Transportation Research Part C 153 (2023) 104196

Contents lists available at ScienceDirect

Transportation Research Part C

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

# Anticipatory rebalancing of RoboTaxi systems ☆

Shir Tavor, Tal Raviv *

*Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel*

ARTICLE INFO

*Keywords:*
Autonomous mobility on demand
Online rebalancing optimization
Car sharing
RoboTaxi
Shared autonomous vehicles

ABSTRACT

This study considers a RoboTaxi service, a futuristic transportation system whereby passengers can place transportation requests in real-time for a private ride. The passengers are picked up from their origins as soon as possible after placing a request and travel directly to their destination, from which the RoboTaxi may immediately continue to serve other customers, remain idle, or move to another location in the city in anticipation of future demand.

The operation of RoboTaxi systems requires online decisions regarding vehicle dispatching and rebalancing. Dispatching assigns cars to requests. Rebalancing is the initiation of empty trips of idle cars to reduce future requests' waiting time and take advantage of off-peak traffic hours. This study proposes an advanced method for rebalancing while assuming a simple yet effective dispatching policy.

The optimal operation of a RoboTaxi fleet aims to minimize two competing values: the passengers waiting time between requests and pickups and the total distance traveled by empty vehicles. In a RoboTaxi system, vehicles may travel empty either when deadheading to pick up a passenger or while making rebalancing trips. There is a trade-off between the rebalancing effort, which includes various types of costs, and the passengers' waiting time, representing the quality of the service provided by the system.

## 1. Introduction

One-way car-sharing systems consist of cars scattered within a predefined service area and can be used for short journeys at any time of the day. A user can rent a car in a self-check-in process near the origin of their journey and return it near their destination. Some systems are station-based, meaning that the car must be checked in and checked out at a designated parking spot (i.e., one of the system's stations). In other free-float systems, the user can leave the car at any valid parking space in the service area.

One-way car-sharing systems have appeared in many cities as a complement to public transit systems. Their availability in a city may encourage residents to own fewer cars and make a larger share of their journeys using public transit. Consequently, scarce parking spaces are saved, and traffic congestion is eased. The appearance of car-sharing systems is one phenomenon in a broader economic and social process of servitization that allows higher utilization of scarce and expensive resources (Shaheen and Chan, 2015).

The demand for journeys in one-way car-sharing systems varies geographically throughout the day and is inherently asymmetric; see, for example, Kaspi et al. (2014). Indeed, in a typical city, many residents wish to travel from residential areas in the morning and toward residential areas in the afternoon. Moreover, since users of the car-sharing system are likely to combine this system with public transit and other means, the demand is not necessarily symmetric, even in the long run. For example, some people travel

☆ This article belongs to the Virtual Special Issue on IG005586:VSI: On-Demand Transportation.

* Corresponding author.
*E-mail address:* talraviv@tau.ac.il (T. Raviv).

https://doi.org/10.1016/j.trc.2023.104196
Received 14 August 2022; Received in revised form 28 May 2023; Accepted 2 June 2023
Available online 12 June 2023
0968-090X/

from home by public transit during the day when the frequency of public transit vehicles is high but prefer to return at night with a shared car. This variability and asymmetry cause an imbalance in the system and reduce the availability of cars where and when they are needed. Moreover, even if the demand were homogeneous in time and symmetric, relatively large numbers of cars would accumulate in zones where demand is low because when a journey ends in such a zone, the car is likely to stay idle there for an extended period until it is rented again.

To mitigate the effect of the inherent imbalance of one-way car-sharing systems, operators employ workers who reposition cars in the system to rebalance it. That is, workers move cars from low-demand areas to locations where they are needed. However, the repositioning process is labor-intensive and very intricate to plan. The relocation of each car requires sending a worker with another car to drive the car to its destination, from which the driver must be picked up again for his next relocation task (Illgen and Höck, 2019).

The emerging technology of driverless automated vehicles (AVs) may be used to create new types of transit systems, collectively referred to as automated mobility on demand (AMoD); see, for example, Turan et al. (2020). In the AMoD system, passengers request journeys using a mobile app, and the system provides the service 24/7 using a centrally managed fleet of AVs. The AMoD system may provide shared rides, where several passengers, possibly with different origins and destinations, are served by the same AV. Alternatively, the AMoD system may provide private rides similar to one-way car-sharing systems and taxis. We refer to such an automated car-sharing service as a RoboTaxi system, but other authors have used the term shared autonomous vehicles (SAV). RoboTaxi systems combine the convenience of taxi service (e.g., there is no need to look for a parking space at the end of the journey) with the economic advantage of a self-driven one-way car-sharing system.

One of the main advantages of RoboTaxi systems over human-driven vehicle (HDV) car-sharing systems is that upon receiving a request, the automated vehicle can travel to the exact location of the passenger's destination rather than requiring the passenger to walk to the car. In this sense, RoboTaxi systems are more similar to ride-hailing services such as Uber and Lyft. However, while a fleet of AVs can be centrally managed, the drivers in a ride-hailing service are independent agents and, thus, are not likely to cooperate with a rebalancing scheme designed by the operator. Moreover, even if the ride-hailing companies were the driver's employer and could send them on rebalancing trips, such trips are unlikely to be profitable because having a driver relocate and wait, possibly for a long time, is very costly. In a RoboTaxi system, a large portion of rebalancing could be achieved during hours in which demand is low, and the alternative cost of AVs is minor. From a social perspective, diverting some empty journeys to low-traffic hours is desirable.

In this study, we develop an algorithm to rebalance RoboTaxi systems dynamically. The algorithm is anticipatory, meaning that the repositioning decisions are made based on a forecast of the demand during the next few hours, as well as on the system's current state. We repeatedly solve a linear program (LP) at the beginning of each period with a rolling horizon and apply its solution to the current period with some adjustments. Our algorithm considers that the travel time between a pair of locations in the service area is time-dependent, enabling us to schedule the movements of empty cars at times during which traffic is low. We build a discrete event simulation model demonstrating the system's operation in a realistic urban environment. The demand is adopted from real-life taxi journeys made in New York City (Kaggle, 2017). We use the simulation model to compare the proposed rebalancing methods with a baseline policy in which no rebalancing occurs and cars are dispatched only on a just-in-time (JIT) basis. In addition, we compare our approach to an established reactive rebalancing method proposed in the literature (Pavone et al., 2012; Marczuk et al., 2016) as well as an anticipatory rebalancing approach proposed by Iglesias et al. (2018). We note that while the recent literature on the dispatching of vehicles in AMoD systems is rich, studies that focus on rebalancing in such systems are sparser. New studies that incorporated rebalancing decisions into their simulation models followed similar methods to Pavone et al. (2012), and Iglesias et al. (2018).

Our approach is shown to significantly outperform the reactive approach in terms of reducing empty car travel time while providing a similar level of service to passengers. Comparing our approach to one presented by Iglesias et al. (2018), we demonstrate some improvement in empty car rides and service levels. When comparing our anticipatory approach to the JIT approach, we observe a significant improvement in service level in addition to some improvement in efficiency, i.e., a reduction in the empty car travel time.

The rest of this paper is organized as follows: In Section 2, we provide scientific background by reviewing the relevant literature and identifying the gap that is addressed by this study. A formal definition of the rebalancing problem and notation are introduced in Section 3. Our LP formulations of the rebalancing problem and a discussion of its adaptation to dynamic settings are presented in Section 4. In addition, in this section, we present our adaptation for the models of Pavone et al. (2012) and Iglesias et al. (2018) to the settings and assumptions of free-float systems. Our simulation model is described in Section 5. In Section 6, we report our numerical study and derive some insights from its results. In Section 7, we present some conclusions and indicate directions for future work.

## 2. Literature review

In this section, we discuss the relevant literature on rebalancing RoboTaxi systems. Some early studies on one-way station-based car-sharing systems modeled the rebalancing operation abstractly and assumed that the cost of relocating a car from station to station depends solely on the travel time between the stations and is not subject to any constraints on the availability of drivers. While this assumption is not realistic for the real-time operation of car-sharing systems, the obtained solutions apply to RoboTaxi systems, and thus, we describe them here.

In free-float RoboTaxi systems, the rebalancing and the *dispatching* policies are interdependent. The dispatching policy is the rule by which cars are assigned and sent to serve passenger requests. Dispatching is not relevant for traditional car-sharing and station-based RoboTaxi systems since, in these systems, the passenger chooses the car that serves them and not the system. It may be the case that an automated car is sent to an empty station where passengers are waiting, but this is a particular case of a rebalancing operation rather than a dispatching operation.

In the rest of this section, we review the literature on rebalancing car-sharing systems as well as dispatching and rebalancing in ride-hailing and AMoD systems with shared and non-shared (taxi) operation strategies. We then identify the gap in the literature that we aim to address in this study.

The problem of rebalancing car-sharing and ride-hailing systems has been thoroughly studied in the past decade; see Illgen and Höck (2019) for a comprehensive literature review. Most of the literature focuses on station-based systems and static settings in which the rebalancing operations are planned in advance, and the demand during the rebalancing process is neglected.

Narayanan et al. (2020) surveyed almost 200 papers on various aspects of AMoD, including land–use, environment, economy, travel behavior, traffic, safety, transport supply, and public policy. On the operational level, they categorized the components of AMoD modeling to demand estimation, fleet size, charging, parking, pricing, dispatching, and rebalancing. Our literature review focuses on dispatching and rebalancing, which are relevant to this work.

Alonso-Mora et al. (2017) presented a mathematical model for assigning vehicles to batches of passengers periodically in a ride-sharing AMoD system. They demonstrate their algorithms via a simulation based on the New York City taxicab public dataset and study the trade-offs between various service-level and operational cost key performance indicators (KPIs). Although they use the term rebalancing to describe journeys that are initiated to serve passenger requests using currently idling vehicles, they do not consider rebalancing in the sense studied in this paper; i.e., no empty rides are initiated with the aim of better serving future demand.

Hyland and Mahmassani (2018) compared six dispatching policies in RoboTaxi systems using a simulation model. Two of these policies are simple heuristics, and the other four require the solution of a mixed-integer linear programming (MILP) model. The authors show that more elaborate models can be used to reduce costs and passenger waiting times.

