of vehicles

Parameters:

d: Denotes the destination/station node

$o(k)$: Denotes the original location of vehicle k, for $k \in \mathcal{K}$

$o(i)$: Denotes the location of request $i \in \mathcal{N}_U$

P_i: Denotes the profit of picking up customer i, for $i \in \mathcal{N}_U$

C: Denotes the unit travel time cost of the vehicle.

β: Denotes the weight parameter of rebalancing reward

V_k: Denotes the number of customers on board of vehicle k at the beginning of the operational period, for $k \in \mathcal{K}$

Q: Denotes the capacity of the vehicle

T: Denotes the start time of the operational period

T_{ij}: Denotes the travel time between locations $o(i)$ and $o(j)$, for $i \in \mathcal{K} \cup \mathcal{N}_U$, $j \in \mathcal{N}_U \cup \mathcal{R} \cup \{d\}$

T_i^A: Denotes the requested arrival time of customer i, for $i \in \mathcal{N}_U$

$T^L := max_i\{T_i^A\}$ for $i \in \mathcal{N}_U$, is the upper bound of the requested arrival time of all the customers

T_k: Denotes the requested arrival time of vehicle k, for $k \in \mathcal{K}$

T_i^P: Denotes the requested pick up time of the customer i, for $i \in \mathcal{N}_P \cup \mathcal{N}_C$

B_i: Denotes an upper bound on the number of vehicles that can move to rebalancing center $i \in \mathcal{R}$

E_i: Denotes the net profit for a vehicle to go to the rebalancing center i, for $i \in \mathcal{R}$

Δ: Denotes the maximum waiting time. Waiting time here is represented by the difference between actual pick up time and requested pick up time of the customer

$\bar{Q}$: Denotes the average number of customer on board during one trip of a vehicle

Decision variables:

x_{ij}^k: Equal to 1 if vehicle k moves directly between $o(i)$ and $o(j)$, 0 otherwise, for $i \in \{k\} \cup \mathcal{N}_U, j \in \mathcal{N}_U \cup \mathcal{R} \cup \{d\}, k \in \mathcal{K}$.

t_k^A: Denotes the actual arrival time of vehicle k to the station, for $k \in \mathcal{K}$

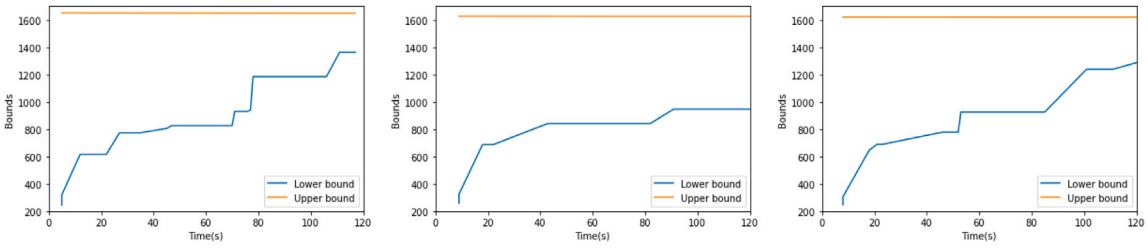
t_i^P: Denotes the actual pick up time of customer i, for $i \in \mathcal{N}_U$

Appendix B. Illustrative example

In what follows we provide two examples to illustrate how the model in Section 3.2 works. The online FMRSP model re-optimizes the vehicles dispatch and rebalancing decisions at fixed time intervals. Assume that the current re-optimization phase start at $T = T_1$, and a snapshot of the system, which includes vehicles and customers positions, is shown in Figs. 12(a) and 13(a). The blue circles denote customers while the yellow squares denote vehicles. Assume all vehicles have a capacity $Q = 2$, and that are all empty except for $D3$ that has currently one customer on board ($V_{D3} = 1$). The triangle denotes the station. Particularly, Fig. 12 describes a scenario without rebalancing, while Fig. 13 describes a scenario where rebalancing is permitted.

