Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Data-Driven Competitor-Aware Positioning in On-Demand Vehicle Rental Networks

Karsten Schroer, Wolfgang Ketter, Thomas Y. Lee, Alok Gupta, Micha Kahlen

To cite this article:
Karsten Schroer, Wolfgang Ketter, Thomas Y. Lee, Alok Gupta, Micha Kahlen (2022) Data-Driven Competitor-Aware Positioning in On-Demand Vehicle Rental Networks. Transportation Science 56(1):182-200. https://doi.org/10.1287/trsc.2021.1097

http://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Vol. 56, No. 1, January–February 2022, pp. 182–200
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Data-Driven Competitor-Aware Positioning in On-Demand Vehicle Rental Networks

Karsten Schroer,[a] Wolfgang Ketter,[a,b] Thomas Y. Lee,[c] Alok Gupta,[d] Micha Kahlen[b]

[a] Faculty of Management, Economics and Social Science, University of Cologne, 50923 Cologne, Germany; [b] Rotterdam School of Management, Erasmus University Rotterdam, 3062 PA Rotterdam, Netherlands; [c] Haas School of Business, University of California at Berkeley, Berkeley, California 94720; [d] Carlson School of Management, University of Minnesota, Minneapolis, Minnesota 55455

Contact: karsten.schroer@wiso.uni-koeln.de, https://orcid.org/0000-0002-5443-1696 (KS); ketter@wiso.uni-koeln.de, https://orcid.org/0000-0001-9008-142X (WK); thomasyl@haas.berkeley.edu (TYL); alok@umn.edu, https://orcid.org/0000-0002-2097-1643 (AG); micha.kahlen@gmail.com (MK)

Received: October 20, 2020
Revised: April 30, 2021; August 20, 2021
Accepted: August 27, 2021
Published Online in Articles in Advance: December 3, 2021

https://doi.org/10.1287/trsc.2021.1097

Abstract. We study a novel operational problem that considers vehicle positioning in on-demand rental networks, such as car sharing in the wider context of a competitive market in which users select vehicles based on access. Existing approaches consider networks in isolation; our competitor-aware model takes supply situations of competing networks into account. We combine online machine learning to predict market-level demand and supply with dynamic mixed integer nonlinear programming. For evaluation, we use discrete event simulation based on real-world data from Car2Go and DriveNow. Our model outperforms conventional models that consider the fleet in isolation by a factor of two in terms of profit improvements. In the case we study, the highest theoretical profit improvements of 7.5% are achieved with a dynamic model. Operators of on-demand rental networks can use our model under existing market conditions to build a profitable competitive advantage by optimizing access for consumers without the need for fleet expansion. Model effectiveness increases further in realistic scenarios of fleet expansion and demand growth. Our model accommodates rising demand, defends against competitors' fleet expansion, and enhances the profitability of own fleet expansions.

Supplemental Material: The online appendices are available at https://doi.org/10.1287/trsc.2021.1097.

Keywords: machine learning • online optimization • optimal positioning • sharing economy • Car2Go

1. Introduction

A key trend in urban mobility is the consumption of mobility as a service and on-demand (Laporte, Meunier, and Wolfler Calvo 2018) heralding the age of shared, fleet-based, transportation platforms and rental networks (Qi and Shen 2019, Benjaafar and Hu 2020). On-demand rental networks have emerged for cars (e.g., Car2Go, DriveNow); bikes (e.g., Nextbike); and, more recently, scooters (e.g., Lime, Bird). Shared mobility is characterized by the fact that demand is not only influenced by price but also by local supply as defined in terms of availability (e.g., wait time or distance to the nearest vehicle) (Benjaafar and Hu 2020). Therefore, regular fleet balancing via physical repositioning actions is required and has been the subject of extensive research (e.g., Lu, Chen, and Shen 2018; He, Mak, and Rong 2019b). A commonly neglected fact in the literature on shared mobility operations is the competitive environment in which shared mobility platforms and rental networks tend to operate in. Participants compete for a finite number of customers who may switch freely between competing networks (multihoming). Given this situation, own vehicle supply should be considered in the wider context of the competitive market in which users select vehicles based on access and availability. It follows that value may be gained from factoring in the competition's supply decisions into a fleet repositioning decision framework. Existing research, however, treats the fleet repositioning problem as a cost-minimizing task aimed at serving an exogenous observed demand. Although overall market demand may indeed be exogenous, this is unlikely to be the case for own observed demand. For example, relocating away from a region or reducing the fleet size to serve the same demand can be reasonably expected to result in a relatively worse competitive positioning of the focal fleet to the benefit of competing networks.

In the new digitalized mobility system the necessary data to incorporate competitive dynamics into the operational decision process of a fleet operator has become abundant. We observe a more and more liberal and standardized provision of real-time data across the mobility industry. Open standards, such as the General Bikeshare Feed Specification (North American Bikeshare Association 2021) or the Mobility Data Specification (Open Mobility Foundation 2021), are increasingly

adopted by mobility operators who, in many cases, provide open application programming interfaces (APIs) to their digital platforms that allow for automated and real-time data retrieval. Alternatively, real-time data streams of most mobility platforms are now available commercially from third party data analytics providers (e.g., fluctuo). Furthermore, innovations in city-specific regulations increasingly mandate mobility providers to participate in open data initiatives (e.g., Los Angeles). Such developments are only expected to improve real-time mobility data availability in the future.

The question then remains whether there is sufficient value to be gained from this real-time data to warrant investments into the development of a real-time data analytics and decision framework that incorporates competitor supply information into the operational decision-making process? In this research we explore this question by focusing on the vehicle positioning problem, a core challenge in the operation of on-demand vehicle rental networks. We explore how fleet operators can leverage real-time competitor information along with other large-scale urban data sources to make optimal competitor-aware vehicle supply decisions to boost overall market share, utilization and profits. What emerges is a novel decision problem which we term the *Competitor-Aware Shared Vehicle Positioning Problem* (CSVP). The key contribution of our work is to compute the value of real-time competitor information in operational decision making of on-demand rental networks by developing a temporally and spatially flexible online learning and optimization framework to solve the CSVP. Specifically, we formulate a novel dynamic mixed-integer nonlinear programming (MINLP) model that produces optimal competitor-aware positioning decisions for a given spatiotemporal resolution. We address the uncertainty associated with the input parameters of this optimization via a high-accuracy learning model and the online nature of our model which incorporates updated contextual information as it becomes available. A core benefit of this approach is that it reduces the complexity of the downstream optimization allowing for close to real-time solution for large problems. Finally, we evaluate our combined model via extensive simulation and counterfactual testing based on a unique combined data set of Car2Go and DriveNow rental transactions in the city of Berlin.

The remainder of this paper is structured as follows: In the next Section (Section 2), we present a short review of related work. We then describe the predictive and prescriptive modeling approaches of our solution approach (Section 3). In Section 4, we introduce the discrete event simulation (DES), which we use for evaluation, and present results. We end with a discussion (Section 5).

2. Related Work

2.1. Introduction to On-Demand Vehicle Rental Networks

Shared mobility is widely regarded as a core building block of the future mobility system (Savelsbergh and Van Woensel 2016; Laporte, Meunier, and Wolfler Calvo 2018). It entails the consumption of mobility as an on-demand service. Asset ownership is relegated to third parties (Paundra et al. 2017). Benjaafar and Hu (2020) distinguish three canonical sharing economy applications: (1) peer-to-peer resource sharing (e.g., BlaBlaCar), (2) on-demand service platforms (e.g., Uber, Lyft), and (3) on-demand rental networks (e.g., Car2Go, Lime). This work focuses on on-demand rental networks. On-demand rental networks in mobility have historically been operated as station-based systems in that vehicles are collected and dropped off at a certain number of fixed stations (Nair and Miller-Hooks 2011). Recently, free floating vehicle sharing (FFVS) is emerging as an alternative system. In such flexible, one-way systems, customers are free to pick up and drop off a vehicle at any point within a predefined operating area, providing the benefit of direct point-to-point travel. From an operations management perspective, on-demand rental networks exhibit a range of unique characteristics. These are (1) the inability to book in advance, (2) the spatial distribution of resources, and (3) the one-way characteristic of rentals that do not have to be returned to their point of origin (Benjaafar and Hu 2020). These characteristics bring with them novel operations management challenges, ranging from strategic issues, such as fleet sizing and service area definition, to operational challenges, such as spatial and temporal fleet positioning (He, Mak, and Rong 2019b). Here we consider FFVS operations when competition is present, with a particular focus on the repositioning challenge.

2.2. Predictive Analytics for On-Demand Vehicle Rental Networks

The emergence of big urban data has resulted in new research opportunities in predictive analytics for operations management (Cohen 2018). Most existing approaches of FFVS positioning rely on known distributions of (1) demand and (2) trip origin-destination pairs across the spatiotemporal network. Machine learning (ML)–based regression techniques allow for a reduction in the uncertainty of future realizations through accurate point estimates. In situations with as many free parameters as the one we study (market level and focal fleet), using point estimates rather than uncertainty sets can significantly enhance performance and scalability while reducing overconservativeness of some robust optimization approaches (Hao et al. 2020).

In the domain of vehicle sharing, predictive analytics work has focused on demand prediction. For repositioning applications, knowledge on vehicle inflows is equally required. Recent work has shown that the same model architectures can be used to predict both supply and demand (e.g., Xu et al. 2018). FFVS predictive analytics research can be differentiated in terms of (1) the type of model used, (2) the spatial discretization, (3) prediction horizon, and (4) whether contextual data are used. In terms of (1) models used, we identify traditional statistical time series approaches using, for example, Seasonal Autoregressive Integrated Moving Average (SARIMA) models (Müller and Bogenberger 2015); traditional machine learning approaches (Kahlen et al. 2017, Willing et al. 2017); and, more recently, deep learning frameworks (Caggiani et al. 2018, Xu et al. 2018, Zhou et al. 2018, Ai et al. 2019). ML models, and in particular deep learning models, have consistently demonstrated superior performance compared with naive and statistical benchmarks. Spatial discretization (2) is relevant for FFVS applications because of the absence of discrete spatial reference points, such as rental stations. A common discretization method is the use of zip codes (e.g., Lu, Chen, and Shen 2018). However, this technique suffers from various limitations, such as arbitrary and nonuniform shapes and sizes of regions. An alternative approach is the use of grids to discretize the service area into regions of uniform size. In most cases, static quadratic (e.g., Zhou et al. 2018) or hexagonal (e.g., Schroer et al. 2019) grids are used. Research in the area of discrete spatial simulations suggests the use of a hexagonal geodesic discrete global grid system (GDGGS), because of uniform adjacency of hexagons and limited distortion of a geodesic grid. Regarding (3) prediction horizons, myopic single-step predictions are the standard scenario being studied, although Zhou et al. (2018) perform a multiperiod prediction using a deep neural network. The use of contextual data varies by type of model. Although statistical and certain deep learning models use historic time series data exclusively (Müller and Bogenberger 2015, Ai et al. 2019), the additional use of contextual data is commonplace. Typical contextual features employed are temporal metadata (time of day, day of week), meteorological data (e.g., Xu et al. 2018), points of interest (e.g., Willing et al. 2017, Schroer et al. 2019), and events (e.g., Kahlen et al. 2017). Existing predictive approaches are commonly evaluated in static environments with arbitrarily defined period length and region size. Consequently, there is limited clarity regarding their performance in varying temporal and spatial resolutions and different prediction horizons, in which data might become more sparse. As these are important parameters for the performance of the CSVP solution framework, we investigate them in detail in our work.

2.3. Prescriptive Analytics and Empty Positioning Problems for On-Demand Vehicle Rental Networks

The problem of vehicle (asset) allocation and positioning is an emerging area of research in operations management (Benjaafar, Li, and Li 2017; He, Mak, and Rong 2019b; Benjaafar and Hu 2020) and is related to the problem class of empty repositioning, a special case of the assignment problem (Erera, Morales, and Savelsbergh 2009). Vehicle allocation and positioning are especially significant in FFVS systems where, because of the unknown target destination of one-way trips, demand and supply imbalances may occur locally (Benjaafar et al. 2017). FFVS are the subject of this work, and we therefore focus our review on this thread of literature. The reader is referred to Pal and Zhang (2017) for a comprehensive review of the literature on repositioning in the context of station-based vehicle sharing systems.