Vosooghi et al. (2019) evaluated the performance of AMoD systems using simulation under shared and individual (RoboTaxi) ride strategies using dispatching and rebalancing policies from the literature. Their evaluation is based on KPIs such as total waiting and in-vehicle time and total distance driven by vehicles. One of their conclusions was that the rebalancing policy plays an important role in the performance of the AMoD system. They also study the effect of fleet size and vehicle capacity on these KPIs.

Ben-Gal and Tzur (2023) solved a dispatching problem for a centrally managed ride-hailing service (an equivalent of RoboTaxi) with the goal of maximizing geographic fairness and system efficiency. Their model uses a data-driven method and distance measures to map the state of the system to vehicle assignment decisions in real-time.

In the rest of this section, we focus on studies on the rebalancing of car-sharing and RobotTaxi systems. Nair and Miller-Hooks (2011) explored optimization methods and proposed a stochastic mixed-integer programming (MIP) model with the objective of generating least-cost car redistribution plans such that a proportion of all near-term demand scenarios is satisfied, acknowledging the strong effect of uncertainty on car-sharing planning. Their model assumes that the rebalancing operation is performed before demand commences.

Repoux et al. (2019) developed a proactive staff-based relocation method for a car-sharing system that aims to reduce imbalances of vehicles and spot availability, as well to prepare the system for the expected future demand. In addition, they developed a proactive prediction-based policy that employs a Markovian model to estimate the expected demand loss while taking into account reservation information.

Jorge et al. (2014) developed a method to jointly determine the size of a fleet, the number of parking spaces, and the rebalancing operations in a station-based car-sharing system. They presented a static mathematical model with the objective of maximizing the system's profit, which is affected by the total traveling time of the cars, the net rebalancing expenses, and the cost of maintaining cars and parking spots. All the demand between stations in the static model is assumed to be satisfied (no rejections). In addition, they presented a dynamic (real-time) rebalancing model based on a forecast of the demand over the next short period and tested it in a simulation. While the study modeled rebalancing operations, its goal was to support long-term decisions regarding the fleet size and the number of parking spots at each station rather than to optimize the rebalancing operation itself.

Pavone et al. (2012) presented a reactive rebalancing policy for station-based RoboTaxi systems. The passengers are assumed to wait at their origin station until a car is available. The policy is based on a solution of a network flow model that minimizes the total distance traveled by cars due to rebalancing while trying to spread idling cars as evenly as possible among the stations in the system. The optimization model is repeatedly solved at the beginning of each short period, and the rebalancing solution is applied. The model assumes there are no proactive dispatching operations. That is, if a passenger arrives at an empty station, they will have to wait until the end of the rebalancing period, and only then will a car be sent to pick them up. Alternatively, a car with a passenger that finishes his trip at the station may arrive and be assigned to the passenger. The performance of the proposed rebalancing policy was tested using a simulation study. The average number of waiting passengers and cars in transit as a function of fleet size and rebalancing policy was estimated. Moreover, the method's robustness to uncertain travel times was demonstrated. It was shown that when the fleet size is sufficiently large, almost all passengers can be served immediately.

Zhang and Pavone (2016), based on Pavone et al. (2012), provided a solution to an offline rebalancing problem for a station-based RoboTaxi system in which passengers leave immediately if they arrive at a station with no available cars. They proposed a queueing-theoretical approach that models a network of AVs as a Jackson network and produces a linear program that equalizes the fleet availability across all stations. Their algorithms were tested using taxi data from New York City, and it was demonstrated that the current taxi demand in New York City could be served with 40% fewer cars.

Marczuk et al. (2016) developed a simulation framework for a station-based RoboTaxi system using the SimMobility simulation environment (Adnan et al., 2016). They proposed dispatching and rebalancing policies. They presented an online dynamic rebalancing policy that aims to minimize the rebalancing effort by solving a mixed-integer linear program (MILP) based on the current availability of cars and the expected demand in the next period. The model is repeatedly solved to determine a rebalancing plan for each period separately.

Rossi et al. (2018) presented a congestion-aware algorithm for routing and rebalancing shared AVs in a RoboTaxi system. Their rebalancing method was built upon previous studies by Pavone et al. (2012) and Zhang and Pavone (2016). The study showed that the passenger routing and rebalancing problems can be decoupled and that, under reasonable assumptions, the rebalancing process does not contribute to congestion.

Hörl et al. (2019) used MATsim (Horni et al., 2016) to evaluate scenarios in which all passenger transportation in a city is served by the AMoD system. They applied reactive rebalancing policies presented by Pavone et al. (2012) adapted to a free-float system with dispatching rules based on the optimal matching of open requests and available cars, which was solved periodically. The dispatching policies were adapted from Bischoff and Maciejewski (2016), and Ruch et al. (2018). They concluded that with the tested operational policies, the service cost is significantly higher than that of public transit and private cars.

Fagnant and Kockelman (2014) investigated the operation of a station-based RoboTaxi system and presented heuristic dispatching and rebalancing rules. They tested their method using simulation in a synthetic grid-like city, with grid cells of quarter-mile squares. The demand is generated periodically, every five minutes, between the centers of grid cells and is served by cars that are available at the origin cells or in cells within a predefined distance. If no such car is available, the passengers wait in a queue until a car becomes available in one of the following periods. The rebalancing rules are based on grouping grid cells into blocks of one- or two-mile squares (16 or 64 cells). A heuristic criterion is used to move cars between neighboring blocks based on the current availability of cars in the block, the number of waiting passengers, and the expected demand in the following short period.

Winter et al. (2017) developed a simulation model for a station-based RoboTaxi system using the MATSim environment (Horni et al., 2016). They used a previously proposed assignment-based dispatching rule (see Maciejewski et al., 2016), implemented five rebalancing policies, and tested their performance on a RoboTaxi system in terms of service level and external impact on congestion in the city. The baseline policy of leaving each car at the end of its last trip until it is requested was shown to outperform the other four tested rebalancing strategies in terms of service level, which implies that a more sophisticated rebalancing scheme is required.

Babicheva et al. (2018) studied the dispatching and rebalancing policy for a station-based RoboTaxi system. They presented a proactive greedy heuristic for the redistribution of empty cars based on the number and seniority of the waiting passengers and a demand forecast for the near future. The model studied in this short paper is different from ours in several respects since we assume a free-float system in which passengers are notified about their pickup time upon placing a request.

Iglesias et al. (2018) solved a dynamic rebalancing problem for a station-based RoboTaxi in a similar setting to that of Pavone et al. (2012). They presented an anticipatory rebalancing method that repeatedly solves an LP at the beginning of each short period in a rolling-horizon fashion. The solution is based on a demand forecast for several future periods. They demonstrated that the anticipatory model performs much better than previously introduced reactive approaches in terms of passenger waiting times. In Section 4.2, we present an adaption of this model to a free-float setting.

In Table 1, we summarize the relevant literature on the rebalancing of car-sharing and RoboTaxi systems and compare them with respect to the most relevant characteristics. For each model, we first present the type of cars in the system: HDVs or AVs. Next, we indicate the network type: station-based or free float. The dispatching policy (applicable only in free-float networks) is presented in the fourth column. Next, we present the forecast horizon. Some studies take a myopic approach that considers only the current state of the system, while others base rebalancing decisions on a demand forecast for one or several short future periods. In the sixth column, we present each study's KPIs, measured using simulation. The last column presents the methodology used to solve the rebalancing problem, namely, LP, ILP, MILP, heuristic, or simulation-based optimization.

It is apparent from Table 1 that most previous studies based rebalancing decisions on myopic or short-sighted considerations. The myopic approach seeks to distribute cars evenly in the service area without considering the temporal demand patterns. It may provide good service when the safety margins are sufficiently wide but incurs the cost of many unnecessary rebalancing trips. The short-sighted approaches address system imbalance only when it is imminent, which can be too late.

The only previous study that considered a forecast horizon of multiple periods in a dynamic setting is Iglesias et al. (2018). The current study extends it in the following dimensions: (1) We present a model that works in a free-float rather than a station-based system. Our model is tested in a simulation environment with an effective dynamic dispatching rule that sends the earliest available car to each passenger, while Iglesias et al. (2018) assumes that passengers wait at stations until the periodic rebalancing mechanism supplies them with cars. (2) While the Iglesias et al. (2018) rebalancing model is based on an integer demand forecast for each future period between each origin–destination pair, we based our rebalancing optimization on an aggregated forecast of the net demand of each zone in every period. This approach allows a more accurate forecast while maintaining the tractable network-flow structure of the problem. (3) Our rebalancing model uses a discounting mechanism to prioritize rebalancing decisions that help meet more imminent demand. Because our model is applied in a rolling-horizon fashion and future rebalancing decisions are revisited, this approach enables the demand to be met more accurately. (4) Our model considers time-varying travel times (based on historical trip data). This information helps the anticipatory rebalancing model initiate rebalancing trips when there is low traffic.

In extensive numerical experiments, we compare the performances of our rebalancing approach to adaptations of the models presented by Pavone et al. (2012) and Iglesias et al. (2018) to our free-float setup and advanced dispatching mechanism. Moreover,

**Table 1**
Literature on rebalancing operations.

| Paper | Car type | Network type | Dispatching | Forecast Horizon | KPIs | Methodology |
|---|---|---|---|---|---|---|
| Nair and Miller-Hooks (2011) | HDV | Station based | NA | One period | Empty car rides | MILP |
| Jorge et al. (2014) | HDV | Station based | NA | One period | Profit | ILP |
| Repoux et al. (2019) | HDV | Station based | NA | One period | Rejections | On-line Heuristic |
| Pavone et al. (2012) | AV | Station based | NA | Myopic | Empty car rides<br>Waiting time | LP<br>Simulation |
| Spieser et al. (2016) | AV | Station based | NA | One period | Empty car rides<br>Waiting time<br>Walks away | LP |
| Zhang and Pavone (2016) | AV | Station based | NA | Myopic | Empty car rides<br>Waiting time | MILP |
| Marczuk et al. (2016) | AV | Station based | NA | One period | Empty car rides<br>Waiting time | MILP |
| Fagnant and Kockelman (2014) | AV | Free float space is discretized | Closest available car[a] | One period | Empty car rides<br>Waiting time<br>Fleet utilization | Heuristic-Rules |
| Winter et al. (2017) | AV | Free float | Assignment Based | Myopic and One period | Waiting time<br>Fleet utilization<br>Deadheading time | Five Heuristic-Rules |
| Babicheva et al. (2018) | AV | Station based | Greedy assignment | One period | Waiting time | Heuristic |
| Iglesias et al. (2018) | AV | Station based | NA | Multi periods | Empty car rides<br>Waiting time | LP |
| This study | AV | Free float | Earliest available car[b] | Multi periods | Empty car rides<br>Waiting time<br>Fleet utilization | LP |

[a]The passengers wait until a car is available at a five-minute travel distance.
[b]Allows dispatching of currently occupied cars.

for a fair comparison, our implementation of the Iglesias et al. (2018) model considers the time-dependent travel time in the same way as in our proposed model. The three rebalancing optimization approaches, and a baseline JIT approach are tested in the same simulation environment we created based on realistic trips and travel time data. The experiment demonstrates the merits of our approach.