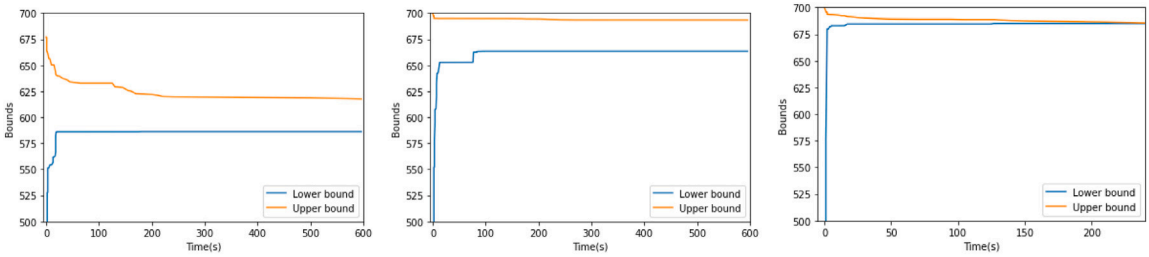
The solution for the case without rebalancing, is illustrated in Fig. 12(b) where the orange circles denote rebalancing centers and their size reflects the expected demand at their location. The solution for the scenario with rebalancing is provided in Fig. 13(b). Consider the solution to the first scenario. We observe that the model provides three routes and that the rebalancing centers are not visited. The route for vehicle $D1$ (Route 1) starts from the current location of the vehicle, $o(D1)$, and visits customer 1 (thus $x_{D1,1}^{D1} = 1$) and customer 2 ($x_{1,2}^{D1} = 1$), in this order, before arriving at the station ($x_{2,d}^{D1} = 1$). In this solution, vehicles $D4$, $D5$ and $D6$ stay idle at their current position and wait for the next re-optimization phase.

Fig. 12(c) depicts a new snapshot of the system at the new re-optimization phase (say $T = T_2$), and the corresponding solution. Customers 1 through 5 from the previous optimization phase are not in the system anymore because they have been by the time picked up. Similarly, vehicles $D1$, $D2$ and $D3$ are not in the system because they are currently on the way to the station and have no extra capacity ($D1$ and $D2$ have filled their two seats and $D3$ has the remaining seat available). New customers $N1$-$N4$ appear in the system. The solution suggests picking up only customer $N1$. The remaining customers are, in fact, too distant from the available vehicles.



a *C60V40* b *C60V50* c *C60V60*

Fig. 10. Progression of upper and lower bounds for the instances *C60V**.



a *C25V15* b *C25V20* c *C25V25*

Fig. 11. Progression of upper and lower bounds for the instances *C25V**.

Table 9
Profit [$] in the simulations with $|\mathscr{R}| = 4$.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	52.85	57.35	60.40	69.60	70.33	71.22	79.85	83.19	85.06
	wRRS	58.07	55.84	56.20	65.54	66.87	67.03	76.68	80.80	80.38
	woR	56.37	56.61	56.42	63.96	67.08	65.72	74.71	77.52	78.71
UCt	wRKC	55.89	54.22	53.71	63.74	62.81	64.74	68.93	75.21	74.94
	wRRS	52.73	52.81	54.21	57.21	63.08	65.36	70.09	71.39	70.86
	woR	50.88	55.03	54.35	61.46	61.68	65.22	66.79	71.78	68.77
NUCn	wRKC	91.50	100.84	103.91	114.30	122.99	124.83	131.54	146.45	157.57
	wRRS	68.29	70.93	90.19	74.72	90.36	93.54	103.25	92.65	120.69
	woR	71.16	72.65	85.50	64.70	82.34	88.40	73.28	76.91	94.33
UCn	wRKC	94.80	99.53	98.20	106.83	111.00	118.21	124.10	131.44	130.02
	wRRS	77.45	77.99	81.28	90.68	91.29	104.36	97.70	87.89	105.68
	woR	70.33	69.94	89.82	77.23	94.43	86.60	90.63	106.04	91.04

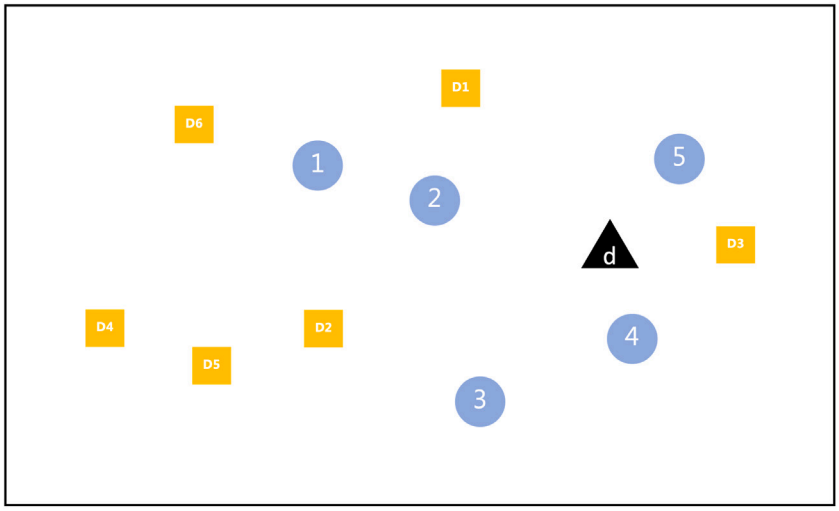
The solution for the scenario with rebalancing (Fig. 13(b)) differs only in the fact that vehicles $D4$, $D5$, and $D6$ are dispatched to rebalancing centers. Therefore, at the next re-optimization phase (Fig. 13(c)), $D4$ and $D6$ are much closer to the new requests $N2$, $N3$, and $N4$, and can pick them up ensuring their requested arrival time is satisfied.

