Data-driven Distributionally Robust Optimization For Vehicle Balancing of Mobility-on-Demand Systems

FEI MIAO, SIHONG HE, and LYNN PEPIN, University of Connecticut
SHUO HAN, University of Illinois at Chicago
ABDELTAWAB HENDAWI, University of Rhode Island
MOHAMED E KHALEFA, Alexandria University
JOHN A. STANKOVIC, University of Virginia
GEORGE PAPPAS, University of Pennsylvania

With the transformation to smarter cities and the development of technologies, a large amount of data is collected from sensors in real time. Services provided by ride-sharing systems such as taxis, mobility-on-demand autonomous vehicles, and bike sharing systems are popular. This paradigm provides opportunities for improving transportation systems' performance by allocating ride-sharing vehicles toward predicted demand proactively. However, how to deal with uncertainties in the predicted demand probability distribution for improving the average system performance is still a challenging and unsolved task. Considering this problem, in this work, we develop a data-driven distributionally robust vehicle balancing method to minimize the worst-case expected cost. We design efficient algorithms for constructing uncertainty sets of demand probability distributions for different prediction methods and leverage a quad-tree dynamic region partition method for better capturing the dynamic spatial-temporal properties of the uncertain demand. We then derive an equivalent computationally tractable form for numerically solving the distributionally robust problem. We evaluate the performance of the data-driven vehicle balancing algorithm under different demand prediction and region partition methods based on four years of taxi trip data for New York City (NYC). We show that the average total idle driving distance is reduced by 30% with the distributionally robust vehicle balancing method using quad-tree dynamic region partitions, compared with vehicle balancing methods based on static region partitions without considering demand uncertainties. This is about a 60-million-mile or a 8-million-dollar cost reduction annually in NYC.

CCS Concepts: • **Mathematics of computing → Stochastic control and optimization**; *Probabilistic algorithms*; • **Networks → Network algorithms**; • **Computer systems organization → Embedded and cyber-physical systems**;

Additional Key Words and Phrases: Distributionally robust vehicle balancing, dynamic region partition, average idle distance, uncertain demand sets

17

This work was supported by NSF CPS-1932250, NSF S&AS-1849246. A conference version of this work was published in title "Data-driven distributionally robust vehicle balancing using dynamic region partitions," in *Proceedings of the 8th International Conference on Cyber-Physical Systems*, pp. 261–271, April 2017.
Authors' addresses: F. Miao, S. He, and L. Pepin, University of Connecticut; emails: fei.miao@uconn.edu, sihong.he@uconn.edu, lynn.pepin@uconn.edu; S. Han, University of Illinois at Chicago; email: hanshuo@uic.edu; A. Hendawi, University of Rhode Island; email: hendawi@uri.edu; M. E Khalefa, Alexandria University; email: khalefa@alexu.edu.eg; J. A. Stankovic, University of Virginia; email: stankovic@virginia.edu; G. Pappas, University of Pennsylvania; email: pappasg@seas.upenn.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
2378-962X/2021/01-ART17 $15.00
https://doi.org/10.1145/3418287

ACM Reference format:
Fei Miao, Sihong He, Lynn Pepin, Shuo Han, Abdeltawab Hendawi, Mohamed E Khalefa, John A. Stankovic, and George Pappas. 2021. Data-driven Distributionally Robust Optimization For Vehicle Balancing of Mobility-on-Demand Systems. *ACM Trans. Cyber-Phys. Syst.* 5, 2, Article 17 (January 2021), 27 pages.
https://doi.org/10.1145/3418287

1 INTRODUCTION

The number of cities is increasing worldwide and the transformation to smarter cities is taking place, which brings an array of emerging urbanization challenges [33]. With the development of technologies, we are able to collect, store, and analyze a large amount of data efficiently [3]. An intelligent transportation system is one example in which sensing data collected in real time provide opportunities for understanding spatial-temporal human mobility patterns. For instance, traffic speed [6], travel time [7, 22], passengers' demand model [31, 47] and origin-destination model of taxi networks [24, 27], and transportation network efficiency [41] are inferred and measured.

Researchers have been working on various approaches to improve the performance of transportation systems. Smart parking systems that allocate and reserve parking space for drivers [18], routing, and motion planning problems for mobile systems [23, 42] have been proposed. By considering future demand predicted with data when making current decisions, optimal vehicle balancing strategies have many advantages compared with approaches that do not balance vehicles from a systemwide coordination perspective. Vehicle balancing methods have been studied for car sharing systems [44] or mobility-on-demand systems [40, 43] by distributing vehicles among regions or reactive trip assignment [5], based on current or predictive demand. Vehicle balancing can reduce the number of vehicles needed to serve all passengers in mobility-on-demand systems [36, 50, 51] and bike-sharing systems [37, 38], or reduce customers' waiting time [36, 51] and taxis' total idle distance [29] with the same number of empty vehicles. However, the limited knowledge we have about demand and mobility patterns [17] affect the performance of vehicle balancing strategies, and make real-time decisions under demand model uncertainties still a challenging and unsolved task. Considering demand uncertainties, although the robust optimal solution shows its advantage in worst-case scenarios compared with non-robust approaches [4, 26, 28], there is still a tradeoff between the system's average and worst-case performance [30].

In this work, we integrate the process of estimating passenger demand based on data and calculating a vehicle balancing solution, to consider demand model uncertainties and compute a vehicle resource allocation solution in real time. In practice, it is difficult to obtain a true probability distribution of the random passenger demand purely based on data, making it impossible to estimate the true expected vehicle balancing cost. Therefore, we minimize the worst-case expected vehicle balancing cost under a set of possible probability distribution functions of passenger demand learned from data, by formulating a data-driven distributionally robust optimization problem. Distributionally robust optimization techniques have been developed for minimizing the worst-case expected (average) cost under a set of probability distributions of the random parameters by solving a semidefinite programming (SDP) problem [14, 19, 35], instead of the worst-case cost under the extreme case value of the parameters in robust optimization in the literature. These algorithms have been shown to provide solutions with better average cost compared with the minimizing worst-case robust optimization approaches [9], hence, they are less conservative than robust optimization algorithms. However, there are no approaches for real-time distributionally robust vehicle balancing, or evaluations to show the ride-sharing service performance improvement by considering prediction uncertainties of the demand probability distribution, especially for complicated demand prediction models such as time-series or deep neural network [25, 31, 47] yet.

Hence, we design a computationally tractable dynamic vehicle balancing method that is robust to uncertainties in the probability distribution of the demand. Efficient algorithms for constructing an uncertainty set of the probability distributions based on data and different demand models are developed, where we utilize a structural property of the covariance of the random demand. A quad-tree dynamic region partition method is used for the first time, and shown to improve performance in the experiments. We then derive an equivalent convex optimization form of the non linear programming (LP) or SDP form of the distributionally robust vehicle balancing problem, and guarantee both average performance of the system and computational tractability. Finally, we evaluate the average costs of the distributionally robust vehicle balancing method, based on uncertainty sets constructed on different region partitions and demand models from real data. We calculate the total idle distance by summing up the distance every vehicle runs without any passenger on it, and the "average total idle distance" by taking the average of all the testing samples.

The contributions of this work are the following:

- We explicitly take the ambiguity of demand probability distribution into account when minimizing vehicle balancing cost. We design a data-driven distributionlly robust dynamic vehicle balancing method to optimize the expected cost over the worst-case distribution of demand. Previous vehicle balancing work either focuses on one specific probability distribution or aims to find a robust solution for a deterministic worst-case demand.
- For the first time, we design a quad-tree dynamic region partition method and efficient algorithms to construct uncertainty sets of probability distributions given different demand models. These sets better capture the spatial-temporal correlations of demand uncertainties based on data.
- We derive an equivalent convex reformulation of the distributionally robust vehicle balancing problem to guarantee computational tractability of finding a solution under demand uncertainties. The original distributionally robust vehicle balancing problem is a minimax problem with an objective function that is convex of the decision variables and linear of the uncertain parameters.
- We evaluate the average cost obtained by adopting the distributionally robust vehicle balancing solutions based on four years taxi trip data of New York City. We show that for the average demand model, the average total idle distance is reduced by 10.05% with static grid region partition. With the quad-tree dynamic region partition, the average total idle distance is reduced by 20% more. This is about 60 million miles or 8 million dollars gas cost reduction annually compared with non-robust solutions. For a more accurate time-series demand model, the average total idle distance is reduced by 7.68% by considering demand prediction uncertainties with a static grid region partition, and is reduced by 19.60% more with a quad-tree dynamic region partition.

The rest of the article is organized as follows. The distributionally robust vehicle balancing problem is presented in Section 2. Efficient algorithms for constructing distributional uncertainty sets for different demand prediction models and a dynamic region partition method are designed in Section 3. An equivalent computationally tractable form of the distributionally robust vehicle balancing problem is proved in Section 4. We show performance improvement in experiments based on a real dataset in Section 5. Concluding remarks are provided in Section 6.