The operations management of FFVS has received increasing attention in recent years. Generally, two distinct approaches to facilitate vehicle repositioning in FFVS exist: (1) operator-based repositioning (e.g., Angelopoulos et al. 2018) and (2) user-based repositioning (e.g., Ströhle, Flath, and Garttner 2019; Schiffer et al. 2021). Li and Liao (2020) consider self-relocation of autonomous vehicles, which can be understood as a form of operator-based repositioning at lower cost. Either of the two approaches rely on periodic instructions regarding the number of vehicles to be relocated to and from each region in the service network but vary in terms of relocation cost factor and the degree to which positioning actions can be assumed to be deterministic (Laporte et al. 2018). The objective of the underlying positioning framework usually revolves around minimizing a fleet operator's cost of service (He et al. 2019a) or maximizing profitability and/or quality of service (Lu, Chen, and Shen 2018).

Positioning problems are typically modelled using temporal-spatial networks (Erera, Morales, and Savelsbergh 2009). Deterministic (Boyaci, Zografos, and Geroliminis 2015; Pal and Zhang 2017; Caggiani et al. 2018; Schiffer et al. 2021), stochastic (Lu, Chen, and Shen 2018), and robust (Laporte, Meunier, and Wolfler 2018; He, Mak, and Rong 2019b; Hao et al. 2020) mathematical programming approaches as well as greedy simulation-based frameworks (Kahlen et al. 2017) have been proposed to solve the positioning challenge for FFVS. Boyaci, Zografos, and Geroliminis (2015) propose a deterministic multiobjective model for fleet sizing under repositioning. Caggiani et al. (2018) propose an integrated prediction and optimization model that uses point estimates of future demand as input into a deterministic optimization-based decision support system. Weikl and Bogenberger (2015) propose a two-stage approach with relocations between macroscopic zones

and between microscopic zones inside macro zones. Pal and Zhang (2017) develop an optimization model for bike-based FFVS that accounts for the possibility of multivehicle repositioning on trucks. Other studies consider the uncertainty inherent in demand and supply prediction and respond with stochastic frameworks for optimization under uncertainty. Lu, Chen, and Shen (2018) develop two-stage stochastic integer programming models for optimizing strategic parking planning and vehicle allocation for station-based and free-floating carshare systems with the objective of maximizing profit and quality of service (i.e., minimizing unserved rentals). A recent example of robust optimization is provided by He, Hu, and Zhang (2019a) who produce a myopic and multiperiod robust optimization model assuming random realizations of demand based on learned uncertainty sets. They evaluate their model empirically on a temporal-spatial network with up to five regions and three periods. Hao et al. (2020) attempt to address the commonly observed overconservativeness of robust approaches. The authors include additional contextual covariates in their estimation of uncertainty sets and derive differentiated demand distributions conditional on weather factors.

We identify various gaps in the existing FFVS literature that this study sets out to address. First, previous studies typically assume that demand and origin-destination probabilities follow learnable unconditional spatiotemporal distributions from which random draws materialize at any given instance. Hao et al. (2020) show how this assumption can lead to overly conservative repositioning decisions as contextual factors, such as weather, points of interest, or events are ignored, all of which have impact on demand (Kahlen et al. 2017). We also find that past research assumes that shared vehicle fleets operate in isolation and that their positioning decision will have no impact on user choice. This is a bold assumption, especially for real-time location-based rental networks, where location (or access) can be regarded as a major source of competitive advantage. Indeed, Braverman et al. (2019) postulate that in ride hailing systems, the availability of vehicles directly corresponds to the probability of a ride being served. However, in competitive situations, availability should be viewed in relative terms, that is, relative to the availability of competitor vehicles (Balac et al. 2019). Spatiotemporal supply decisions can thus be a source of competitive advantage. Finally, existing approaches tend to focus on small networks with correspondingly large regions. In an FFVS network, availability is usually defined in terms of walking distance, making granular networks more beneficial. We develop a scalable solution approach, which can be applied in practice for network sizes of up to 50 regions, greatly exceeding network sizes used in previous research (e.g., He, Hu, and Zhang 2019a) and allowing for highly targeted positioning operations.

3. Model

We formulate a data-driven, three-stage online learning and optimization model. In Stage 1, the system retrieves, processes, and combines real-time data from a range of different sources and updates the underlying multivariate data set. In Stage 2, these data serve as the basis for machine-learning models that are retrained in each period to estimate key input parameters for the downstream prescriptive model. In Stage 3, the estimated parameters are used in a dynamic MINLP optimization framework that computes optimal positioning decisions for all periods in the optimization horizon. Positioning actions for the upcoming period are isolated and executed after which Stage 1 is repeated. In Sections 3.1–3.3, we discuss these three stages in detail. The CSVP relies on the following assumption. First, we assume nonstrategic user behaviour, that is, we assume that users are indifferent between vehicles of all fleets in the market. This assumption is valid, as vehicle types and price levels between competing networks are comparable in most markets. Customers therefore select vehicles based on access and convenience. Both criteria in turn depend on the relative availability of fleet vehicles per competitor in a given region. We, thus, postulate that the probability of capturing a rental within a given region i and period $[t, t+1]$ is directly proportional to the share of vehicles on the ground within that region and period. We test this assumption empirically for the case of Car2Go and DriveNow and find strong support for it (see Online Appendix B). Second, we assume that competitors are nonstrategic, which is currently the case in most markets in which operators do not engage in competitor-aware operations management. From this assumption, it follows that early adopters of CSVP positioning can fully exploit the benefits of our solution approach. We also assume that demand is exogenous and observable, that is, that there is a fixed demand for shared mobility of a certain mode, which can be monitored and predicted (a common assumption in data-driven demand prediction). Finally, we assume that positioning actions are completed within a single planning period and vehicles become available for rental within that same period. We adopt a structured discretization approach in the construction of the underlying spatiotemporal network across which positioning actions are performed. We distinguish between planning times t where new information is retrieved and new assignment schedules are computed and planning periods $[t, t+1]$ for which these assignment schedules are developed. Overall, we define two temporal parameters. First, we define the length of a

period Δt (in unit time). Second, we set the length of the optimization horizon by defining the set of optimization periods as $\mathcal{T}^O$. The length of the optimization horizon is then given by $\Delta O = \Delta t\ |\mathcal{T}^O|$. The spatial framework we adopt is as follows. We construct a hexagonal geodesic discrete global grid system as defined in Sahr, White, and Kimerling (2004). A GDGGS is more flexible and allows for relatively easy adjustment of spatial resolution because of the hierarchical nature of tile sizes, a property which is helpful if the set of regions $\mathcal{H}$ is to be flexibly defined. We opt for hexagonal tiles because of their advantage of uniform adjacency.

3.1. Stage 1: Real-Time Contextual Data Inputs for Predictive and Prescriptive Analytics

On-demand rental networks operate modern app-based digital interfaces via which users can start, terminate, and pay for rentals. A user has access to accurate information at any given point in time about the amount and location of available vehicles in his or her vicinity. These data are also accessible at a larger and more automated scale via APIs. This allows mobility companies to retrieve and periodically archive real-time geo-tagged competitor supply information from which trip information (start point, end point, duration) can be extrapolated with relative ease (see Online Appendix A for more details). Our approach also works if individual vehicle trajectories cannot be tracked, which is sometimes made difficult by the nonstatic vehicle ID used for privacy reasons in some data standards, such as the General Bikeshare Feed Specification (North American Bikeshare Association 2021). It is in fact sufficient to track total supply, inflows, and outflows per area and time period (irrespective of individual vehicle trajectories) by regularly counting individual vehicles (as identified by their location) per area and tracking changes. Note that this work relies solely on data streams that are generated in the public space and that are available freely to any user accessing the digital platform interface. We do not use any price data, which may be critical from a competition regulation standpoint.

One challenge with this approach is that it allows the observation of censored demand only, that is, demand that is limited by the supply of vehicles. We circumvent this challenge in our demand prediction by removing entries where there is either no supply in a region or where supply has been fully exhausted. In addition, we refrain from employing local supply levels as a predictor of demand and subsequent inflows in our predictive model. Our online learning framework relies on the availability of a rich set of independent variables (features) that capture observed and unobserved patterns in mobility behavior. We create spatial and temporal joins between the data sets and obtain rich feature vectors of 238 features for each spatiotemporal instance. These features can be grouped into five sets, which we describe below. A detailed overview is provided in Online Appendix D.

- Temporal feature vector $\mathcal{Z} = (z^1, \ldots, z^z)$: $z = 35$ temporal features, for example, the hour of the day, the day of the week, and whether the day is a weekend or holiday. They capture temporal patterns in shared mobility usage.
- Meteorological feature vector $\mathcal{W} = (w^1, \ldots, w^w)$: $w = 6$ meteorological features on temperature, precipitation, cloud cover, and wind speed are included. Current weather conditions can have a significant impact on mobility choices (Willing et al. 2017).
- Geographical feature vector $\mathcal{G} = (g^1, \ldots, g^g)$: $g = 105$ features related to points of interest (POIs) are retrieved via Google Maps. We include counts and also capture the relative popularity of a location by counting the number of POIs that have a popularity rating and a price rating.
- Competitor fleet feature vector $\mathcal{C} = (c^1, \ldots, c^c)$: We include $c = 46$ lagged outflow variables (in number of vehicles). We use lags of one, two and three periods prior as well as one to seven days prior to the period in scope, which were selected based on autocorrelation with the target. We also include lagged inflows per region, as these will have an influence on rental starts in the next period (e.g., via round trips). Additionally, we include lagged inflows and outflows aggregated across the entire network, which determine the total pool of vehicles that become available for rental or are candidates for inflows in the next periods. We apply lags of one, two, and three periods. For both inflows and outflows of competitor vehicles, we also include the mean and variance over the past seven days as well as weekly differencing variables that capture any trends in inflows and outflows over the past seven weeks.
- Focal fleet feature vector $\mathcal{F} = (f^1, \ldots, f^f)$: We use exactly the same features as for the competitor fleet (i.e., $f = c$).

3.2. Stage 2: Online Predictive Model

We use machine learning models to predict input parameters for the online optimization. We predict total market demand $\hat{D}^M_{i,t}$ in region i during period $[t, t+1]$.. as well as total and focal fleet inflows per region and period ($\hat{I}^M_{i,t}$ and $\hat{I}^O_{i,t}$). Because the proposed feature set captures many observed and latent variables, we use the same comprehensive feature vectors to learn models for each of the above target variables. We propose an online learning approach, retrieving and incorporating new information as it becomes available. This allows dynamic trends and evolution of the system, such as changes in consumer behavior or changes in competitor relocation schedules to be incorporated by

the model as soon as possible. We use a combined data matrix $\mathcal{S}_{[\mathcal{T}^L]}$ of form

$$\mathcal{S}_{[\mathcal{T}^L]} = (\mathbf{z}^1_{[\mathcal{T}^L]}, \dots, \mathbf{z}^z_{[\mathcal{T}^L]}, \mathbf{w}^1_{[\mathcal{T}^L]}, \dots, \mathbf{w}^w_{[\mathcal{T}^L]}, \mathbf{g}^1_{[\mathcal{T}^L]}, \dots, \mathbf{g}^g_{[\mathcal{T}^L]}, \mathbf{c}^1_{[\mathcal{T}^L]}, \dots, \mathbf{c}^c_{[\mathcal{T}^L]}, \mathbf{f}^1_{[\mathcal{T}^L]}, \dots, \mathbf{f}^f_{[\mathcal{T}^L]})$$

to train our models, where $\mathcal{T}^L$ is the set of training periods and the individual element matrices $\mathbf{e}^n_{[\mathcal{T}^L]} = (\boldsymbol{e}^n_{\mathbf{1}}, \dots \boldsymbol{e}^n_{\mathbf{t^L}})$ consist of vectors of form $\mathbf{e}^n_{\mathbf{t}} = (e^n_{i,t})$, that is, the realizations of independent variable e across all regions $i \in \mathcal{H}$ in period $[t, t+1]$. We normalize the data by subtracting the mean and scaling to unit variance to enhance statistical stability. To increase performance while reducing the risk of overfitting, we employ regularization techniques inherent to the respective algorithm (such as ($\ell 1$) regularization in linear models or dropout for deep neural networks).

We learn new models per period and location based on a fixed-length moving training window of length $|\mathcal{T}^L|$, where $|\mathcal{T}^L|$ is problem-specific. Similar to the process of hyperparameter tuning, it can be tailored to the individual application case via a heuristic search over a range of possible lengths. By comparing predictive performance on a validation set, the best lengths can be identified.