## 3. Problem description

A rebalancing policy in a RoboTaxi system is a set of criteria by which idle cars are moved from their current locations. The effectiveness of the policy is measured by two competing KPIs: the average waiting time of the passengers and the total travel time of empty cars. When a passenger places a request, a car is immediately assigned to them. Empty car trips occur due to either deadheading while traveling to pick up a passenger or rebalancing operations.

Our problem definition is based on assumptions regarding the operation of the RoboTaxi system. A passenger can place a request for a ride between any pair of points in a predefined service area. Rides are not shared between passengers unless they request the same ride together from the same origin to the same destination. Upon their requests, passengers are notified of the expected arrival time of the car that will pick them up. We consider an operational model in which requests are accepted for immediate rides only. That is, passengers cannot book future rides. The travel time of the car in the city is time-dependent (e.g., there are longer travel times during rush hours) but deterministic.

A dispatching policy is the set of criteria by which cars are assigned to serve requests. Notably, an optimal rebalancing policy may be affected by the dispatching policy used by the operator. This study focuses on the rebalancing policy and assumes the following dispatching policy: upon receiving a request, the car that can pick up the passenger as early as possible is assigned to this request. Note that the assigned car may be idle, in which case the pickup time consists of only the deadheading time of the car from its current location to the location of the request. Alternatively, a busy car may be assigned to serve the request. In this case, the pickup time consists of the time until the car becomes available and the deadheading time from its availability location to the request's location. We note that the advantage of this dispatching policy is that it allows the passengers to be notified of their expected pickup time when they place a request so that they can arrive at their desired pickup location at the expected arrival time. This implies that passengers never have to wait outside for a car.

It is possible to design a different dispatching policy that can reject some passenger requests, particularly if they require long waiting times. For example, the operator could set a time limit of ten minutes of waiting and reject all requests that cannot be served within this time. Similarly, the passengers' patience may be limited, and they may decline the service immediately after being notified about the car's arrival time if the expected waiting time is too long. The rebalancing method presented in this paper may also apply to these settings. However, we believe that the RoboTaxi system should be designed with sufficient capacity and a proper rebalancing policy such that it can serve all passengers within a short time. Hence, the phenomena of rejections and passengers declining service are very rare. In our numerical experiment in Section 6.2, we demonstrate that almost all requests are served within a few minutes in a realistic system.

Some previous authors (e.g., Alonso-Mora et al., 2017) took a different approach to the dispatching problem. They assumed that requests are batched and cars are assigned to passengers periodically, say every minute. With such a dispatching policy, it could be reasonable to integrate the dispatching and rebalancing decision into a single optimization model. The advantage of assigning cars in batches is that a better assignment solution can be found when multiple requests are considered simultaneously. This may be significant if each batch consists of numerous requests with many possible interactions but clearly cannot improve the assignment if there are few far-apart requests in each batch. Therefore, the batching approach is beneficial in practical settings if the demand rate is very high (e.g., a few requests per minute in each neighborhood). If the demand rate is low, the operator can still form batches by solving the assignment problem for the requests accumulated over a longer period (say every five minutes), but this implies a significant adverse effect on the passengers' experience. Indeed, using batches of relatively long periods not only increase the waiting time for the cars but also slows the response time of the system (i.e., the time between the request is received in the system and the moment the users are notified about their expected pickup time).

We believe that passengers of a RoboTaxi system will expect to be notified about their pickup time a few seconds or at most one minute after placing their request because this is the experience provided by current ride-hailing services such as Uber and Lyft. The rebalancing decisions, however, can be made periodically, say every 15 min, based on aggregated information and forecasts because these decisions do not directly affect the passenger's experience.

The demand of passengers is a nonhomogeneous stochastic arrival process of requests characterized by origin and destination locations. In practice, modeling and estimating such a time-dependent and location-dependent stochastic process is a complicated task, as discussed in Sections 4 and 5.

The RoboTaxi rebalancing problem aims to find a policy that jointly minimizes the expected weighted sum of the average passenger waiting time, the ratio of rejected requests, and the average total empty car travel time.

## 4. Methodology

The operation of a RoboTaxi system, including dispatching and rebalancing activities, is dynamic by nature. At any given moment, the operator reacts to the state of the system and newly available information by sending cars to pick up passengers or rebalancing trips. Our solution to the rebalancing problem is to make rebalancing decisions periodically (say every fifteen minutes) by solving a static optimization problem with the system's current state, including pending dispatching operations and possibly a forecast of demand over the next few periods, as input. We formulate this optimization problem as an LP and apply it in a rolling-horizon fashion at the beginning of each period. Since the rebalancing operation of a car may extend over several periods, the rebalancing operations in progress when the model is solved are also considered as input. We stress that while the rebalancing decision-making process is periodic, the car dispatching decisions are made in real-time, immediately after the arrival of each request. Note that there is no one-to-one mapping of plans made by the rebalancing and operational models. In the rebalancing process, the demand can be met only by cars that are located in the zone of their origin during the period when the request is generated. In the operational model, a request can trigger the relocation of cars between zones, and passengers may wait some time before they are served. The rebalancing model is a simplification of the operational model that is used to match the locations of the cars with the forecasted demand.

To optimize the rebalancing operations, the service area is discretized into zones, where each zone is represented by a single focal location at its geographic center. The system's state is represented by the number of cars idling at or traveling to each zone without fine-grained information about their particular locations. When a rebalancing operation is performed, i.e., a car is relocated from one zone to another, it is sent from its current location to the center point of the destination zone. The travel distance, in this case, is approximated by the travel time between the centers of the zones. We stress that these simplifying assumptions are valid only for the optimization model that guides the rebalancing operation. However, the rebalancing model results are adapted to a realistic situation in which time and space are continuous. The evaluation of the KPIs in the results presented below was obtained from a simulation that periodically employs the discretized rebalancing model to make rebalancing decisions and adapt them to the actual state of the system.

The success of our rebalancing optimization model partly depends on partitioning the service area into zones. While the zoning problem is not the focus of this study, in Appendix, we present the mathematical model used to create the zones in our experiments.

The system maintains a destination location and expected arrival time for each moving car. When a car, either idle or moving, is assigned to a passenger, the availability time of the car is increased by the expected deadheading time and the passenger's journey time. The availability location of the car is updated to the destination of the new journey. On the basis of this information, each car's availability period and zone can be derived when rebalancing optimization occurs. To this end, the availability time of an idling car is set to the current time, and its availability zone is derived from its current location.

The travel time parameter between each pair of zones used by the rebalancing models is time-dependent to represent the time-varying travel time due to expected variations in traffic conditions. We denote the travel time between zones $i$ and $j$ when departing in period $t$ by $\tau_{ijt}$. In the anticipatory models, we also use a discretized version of the travel time parameter, $\tau'_{ijt}$, which represents the travel time in terms of periods, rounded up, for journeys that **end** in period $t$.

The anticipatory models plan situations in which the system may reject passenger requests. If the number of cars in the system and the passengers' patience are limited, such rejections are unavoidable. However, the rebalancing optimization model planning a rejection does not necessarily imply that a passenger will leave unserved in (the simulated) reality. While in the optimization models, cars are dispatched only from the zone of the request, our advanced dispatching policy sends cars to serve passengers from neighboring zones as well.

To demonstrate the merits of our anticipatory zone-based model, we compare it with two alternative approaches previously presented in the literature by Pavone et al. (2012) and Iglesias et al. (2018). Since the previous literature focused on station-based systems and did not assume that cars are dispatched immediately upon request, we adapted the above two models to our free-float settings and dispatching rule. In addition, we tested and compared the performances of these rebalancing policies to that of a simple JIT approach, where idle cars are always left where their last trip ended.

In Sections 4.1 and 4.2, we present our adaptation of the reactive model of Pavone et al. (2012) and the anticipatory trip-based model presented by Iglesias et al. (2018). In Section 4.3, our novel anticipatory zone-based rebalancing model is presented.

### 4.1. Reactive rebalancing model

In this section, we adapt the reactive model presented by Pavone et al. (2012) to our free-float setting. The model is applied periodically, say every fifteen minutes, to move idle cars between zones to spread available cars evenly across the zones. When rebalancing is executed (or simulated), cars are moved from the origin zone to a predefined focal location in the destination zone. The cars selected to be moved are those closest to the location of the origin zone. Since, according to our dispatching policy, a car is assigned immediately upon a passenger's request, there are no outstanding passengers in the system, and cars are considered idle only if they are not already assigned to passengers or on a rebalancing trip. In this respect, our setting is different from that of Pavone et al. (2012), where the number of outstanding passengers must be deduced from the number of available cars.

$\mathcal{N}$ Set of zones

$s_i^0$ Total number of cars in zone $i$ or in transit to it.

$v^d$ Desired number of cars in each zone. In their numerical experiment, Pavone et al. (2012) used $v^d = \left\lfloor \frac{\sum_i s_i^0}{|\mathcal{N}|} \right\rfloor$, and this method is adopted here.

$\tau_{ij}$ Travel time from zone $i$ to zone $j$. In our model, the travel times are time-dependent and are updated for $\tau_{ijt}$ each time the model is solved according to the current state.