2 DYNAMIC DISTRIBUTIONALLY ROBUST VEHICLE BALANCING

In this section, we propose a distributionally robust vehicle balancing problem based on dynamic spatial region partitions. The goal includes balancing vehicles for efficient service and reducing the total costs, such as vehicles' total idle distance or the total number of vehicles sent to other

Table 1. Parameters and Variables of Taxi Dispatch Problem (8)

Parameters of (8)	Description
n^k	the number of regions at time k
τ	model predicting time horizon
n_c	total number of regions for time $\{1, 2, \ldots, \tau\}$
$r_c \in \mathbb{R}^{n_c} \sim F^*, F^* \in \mathcal{F}$	the concatenated demand vector with unknown distribution function F^* for $k = 1, \ldots, \tau$
$W^k \in \mathbb{R}^{n^k \times n^k}$	weight matrix, W^k_{ij} is the distance from region i to j
$P^k_v, P^k_o, Q^k_v, Q^k_o$	region transition matrices from time k to $(k+1)$
$V^1 \in \mathbb{N}^{n^1}$	the initial number of vacant taxis at each region provided by GPS and occupancy status data
$O^1 \in \mathbb{N}^{n^1}$	the initial number of occupied taxis at each region provided by GPS and occupancy status data
$m^k \in \mathbb{R}^+$	the upper bound of distance each taxi can drive idly for picking up a passenger at time k
$M^k \in \mathbb{R}^{n^k \times n^k}$	the structural constraint matrix that restricts $X^k_{ij} = 0$ for far away regions
$\alpha \in \mathbb{R}_+$	the power on the denominator of the objective function
$\beta \in \mathbb{R}_+$	the weight factor of the objective function
Variables of (8)	
$X^k_{ij} \in \mathbb{R}_+$	the number of taxis dispatched from region i to region j during time k
$V^k \in \mathbb{R}^{n^k}_+$	the number of vacant taxis at each region before dispatching at the beginning of time k
$O^k \in \mathbb{R}^{n^k}_+$	the number of occupied taxis at each region before dispatching at the beginning of time k
$S^k \in \mathbb{R}^{n^k}_+$	the number of vacant taxis at each region after dispatching at time k

regions. By considering different probability distribution functions of predicted demand, we take explicitly the ambiguity of demand probability distributions to guarantee the average system performance. Previous work either assumes an explicit demand distribution [36, 38, 50, 51] or aims to find a robust vehicle balancing solution for the worst-case [28, 30, 36] given static city region partitions. The generalization of the vehicle balancing problem formulation in this work is explained in Section 2.2. A list of parameters and variables in the problem formulation is shown in Table 1.

We assume that one day is divided into K time intervals indexed by $t = 1, 2, \ldots, K$ in total. Vehicle balancing or re-balancing decision for a time window of τ intervals is calculated at the beginning of the current time interval in a receding horizon control process. To improve the overall performance of the system, the main decision variables in the proposed algorithm are the number of empty vehicles in each region that will be allocated to other regions under certain objectives and constraints, and the total number of empty vehicles at each region will be changed after re-balancing. Examples of receding horizon control of resource allocation approaches include economic power dispatch [26], taxi dispatch [29], autonomous mobility-on-demand service [51], and so on. In particular, each τ discrete time slots $(t, t+1, \ldots, t+\tau-1)$ is indexed by $k = 1, 2, \ldots, \tau$ when we calculate a vehicle re-balancing solution for empty unassigned vehicles toward demand predicted within time $(t, t+1, \ldots, t+\tau-1)$, respectively. The effect of current decisions to the future re-balancing cost is involved. Only the solution of $k = 1$ for time t is implemented, while the solutions for remaining time slots are not materialized. After one empty vehicle arrives at its dispatched region, a local controller will assign the vehicle to pick up one or several passengers in this region's request queue according to the local control or trip assignment algorithms, such as greedy method or trip assignment algorithms designed in the literature [5, 12, 32]. Local trip assignment algorithm is out of the scope of this work, and our vehicle balancing method is agnostic and thus could also be employed in conjunction with those other local control methods to improve the re-balancing. In this work, we focus on the city level vehicle balancing problem calculated at the beginning each time step t, such that demand prediction uncertainties and expected total resource allocation cost are considered in a computationally tractable optimization problem

format. When the time horizon rolls forward by one time step from t to $(t+1)$, information about uncertain demand is first updated, and vehicle locations and occupancy status are observed again. For notation convenience, the parameters and variables definition in the following parts of this section omit the time index t when there is no confusion.

2.1 Problem Formulation

We assume that for every τ time slots the number of region partitions in the city is either static or changing arbitrarily with time, use superscript k to denote time, and space is partitioned to n^k regions (nodes) at time k. We define $r_j^k \geqslant 0$ as the predicted total amount of demand (e.g., number of passengers for a mobility-on-demand system) of region j during time k, where $j = 1, \ldots, n^k$, $k = 1, \ldots, \tau$, and $r^k = [r_1^k, r_2^k, \ldots, r_{n^k}^k]^T \in \mathbb{R}^{n^k}$ is a column vector of predicted demand of all regions at time k, and we consider r^k as a random vector instead of a deterministic one. To model spatial-temporal correlations of demand during every τ consecutive time slots, we define the concatenation of demand as $r_c = (r^1, r^2, \ldots, r^\tau), \quad n_c = \sum_{k=1}^{\tau} n^k$. We assume that F^* is the true probability distribution of the random vector r_c, i.e., $r_c \sim F^*$. It is worth noting that F^* can vary for different time step t, and r^k, r_c is predicted as real-valued vectors instead of integers according to the current learning models [31, 47]. In Section 3, we will build an uncertainty set for the probability distribution of r_c at each time step t.

We denote by a non-negative matrix X^k the decision matrix at time k, where $X^k \in \mathbb{R}_+^{n^k \times n^k}$, and $X_{ij}^k \geq 0$ is the number of empty unassigned vehicles that will be sent from region i to region j (or node i to node j) at time k according to the demand and service requirements. For notational convenience, we define a set of decision variables as $X^{1:\tau} = \{X^1, X^2, \ldots X^\tau\} \in \mathcal{D}_c$, where $\mathcal{D}_c$ is the convex domain of decision variables defined by constraints. If we have the true probability distribution of demand $r_c \sim F^*$, then minimizing the expected cost of allocating vehicles in the city is defined as a stochastic programming problem:

$$\min_{X^{1:\tau}} \mathbb{E}_{r_c \sim F^*} \left[J(X^{1:\tau}, r_c) \right] \quad \text{s.t.} \quad X^{1:\tau} \in \mathcal{D}_c, \tag{1}$$

where $J(X^{1:\tau}, r_c)$ is the cost of allocating vehicles according to decisions $X^{1:\tau}$ under demand r_c.

However, in many applications we only have limited knowledge about the true distribution F^*. The knowledge of random demand r_c is restricted to a set of independent and random samples—historical or streaming demand data, according to an unknown distribution F^*. We assume that the true lower, upper bound, mean and covariance information lie in a neighborhood of their respective empirical estimates, a common assumption of data-driven optimization problems [14, 19]. It is worth noting that solving problem (1) is computationally expensive, especially for a large-scale transportation network. We then consider to minimize the worst-case expected cost as a robust form of problem (1) defined in the following:

$$\min_{X^{1:\tau}} \max_{F \in \mathcal{F}} \quad \mathbb{E}_{r_c \sim F} \left[J(X^{1:\tau}, r_c) \right] \quad \text{s.t.} \quad X^{1:\tau} \in \mathcal{D}_c. \tag{2}$$

Problem (2) is a form of a distributionally robust optimization problem, in the sense of minimizing the worst case expected cost when we are uncertain about the true probability distribution of the demand, or the parameter r_c [14, 19], instead of the worst-case cost under the extreme case value of the parameters in robust optimization in the literature. It provides solutions with better average cost compared with the minimizing worst-case robust optimization approaches [9] in LP and SDP problems, hence, less conservative than robust optimization algorithms. In the rest of this section, we define concrete forms of the objective function and constraints. In Section 3, we design an algorithm for calculating the set $\mathcal{F}$ such that $F^* \in \mathcal{F}$ with a desired probability.

2.1.1 Service Quality Metric Function J_E. We define $V_j^k \in \mathbb{R}_+, O_j^k \in \mathbb{R}_+$ as the number of vacant and occupied vehicles in region j before balancing or re-balancing at the beginning of time k, respectively, and $V^k, O^k \in \mathbb{R}_+^{n^k}$. When receding the time horizon, we always first update real-time sensing information, such as GPS locations and occupancy status of all vehicles, and $V^1 \in \mathbb{R}_+^{n^1}$ and $O^1 \in \mathbb{R}_+^{n^1}$ are provided by real-time data. We denote $S_i^k > 0$ as the total amount of vehicles available within region i during time k with dispatch decision $\{X^1, \ldots, X^k\}$, and

$$\begin{aligned} S_i^k &= \sum_{j=1}^{n^k} X_{ji}^k - \sum_{j=1}^{n^k} X_{ij}^k + V_i^k > 0, \quad k = 1, \ldots, \tau, \\ V_i^{k+1} &= \sum_{j=1}^{n^k} P_{v,ji}^k S_j^k + \sum_{j=1}^{n^k} Q_{v,ji}^k O_j^k, \quad O_i^{k+1} = \sum_{j=1}^{n^k} P_{o,ji}^k S_j^k + \sum_{j=1}^{n^k} Q_{o,ji}^k O_j^k, \quad k = 1, \ldots, \tau - 1, \end{aligned} \tag{3}$$

where $P_v^k, P_o^k, Q_v^k, Q_o^k \in \mathbb{R}^{n^k \times n^{k+1}}$ are region transition matrices: $P_{v,ji}^k$ ($P_{o,ji}^k$) describes the probability that a vacant vehicle starts from region j at the beginning of time interval k will traverse to region i and being vacant (occupied) at the beginning of time interval $(k+1)$; similarly, $Q_{v,ji}^k$ ($Q_{o,ji}^k$) describes the probability that an occupied vehicle starts from region j at the beginning of time interval k will traverse to region i and being vacant (occupied) at the beginning of time interval $(k+1)$. The current occupied vehicles can drop passengers, turn to empty vehicles and be part of the service in the future. The region transition matrices are learned from historical data, and satisfy that $\sum_{j=1}^{n^k} P_{v,ij}^k + P_{o,ij}^k = 1, \quad \sum_{j=1}^{n^k} Q_{v,ij}^k + Q_{o,ij}^k = 1$. Methods of origin-destination prediction [24, 25], taxi trajectory prediction [27], and region transition matrices calculation have been analyzed in previous work [29, 51]. Though the estimation based on data is not perfectly accurate, at each time step (each $k = 1$), we always update GPS information of current vacant and occupied vehicles before calculating the vehicle dispatch decision, such that V^1, O^1 are always true values. It is a compensation for the inaccurate estimation based on experiments in References [29, 51]. Hence, in this work, we only consider predicted demand uncertainties for time $k = 1, \ldots, \tau$ to avoid unnecessary computational complexity of involving both demand and region transition matrices uncertainties.