We train models in two types of configurations: (1) a single-period prediction and (2) a multiperiod prediction. In the single-period prediction, we predict market demand $\hat{D}^M_{i,t}$ as well as total and focal fleet inflows $(\hat{I}^M_{i,t}, \hat{I}^O_{i,t})$ as a function of the respective spatiotemporal realization of features as $\phi_{i,t} : \mathcal{S}_{i,t} \to \hat{y}_{i,t}$, which is obtained by solving an optimization of form $\min_{\phi^y_{i,t}} \sum_{t \in \mathcal{T}^L} \ell(\phi^y_{i,t}(\mathcal{S}_{i,t}), y_{i,t})$, where ℓ is the selected loss function of the machine learning algorithm. Hence, $\hat{\mathbf{y}}_\mathbf{t} = (\hat{y}_{i,t})$ for all $i \in \mathcal{H}$ is computed as follows:

$$\hat{\mathbf{y}}_\mathbf{t} = \boldsymbol{\phi}_\mathbf{t}(\mathcal{S}_\mathbf{t}).$$

In the multiperiod case, we predict market demand $(\hat{D}^M_{i,t+n})$, total inflows $(\hat{I}^M_{i,t+n})$, and focal fleet inflows $(\hat{I}^O_{i,t+n})$ for region i in any period $[t+n, t+n+1]$. The prediction is repeated for all $(t+n) \in \mathcal{T}^O$, where $\mathcal{T}^O$ is the set of periods in the optimization horizon. We adapt our feature set because our single-period prediction model uses lagged features from the past period that cannot be observed more than one period ahead. One option is to supplement it with predicted values as we move along the prediction horizon. Alternatively, to avoid prediction error propagation, we may iteratively drop lagged features that have become unobservable for the period in scope. We opt for the latter approach and use the reduced feature vector $\mathcal{S}'$ to derive $\hat{y}_{i,t+n}$. Note that $\mathcal{S}'$ will periodically reduce in size the further ahead in time one wishes to predict. Note also that we do not drop any temporal, meteorological or geographical features (i.e., we do not modify $\mathcal{Z}, \mathcal{W}$ and $\mathcal{G}$) as these are either time-independent or we assume that perfect foresight will be available. Thus, for all realizations further than one period ahead we learn models of form $\hat{\mathbf{y}}_{\mathbf{t+n}} = \boldsymbol{\phi}'_{\mathbf{t+n}}$.

Hyperparameter tuning via brute-force grid searches is performed on a fixed training data set using time series cross-validation. Hyperparameters are parameters of a machine learning model that are not learned during training but chosen by the user directly (e.g., tree depth in a random forest model or the number of nodes per layer in a deep neural network). Because model results can be highly sensitive to these hyperparameters, careful tuning on a training and validation data set is required. Test metrics of the final tuned model are reported on a completely unseen test data set. To obtain accurate test metrics, a moving window validation that most accurately reflects the intended real-world application is adopted. We train the model on a fixed amount of periods $|\mathcal{T}^L|$ and evaluate it on the next period. The length of the training window is selected based on predictive performance. We use the mean absolute error (MAE) as the main error. The MAE is a measure for the average absolute error in terms of number of vehicles. Mathematically, it is expressed as $MAE = \frac{1}{|\mathcal{H}||\mathcal{T}^E|} \sum_{i \in \mathcal{H}} \sum_{t \in \mathcal{T}^E} |\hat{y}_{i,t} - y_{i,t}|$, where $\mathcal{H}$ is the set of regions in the data set and $\mathcal{T}^E$ is the set of evaluation periods. One downside of using the MAE is that model performance comparisons between different resolutions and application fields are not possible because the error is not scaled in accordance with the baseline value. The mean absolute percentage error (MAPE) is a popular relative metric, which is scale independent. But the MAPE suffers from several problems; the most serious of which (in relationship to this work) is its inability to handle cases in which the true value is zero (Chen, Twycross, and Garibaldi 2017). Therefore, the symmetric mean absolute percentage error (SMAPE) is used, which we define here as $\frac{1}{|\mathcal{H}||\mathcal{T}^E|} \sum_{i \in \mathcal{H}} \sum_{t \in \mathcal{T}^E} \frac{|\hat{y}_{i,t} - y_{i,t}|}{|\hat{y}_{i,t}| + |y_{i,t}|}$.

3.3. Stage 3: Online Prescriptive Model

To take advantage of our ability to periodically compute more precise forecasts of parameters based on real-time information, we focus on online optimization approaches. We develop a dynamic model formulated as a mixed integer nonlinear program. In a vehicle FFVS context, knock-on effects of positionings on future rental performance may be substantial; a dynamic model is able to exploit these to the fleet manager's advantage. Table 1 defines the nomenclature adopted in the following formulations. Our optimization objective can be expressed as follows: *For the* $|\mathcal{T}^O|$ *upcoming periods, position vehicles across the*

Table 1. Optimization Model Nomenclature

Symbol	Description	Unit
Sets		
$\mathcal{H}$	Set of regions in network with $\mathcal{H} = \{h_0, h_1, \dots h_i\}$ and index i,j	set
$\mathcal{T}^O$	Set of planning times/periods in opt. horizon $\mathcal{T}^O = \{t_1, t_2, \dots t_{t^O}\}$ with index t	set
Parameters		
$A_{i,t}^{M,base}$	Base market availability without positioning in region i at start of period $[t, t+1]$	vehicles
$A_{i,t}^{O,base}$	Base focal fleet availability without positioning in region i at start of period $[t, t+1]$	vehicles
c_{ij}^{reloc}	Cost of relocating one vehicle from region i to region j	USD
c^{var}	Specific variable cost of vehicle movement (fuel, wear, etc.)	USD/hour
d_{ij}	Shortest path between centroids of region i and j across the road network	kilometers (km)
$\hat{D}_{i,t}^M$	Total predicted market demand for trips in region i during period $[t, t+1]$	vehicles
$\Delta D_{i,t}^O$	Gained trips in region i during period $[t, t+1]$ due to relocations	vehicles
$\hat{\delta}_{i,t}^M$	Average expected length per rental in region i during period $[t, t+1]$	hours
$H_{i,t}^{max}$	Maximum available parking capacity of region i during period $[t, t+1]$	vehicles
$\hat{I}_{i,t}^M$	Total predicted inflows of all vehicles in region i during period $[t, t+1]$	vehicles
$\hat{I}_{i,t}^O$	Total predicted inflows of focal fleet vehicles in region i during period $[t, t+1]$	vehicles
M_t^{max}	Maximum number of relocations during period $[t, t+1]$	relocs
ΔO	Length of optimization horizon	unit time
$\psi_{i,t}^{O,base}$	Relative base vehicle availability of focal fleet in region i during period $[t, t+1]$	ratio
r^{rent}	Specific revenue per rental	USD/hour
$s_{i,t}^O$	Density share of focal fleet vehicles in region i during period $[t, t+1]$	%
Δt	Length of an individual period $[t, t+1]$	hours
ΔW	Length of the planning window	hours
$\Delta x_{i,t}$	Net number of relocations to region i during period $[t, t+1]$	vehicles
State variables		
$A_{i,t}^M$	Total availability of all fleet vehicles in region i at time t	vehicles
$A_{i,t}^O$	Total availability of focal fleet vehicles in region i at time t	vehicles
$D_{i,t}^M$	Total market demand in region i during period $[t, t+1]$	vehicles
$I_{i,t}^M$	Total market inflows fleet vehicles in region i during period $[t, t+1]$	vehicles
$I_{i,t}^O$	Total inflows of focal fleet vehicles in region i during period $[t, t+1]$	vehicles
$\psi_{i,t}^O$	Relative vehicle availability of focal fleet in region i during period $[t, t+1]$	ratio
Decision & target variables		
Π	Focal fleet operator's contribution margin over optimization horizon $\mathcal{T}^O$	USD
$x_{ij,t}$	Number of focal fleet vehicles to be relocated from tile i to j in period $[t, t+1]$	vehicles

network in a competitor-aware fashion such that fleet contribution margin is maximized over the optimization horizon. We choose the contribution margin as our objective, as we assume fleet size to be exogenously determined. In such a scenario, the fleet operator has an incentive to maximize the contribution margin, that is, to maximize the proportion of sales revenue not consumed by variable costs and thus able to cover the fixed cost base. What results is a special profit-maximizing form of the assignment problem where $i \in \mathcal{H}$ origin regions are matched by the same number of destination regions $j \in \mathcal{H}$, with $\mathcal{H}$ being the set of regions. Let us define the decision variable $x_{ij,t}$ as the number of vehicles that flow from region i to j during period $[t, t+1]$. The decision variables have both revenue and cost implications. The objective function of our general mathematical model can thus be formulated as follows:

$$\underset{x_{ij,t}}{\text{maximize}}\, \Pi = \sum_{t \in \mathcal{T}^O} \sum_{i \in \mathcal{H}} (D_{i,t}^M \psi_{i,t}^O \hat{\delta}_{i,t}^M (r^{rent} - c^{rent})) - \sum_{t \in \mathcal{T}^O} \sum_{i \in \mathcal{H}} \sum_{j \in \mathcal{H}} (x_{ij,t} c_{ij,t}^{reloc}). \quad (1)$$

The first term in Equation (1) describes the contribution margin of each captured trip across the network, where the number of captured trips is controlled by the focal fleet's relative availability $\psi_{i,t}^O$:

$$\psi_{i,t}^O = \frac{A_{i,t}^O + I_{i,t}^O + \sum_{j \in \mathcal{H}} (x_{ji,t} - x_{ij,t})}{A_{i,t}^M + I_{i,t}^M + \sum_{j \in \mathcal{H}} (x_{ji,t} - x_{ij,t})}. \quad (2)$$

This is equivalent to the assumption that the probability of capturing a rental is proportional to a fleet's relative availability within a region (converged to expectation) and lies at the core of the *competitor awareness* of our model. Repositioning decisions are made based on total market demand $D^M_{i,t}$ and supply $A^M_{i,t} + I^M_{i,t}$ per region, thus, factoring in where competitor vehicles are positioned and how many additional inflows can be expected over the period in scope. The second term of the objective function sums the cost of positionings across the network, which depends on $c^{reloc}_{ij,t}$ and the decision variables $x_{ij,t}$, where $c^{reloc}_{ij,t}$ denotes the cost of relocating a single vehicle from region i to region j during time-period t. Note that we are dealing with a concave objective function, which is maximized, as can be readily seen by taking the second derivative (which is negative for all values of $x_{ji,t} - x_{ij,t}$ that lie within the constraints of this problem). Consequently, a global optimum exists for the CSVP. The resulting mixed integer nonlinear program can be solved with state-of-the-art modeling frameworks (Kröger et al. 2018). Similar to He, Mak, and Rong (2019b), we allow for the fact that the value of an individual rental may vary across time and space, which is captured by the factor $\hat{\delta}^M_{i,t}$.

The optimization is subject to a number of constraints. We first ensure that the decision variables only take on positive integer values.

$$x_{ij,t} \geq 0, \quad \forall i,j \in \mathcal{H}, \quad \forall t \in \mathcal{T}^O \tag{3}$$

$$x_{ij,t} \in \mathbb{Z}, \quad \forall i,j \in \mathcal{H}, \quad \forall t \in \mathcal{T}^O \tag{4}$$

Also, to conserve the number of vehicles, we ensure that the net number of vehicles being relocated from a tile does not exceed the vehicle availability within that tile at the start of a period. This ensures that the relocation schedule can always be executed and that the operator does not have to wait for any natural inflows to arrive in the region i for subsequent repositioning.

$$\sum_{j \in \mathcal{H}} (x_{ij,t} - x_{ji,t}) \leq A^O_{i,t}, \quad \forall i \in \mathcal{H}, \quad \forall t \in \mathcal{T}^O \tag{5}$$

Equally, as market demand will be limited by the available vehicles within a region, we implement the following constraints, which guarantee that $D^M_{i,t}$ will be either equal to the lower of predicted demand $\hat{D}^M_{i,t}$ or total available vehicles. Note that as per the definition of $\psi^O_{i,t}$, this also guarantees that own captured demand $D^M_{i,t}\psi^O_{i,t}$ cannot exceed own vehicle availability.

$$D^M_{i,t} \leq \hat{D}^M_{i,t}, \quad \forall i \in \mathcal{H} \quad \forall t \in \mathcal{T}^O \tag{6}$$

$$D^M_{i,t} \leq A^M_{i,t} + I^M_{i,t} + \sum_{j \in \mathcal{H}} (x_{ji,t} - x_{ij,t}), \quad \forall i \in \mathcal{H}, \quad \forall t \in \mathcal{T}^O \tag{7}$$

Consequently, if market demand is not fulfilled, inflows will reduce. We implement this by adjusting state variable $I^M_{i,t}$ by the lost demand in the previous period $\Delta D^M_{t-1} = \sum_{i \in \mathcal{H}} (D^M_{i,t-1} - \hat{D}^M_{i,t-1})$ allocated proportionately across the network. In our simulation, we consider the stochasticity in this process by allocating inflows in a probabilistic manner.