The example in Fig. 14(a) illustrates how capacity constraints work. The original number of customers on board are shown in green boxes next to the vehicle. Assume the capacity of each vehicle is $Q = 4$. We observe that routes 1 and 3 are feasible with respect to the capacity constraints as in both cases the number of customers on board does not exceed Q. Route 2, on the other hand, will be infeasible, since there are already 3 customers on board vehicle $D2$ at the beginning of the optimization phase, and the route assigns two additional customers to the vehicle, thus violating the capacity constraints.

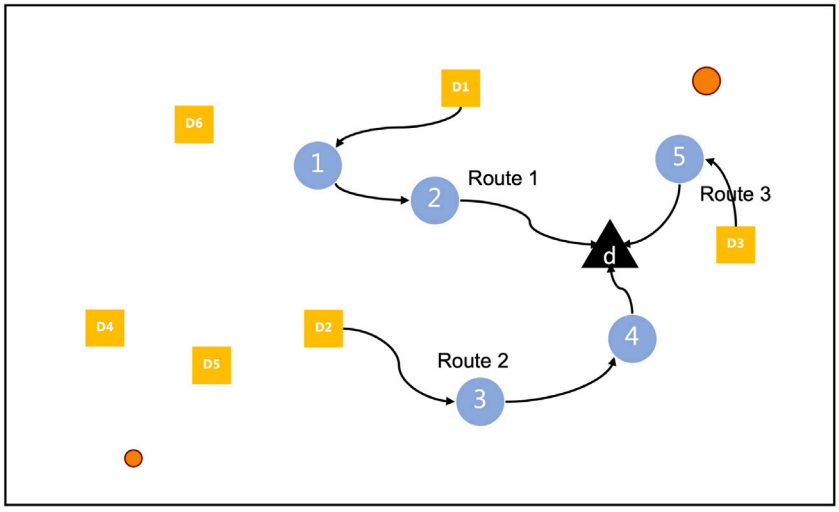
Fig. 14(b) illustrates how requested arrival time constraints are enforced. Assume the re-optimization phase starts at time $T = 0$. The requested arrival time of the vehicles and customers are shown in purple boxes while travel times are shown on top of the arcs. For route 1, the total travel time is $5 + 5 + 4 = 14$, which violates the requested arrival time of the passengers already on board of $D1$, which is 12. Thus the route is infeasible, even if the arrival times of the two customers in route 1 are respected. Routes 2 and 3 are instead feasible as the arrival times of both customers and vehicles are respected.

Appendix C. Results for different numbers of rebalancing centers

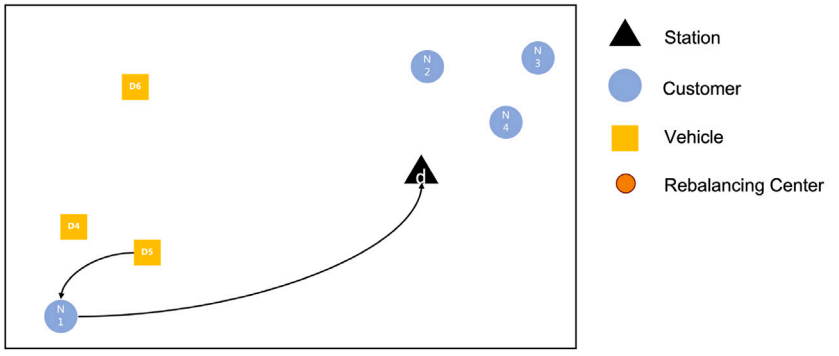
See Tables 9–12.