Balancing the supply–demand ratio across the network is one commonly considered service quality metric for taxi dispatch [29] and autonomous mobility on demand systems [50]. The objective is defined as minimizing the total difference between the local and global demand–supply ratio for τ time intervals

$$\sum_{k=1}^{\tau} \sum_{i}^{n^k} \left| \frac{r_i^k}{S_i^k} - \frac{\sum_{j=1}^{n^k} r_j^k}{\sum_{j=1}^{n^k} S_i^k} \right|. \tag{4}$$

However, function (4) is not concave of the uncertain random parameters r^k, making the problem (2) computationally intractable to find a maximum value of the objective function over r^k given x^k. Hence, we adopt the following equivalent service quality function J_E:

$$J_E(X^{1:\tau}, r^k) = \sum_{k=1}^{\tau} \sum_{i=1}^{n^k} \left(\frac{r_i^k}{(S_i^k)^\alpha} \right), \tag{5}$$

where $\alpha > 0$ is a non-negative value close to 0, and minimizing the objective function (5) approximates minimizing the objective (4). This is because the optimal solution of Equation (5) makes the absolute value function (4) close to 0, as proved by Lemma 1 in Reference [30]. As proved in Theorem 1 of Reference [30], Equation (5) is convex of the decision variables on the denominator.

We then minimize Equation (5) to reach the balancing vehicle objective with other objective and constraint functions in the final formulation of vehicle balancing problem (see Section 2.1.2). With the definition of S_i^k as Equation (3), S_i^k is linear of $X^{1:\tau}$, J_E is a function concave (linear) in r^k and convex in $X^{1:\tau}$ that has the decision variables on the denominator.

2.1.2 Cost of Balancing and Re-balancing. Besides minimizing the service quality function (5), we also consider minimizing the costs (such as idle distance) by sending vacant vehicles according to X^k. Given the city map and region partition structure during time k, we define $W^k \in \mathbb{R}^{n^k \times n^k}$ as the weight matrix, where W_{ij}^k describes the cost of sending one vehicle from region i to region j during time k according to the spatial network model. For instance, when W_{ij}^k is the approximated distance to drive from region i to region j, the *en route* idle distance is considered as the cost for allocating one empty vehicle. When $W_{ij}^k = 1$, the cost of re-balancing a vehicle between any region pair (i, j) is identical that the total number of vacant vehicles balanced between all pairs of (i, j) is considered as the total cost. The across-region balancing cost according to X^k is

$$J_D(X^k) = \sum_{i=1}^{n^k} \sum_{j=1}^{n^k} X_{ij}^k W_{ij}^k. \tag{6}$$

The distance every vehicle can travel is bounded, because of the speed limit during time k and traffic conditions—during congestion hours, the distance each vehicle can go to pick up a passenger should be shorter than normal hours without congestion. Assume that the idle distance upper bound for a vehicle at time k is $m^k > 0$, provided by traffic speed monitors and forecasting models [1, 6], the distance from region i to region j is $dist_{ij}$. Then the following constraint:

$$X_{ij}^k \geqslant 0 \text{ and } X_{ij}^k = 0 \text{ when } dist_{ij} \geqslant m^k \tag{7}$$

indicates a solution that satisfies $X_{ij}^k = 0$ for $dist_{ij} > m^k, i, j = 1, \ldots, n^k$. Hence, a vehicle can only be sent from region i to region j to balance the demand–supply ratio at time k, when the distance between the two regions is smaller than or equal to m^k.

We aim to balance vehicles with minimum idle distance, and define a weight parameter β of two objectives J_D in Equation (6) and J_E in Equation (5). How parameter β will affect the cost of each objective, the idle distance and the balancing of vehicles is discussed and compared by experiments in Reference [29]. With constraints (3) and (7), we consider the following distributionally robust vehicle balancing problem under uncertain probability distributions of random demand:

$$\begin{aligned} \min_{X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau}} \max_{F \in \mathcal{F}} & \mathbb{E}\left[\sum_{k=1}^{\tau} J_D(X^k) + \beta J_E\right] \\ \text{s.t.} \quad & (3), (7), \end{aligned} \tag{8}$$

where $X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau}$ denote variables and $O^2, \ldots, O^\tau$ (V^1 and O^1 are given by sensing information) respectively. The above problem (8) cannot be immediately translated into an LP or SDP form. Only the service requirement J_E has decision variables on the denominator and is directly related to the random demand r^k, balancing cost J_D and all the constraints are linear in the variables and not functions of r^k. The minimization part of problem (8) is convex of the decision variables, since J_E is convex of the decision variables, J_D and the constraints are all linear, and linear constraints and objective function are convex [10].

2.2 Generalization of Problem Formulation

Reducing the dependency of the average performance of solutions on the accuracy of demand model: Problem (8) is one example of a distributionally robust vehicle balancing problem

that does not restrict the specific distribution of random demand. For instance, for queuing models, the average number of waiting customers in the queue is related to the demand–supply ratio or supply–demand ratio for a stable queue [20]. Considering a balanced demand–supply ratio is considering balancing the average number of waiting customers intuitively. Robotic mobility-on-demand systems [48, 50] usually assume a queuing model to describe the passenger arrival rate at region i is λ_i^k. When calculating the arrival rate for one time interval from historical data, λ_i^k equals the total number of requests appearing in one time interval, or r_i^k in this work. Mean and covariance of the estimation of λ_i^k still exist when calculating this arrival rate λ_i^k via data. Hence, when a mobility-on-demand system can be described by a queuing model, solving problem (8) provides a solution for balancing vehicles for λ_i^k in a range instead of a deterministic value. Therefore, we do not restrict the demand model to satisfy a specific distribution and we reduce the dependency of the average performance of solutions to the accuracy of demand model.

Similarly, bicycle balancing and re-balancing problems also require that the demand–supply ratio of each station is restricted inside a range to provide a certain level of service satisfaction [38]. While adjusting the range of demand–supply ratio or supply–demand ratio back and forth is computationally expensive, when we find a feasible solution of Equation (8), the demand–supply ratio of each region should not be far away from the global demand–supply ratio, and falls in a range around the global level. Hence, when the objective is to make the demand–supply ratio of each region all be inside some range without knowing the feasible upper and lower bounds of the range, solving Equation (8) that makes the local ratio all close to the global ratio and will reach an equivalent objective without selecting the range manually.

Balancing vehicles for carpooling or heterogeneous vehicle service: We consider a single type vehicle balancing problem (for instance, each individual empty vehicle is considered to have the same ability) under formulation (8). When each vehicle in the system has a different service ability, for instance, when the capacity of one vehicle is $C_1 = 1$, $C_2 = 2$, $C_3 = 3$ or $C_4 = 4$, we denote $O_{l,i}^k$ as the number of vehicles with capacity C_l before dispatch at region i, and $X_{l,ij}^k$ as the number of vehicles that should go from region i to region j. Then the total number of available seats or supply is $S_i^k = \sum_{l=1}^4 C_l(O_{l,i}^k + \sum_{j=1}^{n^k} X_{l,ji}^k - \sum_{j=1}^{n^k} X_{l,ij}^k) > 0$. With this number S_i^k, objective function J_E defined as Equation (5) is still concave in r^k, convex in X_l^k, $l = 1, 2, 3, 4$, since linear operation preserves convexity [10, Chapter 3.2.2]. The balancing cost (6), constraints about region transition (3) and idle distance bound (7) can be modified accordingly and still be linear of decision variables. Under this scenario, with a modified definition of S_i^k, the vehicle balancing model (8) can be generalized for carpooling or heterogeneous capacity vehicle balancing problems. With periodically re-balancing vehicles every hour or 30-minutes by a centralized distributionally robust optimizer, a local level matching between passengers and vehicles within each region will assign one vehicle to several requests according to its capacity, such as the local carpool algorithm developed in Reference [12].

3 EFFICIENT DISTRIBUTIONAL SET CONSTRUCTION ALGORITHMS

We design efficient algorithms for constructing the uncertainty set $\mathcal{F}$ of probability distributions in problem (8), with spatial-temporal data that provides information about the true distribution F^* of r_c. Empirical estimates of the uncertainty set for predicted variables according to confidence regions of hypothesis testings are acceptable in portfolio management problems [9, 14]. However, vehicle trip or trajectory data are usually large-scale spatial-temporal data, and how to efficiently extract information of mobility demand is a challenging task. Considering the computational cost of building a distributional set for every consecutive τ time slots (the demand prediction and vehicle balancing time lengths) of one day, we leverage the structure property of the covariance matrix

Table 2. Parameters of Algorithm 1 and Algorithm 2

$r_c(t) \in \mathbb{R}^{n_c}$	the concatenated demand at each region from time t to $t+\tau-1$
$\tilde{r}_c(d_l, t, I_p)$	one sample of $r_c(t)$ according to sub-dataset I_p, records at date d_l
$\hat{r}_c \in \mathbb{R}^{n_c}$, $\hat{\Sigma}_c \in \mathbb{R}^{n_c \times n_c}$	the estimated mean and covariance of vector r.v. r_c
$\hat{r}_{c,l}$, $\hat{r}_{c,h}$	the estimated lower and upper bound of vector r.v. r_c
γ_1^B, γ_2^B	the bootstrapped thresholds for accepting hypothesis testing (9)
$\alpha_h \in (0,1)$	significance level of a hypothesis testing
$N^B \in \mathbb{Z}^+$	the time of re-sampling in bootstrap
$N \in \mathbb{Z}^+$	the size of bootstrap sample sets

of r_c to develop efficient construction algorithms for set $\mathcal{F}$ based on different demand prediction models. Then, for prediction methods directly using average historical values as the demand, and more accurate complicated prediction models such as time series, we can describe the prediction uncertainties by a closed and convex set, and use the uncertainty set to make distributionally robust vehicle dispatch decisions based on Equation (8). Furthermore, to reflect the spatial-temporal dynamic properties of demand and index regions efficiently, we build our distributional set based on a dynamic space partition method.