The decision variables and the optimization model are as follows:

$x_{ij}$ Number of cars to relocate from zone $i$ to zone $j$ in the current period

$$\min \sum_{ij} \tau_{ij} x_{ij} \tag{1}$$

subject to

$$s_i^0 + \sum_{j\neq i}(x_{ji} - x_{ij}) \geq v^d \qquad \forall i \in \mathcal{N} \tag{2}$$

$$x_{ij} \geq 0 \qquad \forall i, j \in \mathcal{N}. \tag{3}$$

The objective function (1) minimizes the rebalancing effort, i.e., the total travel time of rebalancing trips. Constraint (2) relocates cars between zones to ensure that a sufficient number of cars will be available at each station after the rebalancing operation is completed. Note that this is a network flow model with a totally unimodular coefficient matrix. Therefore, since $v^d$ is an integer, in an optimal solution, the values $x_{ij}$ are integers.

A straightforward improvement of the reactive model is to set a different target level $v^d$ for each zone and each time of the day based on the expected demand in the zone. However, we note that estimating location and time-dependent desired levels may be a more involved task than including the expected future dynamics of the demand in the optimization model as done by the trip-based model presented in Section 4.2 and by our zone-based model presented in Section 4.3.

### 4.2. Trip-based rebalancing model

This section presents our adaptation and extension of the trip-based anticipatory optimization model presented by Iglesias et al. (2018). Its input consists of all the parameters used in the reactive model, (1)–(3), and the following additional parameters: (1) A forecast of the demand between each pair of zones in each of several following periods. This demand may change from period to period, and the input is based on the best forecast that can be achieved when the model is solved. The forecast for each such demand stream must be an integer for the model to be tractable. (2) The number of cars that are known to become available in each zone in each of the following few periods. A car is considered to be available in a zone in a future period if it is moving toward it either with a passenger or on a rebalancing trip. A car that is planned to arrive at a zone sometime during a period is assumed to be available for new requests only at the end of the period. For the sake of rebalancing planning, passengers are served only by cars that are available at the beginning of the period of their request in their origin zone. We stress that the above assumptions are valid only for the rebalancing optimization model but not in the real (or simulated) system. In our simulation model described below, cars are dispatched immediately, even if they are still busy with previous tasks, and the KPIs of the system are evaluated with the above simplifying assumptions removed.

The trip-based model simultaneously determines the rebalancing tasks and the assignments of cars to serve future demand between pairs of zones. These decisions are tentatively made in advance for several future periods (e.g., twelve periods of fifteen minutes each), referred to as the planning horizon. The objective function aims to minimize the weighted sum of the rebalancing time and the number of passengers the model cannot serve during the planning horizon.

While the trip-based model plans the rebalancing operation for several periods ahead, it is solved at the beginning of each period, and only the rebalancing decisions of the first period of the planning horizon are implemented. The rest of the periods are optimized only to mitigate the end-of-the-horizon effect. Since the rebalancing optimization considers several future periods, the model may take advantage of periods with low traffic to perform rebalancing tasks in preparation for future periods in which the traffic is expected to be slower, and the demand is likely to be higher. The time-dependent travel times make such preparation more likely to occur.

We note that the adaptation of Iglesias et al. (2018) presented here modifies the original model in the following respects: (1) we consider time-dependent travel times; (2) we do not address outstanding passengers waiting for service at the beginning of the planning horizon because our advanced dispatching policy makes this unnecessary; (3) the notation is modified and auxiliary decision variables are introduced to make the formulation clearer and to conform to the notation used in our model presented in Section 4.3.

Next, we formally define the model parameters and decision variables and present the objective function and constraints.

$N$ Set of zones in the service area.

$T$ Planning horizon. $T = \{1, 2, \dots, |T|\}$.

$\lambda_{ijt}$ Expected demand for journeys from zone $i$ to zone $j$ during period $t$ (we assume that all these journeys start at the beginning of the period).

$\tau_{ijt}$ Travel time from zone $i$ to zone $j$ (in seconds) for a journey that starts in period $t$.

$\tau'_{ijt}$ Travel time from zone $i$ to zone $j$ (in units of periods, rounded up to the next integer) for a journey that **ends** in period $t$.

$\alpha$ Weight of the deadheading travel time in the objective function.

$\beta$ Weight of each rejected passenger in the objective function.

$s_{i0}$ Number of cars that are available in zone $i$ at the beginning of the planning horizon.

$s_{it}$ Number of cars that are traveling when the model is solved and are scheduled to be available in zone $i$ by the end of period $t$.

The decision variables and the optimization model are as follows:

$x_{ijt}$ Number of cars to transfer from zone $i$ to zone $j$ starting in period $t$. This number refers only to rebalancing operations.

$y_{ijt}$ Number of cars tentatively assigned in period $t$ to passengers who travel from zone $i$ to zone $j$.

$I_{it}$ Number of cars in zone $i$ at the end of period $t$.

$$\min \sum_{t\in T} \sum_{i,j\in N} \left(\alpha \tau_{ijt} x_{ijt} + \beta \rho^{t-1} (\lambda_{ijt} - y_{ijt})\right) \tag{4}$$

subject to

$$I_{it} = I_{i,t-1} + s_{it} + \sum_{j\in N:\tau'_{jit}<t} \left(y_{ji,t-\tau'_{jit}} + x_{j,i,t-\tau'_{jit}}\right) - \sum_{j\in N} \left(y_{ijt} + x_{ijt}\right) \qquad \forall t \in T, i \in N \tag{5}$$

$$I_{i0} = S_{i0} \qquad \forall i \in N \tag{6}$$

$$\lfloor \lambda_{ijt} \rfloor \geq y_{ijt} \geq 0 \qquad \forall i, j \in N, t \in T \tag{7}$$

$$I_{it} \geq 0 \qquad \forall i \in N, t \in T \tag{8}$$

$$x_{ijt} \geq 0 \qquad \forall i, j \in N, t \in T. \tag{9}$$

The objective function (4) consists of two components. The first component is the total cost of the rebalancing operations, and the second is the penalty for the estimated number of rejected passengers based on the expected demand. When the operator chooses a relatively high value for $\beta$, a rebalancing solution with many rejections is less likely, and the system exerts more effort for rebalancing operations. Since the units of $\tau_{ijt}$ are traveling time in seconds while the units of $\lambda_{ijt}$ and $y_{ijt}$ are numbers of passengers, $\beta$ is typically much greater than $\alpha$. Note that the solution is affected by the ratio between $\alpha$ and $\beta$ and not by their absolute values. Finally, we note that the term $\beta\ \lambda_{ijt}$ is constant and thus can be removed from the objective function without affecting the solution.

The second component of the objective function is discounted by a factor of $\rho$ per period. We note that giving more weight to more imminent rejections makes sense because future rejections may be eliminated later when more reliable predictions about the system's state are available. We do not discount the rebalancing cost component since we do not wish the model to postpone the rebalancing operation when there is sufficient capacity to eliminate imminent rejections.

Constraint (5) is an inventory balance equation for each zone in each period. The number of cars in zone $i$ at the end of period $t$ in the planning horizon ($I_{it}$) equals the number of cars in the zone at the end of the previous period plus the number of cars that are scheduled to arrive in the zone during the period due to decisions that were made before the current horizon ($s_{it}$). Note that $s_{it}$ may include passenger journeys that are already assigned and rebalancing operations that are in progress when the current rebalancing plan is made. The right-hand side of (5) also includes the net number of cars that arrive in the zone at time $t$ due to the assignments of cars to passengers and rebalancing operations that started in earlier periods of the planning horizon: $\sum_{j\in N:\tau'_{jit}<t}\left(y_{ji,t-\tau'_{jit}}+x_{j,i,t-\tau'_{jit}}\right)$. Thanks to the summation condition, $\tau'_{jit}<t$, only periods defined in the current planning horizon are referenced. Finally, the number of cars that are assigned to passengers in the zone in period $t$, i.e., $y_{ijt}$, and cars that are sent to other zones due to rebalancing operations ($x_{ijt}$) are subtracted from the "inventory" of the zone at the end of the period. Eq. (6) sets the initial number of available cars in each zone at the beginning of the planning horizon.

Constraint (7) ensures that the number of passengers served between each origin–destination pair during each period is nonnegative and does not exceed the demand. The expected demand for trips is rounded down to ensure the totally unimodularity of LP (4)–(9). In their paper, Iglesias et al. (2018) assumed that the expected (forecast) demand is an integer. However, this assumption is not compatible with the nature of the demand, especially when many streams between various zones during short periods are considered. Since we assume that the arrival process of each origin–destination pair follows a Poisson distribution and the mode of this distribution is the rounded-down rate parameter, we replace $\lambda_{ijt}$ with $\lfloor\lambda_{ijt}\rfloor$.

Constraint (8) disallows the assignment of cars that are not currently available in each zone. Constraint (9) stipulates that a nonnegative integer number of cars are relocated between zones.

### 4.3. Zone-based rebalancing model

The trip-based model uses the demand forecast of the trips between each pair of zones in the service area during each of the next few short periods. Naturally, the expected number of trips between most zones is usually a small fractional number. However, for the model to be tractable, the demand parameters must be rounded, which broadly distorts the total expected demand. Conversely, the expected net demand faced by a zone, i.e., the difference between the aggregated expected numbers of trips from the zone and to the zone, is a larger number that is only slightly distorted by the rounding operation. Moreover, the estimation of the aggregated net demand of a zone is less noisy than the estimation of the much smaller number of trips between each pair of zones. Therefore, it can be performed reliably with fewer data points.

In this section, we present a rebalancing model that uses a forecast of the aggregated net demand of each zone. In Section 6, we show that it leads to a more effective rebalancing operation when applied in a simulated system. We use the same notation as in the trip-based model and introduce a new decision variable.