a *Snapshot of the system at* $T = T_1$

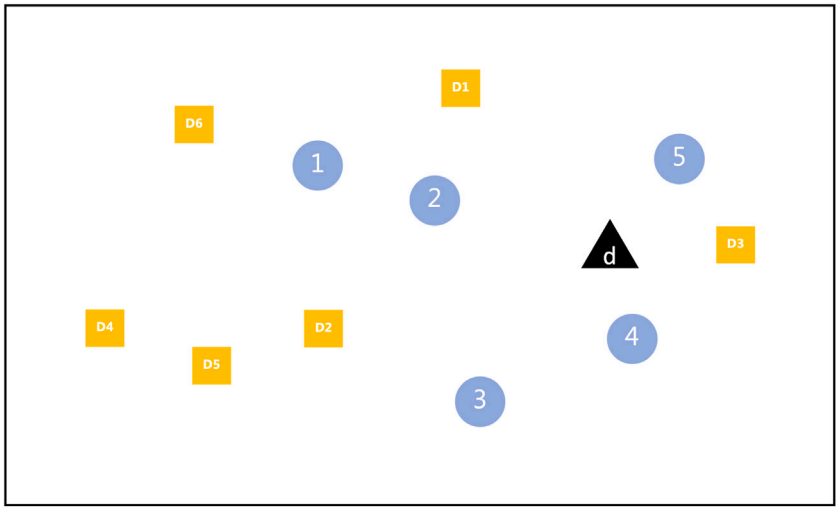


b *Solution for* $T = T_1$

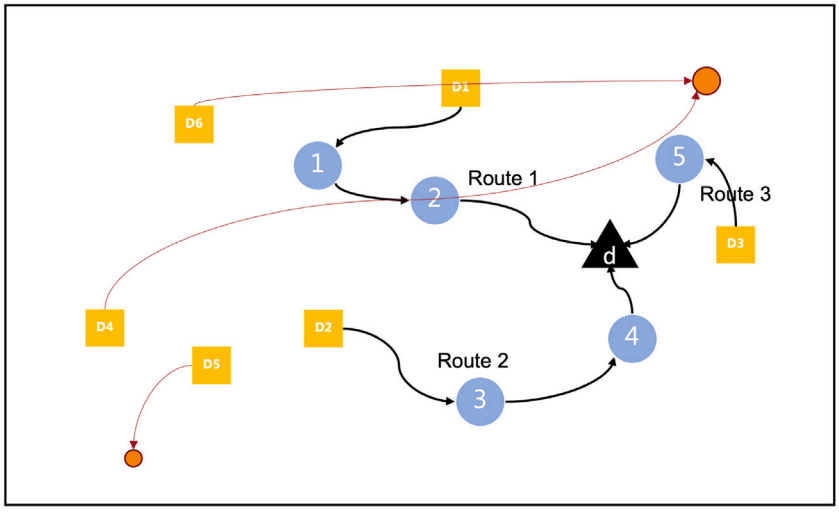


c *Solution for* $T = T_2$

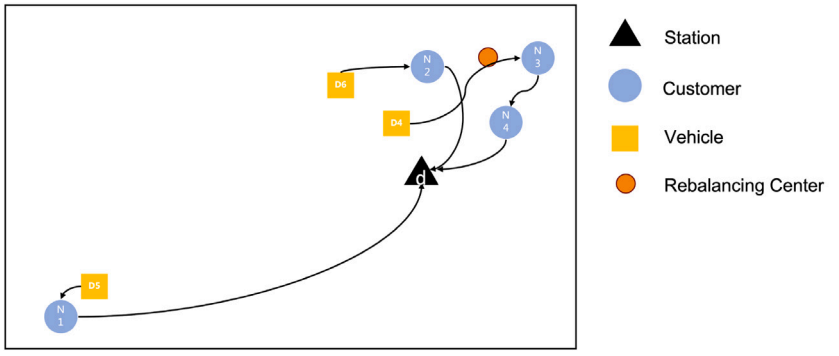
Fig. 12. Example problem in a scenario where rebalancing in not permitted.