3.1 Distributional Set Formulation

We denote one sample of vector $r_c(t) = (r^t, r^{t+1}, \ldots, r^{t+\tau-1})$ at date d_l as $\tilde{r}_c(d_l, t)$, a vector of demand at each region for time $\{t, t+1, \ldots, t+\tau-1\}$, $t = 1, \ldots, K$ of each day. We aim to construct a uncertainty set $\mathcal{F}(t)$ that describes possible probability distributions of $r_c(t)$ based on the support, mean and covariance of random samples of $r_c(t)$. We omit t for the following problem definition when there is no confusion. Possible probability distributions of a random vector r_c is related to the following hypothesis testing H_0: Given mean μ_0, covariance Σ_0, test statistics γ_1, γ_2, and with a given significance level α_h, the random vector r_c satisfies that [14]

$$H_0 : (\mathbb{E}[r_c] - \mu_0)^T \Sigma_0^{-1} (\mathbb{E}[r_c] - \mu_0) \leqslant \gamma_1, \quad \mathbb{E}[(r_c - \mu_0)(r_c - \mu_0)^T] \preceq \gamma_2 \Sigma_0. \tag{9}$$

Without prior knowledge of the true mean μ_0, covariance Σ_0, test statistics γ_1 and γ_2, constructing set $\mathcal{F}$ based on data is an inverse process of hypothesis testing—estimating the mean and covariance and calculating threshold values γ_1 and γ_2 such that Equation (9) is an acceptable hypothesis by the dataset with probability at least $1-\alpha_h$. The problem of constructing $\mathcal{F}$ is formally defined as follows.

Definition 3.1 (Problem 1). Given a sample set of r_c, find values of $\hat{r}_{c,l}, \hat{r}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, \gamma_1^B$, and γ_2^B, such that with probability at least $1-\alpha_h$ with respect to the samples the hypothesis testing Equation (9) is acceptable, i.e., with probability at least $1-\alpha_h$ the true distribution of r_c is contained in the following distributional set $\mathcal{F}$:

$$\begin{aligned} &\mathcal{F}(\hat{r}_{c,l}, \hat{r}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, \gamma_1^B, \gamma_2^B) \\ =&\{(\mathbb{E}[r_c] - \hat{r}_c)^T \hat{\Sigma}_c^{-1} (\mathbb{E}[r_c] - \hat{r}_c) \leqslant \gamma_1^B, \quad \mathbb{E}[(r_c - \hat{r}_c)(r_c - \hat{r}_c)^T] \preceq \gamma_2^B \hat{\Sigma}_c, \ r_c \in [\hat{r}_{c,l}, \hat{r}_{c,h}]\}, \end{aligned} \tag{10}$$

where $\hat{r}_{c,l}$ and $\hat{r}_{c,h}$ is the lower and upper bound of each entry of the demand vector, respectively.

We then design Algorithm 1 (a list of parameters in Table 2) to calculate the bootstrapped [11] estimations of $\hat{r}_{c,l}, \hat{r}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, \gamma_1^B, \gamma_2^B$ for $r_c(t)$, $t = 1, 2, \ldots, K$ of every time step, that makes H_0 in Equation (9) acceptable.

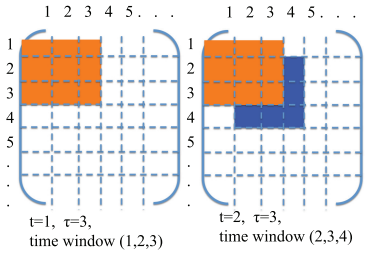


Fig. 1. The process of calculating $\hat{\Sigma} \in \mathbb{R}^{n\times n}$, $n = \sum_{t=1}^{K} n^t$ when receding time horizon. When index moves from $t = 1$ to $t = 2$, only entries in matrix $\hat{\Sigma}$ shown in blue are new and necessary for calculating $\hat{\Sigma}_c(t), t = 2$.

3.2 Reducing Computational Complexity

The dimension of $\hat{r}_c, \hat{\Sigma}_c$ is decided by the number of dynamic regions and the prediction horizon, which can be large-scale for spatial-temporal city transportation data. However, the mean and covariance matrices for $t, t+1, \ldots, t+\tau$ have overlapping components: For instance, $\hat{r}_c(t)$ and $\hat{r}_c(t+1)$ both include estimated mean values of demand in time $(t+1, t+2, \ldots, t+\tau-1)$. Hence, instead of always repeating the process of calculating a mean and covariance for τ time slots for each index t, the key idea of reducing computational cost of constructing $\mathcal{F}(t), t = 1, \ldots, K$ is to calculate the mean and covariance of each pair of time slots of the whole day only once. Then pick the corresponding components needed to construct $\hat{r}_c(t)$ and $\hat{\Sigma}_c(t)$ for each index t.

Specifically, we define the whole day demand vector as $r = (r^1, r^2, \ldots, r^K) \in \mathbb{R}^n, n = \sum_{t=1}^{K} n^t$, i.e., a concatenated demand vector for each time slot of one day. And we denote $\hat{r}$ as the estimated mean of the random vector r. To get all covariance components for each index t, the process is as follows: At $t = 1$, calculate the covariance of $r_c(1)$, store it as $\bar{\Sigma}_{[1:n^1,1:n^1]}$; and every time when rolling the time horizon from t to $t+1$, only calculate the covariance matrix entries between τ pairs of $(r^{t+\tau-k}, r^{t+\tau})$, $k = 0, \ldots, \tau-1$ and store the result as

$$\bar{\Sigma}_{[n^{[1,t+\tau-1]}:n^{[1,t+\tau]},\, n^{[1,t+\tau-k]}:n^{[1,t+\tau-k+1]}]} = \bar{\Sigma}_{[n^{[1,t+\tau-k]}:n^{[1,t+\tau-k+1]},\, n^{[1,t+\tau-1]}:n^{[1,t+\tau]}]} = \mathrm{cov}(r^{t+\tau-k}, r^{t+\tau}), \tag{11}$$

where $n^{[1,t+\tau]} = \sum_{j=1}^{t+\tau} n^j$, the subscript $[b_1 : b_2, b_2 : b_1]$ means entries from the b_1th to the b_2th rows and b_2th to the b_1th columns of matrix $\bar{\Sigma}$ as explained in Figure 1.

Then we have Algorithm 1 that describes the complete process of constructing distributional sets. Given vehicles' service trajectories or trips data, we count the total number of pick up events during one hour at each region as total demand. If the given dataset is the arriving time of each customer at different service nodes of a network, then the total number of customers appearing in every service node during each unit time is the demand. When categorical information such as normal days or holidays/special event days of one year, different weather conditions or a combination of different contexts is available, indexed as $I_p, p = 1, 2, \ldots, P$, we cluster the dataset as subsets first.

For step 3, the process of picking components from the mean and covariance matrices of the whole day demand is

$$\hat{r}_c(t, I_p) = \hat{r}_{[n^{[1,t-1]}:n^{[1,t+\tau-1]}]}(I_p), \quad \hat{\Sigma}_c^j(t, I_p) = \hat{\Sigma}^j_{[n^{[1,t-1]}:n^{[1,t+\tau-1]},\, n^{[1,t-1]}:n^{[1,t+\tau-1]}]}(I_p). \tag{12}$$

For the jth re-sampled subset $\mathcal{S}^j(t, I_p)$, the mean and covariance matrices are $\mathbb{E}[r_c] = \bar{r}_c^j(t, I_p)$ and $\mathbb{E}[r_c r_c^T] = \bar{\Sigma}_c^j(t, I_p)$, respectively. For step 3(2), according to the definition of $\mathcal{F}$ in Equation (10),

ALGORITHM 1: Algorithm for constructing distributional sets

Input: A dataset of spatial-temporal operation records

1. Demand aggregating and sample set partition

Aggregate demand of each region for each time t to get a sample set $\mathcal{S}$ of demand r for the whole day (denote $\mathcal{S}(t)$ as a sample set for $r_c(t)$) from the original data. Partition $\mathcal{S}(or\mathcal{S}(t))$ and denote $\mathcal{S}(I_p) \subset \mathcal{S}$ (or $\mathcal{S}(t, I_p) \subset \mathcal{S}(t)$), $p = 1, \ldots, P$ as the subset of categorical information I_p. Set a significance level $0 < \alpha_h < 1$, the number of bootstrap time $N_B \in \mathbb{Z}_+$ and the size of bootstrap sample set $N \in \mathbb{Z}_+$.

2. Bootstrapping mean and covariance matrix for all time steps t in one day

for $j = 1, \ldots, N_B$ **do**

Re-sample $\mathcal{S}^j(I_p) = \{\tilde{r}(I_p)_1, \ldots, \tilde{r}(I_p)_N\}$ from $\mathcal{S}(I_p)$ with replacement, calculate the sample mean $\bar{r}^j(I_p)$ and covariance $\bar{\Sigma}^j(I_p)$ of the whole day demand vector of set $\mathcal{S}^j(I_p)$ as Equation (11).

end for

Get the bootstrapped mean covariance, and support of the whole day demand vector

$\hat{r}(I_p) = \frac{1}{N_B}\sum_{j=1}^{N_B} \bar{r}^j(I_p)$, $\hat{\Sigma}(I_p) = \frac{1}{N_B}\sum_{j=1}^{N_B} \bar{\Sigma}^j(I_p)$, $\hat{r}_{c,l}(I_p) = min_i \tilde{r}(I_p)_i$, $\hat{r}_{c,h}(I_p) = max_i \tilde{r}(I_p)_i$.

3. Bootstrapping γ_1^B and γ_2^B for each subset $\mathcal{S}(t, I_p)$

for $j = 1, \ldots, N_B$ **do**

Get the mean and covariance vector for the jth re-sampled set $\bar{r}_c^j(t, I_p)$, $\bar{\Sigma}_c^j(t, I_p)$, then get $\hat{r}_c(t, I_p)$, $\hat{\Sigma}_c(t, I_p)$ as (12). Get $\gamma_1^j(t, I_p)$ and $\gamma_2^j(t, I_p)$ by Equations (13) and (14).

end for

Get the $\lceil N_B(1-\alpha_h)\rceil$-th largest value of $\gamma_1^j(t, I_p)$ and $\gamma_2^j(t, I_p)$, $j = 1, \ldots, N_B$, as $\gamma_1^B(t, I_p)$ and $\gamma_2^B(t, I_p)$, respectively.