$d_{it}$ Number of rejected passengers who wish to travel from zone $i$ in period $t$.

$$\min\sum_{t\in T}\sum_{i\in N}\left(\beta\rho^{t-1}\ d_{it}+\alpha\sum_{j\in N:j\neq i}\tau_{ijt}x_{ijt}\right)\tag{10}$$

s.t

$$I_{it}=I_{i,t-1}+s_{it}-\left\lfloor\sum_{j\in N}\lambda_{ijt}-\sum_{j\in N:\tau'_{jit}<t}\lambda_{ji,t-\tau'_{jit}}\right\rfloor+\sum_{j\in N:\tau'_{jit}<t}x_{j,i,t-\tau'_{jit}}-\sum_{j\in N}x_{ijt}+d_{it}$$
$$\forall t\in T,i\in N\tag{11}$$

$$I_{i0}=s_{i0}\qquad\forall i\in N\tag{12}$$

$$I_{it}\geq 0\qquad\forall i\in N,t\in T\tag{13}$$

$$x_{ijt}\geq 0\qquad\forall i,j\in N,t\in T\tag{14}$$

$$d_{it}\geq 0\qquad\forall i\in N,t\in T.\tag{15}$$

The objective function minimizes the weighted sum of the number of rejected passengers and the total time of the planned rebalancing trips. We use the idea of discounting future rejections as in the trip-based model.

Constraint (11) is a flow conservation equation that assigns an inventory level to each zone, $i$, in each period, $t$, in the planning horizon. It is calculated as the sum of the following terms: (1) number of cars available from the previous period ($I_{i,t-1}$); (2) number of cars that are on trips that began before the planning horizon and are expected to arrive in the zone in period $t$ ($S_{it}$); (3) minus the net expected demand for cars in zone $i$ in period $t$ rounded down, including all the trips from zone $i$ in period $t$ minus the number of trips that are expected to arrive in the zone with passengers from other zones, which they leave after the beginning of the planning horizon ($\sum_{j\in N}\lambda_{ijt} - \sum_{j\in N:\tau'_{jit}<t}\lambda_{ji,t-\tau'_{jit}}$); (4) number of cars that, according to the plan, should arrive in the zone due to rebalancing operations ($\sum_{j\in N:\tau'_{jit}<t} x_{j,i,t-\tau'_{jit}}$); (5) minus the number of cars that are planned to leave the zone due to rebalancing operations ($\sum_{j\in N} x_{ijt}$); (6) number of rejected trips, which is used as a slack variable for this constraint to ensure feasibility, i.e., the nonnegativity of $I$ ($d_{it}$).

Eq. (12) initializes the number of cars in each zone at the beginning of the planning horizon. Constraints (13)–(15) define the domains of the decision variables. As in the trip-based model, the coefficient matrix of the linear program (10)–(15) is totally unimodular, and the right-hand-side constants are all integers. Therefore, there is no need to require integrality to obtain integral values of the decision variables, and the model can be solved very quickly.

The zone-based rebalancing model allows rejection as a safety measure to guarantee the feasibility of the solution. However, the effect of rejection on the number of cars that leave and arrive at each zone is neglected for the sake of forecasting the future state of the system. Indeed, rejections are not materialized in the actual system (and in our simulation). We also neglect the possible occurrence of passenger abandonment, assuming that when the system is properly balanced, these occurrences are rare.

Since the zone-based model does not assign future trips to cars, we do not need the variable $y_{ijt}$ used in the trip-based model. The arrival of cars with passengers to a zone is assumed to follow historical patterns expressed by the estimated demand. An arrival that does not occur due to the planned rejection of passengers is neglected by the zone-based model. However, since the rejection phenomenon is very rare if the system is designed well, this does not significantly affect the ability of the model to plan the rebalancing operation efficiently.

## 5. Simulation model

To test our models and evaluate the system performance, we developed a discrete event simulation model for the RoboTaxi system in Python 3. Our documented code is available on GitHub. The simulation model relaxes many of the simplifying assumptions of the rebalancing optimization model. In particular, requests arrive in continuous time according to a nonhomogeneous Poisson process that characterizes the demand between each origin and destination zone. The exact origin and destination locations are sampled from a large set of possible stopping locations in the respective zones.

The demand rate between each pair of zones in each period can be initially estimated from the actual trip information of a similar system, such as a traditional taxi fleet, or other sources, such as location data obtained from a cellular network. In each zone, we define a list of potential stopping locations that are used as the actual exact pick-up and drop-off locations of each trip. Once the system is operational, future demand can be forecasted based on the requests received by the system. For our experiment, we used taxi trips in NYC recorded over six months in 2016. The dataset was obtained from Kaggle (2017) and is referred to as the NYC Taxi Trip Duration dataset. In Section 6.1, we describe how this dataset was processed to create our demand input data.

The time-dependent travel times of both the passengers' journeys and deadheading trips were estimated by combining information queried from the Open Source Routing Machine software package (Luxen and Vetter, 2011) based on geographic data obtained from the Open Street Map project (OpenStreetMap contributors, 2017) and the actual travel times of the trips used to create the demand data.

In the simulation, the system's state is described by each car's availability time and location. If the availability time is earlier than the current time, the car is idle at its availability location. Otherwise, the car is moving (either empty or with a passenger) and is expected to be available at the specified time and location.

The simulation repeatedly processes two major types of events: *new period* and *request* events. The events are stored in a heap, where each event is a record that contains its occurrence time, the type of the event, and possibly some additional information. In each iteration, the event with the earliest time is extracted from the heap, the simulation clock is advanced to the time of the event, and the event is processed. That is, some calculations are performed, the state variables are updated, statistics are gathered, and new events are created and inserted into the heap.

Before the simulation starts, a new period event is created at time 0. The new period event creates multiple request events based on the expected demand of each origin–destination zone and another new period event at the beginning of the next period. In the request event record, we store the origin and destination locations. The number of request events for each origin and destination is sampled from a Poisson distribution with a parameter rate corresponding to the time of the day. The exact times of the requests within the period are generated using a uniform distribution. This creates a nonhomogeneous Poisson arrival process of requests for each origin and destination zone, with arrival rates constant throughout each period. The exact pick-up and drop-off locations in the origin and destination zones are sampled from a list of potential stopping locations defined for each zone.

Next, the rebalancing model is run on the new period events using data collected from the system's current state. Using the solution of this model, rebalancing trips are initiated. We scan the origin–destination zones between which the rebalancing model prescribes moving cars and then greedily move idling cars from the origin zone closest to the destination zone's center. Technically, this means that the availability time of the car is set to the current time plus the travel time from its location to the center of the destination zone. The availability location of the car is also updated accordingly. It is also possible to simulate this system without rebalancing. In such a case, the new period event only creates new requests and does not initiate rebalancing trips.

Since our rebalancing models may take a long time to run, while the rebalancing decision should be made online, we limit the running time of the solver (IBM CPLEX) to a predefined time of 30 s in our experiments.

The "request" event triggers the following calculation: the simulation locates the car that can reach the passenger as quickly as possible. For each car in the system, we calculate the earliest possible arrival time at the location of the request, which is the time until the car becomes available (if it is not currently idle), plus the travel time from its availability location to the request's location. Once a car is assigned (dispatched) to a request, we immediately update the availability time and location of the car.

We also define several additional events to collect statistical information and write it to external files once on each simulated day. The statistics are divided into blocks of one day to allow statistical tests that compare system configurations and analyze the warmup time. We collect the number of trips (accepted requests), the number of rebalancing trips, total rebalancing time, total deadheading time (i.e., the time during which cars travel empty to pick up passengers), total traveling time with passengers, the total waiting time of passengers while their allocated cars are in rebalancing, and total waiting time of passengers while their allocated cars are serving previous passengers. In addition, we collect statistics on the distribution of the waiting time and fleet utilization for each period of the day. We note that the travel and waiting time statistics are assigned to the day on which the corresponding journey started.

## 6. Numerical experiment

In this section, we present the results of a numerical experiment that explores the performance of each of the three rebalancing models and a baseline policy without rebalancing. These four policies are tested using the simulation framework. The major KPIs are the average passenger waiting time, the deadheading time, the rebalancing time, and the total empty car travel time. The first KPI represents the quality of service, and the last three represent operational costs and negative externalities from the system due to additional traffic congestion caused by empty car trips.

### 6.1. Experimental settings

The simulation was run continuously for 30 simulated days. The initial locations of the cars in the fleet were set to the centers of the zones. For each car, we randomly picked one of the zones. The time-dependent demand and travel times followed cyclic periods of 24 h and were based on the characteristics of regular working days. The demand rates and travel times were changed at the beginning of every hour. The trip-based and zone-based models were based on periods of fifteen minutes and a planning horizon of six (90 min) or twelve (180 min) periods. Each of the three rebalancing LP models was solved and applied in every period with a time limit of 30 s. CPLEX always converged to the optimal solutions of the zone-based and trip-based models.

The weights of the objective functions of the zone-based and trip-based models were selected as follows. The weights of the empty travel and passenger waiting times were set to $\alpha = 1$. The weight of passenger rejection ($\beta$) was fine-tuned by checking various values and was set to 3900. Only the ratio between $\alpha$ and $\beta$ affects the optimal solution. There is only one degree of freedom when selecting the weight parameters for the zone-based and trip-based models. Another parameter that should be tuned is the discount factor $\rho$. In our experiment, we selected $\rho = 0.99$ per period, yielding promising results for many problem instances.

The optimization models assume and plan rejections if it is impossible to fulfill requests using cars that are available in their origin zone and period. However, in our simulation, requests are served by cars from other zones, and the patience of passengers is assumed to be unlimited; hence, there are no rejections. The size of the fleet is planned such that all the requests are served in a reasonable time (typically a few minutes).

Our experiment was based on the actual geography and transportation demand in Manhattan. The service area was divided into 70 zones using the model described in Appendix. The centers of the zones were selected from a list of the 1262 bus stations in the service area obtained from Open Street Map (OpenStreetMap contributors, 2017).

We created time-dependent demand input data between each origin and destination zone for the simulation using the NYC Taxi Trip Duration dataset from Kaggle. The dataset consists of 1,458,644 trip records in greater NYC. Our experiment defined the service zone as Manhattan; thus, we filtered from the dataset approximately one million trips that started and ended inside this zone. For each trip, the data contain the exact origin and destination locations (latitude and longitude), the start time, and the trip duration. The arrival rate for each origin–destination pair of zones and each of the 24 h of the day on regular weekdays (i.e., Monday to Friday) were estimated from the data by counting the number of occurrences of such trips and dividing it by the number of weekdays in the dataset. In addition, for each zone, we created a list of locations where actual trips can start and end. These lists contain all the locations observed in the data set in each particular zone. A location that is observed several times appears more often in the list and hence is more likely to be sampled.