a *Snapshot of the system at* $T = T_1$

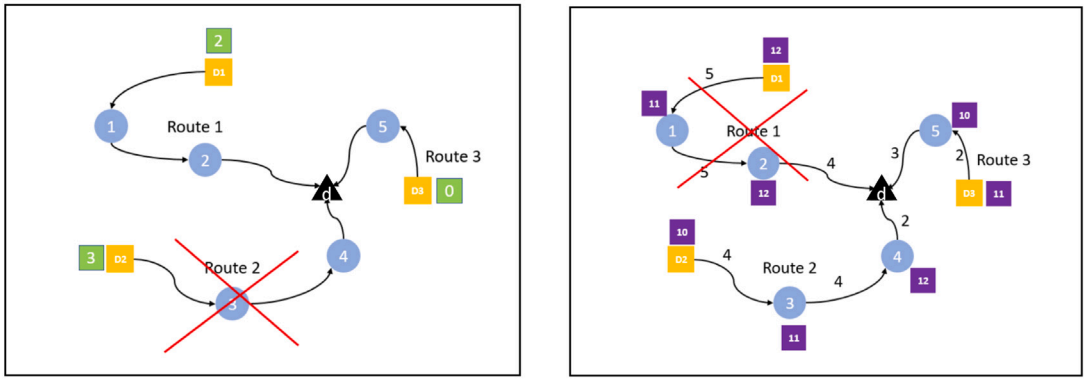


b *Solution for* $T = T_1$



c *Solution for* $T = T_2$

Fig. 13. Example problem in a scenario where rebalancing is permitted.



a *Capacity constraints* b *Arrival time Constraints*

Fig. 14. Illustration of capacity and arrival time constraints.

Table 10
Service rate [%] in the simulations with $|\mathscr{R}| = 4$.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	91%	98%	98%	95%	97%	97%	93%	96%	98%
	wRRS	95%	94%	93%	91%	92%	91%	89%	93%	94%
	woR	94%	94%	94%	89%	94%	91%	90%	90%	92%
UCt	wRKC	97%	97%	96%	95%	96%	98%	94%	98%	99%
	wRRS	92%	94%	95%	92%	93%	94%	92%	94%	94%
	woR	91%	95%	94%	92%	95%	95%	91%	96%	92%
NUCn	wRKC	79%	84%	86%	79%	84%	86%	77%	84%	89%
	wRRS	59%	62%	77%	56%	63%	65%	62%	57%	72%
	woR	63%	66%	72%	48%	60%	62%	47%	48%	57%
UCn	wRKC	86%	87%	87%	81%	88%	87%	82%	87%	90%
	wRRS	67%	70%	72%	72%	72%	78%	64%	62%	70%
	woR	66%	69%	78%	63%	76%	68%	66%	72%	65%

Table 11
Profit [$] in the simulations with $|\mathscr{R}| = 5$.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	55.59	58.84	57.78	68.43	67.63	70.83	76.62	81.41	82.31
	wRRS	56.72	55.37	53.75	64.77	68.90	66.49	73.98	79.11	80.38
	woR	56.37	56.61	56.42	63.96	67.08	65.72	74.71	77.52	78.71
UCt	wRKC	54.50	51.97	56.89	64.16	64.61	63.51	67.93	74.05	72.42
	wRRS	53.76	51.60	53.94	61.15	60.48	62.18	67.57	67.84	70.33
	woR	50.88	55.03	54.35	61.46	61.68	65.22	66.79	71.78	68.77
NUCn	wRKC	96.28	89.64	97.71	117.68	120.27	128.11	127.05	149.52	148.79
	wRRS	73.74	79.37	87.74	86.42	77.28	97.84	91.69	103.02	115.89
	woR	71.16	72.65	85.50	64.70	82.34	88.40	73.28	76.91	94.33
UCn	wRKC	90.33	86.50	94.46	101.61	112.41	106.93	114.81	123.93	125.85
	wRRS	61.13	84.17	81.85	72.83	78.28	91.78	86.65	88.83	103.16
	woR	70.33	69.94	89.82	77.23	94.43	86.60	90.63	106.04	91.04

Table 12
Service rate [%] in the simulations with $|\mathscr{R}| = 5$.

		C6V10	C6V12	C6V14	C7V10	C7V12	C7V14	C8V10	C8V12	C8V14
NUCt	wRKC	93%	99%	96%	94%	95%	97%	89%	95%	96%
	wRRS	93%	94%	91%	91%	94%	92%	90%	93%	93%
	woR	94%	94%	94%	89%	94%	91%	90%	90%	92%
UCt	wRKC	98%	94%	98%	95%	98%	95%	93%	99%	98%
	wRRS	94%	94%	97%	95%	93%	94%	92%	90%	95%
	woR	91%	95%	94%	92%	95%	95%	91%	96%	92%
NUCn	wRKC	78%	78%	81%	81%	83%	86%	78%	86%	84%
	wRRS	63%	68%	75%	61%	59%	70%	59%	63%	69%
	woR	63%	66%	72%	48%	60%	62%	47%	48%	57%
UCn	wRKC	81%	75%	82%	81%	87%	83%	77%	82%	85%
	wRRS	61%	78%	76%	62%	66%	76%	64%	68%	73%
	woR	66%	69%	78%	63%	76%	68%	66%	72%	65%

References

Al-Abbasi, Abubakr O., Ghosh, Arnob, Aggarwal, Vaneet, 2019. DeepPool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. IEEE Trans. Intell. Transp. Syst. 1–14.