Output: Distributionally uncertainty sets (10).

we get $\gamma_1^j(t, I_p)$ by the following equation:

$$\gamma_1^j(t, I_p) = [\bar{r}_c^j(t, I_p) - \hat{r}_c(t, I_p)]^T \hat{\Sigma}_c^{-1}(t, I_p)[\bar{r}_c^j(t, I_p) - \hat{r}_c(t, I_p)]. \tag{13}$$

According to definition (10), the left part of the inequality related to γ_2^B satisfies that

$$\mathbb{E}[(r_c - \hat{r}_c)(r_c - \hat{r}_c)^T] = \mathbb{E}[r_c r_c^T] - \hat{r}_c \mathbb{E}[r_c^T] - \mathbb{E}[r_c]\hat{r}_c^T + \hat{r}_c \hat{r}_c^T = \bar{\Sigma}_c - \hat{r}_c \hat{r}_c^T.$$

Then we get γ_2^j for index (t, I_p) by solving the following convex optimization problem:

$$\min_{\gamma_2} \quad \gamma_2, \qquad \text{s.t} \quad \bar{\Sigma}_c^j(t, I_p) - [\hat{r}_c(t, I_p)][\hat{r}_c(t, I_p)]^T \leq \gamma_2 \hat{\Sigma}_c(t, I_p). \tag{14}$$

3.3 Constructing Uncertainty Sets for a General Demand Prediction Model

Besides directly using the estimated moments of the concatenated demand vector $r_c(t, I_p)$ for each index (t, I_p), methods that predict demand for time t based on the latest observation of time $t-1, t-2, \ldots$ or streaming data have also been applied in areas such as transportation networks [31, 49], power networks [26, 39], and health care systems [2]. Complicated models can be more accurate than the average value prediction. It is critical to develop an uncertainty set constructing algorithm for general demand modeling techniques, and explore the effects of considering uncertainties to improve performance. In this subsection, we design a process of constructing distributional uncertainty sets for a general demand prediction model, and introduce an example of multivariate time-series demand predicting model based on streaming data.

3.3.1 Uncertainty Set of a General Demand Prediction Model. We do not restrict the learning or modeling method to predict demand, and assume that $f_r : \mathcal{O}_{[t-1-l,t-1]} \to \mathbb{R}^{\tau n}$ is a function of mapping sensing or observation data available to the system by time t (from time $(t-1-l)$ to time $(t-1)$) to predicted concatenated demand at time t. The prediction function f_r is unknown

and can only be estimated from data. We would like to quantify the estimation uncertainty and consider possible estimation errors when providing ride-sharing service. Then we have the following relation between the deterministic component of predicted demand $\hat{r}_c(t)$ and the true demand $r_c(t)$:

$$r_c(t) = f_r(O_{t-1}), \quad r_c(t) = \hat{r}_c(t) + \delta_c(t). \tag{15}$$

Here $\delta_c(t) \in \mathbb{R}^{\tau n}$ is considered as the estimation residual that measures the difference between the true demand and the estimated value. The available data O_{t-1} can include not only demand data of each region but also weather and traffic conditions that can act as exogenous input of the prediction model. The time index $(t-1)$ of the observation data O_{t-1} used to predict $r_c(t)$ can include purely historical data of demand at each day of time t, streaming demand data of the same day before time t or both. Then we compare our estimation of $r_c(t)$ based on data for each sample $\tilde{r}_c(t)$ of $r_c(t)$ with the true sample vector value $r_c(t)$, and get a corresponding sample of estimation residual as

$$\tilde{\delta}_c(t) = \tilde{r}_c(t) - \hat{r}_c(t). \tag{16}$$

With a subset of training data $\mathcal{S}_{tr}(t) = \{\tilde{r}_c(t), \tilde{O}_{t-1}\}$ that includes both observations $\tilde{O}_{t-1}$ until time t and demand $\tilde{r}_c(t)$ sampled from multiple days, we get an estimation of function $f_r(O_{t-1})$. Then for each subset of testing samples $\mathcal{S}_{te}(t) = \{\tilde{r}_c(t), \tilde{O}_{t-1}\}$, according to Equation (15), we have a set $\mathcal{S}_r(t) = \{\hat{r}_c(t)\}$ as a sample set of estimated or predicted demand and a set $\mathcal{S}_\delta(t) = \{\tilde{\delta}_c(t)\}$ as a sample set of residuals $\delta_c(t)$. We also have the corresponding mean and covariance for the residuals in set $\mathcal{S}_\delta(t)$.

We consider each $\tilde{\delta}_c(t) \in \mathcal{S}_\delta$ as one sample of the random residual vector $\delta_c(t)$. Since $\hat{r}_c(t)$ is deterministic for time index t, the following equations hold (the time index t is omitted for notation convenience), which indicates the relationship between $\mathbb{E}[r_c]$ and $\mathbb{E}[\delta_c]$, Σ_c and Σ_δ, respectively:

$$\begin{aligned} &\mathbb{E}[r_c] - \hat{r}_c = \mathbb{E}[\delta_c], \quad \mathbb{E}[(r_c - \hat{r}_c)(r_c - \hat{r}_c)^T] = \mathbb{E}(\delta_c \delta_c^T), \quad \hat{r}_{c,l} = \hat{r}_c + \hat{\delta}_{c,l},\ \hat{r}_{c,h} = \hat{r}_c + \hat{\delta}_{c,h}, \\ &r_c - \mathbb{E}[r_c] = \hat{r}_c + \delta_c - (\hat{r}_c + \mathbb{E}[\delta_c]), \quad \Sigma_c = \mathbb{E}[(r_c - \mathbb{E}[r_c])(r_c - \mathbb{E}[r_c])^T] = \Sigma_\delta,\ \hat{\Sigma}_c = \hat{\Sigma}_\delta, \end{aligned} \tag{17}$$

where Σ_c and Σ_δ are the unknown true covariance of r_c and δ_c; $\hat{\Sigma}_c$ and $\hat{\Sigma}_\delta$ are the estimated matrices for Σ_c and Σ_δ; and $\delta_{c,l}$ and $\delta_{c,h}$ are the lower and upper bound of the estimation residual, respectively. To build an uncertainty set for the demand distribution $r_c(t)$, the problem is equivalent to describe the distributional uncertainty set as equations and inequalities of statistics of δ_c.

Hence, according to the definition of distributional uncertainty set (10) defined based on the range of mean and covariance of r_c, we define the following problem of constructing distributional uncertainty set for r_c with the estimated support, mean and covariance values of the residual δ_c.

Definition 3.2 (Problem 2). Given a sample set of r_c, for a prediction method f_r, find the values of $\hat{\delta}_{c,l}, \hat{\delta}_{c,h}, \hat{r}_c, \hat{\Sigma}_\delta, \gamma^B_{\delta,1}$, and $\gamma^B_{\delta,2}$, such that with probability at least $1 - \alpha_h$ with respect to the samples, the true distribution of r_c is contained in the following distributional set $\mathcal{F}$:

$$\begin{aligned} &\mathcal{F}(\hat{\delta}_{c,l}, \hat{\delta}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, \gamma_1^B, \gamma_2^B) \\ =&\{r_c \in [\hat{r}_c + \hat{\delta}_{c,l}, \hat{r}_c + \hat{\delta}_{c,h}] : (\mathbb{E}[\delta_c])^T \hat{\Sigma}_\delta^{-1} \mathbb{E}[\delta_c] \leqslant \gamma^B_{\delta,1}, \quad \mathbb{E}(\delta_c \delta_c^T) \leq \gamma^B_{\delta,2} \hat{\Sigma}_\delta\}. \end{aligned} \tag{18}$$

For a general modeling method f_r, we design the following Algorithm 2 to build an uncertainty set of r_c based on bootstrapped estimated support, mean and covariance values of residual δ_c [11, 45].

ALGORITHM 2: Algorithm for constructing distributional sets for a general prediction method

Input: A dataset of spatial-temporal operation records

1. Demand aggregating and sample set partition similar as Algorithm 1

2. Estimate the bootstrapped mean and covariance of the residual vector for all t in one day.

for $j = 1, \ldots, N_B$ **do**

Re-sample $\mathcal{S}^j(I_p) = \{\tilde{r}(I_p)_1, \ldots, \tilde{r}(I_p)_N\}$ from $\mathcal{S}(I_p)$ with replacement, estimate parameters of a prediction function $f_r(O_{t-1})$, calculate the estimation residual set $\mathcal{S}^j_\delta(I_p) = \{\tilde{\delta}(I_p)_i\}$ of all samples based on prediction function f_r, where $\tilde{\delta}(I_p)_i = \tilde{r}(I_p)_i - \hat{r}(I_p)_i$, then get the sample mean $\bar{\delta}^j(I_p)$, sample covariance $\bar{\Sigma}^j_\delta(I_p)$, and $\mathbb{E}^j[\delta_c\delta_c^T](I_p) = \frac{1}{N}\sum_{i=1}^{N}\tilde{\delta}(I_p)_i(\tilde{\delta}(I_p)_i)^T$ of residual for all time steps t.

end for

Get the bootstrapped mean, covariance, and support of the residual vector

$\mathbb{E}[\delta](I_p) = \frac{1}{N_B}\sum_{j=1}^{N_B}\bar{\delta}^j(I_p)$, $\hat{\Sigma}_\delta(I_p) = \frac{1}{N_B}\sum_{j=1}^{N_B}\bar{\Sigma}^j_\delta(I_p)$, $\hat{\delta}_{c,l}(I_p) = min_i\tilde{\delta}(I_p)_i$, $\hat{\delta}_{c,h}(I_p) = max_i\tilde{\delta}(I_p)_i$.

3. Bootstrapping $\gamma^B_{\delta,1}$ and $\gamma^B_{\delta,2}$ for each subset $\mathcal{S}_\delta(t, I_p)$

for $j = 1, \ldots, N_B$ **do**

Get the statistics of residual vector for the jth re-sampled set, $\bar{\delta}^j_c(t, I_p)$, $\hat{\Sigma}_\delta(t, I_p)$, $\mathbb{E}^j[\delta_c\delta_c^T](t, I_p)$ by picking up the corresponding entries for time index t from $\bar{\delta}^j(I_p)$, $\hat{\Sigma}_\delta(I_p)$, and $\mathbb{E}^j[\delta_c\delta_c^T](I_p)$. Calculate:

$\gamma^j_{\delta,1}(t, I_p) = \underset{\gamma_1}{\text{argmin}}[(\bar{\delta}^j_c(t, I_p))^T(\hat{\Sigma}^j_\delta(t, I_p))^{-1}\bar{\delta}^j_c(t, I_p)]$, $\gamma^j_{\delta,2}(t, I_p) = \underset{\gamma_2}{\text{argmin}}[\mathbb{E}^j[\delta_c\delta_c^T](t, I_p) \leqslant \gamma_2\hat{\Sigma}_\delta(t, I_p)]$.

end for

Get the $\lceil N_B(1-\alpha_h)\rceil$-th largest value of $\gamma^j_{\delta,1}(t, I_p)$ and $\gamma^j_{\delta,2}(t, I_p)$, $j = 1, \ldots, N_B$, as $\gamma^B_{\delta,1}(t, I_p)$ and $\gamma^B_{\delta,2}(t, I_p)$, respectively.