The expected total number of trips per day in the demand matrix estimated from the NYC Taxi Trip Duration dataset is 4879. In our experiment, we controlled the demand rate by multiplying these rates by a factor to create scenarios with 100,000 and 200,000 requests per day. We considered the first day (of the 30 simulated days) to be a warmup period. We simulated different settings and rebalancing policies using the same random-number seed to generate the same set of trip requests. This technique reduces the variability in the experiment and allows a more accurate comparison. We simulate the system with various fleet sizes determined by a *load factor*, which is the ratio between the daily number of trips and the number of cars in the fleet. The load factor ranges from 35 to 50 daily trips per car.

The time-dependent travel times were estimated by combining information from OSRM and the NYC Taxi Trip Duration dataset described below. Since OSRM provides only a free-flow travel time that is constant over a whole day and we wished to use more

**Table 2**
Comparison of rebalancing models.

| Demand | Load factor (trips/ car) | Model | Number of future periods | Average waiting time (seconds) | Deadheading time (seconds/ trip) | Rebalancing time (seconds/ trip) | Empty trip time (seconds/ trip) |
|---|---|---|---|---|---|---|---|
| Moderate demand (100k trips per day) | 35 | JIT | – | 474 | 147 | 0 | 147 |
| | | Reactive | – | 118 | 105 | 71 | 177 |
| | | TB | 6 | 198 | 125 | 14 | 139 |
| | | TB | 12 | 177 | 120 | 16 | 136 |
| | | ZB | 6 | 142 | 109 | 21 | 130 |
| | | ZB | 12 | 125 | 104 | 23 | 128 |
| | 40 | JIT | – | 479 | 147 | 0 | 147 |
| | | Reactive | – | 128 | 110 | 70 | 180 |
| | | TB | 6 | 206 | 126 | 14 | 141 |
| | | TB | 12 | 191 | 124 | 16 | 140 |
| | | ZB | 6 | 156 | 114 | 21 | 135 |
| | | ZB | 12 | 147 | 111 | 23 | 135 |
| | 45 | JIT | – | 524 | 151 | 0 | 151 |
| | | Reactive | – | 143 | 115 | 66 | 182 |
| | | TB | 6 | 233 | 132 | 15 | 147 |
| | | TB | 12 | 215 | 129 | 17 | 146 |
| | | ZB | 6 | 185 | 120 | 21 | 142 |
| | | ZB | 12 | 174 | 118 | 23 | 142 |
| | 50 | JIT | – | 595 | 156 | 0 | 156 |
| | | Reactive | – | 171 | 122 | 60 | 183 |
| | | TB | 6 | 266 | 137 | 15 | 152 |
| | | TB | 12 | 258 | 136 | 17 | 154 |
| | | ZB | 6 | 231 | 129 | 19 | 149 |
| | | ZB | 12 | 218 | 127 | 21 | 149 |
| High demand (200k trips per day) | 35 | JIT | – | 551 | 132 | 0 | 132 |
| | | Reactive | – | 107 | 93 | 65 | 159 |
| | | TB | 6 | 155 | 103 | 19 | 122 |
| | | TB | 12 | 138 | 98 | 20 | 118 |
| | | ZB | 6 | 148 | 99 | 20 | 119 |
| | | ZB | 12 | 122 | 93 | 22 | 115 |
| | 40 | JIT | – | 554 | 132 | 0 | 132 |
| | | Reactive | – | 113 | 96 | 64 | 160 |
| | | TB | 6 | 158 | 104 | 19 | 123 |
| | | TB | 12 | 151 | 102 | 20 | 123 |
| | | ZB | 6 | 155 | 101 | 20 | 121 |
| | | ZB | 12 | 140 | 98 | 21 | 119 |
| | 45 | JIT | – | 580 | 134 | 0 | 134 |
| | | Reactive | – | 125 | 99 | 62. | 162 |
| | | TB | 6 | 180 | 108 | 21 | 129 |
| | | TB | 12 | 175 | 108 | 22 | 130 |
| | | ZB | 6 | 178 | 105 | 20 | 125 |
| | | ZB | 12 | 166 | 103 | 21 | 125 |
| | 50 | JIT | – | 641 | 138 | 0 | 138 |
| | | Reactive | – | 147 | 104 | 58 | 163 |
| | | TB | 6 | 209 | 113 | 21 | 135 |
| | | TB | 12 | 204 | 112 | 22 | 135 |
| | | ZB | 6 | 223 | 111 | 19 | 131 |
| | | ZB | 12 | 209 | 110 | 21 | 131 |

The other parameters for the anticipatory models are $\alpha = 1, \beta = 3900, \rho = 0.99$.
JIT — Just In Time ; TB — Trip Based ; ZB — Zone Based.

realistic data that correctly considered delays due to congestion, we adjusted the travel times obtained from OSRM by inflating them by a factor that was specific to each origin and destination zone and each hour of the day. To calculate this factor, we observed the average ratio between the actual travel times of the journeys included in the dataset and the times obtained from OSRM. We repeated this process for each triplet of origin zone, destination zone, and hour of the day. We estimated the factor by interpolating between earlier and later hours for triplets that were not observed in the dataset. We rounded the factor up to one in the very few cases in which it was less than one, which was probably due to errors in the dataset or in OSRM.

## 6.2. Results

In this section, we present the main results of our experiments and derive some insights. We compare the KPIs of the different rebalancing approaches and analyze the effect of the size of the fleet. The four tested rebalancing approaches are JIT, reactive,

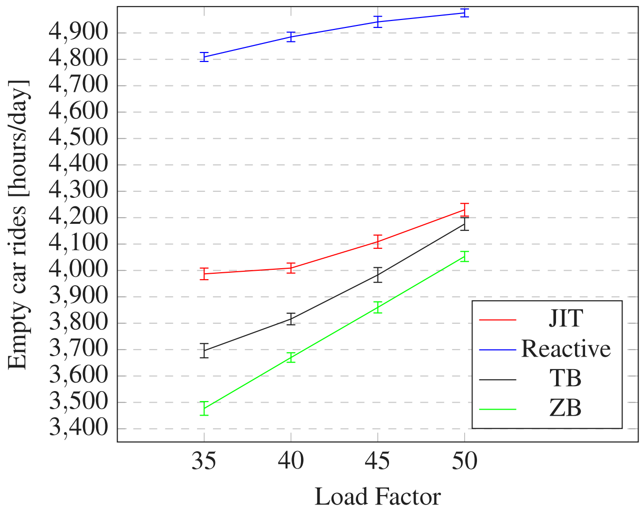


**Fig. 1.** Empty travel time as a function of the load factor under moderate demand (with 95% confidence intervals).

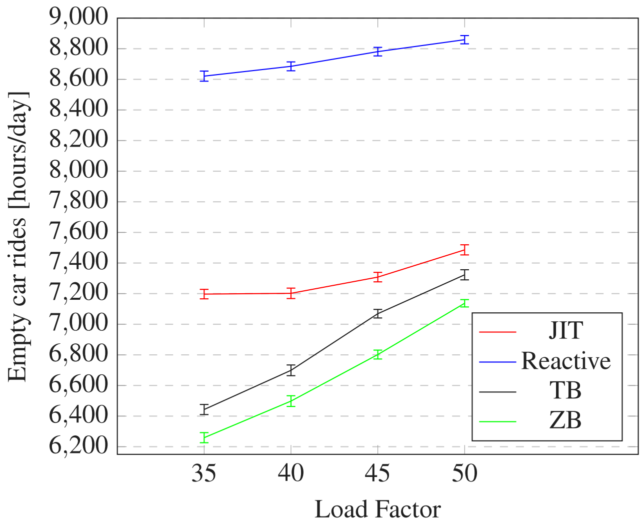


**Fig. 2.** Empty travel time as a function of the load factor under high demand (with 95% confidence intervals).

trip-based, and zone-based. The JIT policy leaves each car at its destination at the end of each trip and dispatches the car that can serve the request at the earliest possible time when the demand arises. The reactive, trip-based, and zone-based approaches are presented in Section 4.

In Table 2, we present the results of the numerical experiments and a comparison between the different approaches. The first two columns of the table describe the dimension of the problem in terms of the demand (number of trips per day) and load factor (ratio between the demand and the fleet size). The third column presents the tested rebalancing policy, and the fourth is the number of future periods used by the anticipatory models. The rest of the columns present the KPIs estimated in our experiment. The fifth column presents the average waiting time of the passengers, which is a measure of the level of service provided by the system. The sixth and seventh columns present the average time per trip that cars are occupied with non-productive tasks: deadheading to passengers and rebalancing, respectively. The sum of these two columns, the total empty travel time per trip, is presented in the rightmost column.

We observe from Table 2 that using a longer planning horizon in the anticipatory models generally improves both the service level and efficiency KPIs. Therefore, our analysis below is based on the results obtained from the 12-period planning horizon.