Alonso-mora, Javier, Wallar, Alex, Frazzoli, Emilio, Rus, Daniela, 2018. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. USA 115 (3), E555.

Balas, Egon, 1989. The prize collecting traveling salesman problem. Networks 19 (6), 621–636.

Beirigo, Breno A, Negenborn, Rudy R, Alonso-Mora, Javier, Schulte, Frederik, 2022. A business class for autonomous mobility-on-demand: Modeling service quality contracts in dynamic ridesharing systems. Transp. Res. C 136, 103520.

Berbeglia, Gerardo, Cordeau, Jean François, Laporte, Gilbert, 2010. Dynamic pickup and delivery problems. European J. Oper. Res. 202 (1), 8–15.

Bertsimas, Dimitris, Jaillet, Patrick, Martin, Sébastien, 2019. Online vehicle routing: The edge of optimization in large-scale applications. Oper. Res. 67 (1), 143–162.

Bian, Zheyong, Liu, Xiang, 2019a. Mechanism design for first-mile ridesharing based on personalized requirements part I: Theoretical analysis in generalized scenarios. Transp. Res. B 120, 147–171.

Bian, Zheyong, Liu, Xiang, 2019b. Mechanism design for first-mile ridesharing based on personalized requirements part II: Solution algorithm for large-scale problems. Transp. Res. B 120, 172–192.

Bian, Zheyong, Liu, Xiang, Bai, Yun, 2020. Mechanism design for on-demand first-mile ridesharing. Transp. Res. B 138, 77–117.

Bongiovanni, Claudia, Kaspi, Mor, Cordeau, Jean-Francois, Geroliminis, Nikolas, 2022. A machine learning-driven two-phase metaheuristic for autonomous ridesharing operations. Transp. Res. E 165, 102835.

Braekers, Kris, Ramaekers, Katrien, Van Nieuwenhuyse, Inneke, 2016. The vehicle routing problem: State of the art classification and review. Comput. Ind. Eng. 99, 300–313.

Bräysy, Olli, Gendreau, Michel, 2005. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Transp. Sci. 39 (1), 104–118.

Chemla, Daniel, Meunier, Frédéric, Calvo, Roberto Wolfler, 2013. Bike sharing systems: Solving the static rebalancing problem. Discrete Optim. 10 (2), 120–146.

Chen, Yiwei, Wang, Hai, 2018. Pricing for a last-mile transportation system. Transp. Res. B 107, 57–69.

Chen, Shukai, Wang, Hua, Meng, Qiang, 2020. Solving the first-mile ridesharing problem using autonomous vehicles. Comput.-Aided Civ. Infrastruct. Eng. 35 (1), 45–60.

Commission, New York City Taxi, Limousine, 2023. New York City Taxi and Limousine Commission. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. (Online; Accessed 10 October 2023).

Cordeau, Jean François, Laporte, Gilbert, 2003. The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. 4OR 1 (2), 89–101.

Danassis, Panayiotis, Sakota, Marija, Filos-Ratsikas, Aris, Faltings, Boi, 2022. Putting ridesharing to the test: Efficient and scalable solutions and the power of dynamic vehicle relocation. Artif. Intell. Rev. 55 (7), 5781–5844.

Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. Manage. Sci. 6 (1), 80–91.

Elting, Steffen, Ehmke, Jan Fabian, 2021. Potential of shared taxi services in rural areas–A case study. Transp. Res. Procedia 52, 661–668.

English, Sandy, 2008. High fuel prices impoverish new york city taxi drivers. https://www.wsws.org/en/articles/2008/09/taxi-s04.html. (Online; Accessed 10 October 2022).

Faghih-Imani, Ahmadreza, Hampshire, Robert, Marla, Lavanya, Eluru, Naveen, 2017. An empirical analysis of bike sharing usage and rebalancing: Evidence from Barcelona and Seville. Transp. Res. A 97, 177–191.

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

Folkestad, Carl Axel, Hansen, Nora, Fagerholt, Kjetil, Andersson, Henrik, Pantuso, Giovanni, 2020. Optimal charging and repositioning of electric vehicles in a free-floating carsharing system. Comput. Oper. Res. 113, 104771.