$$I^M_{i,t_0} = \hat{I}^M_{i,t_0}, \quad \forall i \in \mathcal{H} \tag{8}$$

$$I^M_{i,t} \leq \hat{I}^M_{i,t} + \Delta D^M_{t-1} \frac{\hat{I}^M_{i,t}}{\sum_{i \in \mathcal{H}} \hat{I}^M_{i,t}}, \quad \forall i \in \mathcal{H}, \ \forall t \in \{t_1, t_2, \ldots t_{t^O}\} \tag{9}$$

We also impose $I^M_{i,t} \geq 0$ as per definition. A further practical constraint is added to the model, reflecting relocation capacity in terms of the maximum number of relocations M^{max}_t:

$$\sum_{i \in \mathcal{H}} \sum_{j \in \mathcal{H}} (x_{ij,t}) \leq M^{max}_t, \quad \forall\, t \in \mathcal{T}^O. \tag{10}$$

Finally, we consider the fact that parking capacity $H^{max}_{i,t}$ may be limited in a specific region, for which we add the following constraint:

$$A^M_{i,t} + I^M_{i,t} + \sum_{j \in \mathcal{H}} (x_{ji,t} - x_{ij,t}) - D^M_{i,t} \leq H^{max}_{i,t}, \ \forall i \in \mathcal{H}, \ \forall t \in \mathcal{T}^O. \tag{11}$$

We also add constraints to initialize and update the $A^O_{i,t}$ and $A^M_{i,t}$ state variables in line with the relocation decisions as well as inflows and outflows.

$$A^O_{i,t_0} = A^{O,base}_i, \quad \forall i \in \mathcal{H} \tag{12}$$

$$A^O_{i,t} = A^O_{i,t-1} + I^O_{i,t-1} + \sum_{j \in \mathcal{H}} (x_{ji,t-1} - x_{ij,t-1}) - D^M_{i,t-1}\psi^O_{i,t-1}, \quad \forall i \in \mathcal{H}, \forall t \in \{t_1, t_2, \ldots t_{t^O}\} \tag{13}$$

$$A^M_{i,t_0} = A^{M,base}_i, \ \forall i \in \mathcal{H} \tag{14}$$

$$A^M_{i,t} = A^M_{i,t-1} + I^M_{i,t-1} + \sum_{j \in \mathcal{H}} (x_{ji,t-1} - x_{ij,t-1}) - D^M_{i,t-1}, \quad \forall i \in \mathcal{H}, \forall t \in \{t_1, t_2, \ldots t_{t^O}\} \tag{15}$$

As positioning actions typically result in a delta in rentals captured by the focal firm, focal fleet inflows will change by the same amount. Because these effects will not yet be reflected in the forecast of focal fleet inflows ($\hat{I}^O_{i,t}$), we use a state variable $I^O_{i,t}$ to track additional inflows that result from additional or lost rentals. We initialize and update $I^O_{i,t}$ as follows:

$$I^O_{i,t_0} = \hat{I}^O_{i,t_0}, \quad \forall\, i \in \mathcal{H}, \tag{16}$$

$$I^O_{i,t} \leq \hat{I}^O_{i,t} + \Delta D^O_{t-1} \frac{\hat{I}^M_{i,t}}{\sum_{i \in \mathcal{H}} \hat{I}^M_{i,t}}, \quad \forall i \in \mathcal{H}, \ \forall t \in \{t_1, t_2, \ldots t_{t^O}\}. \tag{17}$$

We also impose $I^O_{i,t} \leq I^M_{i,t}$ as per the definition. Parameter ΔD^O_t is evaluated against an expected total base

demand across all regions $\hat{D}_t^{O,base}$ that would have materialized if no positionings had taken place. It is defined as

$$\Delta\hat{D}_t^O = \sum_{i\in\mathcal{H}} \hat{D}_{i,t}^M \psi_{i,t}^O - \sum_{i\in\mathcal{H}} \hat{D}_{i,t}^M \psi_{i,t}^{O,base}, \tag{18}$$

where $\psi_{i,t}^{O,base}$ is computed analogously to $\psi_{i,t}^O$ in Equation (2) but substituting $A_{i,t}^{O,base}, A_{i,t}^{M,base}$ and $\hat{I}_{i,t}^O$, where $A_{i,t}^{O,base}$ and $A_{i,t}^{M,base}$ are initialized and updated analogously to $A_{i,t}^M$ and $A_{i,t}^O$ in Equations (12)–(15) but letting $x_{ij,t} = 0$ for all $i, j \in \mathcal{H}$ and $t \in \mathcal{T}^O$.

4. Experiments

We employ a simulation experiment to test our model. We simulate positioning actions over a full week. This allows us to capture daily and weekly patterns and to investigate the impact of any cascading effects of positioning actions in later periods. We draw on the case of Car2Go (focal fleet) and DriveNow (competitor fleet), two major free-floating car-sharing operators to test our model. Given that fleet size and market shares are comparable for both fleets, the choice of focal fleet is random in our case. We select Berlin, a large competitive car-sharing market, as the geographical focus. Although Car2Go and DriveNow have since merged, the case of competitor awareness in Berlin remains relevant because of the recent entry of WeShare and Miles, two new free-floating car rental networks. In this section, we describe the setup of our simulation framework; we introduce the empirical real-time data (Stage 1); we report results of the learning model (Stage 2) and discuss the results of the online optimization (Stage 3).

4.1. Discrete Event Simulation Framework

For a realistic evaluation of our model, we construct a discrete event simulation framework. Exogenous parameters, such as market-level demand $D_{i,t}^M$, are retrieved and used as observed in the real world. Factors that are endogenous to the positioning decisions made by our model, that is, vehicle availability and focal fleet inflows, are updated periodically. We thus capture any future downstream effects that arise from relocation decisions made in a given period. The simplified simulation algorithm is detailed in Online Appendix C.

We simulate over a randomly selected one-week window to ensure that weekly and daily seasonality are reflected and that all cascading effects from previous positioning actions are captured. To validate our DES, we run baseline simulations without any positioning actions. These baseline simulations result in minor discrepancies in the number of rentals captured by the focal fleet of just 1.3% versus observed values. Our simulation model thus reflects reality very closely.

4.2. Descriptive Statistics on Real-time Contextual Data (Stage 1)

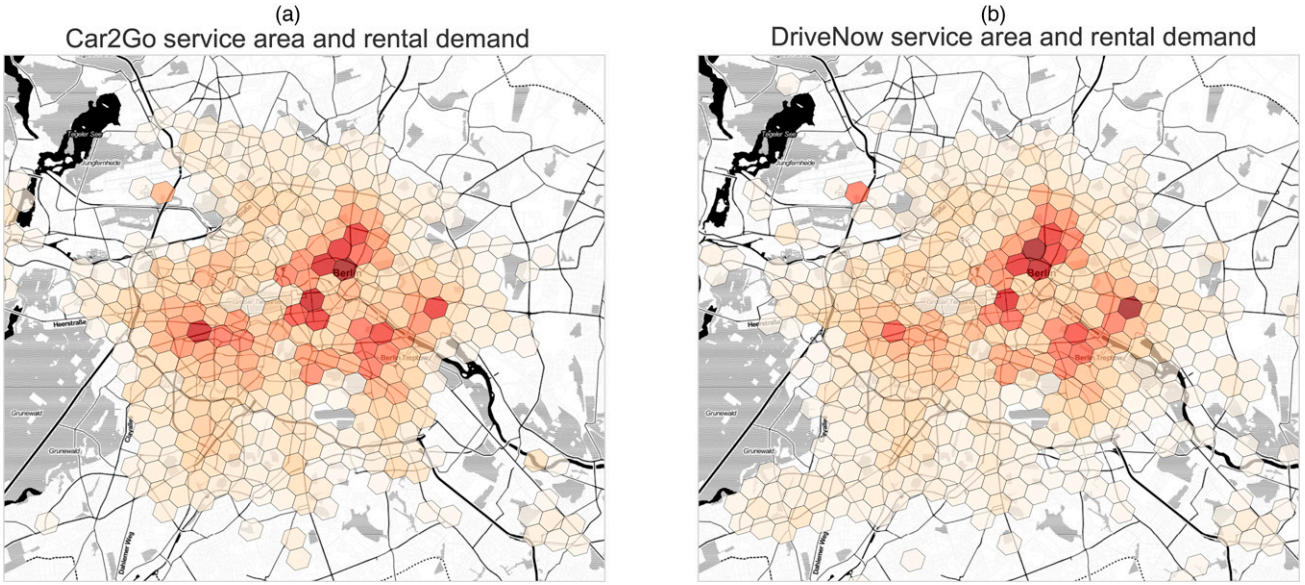
We utilize a roughly five-month window of car-sharing demand data covering the period from December 1st, 2016, to April 26th, 2017. The data contain information on availability of vehicles (identifiable via a vehicle ID) and their position at a granularity of five minutes. To construct the hexagonal GDGGS and to create spatiotemporal merges between the data sets, we draw on open source geographic information systems (GIS) libraries in Python (particularly H3, Geopandas (Jordahl et al. 2020)). We retrieve a network graph representation of Berlin's road network via OpenStreetMap. Distance d_{ij} between regions are determined via shortest path computation across this road network.

We define three spatial sets. First, we define a fine set $\mathcal{H}^f$ with $|\mathcal{H}^f| = 50$ (approximately 1.0 kilometer edge length). Using the properties of the underlying GDGGS, we aggregate up to a medium-coarse set $\mathcal{H}^m$ with $|\mathcal{H}^m| = 15$ (approximately 3.0 kilometer edge length) and a coarse set $\mathcal{H}^c$ with $|\mathcal{H}^c| = 4$ (approximately 8.5 kilometer edge length). For each spatial resolution, we compile two different sets of varying period length with $\Delta t = 1$ hour and $\Delta t = 6$ hours. We choose these to explore trade-offs between temporal resolution and predictive performance. From a practical perspective, different applications may require different resolutions. Table 2 shows the information content per spatiotemporal resolution. We report the average number of inflows/outflows per record and their standard deviation (in parentheses). Information on the sparsity of data are crucial in the interpretation of predictive performance.

Table 2. Information Content (Mean and Standard Deviation (in Parentheses) of Outflows/Inflows per Record) per Spatiotemporal Resolution

	$\mathcal{H}^c$		$\mathcal{H}^m$		$\mathcal{H}^f$	
Parameter	1 h	6 h	1 h	6 h	1 h	6 h
Density of $D_{i,t}^M$	152.7 (155.3)	916.2 (878.4)	40.6 (61.3)	243.8 (351.1)	12.1 (14.5)	72.4 (79.9)
Density of $I_{i,t}^M$	152.7 (155.2)	916.2 (872.9)	40.6 (61.7)	243.8 (351.5)	12.1 (14.9)	72.4 (81.4)
Density of $I_{i,t}^O$	70.5 (74.5)	422.9 (422.2)	18.8 (28.1)	112.7 (159.6)	5.6 (6.9)	33.6 (36.6)

Figure 1. (Color online) Spatial Patterns of Rental Demand in Berlin (Car2Go (a) and DriveNow (b))



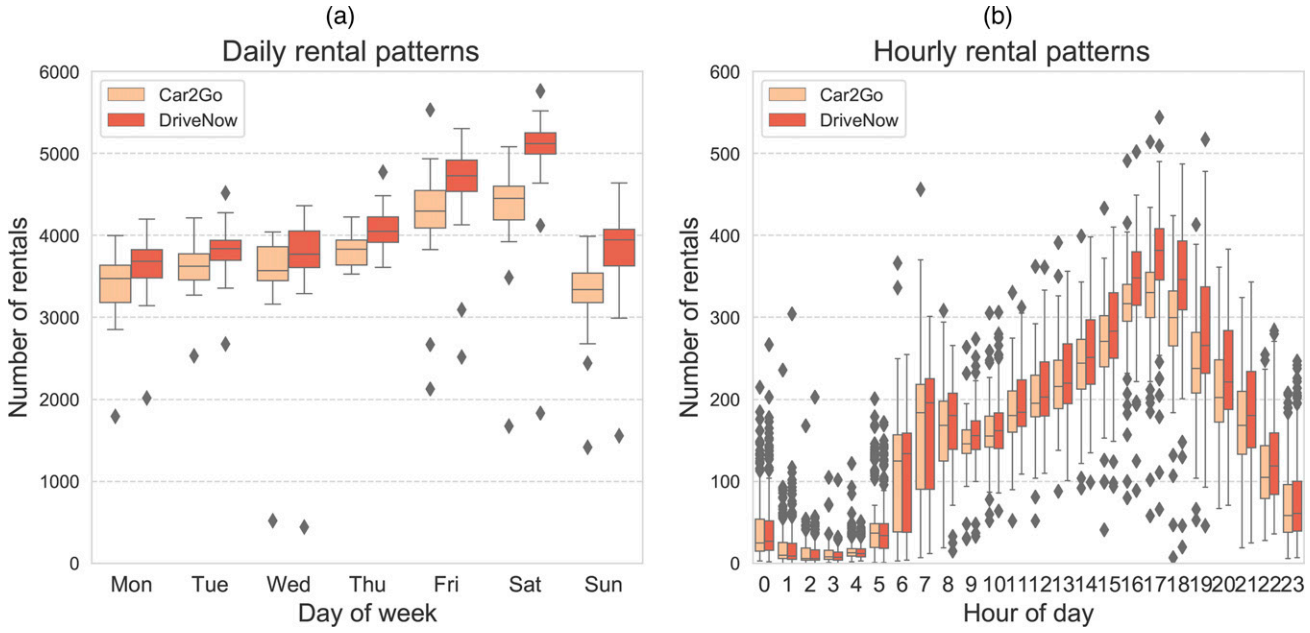
We also explore the spatiotemporal patterns in rental demand (inflows exhibit very similar patterns and are not shown here). Figure 1 illustrates the considerable similarities in spatial rental patterns clustered along the several subcenters of Berlin. Figure 2 provides an overview of the temporal demand patterns. A weekly trend can be observed in the distributions of daily demand with peaks on Fridays and Saturdays followed by lows on Sundays. Intradaily seasonality also follows clear patterns with peaks at 5 p.m. and valleys during the night.