The two most important KPIs presented in Table 2 are the average waiting time and the empty travel time, representing the service level and the operational efficiency, respectively. The empty travel time as a function of the load factor is plotted in Figs. 1 and 2. The waiting time as a function of the load factor is plotted in Figs. 3 and 4 for the moderate- and high-demand cases, respectively. The data points in Figs. 1–4 present the average daily values obtained from the 30 days simulation, excluding the

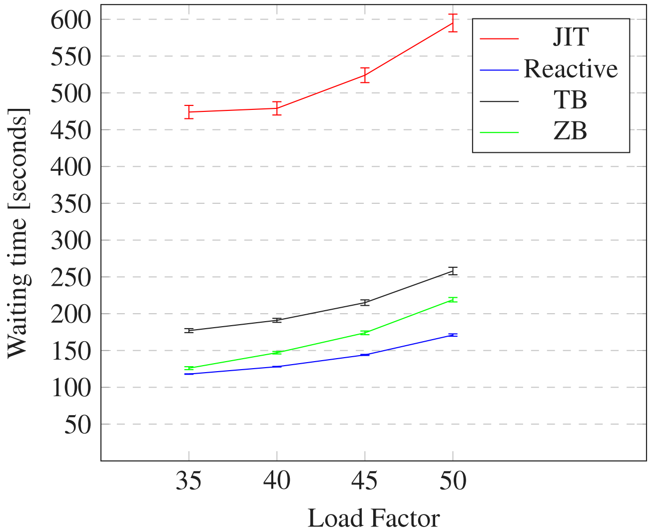


**Fig. 3.** Average waiting time as a function of the load factor under moderate demand (with 95% confidence intervals).

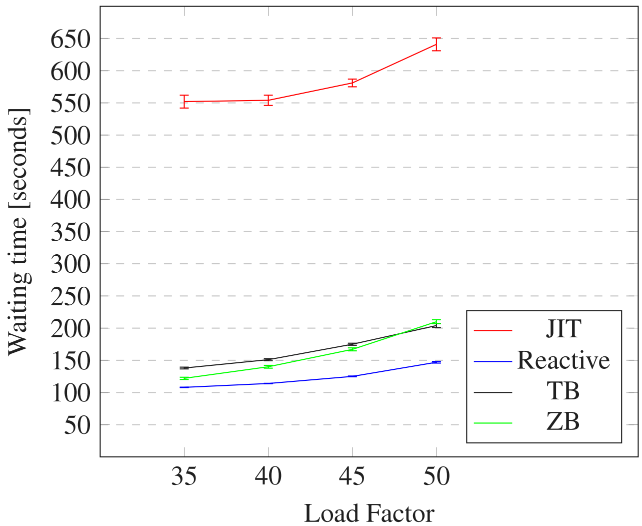


**Fig. 4.** Average waiting time as a function of the load factor under high demand (with 95% confidence intervals).

first warmup day. The error bars, in Figs. 1–4, represent 95% confidence intervals on the mean values. However, in some of the cases, the intervals are too narrow to be observed on the scale of the figures. In Figs. 5 and 6, we plot the efficiency frontier of these two KPIs for a system with moderate demand and two different fleet sizes. While the zone-based and reactive policies are not dominated by the other policies, the JIT policy is far from the Pareto frontier, and the trip-based policy is somewhat dominated by the zone-based policy.

From the results presented in Table 2 and the corresponding figures, it is apparent that the JIT policy leads to a very poor service level, with passenger waiting times of approximately 8-11 min on average, compared with an average waiting time of 2-4 min for all other policies. The total empty travel time under the JIT model is shorter than that in the reactive model but longer than that under the anticipatory models. That is, the anticipatory rebalancing policies dominate JIT in terms of both the service level and efficiency KPIs.

When comparing the zone and trip-based policies, the former appears to dominate the latter, as both the waiting time and empty travel time of the zone-based model are shorter in almost all our experiments. The only case that contradicts this finding is the case with high demand and the highest load factor, where the waiting time under the trip-based model is slightly longer.

Therefore, we are left with a comparison between the zone-based model and the reactive one. They are not comparable because the former is better in terms of efficiency (empty travel time), but the latter is better in terms of service (passenger waiting time). Indeed, the reactive policy under moderate demand saves 7–47 s of waiting time, and under high demand, it saves 15–62 s

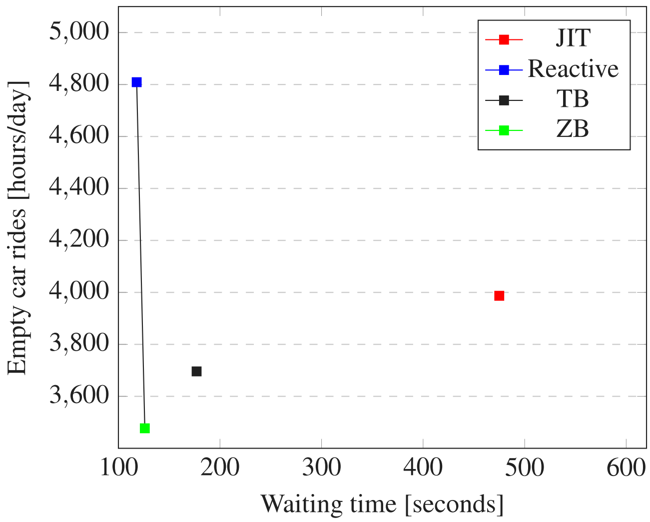


**Fig. 5.** Pareto frontier (black line) - empty car rides and average waiting time for moderate demand and load factor 35 (fleet size 2787 cars).

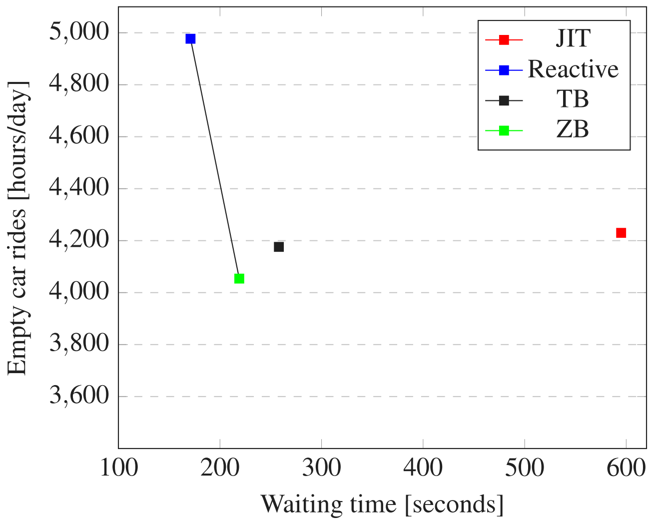


**Fig. 6.** Pareto frontier (black line) - empty car rides and average waiting time for moderate demand load factor 50 (fleet size 1951 cars).

(depending on the load factor). In terms of person-hours per day, this is 194–1305 and 1833–3444 for moderate and high demand, respectively. The empty travel time saved when using the zone-based policy is 923–1333 and 1722–2363 car hours per day for moderate and high demand, respectively. The empty travel time is responsible for 12%–15% of the total travel time of the fleet under the zone-based rebalancing policy and 16%–18% under the reactive policy.

It is difficult to compare the values of car- and person-hours, especially because empty trips not only cost money for the operator but also create negative externalities and, in particular, congestion that causes the waste of additional person-hours. However, as the fleet size decreases relative to the demand (i.e., the load factor increases), the attractiveness of the reactive policy increases. This is because the efficiency gap between the two policies decreases with the load factor, while the service level gap increases.

The average waiting time under the reactive and anticipatory policies was never more than a few minutes, which appears to be an acceptable amount of time for a passenger to wait for a taxi. In many cases, it is less than the amount of time it will take them to leave the origin (e.g., home or office) and get to the pickup point on the street. However, passengers are sensitive to the entire waiting time distribution rather than its mean.

Interestingly, the reactive model yields shorter waiting times than the trip-based and zone-based models at the cost of many more empty car rides. This phenomenon can be explained by the fact that the two models move cars based on the expected demand and do not consider the randomness directly. The reactive model, on the other hand, ensures that enough cars are available in any zone in the service area and is thus able to react to each request more quickly.

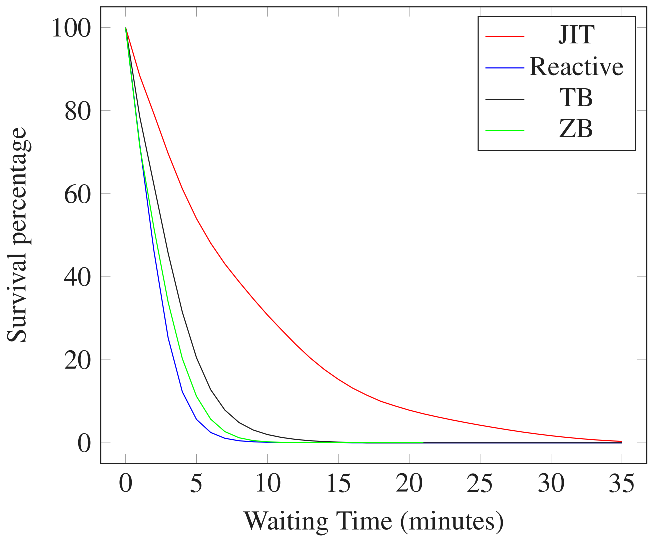


**Fig. 7.** Waiting time distribution with moderate demand and a load factor of 40.

In Figs. 7, we plot the survival function of the waiting time of passengers under the four rebalancing policies with a load factor of 40, that is, the probability that a passenger will wait for a certain amount of time or more. It is apparent that under the reactive and zone-based policies, the fraction of passengers who have to wait more than ten minutes is negligible. Moreover, under the reactive policy, only 5.64% of passengers will have to wait more than five minutes, and under the zone- (resp., trip-) based policy, the fraction of these passengers increases to 11.2% (resp., 20.5%). The waiting time distribution under the JIT policy has a much heavier tail, which implies relatively long waiting times for many passengers. We believe that a waiting time of more than ten minutes for a substantial fraction of trips is not acceptable in on-demand mobility services.

Moreover, in a realistic service system, customers may have limited patience. In particular, in the RoboTaxi system studied in this paper, the passengers are informed about their expected pickup time in advance and may decide immediately after the information is revealed to abandon their trip. In the presence of passenger abandonment, the total load on the system is reduced, and deadheading journeys may be saved. However, we believe that if the rebalancing policy is effective and the fleet size is sufficiently large to provide short waiting times to the passengers, the effect of abandonment will be negligible.

## 7. Conclusion

A RoboTaxi system can provide a service level that is competitive with that obtained by the universal ownership of private cars while using a small fraction of the vehicles and parking spots. In our simulation experiment, each car served approximately 35–50 trips per day on average, compared with a load factor of 2–3 trips per day made by private cars. The average waiting time of passengers in a properly balanced system was 2-3 min, and only a negligible fraction of passengers had to wait more than 10 min. Therefore, RoboTaxi is a very promising technology for future mobility in an urban environment. Such a transition will improve the mobility of passengers in cities, save land used for parking and possibly reduce traffic resulting from searching for parking spots. However, RoboTaxis make empty journeys, which may contribute to congestion.