Output: Distributionally uncertainty set (18) for prediction function f_r.

Examples of demand prediction models. Taxi demand prediction methodologies designed based on time-series analysis [31] and deep neural network [47] are examples of function (15) that model the spatial-temporal relation of the complex taxi network. For instance, time-series analysis is also widely applied method for predicting demand in resource allocation problems [2, 26, 31], such as autoregressive integrated moving average (ARIMA) model used in Reference [31]. In general, an ARIMA model is denoted with orders (p, d, q), where p is the order of the Auto Regressive term, q is the order of the Moving Average term, d is the minimum number of differencing needed to make the time-series data stationary. If the time-series data are already stationary, then d is supposed to be 0. The coefficients of ARIMA model for function (15) can be fitted by maximum likelihood estimation. Then through the analytic expression of prediction function f_r, we can get the residual $\delta_c(t)$, which covers the random error components by equation $\delta_c(t) = r_c(t) - \hat{r}_c(t)$. By estimating uncertainty of $\delta_c(t)$ via Algorithm 2, we describe how the true demand can deviate from our prediction through repeated data experiments to solve the Problem 2.

3.4 Dynamic Space Partitioning

A grid file [34] is a static data structure that divides the underlying space into a grid of adjacent cells. These cells have equal dimensions. Each cell stores spatial objects (e.g., total number of vehicle requests), within its boundaries. The number of objects in each cell is unbounded. Vehicle balancing approaches based on static spatial partitions has reduced total idle driving distance of all taxis in the network and increased service fairness [28, 29, 50]. However, when we capture the reality of spatial-temporal vehicle balancing problems like the taxi requests we address in this article, we can easily notice that those requests are dynamic. This dynamic nature spans both the space and time. For example, suburbanites tend to go to their business in the metropolitan area in the morning and return in the afternoon. This makes vehicle requests in down-town higher

in the afternoon. This pattern might change depending on the occurrence of other events, (e.g., a state fair, or a football game). This leads to the following two major challenges. (1) It is not only necessary to index those mobility requests, but also to reflect their spatial-temporal dynamic properties on the employed index. (2) It is also a real burden to do that while achieving high efficiency. Since the grid structure enforces a fixed partitioning schema with fixed boundaries regardless of the data distributions, we build our solution based on a different but dynamic index structure, the quad-tree [16].

The quad-tree [16] is a dynamic hierarchical data structure, where the space is recursively decomposed into disjoint equal-sized partitions. Each non-leaf node has 2^d children, where d is the number of dimensions, typically $d = 2$ for modeling the spatial dimensions. For spatial data, a non-leaf node A that covers a rectangle determined by $((x_{min}, y_{min}), (x_{max}, y_{max}))$ is spatially divided into adjacent disjoint nodes: $((x_{min}, y_{min}), (x_{mid}, y_{mid}))$, $((x_{mid}, y_{mid}), (x_{max}, y_{max}))$, $((x_{mid}, y_{min}), (x_{max}, y_{mid}))$, and $((x_{min}, y_{mid}), (x_{mid}, y_{max}))$, where $x_{mid} = avg(x_{min}, x_{max})$ and $y_{mid} = avg(y_{min}, y_{max})$. A leaf node stores a maximum of M points or items that are within its boundaries. If the number of items exceeds the threshold, then the node splits. The quad-tree is unbalanced, but it has good support for skewed data. Practically, real-world spatial datasets are highly skewed.

Both the quad-tree and grid files can be classified as space partitioning techniques, as opposed to data partitioning techniques (e.g., R-tree [21]). The advantage of using a quad-tree to index the demand locations is that a quad-tree provides data-sensitive clustering while partitioning the underlying space and time. It is also efficient to handle data sparseness when some regions have dense data points, (i.e., pick up requests), and others have few. In addition, unlike the static and fixed partitions produced by the grid structure, the partitions produced by quad-tree are dynamic depending on the distribution of the underlying dataset. This means for the same given space if the data points changed, the resultant regions from quad-tree partitioning will vary in shapes, sizes, and numbers. Here, we leverage a $3d$-quad-tree. Two dimensions are used to store the taxi pickup locations and the third represents the time of the day, i.e., the three dimensions for partitioning data include $(latitude, longitude, time - interval)$. The time dimension is divided into fixed intervals to provide a fair comparison with the grid structure, and the $(latitude, longitude)$ dimensions are partitioned according to the non-leaf node split process described above. In experiments we use various time intervals to show the effect of fixed time interval partitioning on the quality of the modeling process, or the uncertainty set of the random demand vector.

In this work, we evaluate a dynamic space partition method using a quad-tree that is compatible with the distributionally robust vehicle balancing problem (8) and the distributional set construction, Algorithm 1. The quad-tree-based method further reduces idle distance according to experiments.

4 COMPUTATIONALLY TRACTABLE FORM

In this section, we derive the main theorem of this work—an equivalent computationally tractable form of the distributionally robust optimization problem (8) via strong duality. Only $J_E(X^{1:\tau}, r_c)$ part of problem (8) is related to the random demand r_c. The objective function of (8) is convex over the decision variables and concave (linear) over the random parameter, with decision variables on the denominators. This form is not a LP or a SDP problem examined by previous work [8, 9, 13]. Hence, the form of $J_D(X^k)$ is the same and the process of deriving a standard convex optimization problem that is equivalent to problem (8) is mainly to analyze the $J_E(r^k, X^{1:\tau})$ part, as shown in the following theorem.

Table 3. New York City Data Used in This Evaluation Section

Taxi Trip Data		
Collecting Period	Data Size	Record Number
01/01/2010–12/31/2013	100 GB	700 million
Data Format		
Trip Information	Time Resolution	Trip Locations
Start and end points	Second	GPS coordinates

Theorem 4.1. *The distributionally robust resource allocation problem (8) with a distributional set (10) is equivalent to the following convex optimization form:*

$$
\begin{aligned}
\textit{min.} \quad & \beta(v+t) + \sum_{k=1}^{\tau} J_D(X^k) \\
\textit{s.t.} \quad & \begin{bmatrix} v + (y_1^+)^T \hat{r}_{c,l} - (y_1^-)^T \hat{r}_{c,h} & \frac{1}{2}(q-y-y_1)^T \\ \frac{1}{2}(q-y-y_1) & Q \end{bmatrix} \succeq 0 \\
& t \geqslant (\gamma_2^B \hat{\Sigma}_c + \hat{r}_c \hat{r}_c^T) \cdot Q + \hat{r}_c^T q + \sqrt{\gamma_1^B} \|\hat{\Sigma}_c^{1/2}(q + 2Q\hat{r}_c)\|_2 \\
& \frac{a_{ik}}{(S_i^k)^\alpha} \leqslant y_i^k, \; y = [y_1^1, y_2^1, \ldots, y_1^\tau, y_2^\tau, \ldots, y_{n^\tau}^\tau]^T, \\
& y_1 = y_1^+ - y_1^-, \quad y_1^+, y_1^-, y \geqslant 0, \quad Q \succeq 0, \quad X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau} \in \mathcal{D}_c.
\end{aligned}
\tag{19}
$$

Proof. See Appendix A.1.

Specifically, with the constraints of problem (8) to represent the constraint $X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau} \in \mathcal{D}_c$ in Equation (19), we have a computationally tractable form for the distributionally robust taxi dispatch problem (8).

5 EVALUATIONS WITH TAXI TRIP DATA

We evaluate the performance of the distributionally robust vehicle balancing framework (8) considered in this work based on four years of taxi trip data in New York City (NYC) [15], by simulations if the algorithm can be implemented for city-level taxi or ride-sharing service. Information for every record includes the GPS coordinators of locations, and the date and time (with precision of seconds) of pick up and drop off locations, the number of taxis involved is about 10K, as summarized in Table 3. We construct distributional uncertainty sets according to Algorithm 1 and Algorithm 2, solve Equation (19), the equivalent convex optimization form of problem (8) to get vehicle balancing solutions across regions. A region is partitioned by either a static equal-area grid or a dynamic quad-tree method, demand is predicted by either directly using the average value of historical data, the ARIMA model, and neural network–based Spatio-Temporal Dynamics Network (STDN) [46] model. After reaching the dispatched regions, we assume that each driver pick up one passenger or passenger group according to the local controller algorithm in Reference [12], and add this inside region idle distance to the across-region idle distance of all taxis for calculating the total idle distance. To compare the average performance of different methods, we use the idea of cross-validation from machine learning. All data are separated as a training subset for constructing the uncertain distribution set and a testing subset for comparing the true vehicle balancing costs and average total idle distance of all testing samples (we use 200 weekday's data for testing). We use taxi operational data for experiments, because this dataset is public, contains information about peoples' mobility pattern, and we show the advantage of vehicle service provided according

Table 4. Comparing Thresholds γ_1^B and γ_2^B for Different N_B and Dimensions of r_c

		γ_1^B	γ_2^B
$N_B = 10$	$n = 50, \tau = 2$	0.739	5.24
$N_B = 100$	$n = 50, \tau = 2$	0.368	2.47
$N_B = 1000$	$n = 50, \tau = 3$	0.013	1.56
$N_B = 5000$	$n = 50, \tau = 3$	0.012	1.49

to our framework by bridging the gap between demand data to a balanced supply. The evaluation results validate the efficiency improvement for taxi service system in a city by using the algorithm proposed in this work. The application of our framework can be taxis, autonomous mobility-on-demand systems [50], or bike sharing [37], depending on what kind of demand data are available. Balancing autonomous vehicles with a predicted demand probability distribution in a city outperforms other vehicle dispatch algorithms such as nearest-neighbor or collaborative taxi dispatch algorithm in the literature, as compared based on NYC data [50]. Though not considering any prediction uncertainties, applying the estimation of future demand to make decisions still improves mobility service systems' performance. Hence, we only compare our method that considers uncertainties of demand probability distributions with the method of using the predicted demand model as the true demand model in this section.