4.3. Performance of Online Predictive Model (Stage 2)

We benchmark six different ML models, which we briefly explain in the following.

- Linear model (ML-LIN): Previous studies have successfully employed linear models for mobility demand

Figure 2. (Color online) Daily (a) and Hourly (b) Patterns of Rental Demand in Berlin



prediction in a car-sharing context (e.g., Willing et al. 2017). To control model complexity and avoid overfitting, we employ $\ell 1$-regularization. The regularization factor λ is determined via a grid search.

• Support vector machine (ML-SVM): Support vector machines have shown to be promising in regression settings. We use a radial-basis-function kernel, which is able to capture nonlinear relationships. We set ϵ (the margin within which no error is attributed in the training loss function) to 0.4. This will result in no error once rounding has been applied. The regularization C (corresponding to $\ell 2$-regularization) is determined via a grid search.

• Gradient tree boosting (ML-TREE): Gradient-boosted tree-based regression is a nonparametric ML method, using tree-based decision rules to make predictions. Again, inherently nonlinear relationships can be represented. Gradient boosting is an ML technique using ensembles of weak models (in our case decision trees) in a step-wise fashion, with each subsequent model minimizing the residuals of the preceding ensemble. We choose MAE as the loss function and determine hyperparameters, including the number of decision trees, the learning rate, and the maximum tree depth via a grid search.

• Neural network (NN) (ML-NN1/2/3): We also test three different (deep) neural network topologies with one, two, and three hidden layers, respectively. We implement the NNs using Keras (Chollet et al. 2015) and use grid searches to tune the typical hyperparameters in a NN, that is, batch size, number of epochs, dropout rate (regularization), and the number of neurons per layer. Batch size and epochs define the learning process of the NN. NNs are (re)trained iteratively on subsets of the training data (batches). One epoch is a full training cycle over the entire data set. Regularization (controlled via dropout rate) is a method for controlling overfitting of an ML model. In the case of NNs, this can be achieved by dropping selected nodes to reduce overall model complexity, thus improving predictive performance. We select ADAM (Kingma and Ba 2014), a common optimization algorithm for deep neural networks, as optimizer and use rectifier linear units (ReLU) as activation functions. Each node (neuron) in an NN transmits a signal to the next node only if it is activated by its preceding inputs. The threshold at which this activation happens is determined by the activation function. In NNs, nonlinear functions with nonconstant derivatives are typically used, of which ReLU is one of the most common and efficient choices. Our loss function is again the MAE, as this is the core metric we want to optimize for (i.e., we want to minimize the absolute number of vehicles the model is off on average).

As additional benchmarks, we use the following three models:

• Persistence (PER) model: The PER naively predicts that the demand of the next period is equal to the realized demand of the previous period.

• Historical average (HA) model: The HA uses a moving average of the three most correlated historical periods to predict the next period. The respective periods are chosen based on autocorrelation with the target over the 80-day training period (see Online Appendix D).

• Seasonal Autoregressive Integrated Moving Average model: A SARIMA model is used as a third benchmark.

All grid searches are performed using time series cross-validation on the 80-day training period (starting from the beginning of the data collection window). The results are documented in Online Appendix D. To obtain realistic test metrics, we use a moving window validation as described in Section 3.2. We keep our training window fixed at 80 days of data (see Online Appendix D for details on choice of training period). All prediction results are postprocessed by rounding to the nearest integer. In Table 3, we report MAE and SMAPE per model as evaluated on the full data set minus the 80-day training window. All ML approaches are superior to both naive prediction methods (PER, HA) as well as statistical time series techniques (SARIMA) across all spatiotemporal resolutions for which we test. This highlights the value of using contextual and competitor data to enhance prediction quality and reduce any uncertainty related to the future realizations of decision parameters. Among the models we test for, deep NN topologies along with the linear model seem to perform best. The advantage of NNs is particularly pronounced where information density is high (coarse spatial and low temporal resolution), a factor that is commonly observed with relatively data-hungry frameworks, such as deep learning. The best-in-class models for each resolution are highlighted in Table 3. We use these in all further simulations. All best-in-class models can be trained in under three minutes for their respective spatiotemporal resolution (on a machine with 128-GB RAM and 24 cores) and are therefore suitable for online learning.

In Figure 3, we review the impact of resolution on ML performance. We obtain high predictive accuracy achieving a SMAPE of considerably below 10% using the coarsest set of regions $\mathcal{H}^c$ and lowest temporal resolution. This is in line with our expectations, as high aggregation levels tend to smooth out the variance in the data, leading to better predictability. Overall, we achieve nearly identical predictive performance for outflows and inflows, which seems intuitive given their strong time-shifted correlation per location. We also observe performance trade-offs for temporal resolution, with six-hour period lengths being

Table 3. Benchmark Model Performance in Terms of MAE and SMAPE (in Parentheses) for Different Spatiotemporal Resolutions (Shown Here for $D^M_{i,t}$)

Prediction algorithm	$\mathcal{H}^c$		$\mathcal{H}^m$		$\mathcal{H}^f$	
	1 h	6 h	1 h	6 h	1 h	6 h
PER	29.0 (17.4%)	553.6 (37.7%)	9.4 (24.7%)	149.1 (35.6%)	4.1 (29.9%)	45.8 (36.9%)
HA	29.3 (18.1%)	394.4 (26.8%)	9.1 (24.5%)	106.3 (25.4%)	3.8 (28.1%)	32.6 (26.1%)
SARIMA	18.3 (13.2%)	125.8 (11.4%)	7.0 (21.7%)	38.8 (14.7%)	3.5 (26.3%)	14.9 (15.8%)
ML-LIN	12.7 (10.2%)	54.7 (6.3%)	5.0 (18.1%)	18.7 (9.0%)	2.7 (22.7%)	9.5 (11.1%)
ML-SVM	13.7 (11.1%)	60.4 (6.8%)	5.8 (20.3%)	21.2 (9.7%)	3.1 (24.5%)	10.1 (11.3%)
ML-TREE	13.6 (11.7%)	62.2 (5.7%)	5.3 (18.9%)	19.7 (8.7%)	2.8 (22.9%)	**9.2** (10.4%)
ML-NN1	13.9 (10.8%)	75.1 (6.1%)	5.1 (19.4%)	22.1 (9.5%)	**2.7** (22.2%)	10.4 (10.6%)
ML-NN2	12.7 (10.4%)	**49.3** (5.5%)	**5.0** (17.9%)	18.4 (9.0%)	3.0 (24.7%)	9.5 (10.3%)
ML-NN3	**11.6** (10.5%)	49.8 (5.7%)	5.4 (19.9%)	**18.0** (9.0%)	3.0 (23.9%)	9.3 (10.6%)

Note. Best-in-class models highlighted in bold font.

considerably more predictable. We find that the performance penalty attributable to temporal resolution is largely constant for the different spatial resolutions (trade-off curve shifted upward in Figure 3). Predictive results obtained for $I^O_{i,t}$, while exhibiting similar trade-off characteristics, are typically slightly worse than those obtained for the market-level targets. We explain this with the lower amounts of information content in our data and higher variance compared with the combined market-level data (see Table 2). Figure 4 shows the relative performance penalty of forecasting further into the future as compared with the MAE of predicting just one period ahead. In general, predicting further ahead into the future comes at a penalty (Ketter et al. 2012). This is to be expected as highly predictive features such as previous realizations become unobservable as we move further into the future. In our case, this penalty can be up to 18% higher MAE versus a single-period forecast for certain spatiotemporal resolutions. We review the impact of these performance penalties on relocation decisions in later simulations.

4.4. Performance of Online Optimization Model (Stage 3)

We implement the online optimization as defined mathematically in Section 3.3 in Julia using the Julia Mathematical Programming framework (Lubin and Dunning 2015) and the Juniper branch and bound MINLP solver (Kröger et al. 2018) with IPOPT (Wächter and Biegler 2006) as the nonlinear programming solver and CPLEX as the mixed-integer programming solver used in the feasibility pump (Kröger et al. 2018). For practical reasons and given the online nature of our approach, we impose a time limit of 900 seconds on the optimization. The optimization runs

Figure 3. (Color online) SMAPE of Best-in-Class Model for Different Spatiotemporal Resolutions and Targets (Total Demand (a), Total Inflows (b), Focal Fleet Inflows (c)) (Error Bars Indicate Range Across Prediction Horizons)

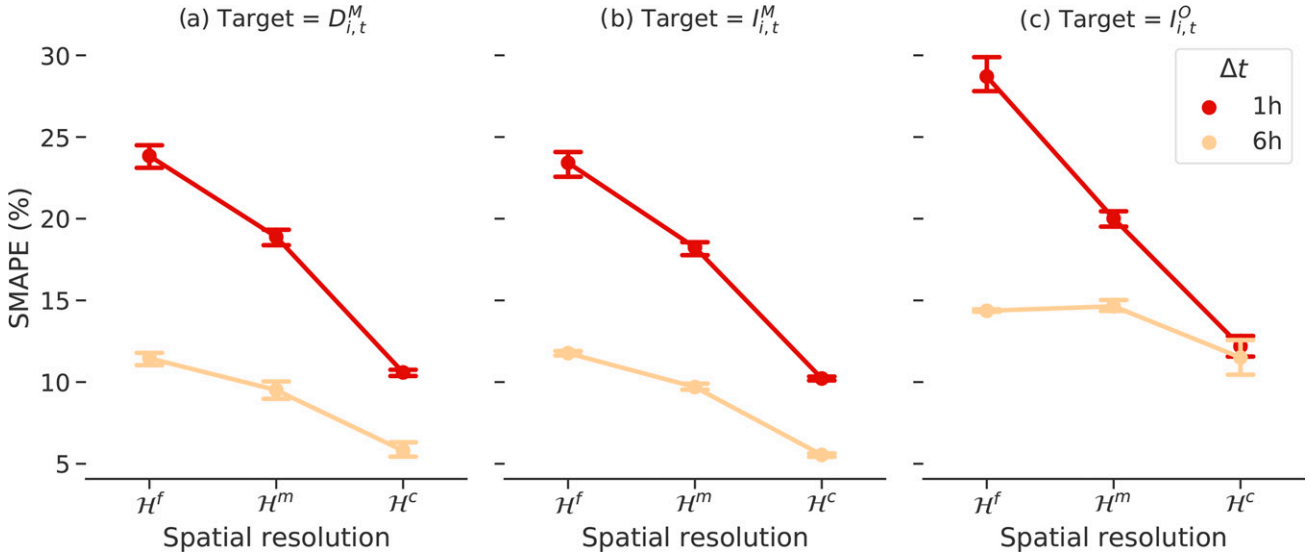
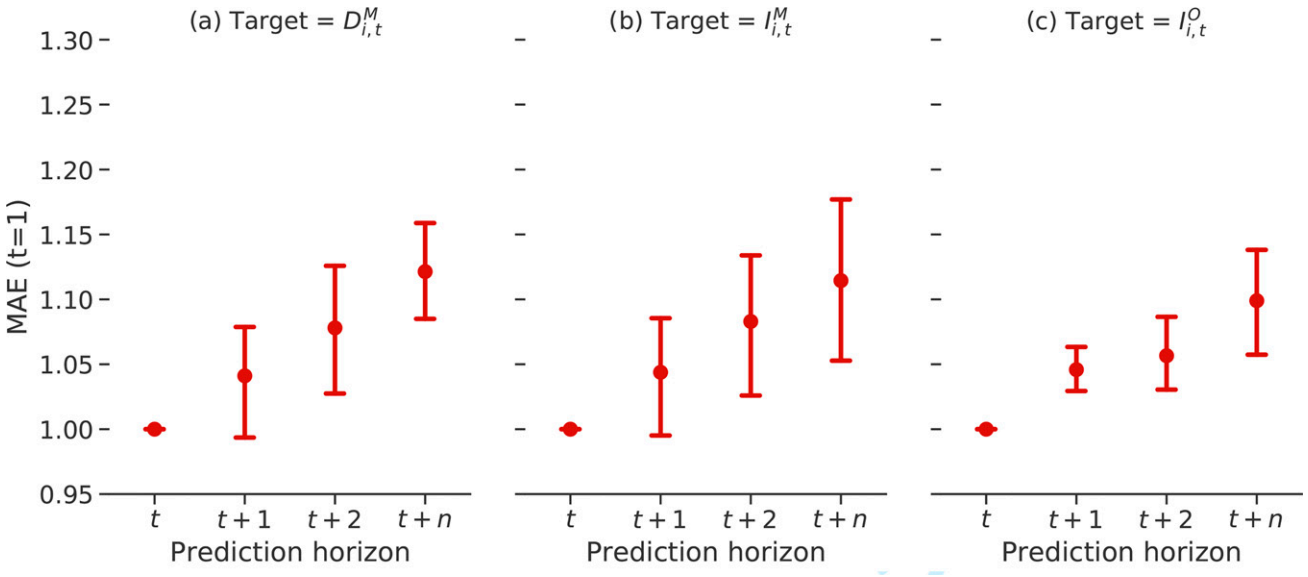