The asymmetric and irregular distribution of the demand for taxi trips in a city leaves many cars in regions that are unlikely to be needed. Since a RoboTaxi can travel to pick up passengers upon request, it can always be available for passengers if they are ready to wait. However, this may come at the cost of a long waiting time while the cars are deadheading to the passengers. Previous studies presented methods of rebalancing the system and distributing the cars evenly so that requests could be served more quickly. Such approaches allow a dramatic decrease in the waiting time but at the cost of significantly increasing the total empty car traveling time. Empty rides are a significant variable cost component of the operator and cause negative externalities due to their contribution to traffic congestion in the city.

In this study, we propose a rebalancing policy that can provide a service level competitive with that of previous approaches while requiring a much shorter total empty travel time. Our rebalancing policy uses short-term forecasts regarding trip demand in the next few short periods. Cars are moved only to locations where they are expected to be needed. Moreover, in this study, rebalancing policies from the literature are adapted and tested in a free-float setting rather than in the station-based system studied in the past. We also introduce a simple online dispatching method that allows requests to be accepted by cars that are currently occupied with other passengers and are expected to be available soon in close proximity to the origin of the new request.

RoboTaxi systems are still a futuristic concept, and many important issues remain to be considered. A suitable forecasting method for demand should be developed for an effective anticipatory model. We note that the required forecast is for a very short horizon (an hour and a half or three hours in our experiment). Thus, forecasting approaches that adapt in real-time to changing environmental

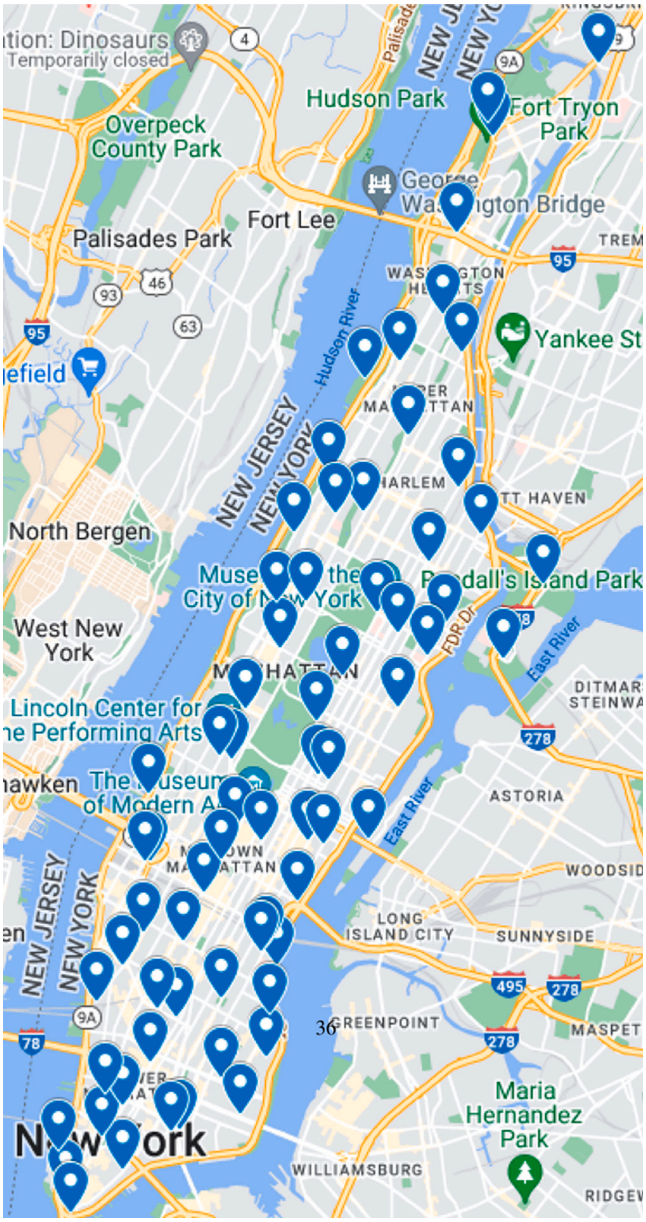

**Fig. 8.** Centers of the 70 zone obtained by our model.

conditions (current traffic, weather, special events in the city, road accidents) can be very beneficial. Moreover, since any forecast is subject to errors, the robustness of the rebalancing solution is another important avenue for future research.

When the RoboTaxi fleet is responsible for a large share of traffic in the city, the operators will have to consider its effect on congestion directly. To reduce congestion, the operator can consider the routes of the cars and create proper incentives for passengers to help balance demand throughout the day. From the policymaker's perspective, integrating the RoboTaxi system with a city's public transit system is crucial. This raises many interesting problems for the operations research community.

## CRediT authorship contribution statement

**Shir Tavor:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Methodology, Visualization, Validation. **Tal Raviv:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Software, Visualization, Validation.

## Data availability

We have shared a link to the code in the article. The data is available from Kaggle.

## Acknowledgments

This work was supported by a grant from the Tel Aviv University Center for AI and Data Science (TAD), Israel.

The first author was partially supported by a scholarship from the Shlomo-Shmeltzer Institute for Smart Transportation at Tel Aviv University, Israel.

## Appendix. Zoning model

To split the service area into zones as required by all the rebalancing approaches tested in this study when applied in the free-float setting, we formulated an ILP model. First, we identified a set of locations, $I$, in the service area as candidates to be used as centers of zones. Next, we created a list, $N$, of pairs of locations in $I$ between which the free-flow travel time is less than a predefined value. We defined a binary decision variable $x_i$ that equals 1 if location $i \in I$ is selected as the center of an actual zone. The ILP model (16)–(18) chose the set of locations to be used as zone centers. Each location in the city was then assigned to the zone of the closest selected center.

$$\max \sum_{i \in I} x_i \tag{16}$$

s.t

$$x_i + x_j \leq 1 \qquad \forall (i,j) \in N \tag{17}$$

$$x_i \in \{0,1\} \qquad \forall (i,j) \in N. \tag{18}$$

The objective function (16) maximizes the number of selected zone centers (and, implicitly, the number of zones). Constraint (17) stipulates that no two zones are defined with centers excessively close to each other.

The solution of ILP (16)–(18) divides the city into as many zones as possible, ensuring that each zone is the largest possible area in which passenger requests can be quickly served by cars located in the same zone.

We solved the model for the island of Manhattan with various distance requirements between the centers and decided to use a three-minute travel time that yielded 70 zones. The zoning solution may significantly affect the system's performance and should be further studied.

We can see the solution of the model in Fig. 8 and in the following link.

## References

Adnan, M., Pereira, F.C., Azevedo, C.M.L., Basak, K., Lovric, M., Raveau, S., Zhu, Y., Ferreira, J., Zegras, C., Ben-Akiva, M., 2016. Simmobility: A multi-scale integrated agent-based simulation platform. In: 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record.

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

Babicheva, T., Burghout, W., Andreasson, I., Faul, N., 2018. The matching problem of empty vehicle redistribution in autonomous taxi systems. Procedia Comput. Sci. 130, 119–125, The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.

Ben-Gal, S., Tzur, M., 2023. Data-driven policies for the online ride- hailing problem with fairness.

Bischoff, J., Maciejewski, M., 2016. Simulation of city-wide replacement of private cars with autonomous taxis in Berlin. Procedia Comput. Sci. 83, 237–244.

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

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

Horni, A., Nagel, K., Axhausen, K., 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity Press.

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

Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., Pavone, M., 2018. Data-driven model predictive control of autonomous mobility-on-demand systems. In: 2018 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 6019–6025.

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

Jorge, D., Correia, G.H., Barnhart, C., 2014. Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. IEEE Trans. Intell. Transp. Syst. 15 (4), 1667–1675.

Kaggle, 2017. New york city taxi trip duration.

Kaspi, M., Raviv, T., Tzur, M., 2014. Parking reservation policies in one-way vehicle sharing systems. Transp. Res. B 62, 35–50.

Luxen, D., Vetter, C., 2011. Real-time routing with OpenStreetMap data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '11, ACM, New York, NY, USA, pp. 513–516, URL http://doi.acm.org/10.1145/2093973.2094062.

Maciejewski, M., Bischoff, J., Nagel, K., 2016. An assignment-based approach to efficient real-time city-scale taxi dispatching. IEEE Intell. Syst. 31 (1), 68–77.

Marczuk, K.A., Soh, H.S., Azevedo, C.M., Lee, D.-H., Frazzoli, E., 2016. Simulation framework for rebalancing of autonomous mobility on demand systems. In: MATEC Web of Conferences, Vol. 81. EDP Sciences, p. 01005.

Nair, R., Miller-Hooks, E., 2011. Fleet management for vehicle sharing operations. Transp. Sci. 45 (4), 524–540.

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transp. Res. C 111, 255–293.
OpenStreetMap contributors, 2017. Planet dump retrieved from . https://www.openstreetmap.org.
Pavone, M., Smith, S.L., Frazzoli, E., Rus, D., 2012. Robotic load balancing for mobility-on-demand systems. Int. J. Robot. Res. 31 (7), 839–854.
Repoux, M., Kaspi, M., Boyaci, B., Geroliminis, N., 2019. Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations. Transp. Res. B 130, 82–104.
Rossi, F., Zhang, R., Hindy, Y., Pavone, M., 2018. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. Auton. Robots 42 (7), 1427–1442.
Ruch, C., Hörl, S., Frazzoli, E., 2018. Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 3639–3644.
Shaheen, S., Chan, N., 2015. Mobility and the Sharing Economy: Impacts Synopsis. Technical Report.
Spieser, K., Samaranayake, S., Gruel, W., Frazzoli, E., 2016. Shared-vehicle mobility-on-demand systems: A fleet operator's guide to rebalancing empty vehicles. In: Transportation Research Board 95th Annual Meeting, number 16-5987. Transportation Research Board.
Turan, B., Pedarsani, R., Alizadeh, M., 2020. Dynamic pricing and fleet management for electric autonomous mobility on demand systems. Transp. Res. C 121, 102829.
Vosooghi, R., Puchinger, J., Jankovic, M., Vouillon, A., 2019. Shared autonomous vehicle simulation and service design. Transp. Res. C 107, 15–33.
Winter, K., Cats, O., Van Arem, B., Martens, K., 2017. Impact of relocation strategies for a fleet of shared automated vehicles on service efficiency, effectiveness and externalities. In: 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, pp. 844–849.
Zhang, R., Pavone, M., 2016. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. Int. J. Robot. Res. 35 (1–3), 186–203.