How does the number of samples affect the distribution set: We partition the map of NYC into different number of equal-area grids to compare the values of γ_1^B and γ_2^B of Algorithm 1. Algorithm 1 captures information about the support, the first and second moments of the random demand, $\alpha_h = 0.1$. We show the value of γ_1^B and γ_2^B with different values of sample number N_B and the dimension of r_c (τn) in Table 4. For the same region partition and prediction time horizon τ, when the values of γ_1^B, γ_2^B are smaller, the volume of the uncertainty set is smaller, the demand prediction is more accurate. Comparing the first two lines in Table 4, when the value of N_B is increased, values of γ_1^B and γ_2^B are reduced, which means the volume of the distributional set is smaller and the demand prediction is more accurate. For a large enough N_B, the value of τn does not affect γ_1^B and γ_2^B much, as shown in the third and fourth lines of Table 4.

5.1 Performance of Distributionally Robust Solutions

We compare three vehicle balancing methods, include the distributionally robust framework (8), the robust method of Reference [30], and the non-robust method with the average requests number during each unit time as the demand model [29] (equivalent to the passenger arrival rate of a queueing model in each unit time [50, 51]). The optimal cost of each method is a weighted sum of the demand–supply ratio mismatch error and estimated total idle driving distance. For each testing sample data r^k, we use the demand–supply ratio mismatch error (Equation (4)) to measure how well the optimal solution balances the vehicle toward the true supply. The idle distance of each taxi between two trips with passengers is approximated as the distance between one drop-off event and the following-up pick-up event.

We compare the average costs of cross-validation tests in Figure 2. The average costs show the performance when we applying the optimal solution of each method to balance taxis under all testing samples of r_c aggregated from weekdays' data from 5 pm to 8 pm. The region partition method is a static equal-area grid partition and the distributional uncertainty set is constructed via Algorithm 1. The minimum average cost of a second-order-cone (SOC) robust solution [30] is close to the average cost of the distributionally robust solutions of Equation (19). They both use the

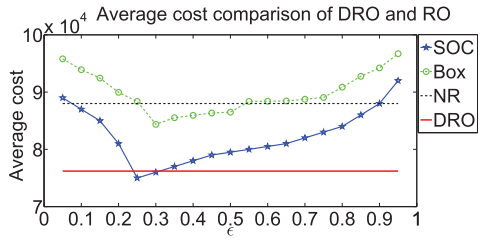


Fig. 2. The average cost of cross-validation tests for the distributionally robust solutions via solving Equation (19) ("DRO" line), two types of uncertainty sets of the robust solutions (lines SOC and Box) and non-robust solutions.

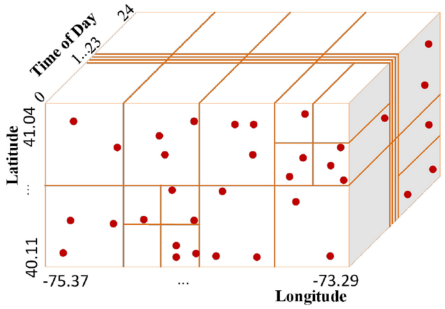


Fig. 3. One-hour Interval Quad-Tree for Demand.

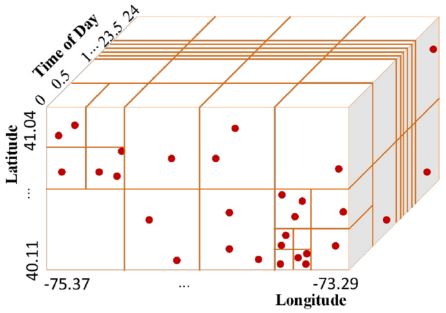


Fig. 4. Half-hour Interval Quad-Tree for Demand.

first and second moments information of the random demand. In particular, the average demand–supply ratio mismatch error is reduced by 28.6%, and the average total idle driving distance is reduced by 10.05%, the weighted-sum cost of the two components is reduced by 10.98% compared with non-robust solutions.

In Figure 2, robust solutions with the box type of uncertainty set and the SOC type of uncertainty set provide a desired level of probabilistic guarantee—the probability that an actual dispatch cost under the true demand vector being smaller than the optimal cost of the robust vehicle balancing solutions is greater than $(1 - \epsilon)$. However, they do not directly minimize the average performance of the solutions and we need to tune the value of ϵ and test the average cost. The horizontal lines show the average cost of distributionally robust solutions and non-robust solutions, since these costs are irrelevant to ϵ. The average cost of solutions of Equation (19) is always smaller than costs of robust balancing solutions based on the box type uncertainty set, which only uses information about the range of demand at each region. This result indicates that the second order moment information of the random variable should be included for modeling the uncertainty of the demand and calculating an optimal solutions. The distributionally robust method (Equation (8)) directly provides a better guarantee for the average performance under uncertain demand, and the SOC robust method designed in Reference [30] provides a probabilistic guarantee for the single-point worst-case performance of the demand.

5.2 Grid Partition Compared with Quad-Tree Partition

As provided in Figure 3, the quad-tree covers from −75.37 to −73.29 for longitude and from 40.11 to 41.04 for the latitude in New York city area. The time in this figure is divided into 1-hour intervals. Figure 4 gives a snapshot for the quad-tree partitions when we change the time dimension to be

Table 5. Comparison of γ_1^B and γ_2^B Values with a Dynamic Quad-tree Partition Method and a Static Equal-area Grid Partition for Different Time Intervals t, Where Unit "h" Means Hour and "m" Means Minute

	Grid	Quad-Tree	Change Rate
$t = 2$ h, γ_1^B	0.016	0.021	31.25%
$t = 2$ h, γ_2^B	1.73	2.05	18.50%
$t = 1$ h, γ_1^B	0.0130	0.0110	−15.38%
$t = 1$ h, γ_2^B	1.56	1.35	−13.46%
$t = 50$ m, γ_1^B	0.0128	0.0107	−16.41%
$t = 50$ m, γ_2^B	1.53	1.32	−13.73%
$t = 40$ m, γ_1^B	0.0125	0.0102	−18.40%
$t = 40$ m, γ_2^B	1.49	1.26	−15.44%
$t = 30$ m, γ_1^B	0.0121	0.0095	−21.49%
$t = 30$ m, γ_2^B	1.46	1.21	−17.12%
$t = 20$ m, γ_1^B	0.0119	0.0120	0.84%
$t = 20$ m, γ_2^B	1.41	1.48	4.96%
$t = 15$ m, γ_1^B	0.0120	0.0123	2.50%
$t = 15$ m, γ_2^B	1.40	1.50	7.14%

Change Rate is calculated via $(V_{Quad-Tree} - V_{Grid})/V_{Grid}$, where $V_{\{\cdot\}}$ means the values in the corresponding column.

in 30-minute intervals, which is different from the one-hour quad-tree in Figure 3. The red dots in both figures represent taxi-requests distributed over the space and time of the day. We fixed the time interval as 2 hours down to 15 minutes as shown in Table 5, and get different partitions on (longitude, latitude) dimensions. We then use demand vectors after these partitions to calculate the uncertain set of probability distributions for 5–8 pm of weekdays, to show the effect of time-interval length on the quality of the quad-tree.

Table 5 shows the comparison of γ_1^B and γ_2^B values with a dynamic quad-tree partition method and a static simple equal-area grid partition method for different values of time interval t. For the same region partition and prediction time horizon τ, when the values of γ_1^B, γ_2^B are smaller, the volume of the uncertainty set is smaller, the demand prediction is more accurate. After region partition and pick-up events aggregation, the demand of each hour is predicted by directly calculating the average of all training data. For the following experiments, we use the same values of $\tau = 4$, $N_s = 1000$, and $\alpha_h = 0.1$. According to the results of $t = 2$ h and $t = 1$ h shown in Table 5 for weekdays' demand data from 5 pm to 8 pm, we conclude that the granularity of time also affects demand prediction accuracy. For $t = 1$ h, with static equal-area grid partition and the average requests number during each unit time as the demand model, the Mean Average Percentage Error (MAPE) is 32.6%. When the length of one time instant is appropriate, the quad tree partition method improves the accuracy of demand prediction. The volume of uncertainty sets shrink, with

Table 6. Comparison of Average Total Idle Distance (Weekdays 5 pm–8 pm) with Distributionally Robust Dispatch Solutions by Solving Equation (19) (Equivalent Form of Equation (8))

Region division	Grid	Quad-tree	Change rate
$t = 1$ h	7.63×10^4	6.62×10^4	−13.1%
$t = 30$ min	6.84×10^4	5.47×10^4	−20.0%

smaller γ_1^B and γ_2^B values when we use the quad tree partition method, according to the results when $t = 50$ m, $t = 40$ m, and $t = 30$ m. However, when the length of one time instant is too short, predicting demand based on the quad tree method is worse than that based on the simple equal-area grid partition. The values of γ_1^B and γ_2^B for time lengths $t = 20$ m and $t = 15$ m show that the values of γ_1^B and γ_2^B are increased by quad tree partition.

In Table 6, we compare the average total idle distance with distributionally robust dispatch solutions by solving Equation (19) (equivalent form of Equation (8)), based on an equal-area grid region partition and a quad-tree region partition method. For a fixed time interval of 1 hour, the quad-tree region partition method can reduce average total idle distance by 13.1%, and for a fixed 30-minute interval, the reduction rate is 20%. This is about a 30% or 60 million miles reduction of total idle distance or 8 million dollar cost reduction annually for all taxis in NYC, compared with the method of balancing taxis in the city with average requests number that does not consider demand uncertainties. By partitioning the regions with a data-sensitive quad-tree method from the beginning, the distributional set better captures the spatial-temporal properties of demand. The performance of the data-driven vehicle balancing method is then significantly improved.