Figure 4. (Color online) Normalized MAE (t = 1) of Best-in-Class Model for Different Prediction Horizons and Targets (Total Demand (a), Total Inflows (b), Focal Fleet Inflows (c)) (Error Bars Indicate Range Across Different Spatiotemporal Resolutions)



all solve within that time limit. For a thorough evaluation, we vary parameters related to spatial resolution ($\mathcal{H} = [\mathcal{H}^c, \mathcal{H}^m, \mathcal{H}^f]$), temporal resolution ($\Delta t = [1\text{h}, 6\text{h}]$), foresight ($|\mathcal{T}^O| = [1,2]$), and information availability $\mathcal{I}$ (perfect parameter information (PI) and forecast parameters (FI)). This parameterization results in a problem size that is significantly larger than those considered in previous research. He , Hu, and Zhang (2019a), for example, test prediction horizons of up to three periods in a network with up to five regions. We argue, however, that in a micromobility setting where location is defined in terms of walking distance, smaller regions are more realistic and accurate.

We parameterize two types of relocation cost c^{reloc}_{ij}: one cost parameter for human operator-based relocation ($c^{reloc,hum}_{ij} = d_{ij}\ (c^{var} + c^{hum})$) and one cost for free relocation (e.g., free rides offered to users or autonomous vehicles) case ($c^{reloc,aut}_{ij} = d_{ij}\ (c^{var} + 0)$). We let the specific variable cost per kilometer c^{var} be equal to USD 0.175 per kilometer (includes fuel and degradation cost) and the operator-based relocation cost c^{hum} be equal to USD 0.7 per kilometer (i.e., roughly 50% of a New York City taxi fee). We define r^{rent} as USD 0.4 per minute, which corresponds to typical fee levels in car sharing. We let $H^{max}_{i,t}$ be equal to the maximum observed market availability plus inflows per region i in the data set. We do not set a limit for M^{max}_t. For the purpose of the optimization, we let $\hat{\delta}^M_{i,t} = \bar{\delta}^M_{i,t}$, where $\bar{\delta}^M_{i,t}$ is the average rental length across the three most correlated historic periods per region. In the simulation, we use the true observed realizations of $\delta^M_{i,t}$. For performance evaluation, we use three key performance indicators (KPIs) (all evaluated against the base case of current policies as observed in the data): (1) average number of positioning actions N per period over the simulation horizon; (2) percentage gain in unit revenue ρ over the simulation horizon; and (3) percentage gain in unit profit π over the simulation horizon, where profit is defined as the contribution margin Π minus depreciation and fixed costs directly attributable to the vehicle. Tables 4 and 5 summarize the results. We observe that the finer the spatiotemporal granularity, the higher the theoretical benefits that can be obtained via CSVP positioning. With finer granularity, the potential for identifying local imbalances and opportunities for local availability optimizations rises. As distances between regions become smaller and relocation costs decrease, more positioning actions are initiated to benefit from these opportunities in a profitable manner. This is especially true for high temporal granularity, in which the larger amount of positioning windows over a day drives CSVP performance. Theoretical unit profit gains at the finest granularity ($\Delta t = 1$ hour, $\mathcal{H}^f$) are up to 7.5% for free user-based positioning (FREE) and 3.0% for operator-based (OPR) positioning. Dynamic models are more effective than myopic models in all cases. We also find that prediction quality has a strong impact on performance. Although results using predicted values are on par with the perfect information case for the $\mathcal{H}^c$ spatial resolution, the performance penalties at $\mathcal{H}^f$ are considerable—particularly for the dynamic models

Table 4. Optimization Results for Different Spatial Resolutions ($\mathcal{H}$), Different Optimization Horizon Lengths ($|\mathcal{T}^O|$) and Different Information Cases ($\mathcal{I}$) at $\Delta t = 6$ Hours

Spatial resolution ($\mathcal{H}$)	Optimization horizon ($|\mathcal{T}^O|$)	Information case ($\mathcal{I}$)	Avg. positioning actions (#/period)		Unit revenue gain (%)		Unit profit gain (%)	
			FREE	OPR	FREE	OPR	FREE	OPR
$\mathcal{H}^f$	MYO	PI	170.9	56.9	3.4	2.4	3.2	2.1
		FI	130.3	61.8	1.7	1.3	1.4	0.7
	DYN	PI	239.5	62.2	4.5	2.7	4.5	2.5
		FI	252.5	70.5	0.4	1.1	0.2	0.5
$\mathcal{H}^m$	MYO	PI	58.1	4.0	3.0	1.1	3.1	1.2
		FI	61.1	4.0	2.6	1.0	2.6	1.1
	DYN	PI	43.0	5.1	3.4	1.5	3.6	1.6
		FI	46.0	6.0	2.7	1.4	2.8	1.5
$\mathcal{H}^c$	MYO	PI	3.9	0.0	0.7	0.0	0.7	0.0
		FI	2.9	0.0	0.7	0.0	0.7	0.0
	DYN	PI	6.2	0.0	0.9	0.0	1.0	0.0
		FI	4.3	0.0	0.9	0.0	1.0	0.0

Note. Avg., average; DYN, dynamic; MYO, myopic; OPR, operator based.

where prediction errors seem to amplify over the optimization horizon. Another effectiveness driver is positioning costs. Although FREE positioning allows for more frequent positioning actions, it is also more susceptible to prediction error compared with operator-based positioning, which compensates prediction errors as a result of more conservative positioning strategies due to higher cost.

At $\Delta t = 6$ hours, the best performing model using forecast parameters is the dynamic model at high spatiotemporal resolution in which a good trade-off is reached and profit gains of 2.8% and 1.5%, respectively, are achieved. We observe similar patterns for a temporal resolution of $\Delta t = 1$ hour. Although theoretical improvements are higher owing to the larger amount of positioning windows over a day, absolute losses due to prediction errors are also higher. We achieve the best performance under FREE positioning for predicted values of 3.1% for a dynamic model and $\mathcal{H}^f$. Under OPR positioning, improvements of 1.8% are feasible using a dynamic model and the best performing model at $\Delta t = 6$ hours is not reached. Given the relatively small improvement for hourly positioning that comes at the cost of considerably more positioning actions and much longer computation times (especially for FREE positioning), we opt for a temporal resolution of $\Delta t = 6$ hours in all following simulations. This corresponds to four positioning windows per day with optimization times below one minute, which is considered feasible in practice.

4.5. Benchmarks and What-If Analyses

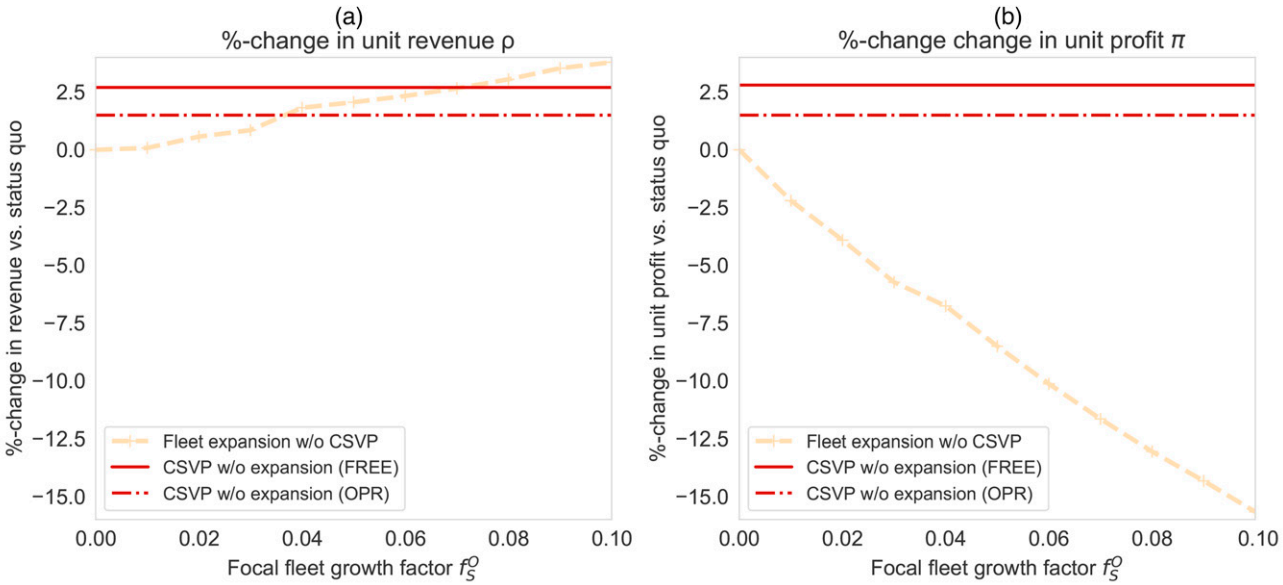
We benchmark our results against relevant alternative approaches and test a range of what-if analyses to explore sensitivities. For all instances, we choose the previously identified best trade-off model setting under predicted parameters as the test case (dynamic, $\mathcal{H} = \mathcal{H}^m$, $\Delta t = 6$ hours). First, we compare our model against the alternative of purchasing and deploying more vehicles. We increase the focal fleet's vehicle supply by

Table 5. Optimization Results for Different Spatial Resolutions ($\mathcal{H}$), Different Optimization Horizon Lengths ($|\mathcal{T}^O|$) and Different Information Cases ($\mathcal{I}$) at $\Delta t = 1$ Hour

Spatial resolution ($\mathcal{H}$)	Optimization horizon ($|\mathcal{T}^O|$)	Information case ($\mathcal{I}$)	Avg. positioning actions (#/period)		Unit revenue gain (%)		Unit profit gain (%)	
			FREE	OPR	FREE	OPR	FREE	OPR
$\mathcal{H}^f$	MYO	PI	118.6	22.7	5.5	2.7	4.4	1.4
		FI	101.3	18.6	4.1	1.7	3.1	0.7
	DYN	PI	120.2	23.4	8.2	4.1	7.5	3.0
		FI	111.8	19.7	4.3	2.0	3.1	0.8
$\mathcal{H}^m$	MYO	PI	18.9	0.4	4.8	1.3	4.8	1.4
		FI	15.7	0.5	2.5	1.2	2.3	1.3
	DYN	PI	20.5	0.9	4.9	2.6	4.9	2.8
		FI	18.0	0.9	2.0	1.7	1.6	1.8
$\mathcal{H}^c$	MYO	PI	1.0	0.0	1.4	0.0	1.5	0.0
		FI	0.9	0.0	1.3	0.0	1.4	0.0
	DYN	PI	1.8	0.0	1.2	0.0	1.2	0.0
		FI	1.7	0.0	1.3	0.0	1.3	0.0

Note. Avg., average; DYN, dynamic; MYO, myopic; OPR, operator based.