Ho, Sin C., Szeto, W.Y., Kuo, Yong-hong Hong, Leung, Janny M.Y.Y., Petering, Matthew, Tou, Terence W.H.H., 2018. A survey of dial-a-ride problems: Literature review and recent developments. Transp. Res. B 111, 395–421.

Huang, Yan, Bastani, Favyen, Jin, Ruoming, Wang, Xiaoyang Sean, 2014. Large scale realtime ridesharing with service guarantee on road networks. In: Proceedings of the VLDB Endowment.

Illgen, Stefan, Höck, Michael, 2019. Literature review of the vehicle relocation problem in one-way car sharing networks. Transp. Res. B 120, 193–204.

INSHUR, 2022. How much does an Uber driver make in New York City?. https://inshur.com/blog/how-much-does-an-uber-driver-make-in-new-york-city/. (Online; Accessed 10 October 2022).

Jamshidi, Helia, Correia, Gonçalo HA, van Essen, J Theresia, Nökel, Klaus, 2021. Dynamic planning for simultaneous recharging and relocation of shared electric taxies: A sequential MILP approach. Transp. Res. C 125, 102933.

Kash, Ian A., Wen, Zhongkai, Zuck, Lenore D., 2022. Dynamic relocation in ridesharing via fixpoint construction. In: Cussens, James, Zhang, Kun (Eds.), Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence. In: Proceedings of Machine Learning Research, vol. 180, PMLR, pp. 980–989, URL https://proceedings.mlr.press/v180/kash22a.html.

Kumar, Suresh Nanda, Panneerselvam, Ramasamy, 2012. A survey on the vehicle routing problem and its variants. Intell. Inf. Manage. 04 (03), 66–74.

Li, Xihan, Zhang, Jia, Bian, Jiang, Tong, Yunhai, Liu, Tie-Yan, 2019. A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network. In: International Foundation for Autonomous Agents and Multiagent Systems. AAMAS '19, pp. 980–988.

Lin, Canhong, Choy, K.L., Ho, G.T.S., Chung, S.H., Lam, H.Y., 2014. Survey of Green vehicle routing problem: Past and future trends. Expert Syst. Appl. 41 (4 PART 1), 1118–1138.

Lin, Kaixiang, Zhao, Renyu, Xu, Zhe, Zhou, Jiayu, 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1774–1783.

Liu, Junming, Sun, Leilei, Chen, Weiwei, Xiong, Hui, 2016. Rebalancing bike sharing systems: A multi-source data smart optimization. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1005–1014.

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

Lotfi, Sepide, Abdelghany, Khaled, 2022. Ride matching and vehicle routing for on-demand mobility services. J. Heuristics 28 (3), 235–258.

Ma, Tai-yu Yu, Rasulkhani, Saeid, Chow, Joseph Y.J. J, Klein, Sylvain, 2019a. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. Transp. Res. E 128 (December 2018), 417–442.

Ma, Ruimin, Yao, Lifei, Song, Lijun, Jin, Maozhu, 2019b. A novel algorithm for peer-to-peer ridesharing match problem. Neural Comput. Appl. 31 (s1), 247–258.

Mahendru, Khyati, 2019. How to determine the optimal K for K-means?. https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb. (Online; Accessed 10 October 2022).

Mao, Chao, Liu, Yulin, Shen, Zuo-Jun Max, 2020. Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. Transp. Res. C 115, 102626.

Masoud, Neda, Jayakrishnan, R., 2017. A decomposition algorithm to solve the multi-hop Peer-to-Peer ride-matching problem. Transp. Res. B 99 (May), 1–29.

Masoud, Neda, Lloret-Batlle, Roger, Jayakrishnan, R., 2017. Using bilateral trading to increase ridership and user permanence in ridesharing systems. Transp. Res. E.

Mourad, Abood, Puchinger, Jakob, Chu, Chengbin, 2019. A survey of models and algorithms for optimizing shared mobility. Transp. Res. B 123, 323–346.

Noruzoliaee, Mohamadhossein, Zou, Bo, 2022. One-to-many matching and section-based formulation of autonomous ridesharing equilibrium. Transp. Res. B 155, 72–100.

Osorio, Jesus, Lei, Chao, Ouyang, Yanfeng, 2021. Optimal rebalancing and on-board charging of shared electric scooters. Transp. Res. B 147, 197–219.

Pantuso, Giovanni, 2022. Exact solutions to a carsharing pricing and relocation problem under uncertainty. Comput. Oper. Res. 144, 105802.

Parragh, Sophie N., Doerner, Karl F., Hartl, Richard F., 2008. A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. J. Betr. 58 (2), 81–117.

Pfeiffer, Christian, Schulz, Arne, 2022. An ALNS algorithm for the static dial-a-ride problem with ride and waiting time minimization. OR Spectrum 44 (1), 87–119.

Pillac, Victor, Gendreau, Michel, Guéret, Christelle, Medaglia, Andrés L., 2013. A review of dynamic vehicle routing problems. European J. Oper. Res. 225 (1), 1–11.

Qin, Zhiwei, Tang, Xiaocheng, Jiao, Yan, Zhang, Fan, Wang, Chenxi, Li, Qun, 2019. Deep reinforcement learning for ride-sharing dispatching and repositioning. In: IJCAI International Joint Conference on Artificial Intelligence. 2019-Augus, pp. 6566–6568.

Ritzinger, Ulrike, Puchinger, Jakob, Hartl, Richard F., 2016. A survey on dynamic and stochastic vehicle routing problems. Int. J. Prod. Res. 54 (1), 215–231.

Ropke, Stefan, Cordeau, Jean-François François, 2009. Branch and cut and price for the pickup and delivery problem with time windows. Transp. Sci. 43 (3), 267–286.

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

Sayarshad, Hamid R., Chow, Joseph Y.J., 2017. Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. Transp. Res. E 106, 60–77.

Shen, Yu, Zhang, Hongmou, Zhao, Jinhua, 2018. Integrating shared autonomous vehicle in public transportation system: A supply-side simulation of the first-mile service in Singapore. Transp. Res. A 113, 125–136.

Stiglic, Mitja, Agatz, Niels, Savelsbergh, Martin, Gradisar, Mirko, 2015. The benefits of meeting points in ride-sharing systems. Transp. Res. B 82, 36–53.

Stiglic, Mitja, Agatz, Niels, Savelsbergh, Martin, Gradisar, Mirko, 2018. Enhancing urban mobility: Integrating ride-sharing and public transit. Comput. Oper. Res. 90, 12–21.

Tang, Xiaocheng, Qin, Zhiwei Tony, Zhang, Fan, Wang, Zhaodong, Xu, Zhe, Ma, Yintai, Zhu, Hongtu, Ye, Jieping, Labs, A I, Chuxing, Didi, Tang, Xiaocheng, Qin, Zhiwei Tony, Zhang, Fan, Wang, Zhaodong, Xu, Zhe, Ma, Yintai, Zhu, Hongtu, Ye, Jieping, 2019. A deep value-network based approach for multi-driver order dispatching. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1780–1790.

Taniguchi, Eiichi, Thompson, Russell G., Yamada, Tadashi, 2014. Recent trends and innovations in modelling city logistics. Procedia Soc. Behav. Sci. 125, 4–14, URL https://www.sciencedirect.com/science/article/pii/S187704281401489X.

Toth, Paolo, Vigo, Daniele, 2002. The Vehicle Routing Problem. SIAM.

Wallar, Alex, Van Der Zee, Menno, Alonso-Mora, Javier, Rus, Daniela, 2018. Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In: IEEE International Conference on Intelligent Robots and Systems. pp. 4539–4546.

Wang, Hai, 2019. Routing and scheduling for a last-mile transportation system. Transp. Sci. 53 (1), 131–147.

Wang, Xing, Agatz, Niels, Erera, Alan, 2018. Stable matching for dynamic ride-sharing systems. Transp. Sci. 52 (4), 850–867.

Wen, Jian, Zhao, Jinhua, Jaillet, Patrick, 2018. Rebalancing shared mobility-on-demand systems: A reinforcement learning approach. In: IEEE Conference on Intelligent Transportation Systems, Proceedings. ITSC, 2018-March, (October), pp. 220–225.

Xu, Zhe, Li, Zhixin, Guan, Qingwen, Zhang, Dingshui, Li, Qiang, Nan, Junxiao, Liu, Chunyang, Bian, Wei, Ye, Jieping, 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '18, Association for Computing Machinery, New York, NY, USA, pp. 905–913.

Zhao, Meng, Yin, Jiateng, An, Shi, Wang, Jian, Feng, Dejian, 2018. Ridesharing problem with flexible pickup and delivery locations for app-based transportation service: Mathematical modeling and decomposition methods. J. Adv. Transp. 2018.

Zheng, Minyi, Pantuso, Giovanni, 2023. Trading off costs and service rates in a first-mile ride-sharing service. Transp. Res. C 150, 104099.