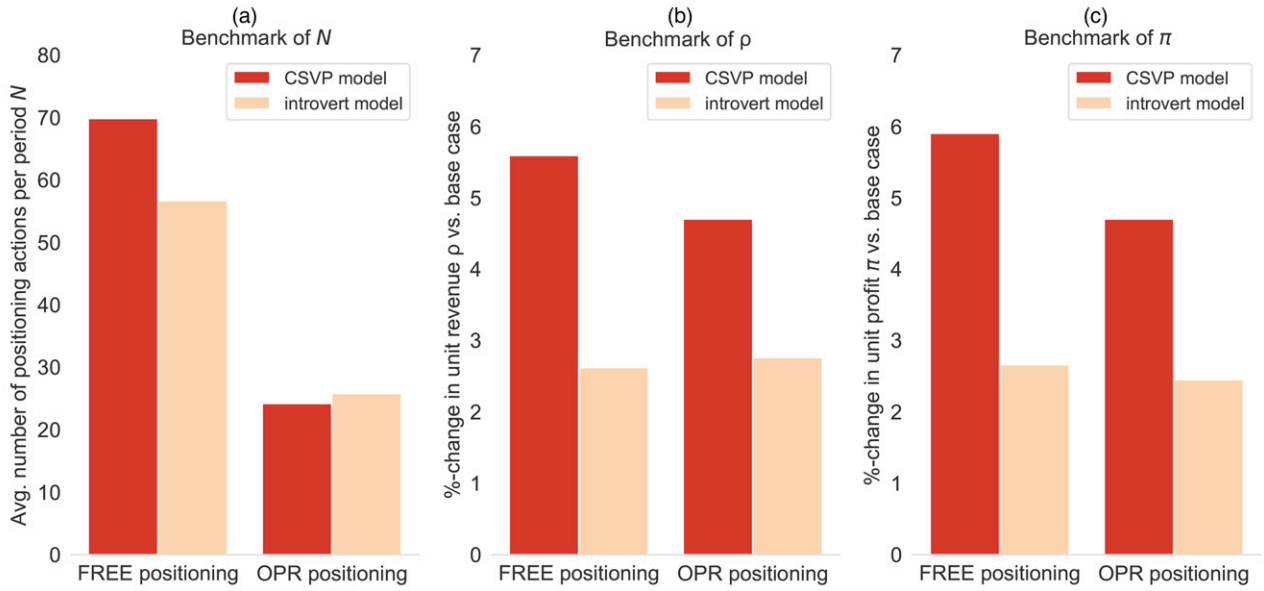
Figure 5. (Color online) Benchmark of Fleet Expansion vs. CSVP Model Without Fleet Expansion (Dynamic, Forecast Parameters, $\mathcal{H} = \mathcal{H}^m$ and $\Delta t = 6$ Hours) in Terms of Revenue Gain (a) and Unit Profit Gain (b)



$f_S^O S^M$ proportionately across the network, where f_S^O is the focal fleet vehicle supply growth factor. Our simulations show (see Figure 5) that in the absence of smart competitor-aware positioning, the focal fleet would have to increase fleet size by an f_S^O of 7% (FREE positioning) and 4% (OPR positioning) to achieve a comparable increase of 2.7% or 1.5% in absolute revenue, respectively. This corresponds to an absolute growth of the fleet of 16.2% and 9.2%. Increasing fleet size, however, has adverse effects on unit economics as shown in Figure 5(b)). Under fleet expansion, unit profit would decline by −13.5% or −7.8% depending on the positioning case. At the same revenue growth, the CSVP model would incur unit profit gain of +2.8% or +1.5%, thus highlighting the value of our approach.

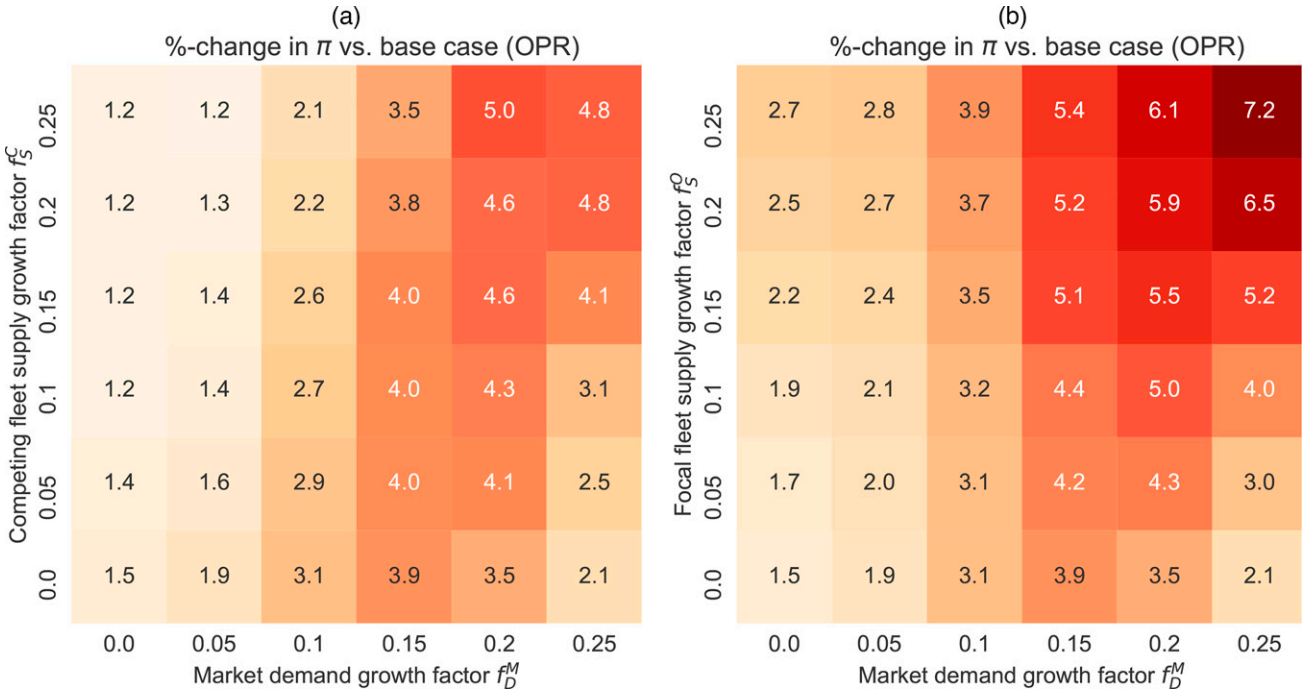
Second, we benchmark against a conventional positioning framework, which considers the focal fleet in isolation. We term this an "introvert" model. In the introvert case, we let $D_{i,t}^M = D_{i,t}^O$, $I_{i,t}^M = I_{i,t}^O$ and $A_{i,t}^M = A_{i,t}^O$ in Equations (1)–(18), whereas the CSVP model formulation remains unchanged. For illustrative purposes and to abstract from the prediction performance, we choose a perfect information case. We also artificially inflate demand by 20% to provide sufficient opportunity for positioning actions in an introvert case. Figure 6 shows the results. In an introvert case, positioning benefits stem exclusively from capturing some of the additional demand that otherwise would have been lost or from repositioning to more valuable areas where expected rental length is higher. However, given that competitor vehicles will be present, actual captured demand will diverge from what is expected a priori. By taking a market perspective, these relative availabilities are considered in the positioning decisions. It also enables the identification of additional demand pockets that would remain unobserved otherwise. As a result, a fleet operator operating on a CSVP positioning model can achieve more than double (FREE positioning) or just under double (OPR positioning) the unit revenue and profit increases versus a non-competitor-aware model for the scenario we test. Third, we explore our model's performance in various realistic scenarios by varying (1) the level of market demand $\mathbf{D^M}_{[\mathbf{T}]}$ and (2) the vehicle supply S^M. We vary demand by increasing $\mathbf{D^M}_{[\mathbf{T}]}$ by factor $(1 + f_D^M)$ proportionally across the network, where f_D^M is the demand growth factor. $\mathbf{I^M}_{[\mathbf{T}]}$ and $\mathbf{I^O}_{[\mathbf{T}]}$ are adjusted depending on how much induced demand is actually served. We increase competitor vehicle supply by adding $f_S^C S^M$ to the initialized availability, where f_S^C is the competitor vehicle supply growth factor. As before, we increase focal fleet vehicle supply by $f_S^O S^M$. Figure 7(a) provides insights into the effects of growing the competitor fleet vehicle supply under different demand growth scenarios. Competing fleet operators may choose to expand their fleets to capture market share (shown for the conservative, high-cost case of operator-based positioning). Alternatively, new entrants may raise the supply of competitor vehicles. Market demand may or may not rise in such future scenarios. Comparing the unit profit π against the base case of not engaging in CSVP positioning, we find improvements at all levels of f_S^C and f_D^M we test for. The isolated effect of a demand increase follows a bell-shaped curve (see bottom rows of Figure 7) with highest effects versus base case at moderate

Figure 6. (Color online) Benchmark Positioning Actions (a), Unit Revenue (b), and Unit Profit (c) Obtained by Noncompetitor-Aware (Introvert) vs. CSVP Model (Dynamic, Perfect Information, $\mathcal{H} = \mathcal{H}^m$ and $\Delta t = 6$ Hours) at 20% Increase of $\mathbf{D}^{\mathbf{M}}{}_{[\mathbf{T}]}$ and $\mathbf{D}^{\mathbf{O}}{}_{[\mathbf{T}]}$



increases. For large demand increases, the market becomes relatively undersupplied and there is less scope for further improvement. Looking at growth in competitor supply alone (first column), we see stable- to moderately decreasing effectiveness of CSVP positioning. This small reduction in effectiveness is noteworthy as

Figure 7. (Color online) Heat Map of Focal Fleets' Change in Unit Profit π Under Operator-Based (OPR) CSVP Positioning vs. Base Case for (a) Different f_D^M and f_S^C and (b) Different f_D^M and f_S^O



a factor $f_S^C = 0.25$ corresponds to a significant loss in relative availability of approximately 10 percentage points. Our framework remains effective at largely constant rates even as relative availability is considerably reduced. The highest uplifts versus the base case are achieved in scenarios with moderate demand increase and large competitor fleet expansions. The high loss in relative availability, coupled with additional rentals to be captured at a constant own fleet size, provides ample improvement opportunities via CSVP positioning and can result in profit increases versus a base case of 5.0%. The CSVP framework thus provides an effective defensive mechanism against competitor fleet expansion by reducing the impact of the loss in relative availability and by amplifying the decline in unit profitability experienced by the competitor fleet. The CSVP is therefore well suited for accommodating an anticipated growth in market demand even if competitors decide to increase fleet size proportionately. Figure 7(b) depicts the unit profit gains under operator-based CSVP positioning in the opposite case of focal fleet expansion and different demand growth scenarios. We explore whether the framework can enhance the effectiveness of the focal fleet operator's physical fleet expansions in the market we study by dampening adverse unit economic effects associated with fleet expansion. We find similar effectiveness patterns for changes in f_S^O. However, focal fleet supply effects in isolation (column one in Figure 7(b)) result in relatively higher effectiveness gains. This can be attributed to the larger vehicle pool the fleet operator can use for CSVP positioning. Effectiveness rises to up to 7.2% unit profit increase in an OPR positioning scenario. Improvements for FREE positioning are even higher and are shown in Online Appendix E. From these what-if analyses, we learn that (1) the level of relative oversupply in the market plays an important role. As demand growth picks up, CSVP effectiveness rises up to a point when the market becomes undersupplied and the scope for additional gain diminishes. We also find that (2) the relative availability share seems not to have a significant impact on effectiveness. As competitor supply rises, the CSVP remains effective at similar levels. Finally, we find evidence that (3) the vehicle pool size plays a role. As the fleet operator increases his or her fleet size, he or she not only improves relative availability but also the absolute size of the vehicle pool that can be used for positioning, which drives effectiveness. The CSVP model can thus be expected to perform well in scenarios of growing market demand, growing competitor supply, and own fleet expansion. All simulations above assume uninterrupted data flow from competitor APIs. In practice, these may experience occasional outages and malfunctions. In Online Appendix E, we run extensive simulations to test the sensitivity of our results to such outages. We find that our results are highly robust even to reasonably high outage probabilities of 50%. These findings further highlight the real-world readiness of our approach.

5. Discussion and Future Work

This research explores how operators of on-demand rental networks can leverage real-time and geo-tagged competitor information in the operations management of their network. Market-level and high-resolution mobility data are widely and publicly available in many geographies. The trend of open-sourcing mobility operator data is continuing (e.g., Mobility Data Specification project or city-level open data initiatives). In an environment evolving toward increased sharing, the question emerges whether operators can profit from incorporating detailed competitor supply and demand information into their operational decision processes. To address this question, we focus on the repositioning challenge, a core operational concern in on-demand rental networks. We develop a novel, data-driven, and scalable three-stage framework that enables operators of on-demand rental networks to improve market share and profitability through competitor-aware positioning actions—independent of market structure. We build our model on the assumption that local relative product availability determines the market share that is captured by a fleet in a competitive environment. We test that assumption and develop a machine learning framework to predict market-level demand and supply parameters. Those predictions are input to a multiperiod MINLP online optimization framework that periodically determines optimal positioning schedules. Our work offers a new perspective on the shared vehicle positioning problem by incorporating competitor awareness into the decision process. We demonstrate the effectiveness of our model via benchmarks and counterfactuals tested on real-world data from the case of Car2Go and DriveNow in Berlin. CSVP positioning outperforms the base case observed in our data both in terms of revenue gain and profitability at all spatial and temporal discretizations under consideration. The upper bound is 7.5% unit profit growth. However, given that predictive accuracy diminishes at larger network sizes, the best real-world results are obtained with a dynamic model at medium spatial and low temporal resolution. The optimal unit profit increase ranges from 2.8% to 1.5% depending on positioning cost. We benchmark this setup against the alternative of fleet expansion and against an introverted player that neglects competitor information. Both benchmarks are significantly outperformed. We also find that model effectiveness typically rises under realistic combinations of defensive (competing fleet growth, new entrants), offensive (focal fleet expansion), and demand growth scenarios. We also show that our model readily handles data outages (see Online Appendix E).

Our work is subject to several limitations, which provide scope for interesting follow-up work. First, our approach relies on the availability of real-time system-level shared mobility data. Throughout this paper, we have argued that such data are increasingly and widely accessible. In many cases, real-time mobility data streams are provided by the mobility operators themselves via open APIs—either voluntarily or in response to regulatory pressures from governmental authorities. We have also referenced alternative, real-time data sources, such as market research companies. Yet, the fact remains that our approach is data hungry. This opens up research opportunities to extend our model to situations where the market environment is only partially observable in real time, that is, where competitor data streams remain untapped. Future work may therefore consider a scenario where competitor demand and supply information is opaque or unavailable in real time. Only static, historical data are available. Here, we briefly elaborate on two possible approaches for dealing with such a challenge. A promising methodological extension to handle partial observability is the use of reinforcement learning (RL), which has received increasing attention in the OR/OM community as a scalable approach in stochastic and sequential environments (Gosavi 2009, Powell 2011). One can cast the CSVP as a single-agent, partially observable Markov decision process. An RL algorithm can then find repositioning policies that maximize a certain reward function (i.e., fleet operator profit) and implicitly incorporate (unobservable) competitor supply and demand information. By learning over many episodes of historical system states, optimal repositioning policies could likely be found. Although such an approach is computationally expensive, once trained, the RL model executes in real time and would therefore offer real-world utility. The latent modeling technique of economic regimes (ER) proposed by Ketter et al. (2009, 2012) also offers a promising avenue. ER models are based on hierarchical hidden Markov models and predict distributions of key market conditions (such as demand and supply volumes) based on limited observable market information (such as a single competitor's supply and demand conditions). Probabilistic modeling is then used to detect and predict changes in economic regimes and derive operational decisions tailored to the specific market/competitive condition.

Second, our model assumes that only the focal fleet conducts competitor-aware positioning actions. This work enables us to quantify the value of real-time market-level data and show that the value is sufficient to warrant investment in the development of a real-time data analytics and decision framework that incorporates competitor supply information. Indeed, as an early adopter, the fleet operator can exploit the full value of the data. Previous research also shows that the overall market efficiency frontier can increase even if only one actor uses the derived competitive strategy (Ketter et al. 2012). The question remains, however, how the situation changes as other competitors with symmetric information may themselves begin to engage in CSVP positioning. One approach to exploring these dynamics is a game-theoretic framework that formalizes interactions between several rational and competitive agents. Because of mathematical tractability, a game-theoretic approach would be limited to evaluating a small number of players and limited set of strategies. A more flexible and scalable alternative is competitive multiagent simulation (MAS). MAS can find and evaluate different agent strategies and stable combinations thereof (equilibrium) in an evolutionary manner. The MAS approach could also leverage the RL or ER methods mentioned above to derive competitive agent strategies. Several outcomes are conceivable. In a zero sum game, as more and more competitors apply CSVP, the value will increasingly be shared between platforms to the disadvantage of those fleets that either do not use CSVP or those that utilize comparatively worse prediction models. Where competition is not a zero sum game, perhaps because the system is not operating at the efficient frontier, if all operators use all available information to make more efficient allocations, overall efficiency of the system could increase and a win-win situation might be created. In either situation, although first movers can fully and exclusively benefit from CSVP today, there is also a clear long-term competitive disadvantage from not using competitor-aware vehicle positioning approaches, which highlights the value of the presented approach.

Finally, there is an opportunity to test the CSVP framework on other on-demand rental networks, such as bikes or e-scooters. On-demand bike-rental networks typically experience more rental activity and have cheaper positioning cost factors (due to pooling), which could allow for the use of lower network resolutions and more positioning actions, thus increasing effectiveness. On-demand service networks (Uber, Lyft) could also benefit from the proposed framework.

References

Ai Y, Li Z, Gan M, Zhang Y, Yu D, Chen W, Ju Y (2019) A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. *Neural Comput. Appl.* 31(5):1665–1677.

Angelopoulos A, Gavalas D, Konstantopoulos C, Kypriadis D, Pantziou G (2018) Incentivized vehicle relocation in vehicle sharing systems. *Transportation Res. Part C: Emerging Tech.* 97 (November 2017):175–193.

Balac M, Becker H, Ciari F, Axhausen KW (2019) Modeling competing free-floating carsharing operators: A case study for Zurich, Switzerland. *Transportation Res. Part C: Emerging Tech.* 98:101–117.

Benjaafar S, Hu M (2020) Operations management in the age of the sharing economy: What is old and what is new? *Manufacturing Service Oper. Management* 22(1):93–101.

Benjaafar S, Jiang D, Li X, Li X (2017) Dynamic inventory repositioning in on-demand rental networks. Preprint, submitted March 30; revised July 29, 2021, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2942921.

Boyaci B, Zografos KG, Geroliminis N (2015) An optimization framework for the development of efficient one-way car-sharing systems. *Eur. J. Oper. Res.* 240(3):718–733.

Braverman A, Dai JG, Liu X, Ying L (2019) Empty-car routing in ridesharing systems. *Oper. Res.* 67(5):1437–1452.

Caggiani L, Camporeale R, Ottomanelli M, Szeto WY (2018) A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Res. Part C: Emerging Tech.* 87:159–182.

Chen C, Twycross J, Garibaldi JM (2017) A new accuracy measure based on bounded relative error for time series forecasting. *PLoS ONE* 12(3):1–23.

Chollet F (2015) Keras. Accessed September 17, 2021, https://github.com/fchollet/keras.

Cohen MC (2018) Big data and service operations. *Production Oper. Management* 27(9):1709–1723.

Erera AL, Morales JC, Savelsbergh M (2009) Robust optimization for empty repositioning problems. *Oper. Res.* 57(2):468–483.

Gosavi A (2009) Reinforcement learning: A tutorial survey and recent advances. *INFORMS J. Comput.* 21(2):178–192.

Hao Z, He L, Hu Z, Jiang J (2020) Robust vehicle pre-allocation with uncertain covariates. *Production Oper. Management* 29(4): 955–972.

He L, Hu Z, Zhang M (2019a) Robust repositioning for vehicle sharing. *Manufacturing Service Oper. Management* 22(2):241–256.

He L, Mak HY, Rong Y (2019b) Operations management of vehicle sharing systems. Hu M, ed. *Sharing Economy* (Springer Nature, Berlin), 461–484.

Jordahl K, den Bossche JV, Fleischmann M, Wasserman J, McBride J, Gerard J, Tratner J, et al. (2020) geopandas/geopandas: v0.8.1. Accessed September 17, 2021, http://dx.doi.org/10.5281/zenodo.3946761.

Kahlen M, Lee TY, Ketter W, Gupta A (2017) Optimal prepositioning and fleet sizing to maximize profits for one-way transportation companies. *Thirty-Eighth Internat. Conf. Inform. Systems: Transforming Soc. Digital Innovation, ICIS 2017* (Association for Information Systems, Atlanta), 70–86.

Ketter W, Collins J, Gini M, Gupta A, Schrater P (2009) Detecting and forecasting economic regimes in multi-agent automated exchanges. *Decision Support Systems* 47(4):307–318.

Ketter W, Collins J, Gini M, Gupta A, Schrater P (2012) Real-time tactical and strategic sales management for intelligent agents guided by economic regimes. *Inform. Systems Res.* 23(4): 1263–1283.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. Preprint, submitted December 22; revised January 30, 2017, https://arxiv.org/abs/1412.6980.

Kröger O, Coffrin C, Hijazi H, Nagarajan H (2018) *Juniper: An open-source nonlinear branch-and-bound solver in Julia. van Hoeve WJ, ed. Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Lecture Notes in Computer Science, vol. 10848 (Springer, Cham), 377–386.

Laporte G, Meunier F, Wolfler Calvo R (2018) Shared mobility systems: An updated survey. *Ann. Oper. Res.* 271(1):105–126.

Li Q, Liao F (2020) Incorporating vehicle self-relocations and traveler activity chains in a bi-level model of optimal deployment of shared autonomous vehicles. *Transportation Res. Part B: Methodological* 140:151–175.

Lu M, Chen Z, Shen S (2018) Optimizing the profitability and quality of service in carshare systems under demand uncertainty. *Manufacturing & Service Oper. Management* 20(2):162–180.

Lubin M, Dunning I (2015) Computing in operations research using Julia. *INFORMS J. Comput.* 27(2):238–248.

Müller J, Bogenberger K (2015) Time series analysis of booking data of a free-floating carsharing system in Berlin. *Transportation Res. Procedia* 10(July):345–354.

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

North American Bikeshare Association (2021) General Bikeshare Feed Specification (GBFS). Accessed September 17, 2021, https://github.com/NABSA/gbfs.

Open Mobility Foundation (2021) Mobility data specification (MDS). Accessed September 17,2021, https://github.com/openmobilityfoundation/mobility-data-specification.

Pal A, Zhang Y (2017) Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. *Transportation Res. Part C: Emerging Tech.* 80:92–116.

Paundra J, Rook L, van Dalen J, Ketter W (2017) Preferences for car sharing services: Effects of instrumental attributes and psychological ownership. *J. Environment. Psych.* 53:121–130.

Powell WB (2011) *Approximate Dynamic Programming*, Wiley Series in Probability and Statistics (John Wiley & Sons, Inc., Hoboken, NJ).

Qi W, Shen ZJM (2019) A smart-city scope of operations management. *Production Oper. Management* 28(2):393-406.

Sahr K, White D, Kimerling AJ (2004) Geodesic discrete global grid systems. *Cartography Geographic Inform. Sci.* 30(2):121–134.

Savelsbergh M, Van Woensel T (2016) 50th Anniversary invited article—city logistics: Challenges and opportunities. *Transportation Sci.* 50(2):579–590.

Schiffer M, Hiermann G, Rüdel F, Walther G (2021) A polynomial-time algorithm for user-based relocation in free-floating car sharing systems. *Transportation Res. Part B: Methodological* 143:65–85.

Schroer K, Ketter W, Lee TY, Gupta A, Kahlen M (2019) An online learning and optimization approach for competitor-aware management of shared mobility systems. *Proc. SIG GREEN Workshop*. Accessed September 17, 2021, https://aisel.aisnet.org/sprouts_proceedings_siggreen_2019/2/.

Ströhle P, Flath CM, Gärttner J (2019) Leveraging customer flexibility for car-sharing fleet optimization. *Transportation Sci.* 53(1):42–61.

Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Programming* 106(1):25–57.

Weikl S, Bogenberger K (2015) A practice-ready relocation model for free-floating carsharing systems with electric vehicles: Mesoscopic approach and field trial results. *Transportation Res. Part C: Emerging Tech.* 57:206–223.

Willing C, Klemmer K, Brandt T, Neumann D (2017) Moving in time and space: Location intelligence for carsharing decision support. *Decision Support Systems* 99:75–85.

Xu C, Ji J, Liu P (2018) The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transportation Res. Part C: Emerging Tech.* 95:47–60.

Zhou X, Shen Y, Zhu Y, Huang L (2018) Predicting multi-step citywide passenger demands using attention-based neural networks. *Proc. 11th ACM Internat. Conf. Web Search Data Mining* (ACM, New York), 736–744.