5.3 Time-series Demand Prediction and Distributional Uncertainty Sets

In this subsection, we show the demand prediction error at different times of one weekday using the ARIMA time-series model, the demand distributional uncertainty sets constructed based on Algorithm 2 based on grid and quad-tree region partition methods, and considering demand prediction uncertainties reduces the total idle distance of all taxis in NYC compared with service provided by not considering prediction uncertainties. Though some recent developed complicated models in the literature such as [31, 46, 47] provide a relatively more accurate model, the MAPE can still be 15.5% by deep neural network [46, 47] or 22% by assembled time-series model [31] with fine tuned parameters. To make reliable vehicle dispatch decisions, the prediction error should not be neglected. The focus of this work is to show the benefit of the proposed dynamic region partitioning and distributionally robust optimization methods, instead of comparing demand prediction models. Hence, we use the widely applied ARIMA model to predict demand in this subsection and the deep neural network–based STDN demand prediction model in the next subsection. We show that our proposed framework works for different types of demand prediction model when building the uncertainty sets, and the optimal solution guarantees the average cost of a taxi dispatch system.

We first compare the true demand and predicted demand via ARIMA model for different time of weekdays in Figures 6, 7, and 8. Figure 5 shows a static equal-area grid region partition for Manhattan and the positions of Regions 13, 24, and 42. Downtown and midtown Regions 13 and 24 are relatively busier especially during daytime compared with Region 42, and have relatively small prediction errors than the not busy upper town Region 42. When demand is predicted by a time-series model and uncertainty sets are constructed by Algorithm 2, Table 7 shows the comparison of $\gamma_{\delta,1}^B$ and $\gamma_{\delta,2}^B$ values with a dynamic quad-tree partition method and a static simple equal-area grid partition method for different values of time interval t. When the values are smaller, the volume of

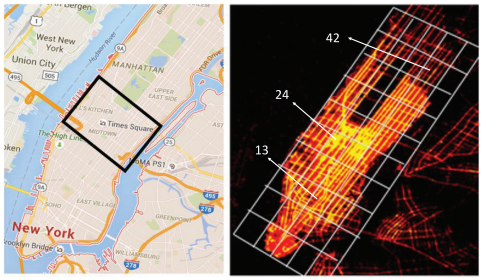


Fig. 5. Heat map of demand in Manhattan area: lighter means more demand. Regions 13, 24, and 42 via grid partition (50 regions in total) are denoted in the right figure.

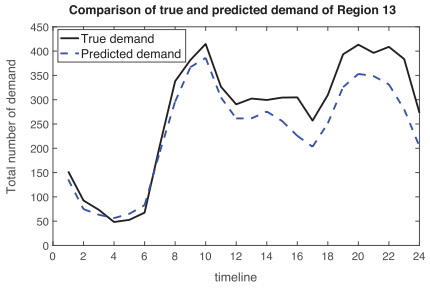


Fig. 6. Comparison of the true demand and demand predicted by ARIMA model in region 13 in one day. The MAPE at each hour is from 5.23% to 19.57%.

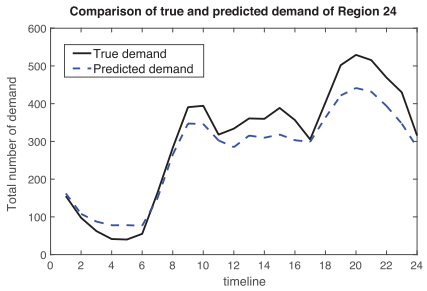


Fig. 7. Comparison of the true demand and demand predicted by ARIMA model in region 24 in one day. The MAPE at each hour is from 4.68% to 18.35%.

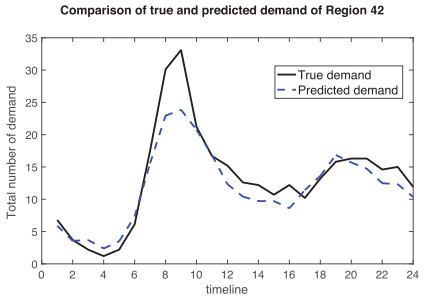


Fig. 8. Comparison of the true demand and demand predicted by ARIMA model in region 42 in one day. The MAPE at each hour is from 6.79% to 26.14%.

the uncertainty set is smaller. For other parameters of the experiments, we use $\tau = 4$ for weekdays 5–8 pm, $N_s = 1{,}000$, and $\alpha_h = 0.1$ for all comparison.

When the length of one time instant is appropriate, the quad tree partition method improves the accuracy of demand prediction. The volume of uncertainty sets shrink, with smaller $\gamma^B_{\delta,1}$ and $\gamma^B_{\delta,2}$ values when we use the quad tree partition method, according to the results when $t = 50$ m, $t = 40$ m, and $t = 30$ m. However, when the length of one time instant is too long such as $t = 2$ h and $t = 1$ h, or too short, such as $t = 20$ m and $t = 15$ m, predicting demand based on the quad tree method is worse than that based on the simple equal-area grid partition. The phenomenon of accuracy changes with the granularity of time step length has also been revealed in other prediction methods [25, 31], that neither a too fine or too coarse prediction time step length works, and 30 minutes or 1 hour is often selected. The values of $\gamma^B_{\delta,1}$ and $\gamma^B_{\delta,2}$ are increased by the dynamic region partition method. Table 7 also shows the generality and compatibility of the dynamic quad-tree region partition method with different demand prediction models.

When demand is predicted by ARIMA, we compare the average total idle distance with distributionally robust dispatch solutions by solving Equation (19) (equivalent form of Equation (8)), based on equal-area grid region partition and quad-tree region partition methods in Table 8. For a fixed time interval of 1 hour, quad-tree region partition method can reduce average total idle distance by 11.05%, and for a fixed 30-minutes interval, the reduction rate is 19.60%. When we use the grid method for region partitioning, compared with vehicle dispatch decisions not considering

Table 7. Comparison of $\gamma^B_{\delta,1}$ and $\gamma^B_{\delta,2}$ Values with a Dynamic Quad-tree Partition Method and a Static Equal-area Grid Partition for Different Time Intervals t, Where Demand Is Predicted by ARIMA Model, Unit "h" Means Hour, and "m" Means Minute

	Grid	Quad-Tree	Change Rate
$t = 2$ h, $\gamma^B_{\delta,1}$	0.015	0.019	26.67%
$t = 2$ h, $\gamma^B_{\delta,2}$	1.67	2.02	20.96%
$t = 1$ h, $\gamma^B_{\delta,1}$	0.0127	0.0106	−16.54%
$t = 1$ h, $\gamma^B_{\delta,2}$	1.53	1.31	−14.38%
$t = 50$ m, $\gamma^B_{\delta,1}$	0.0125	0.0103	−17.60%
$t = 50$ m, $\gamma^B_{\delta,2}$	1.51	1.30	−14.00%
$t = 40$ m, $\gamma^B_{\delta,1}$	0.0123	0.0101	−17.89%
$t = 40$ m, $\gamma^B_{\delta,2}$	1.47	1.23	−16.33%
$t = 30$ m, $\gamma^B_{\delta,1}$	0.0119	0.0092	−22.70%
$t = 30$ m, $\gamma^B_{\delta,2}$	1.45	1.19	−17.93%
$t = 20$ m, $\gamma^B_{\delta,1}$	0.0120	0.0121	0.83%
$t = 20$ m, $\gamma^B_{\delta,2}$	1.42	1.47	3.52%
$t = 15$ m, $\gamma^B_{\delta,1}$	0.0121	0.0123	1.70%
$t = 15$ m, $\gamma^B_{\delta,2}$	1.43	1.51	5.59%

Change Rate is calculated via $(V_{Quad-Tree} - V_{Grid})/V_{Grid}$, where $V_{\{\cdot\}}$ means the values in the corresponding column.

Table 8. Comparison of Average Total Idle Distance (Weekdays 5 pm–8 pm) with Distributionally Robust Dispatch Solutions by Solving Equation (19) (Equivalent form of Equation (8)), Demand Predicted by the ARIMA Model

Region division	Grid	Quad-tree	change rate
$t = 1$ h	7.15×10^4	6.36×10^4	−11.05%
$t = 30$ m	6.58×10^4	5.29×10^4	−19.60%

the demand prediction error by ARIMA, the average total idle driving distance is reduced by 7.68%, though the ARIMA model is more accurate than the bootstrapped average demand model we use in Table 6 (the idle distance reduction rate of the distributionally robust compared with non-robust solutions is 10.05%, when using grid method for region partition and the average requests number during each unit time as the demand model [29], equivalent to the passenger arrival rate of a queueing model in each unit time [50, 51]). Hence, using a data-sensitive quad-tree method from the beginning for region partition and a distributional set better captures the spatial-temporal correlated uncertainties of demand helps to reduce the total idle distances, even the demand prediction model is a relatively accurate time-series model. They together provides a 27.27% mileage

Table 9. Comparison of $\gamma^B_{\delta,1}$ and $\gamma^B_{\delta,2}$ Values with a Dynamic Quad-tree Partition Method and a Static Equal-area Grid Partition for Different Time Intervals t, Where Demand Is Predicted by STDN Model, Unit "h" Means Hour, and "m" Means Minute

	Grid	Quad-Tree	Change Rate
$t = 2$ h, $\gamma^B_{\delta,1}$	0.014	0.017	21.43%
$t = 2$ h, $\gamma^B_{\delta,2}$	1.61	1.92	19.25%
$t = 1$ h, $\gamma^B_{\delta,1}$	0.0113	0.0099	−12.39%
$t = 1$ h, $\gamma^B_{\delta,2}$	1.38	1.21	−12.32%
$t = 50$ m, $\gamma^B_{\delta,1}$	0.0108	0.0097	−10.19%
$t = 50$ m, $\gamma^B_{\delta,2}$	1.31	1.17	−10.69%
$t = 40$ m, $\gamma^B_{\delta,1}$	0.0101	0.0090	−10.89%
$t = 40$ m, $\gamma^B_{\delta,2}$	1.26	1.13	−10.32%
$t = 30$ m, $\gamma^B_{\delta,1}$	0.0095	0.0086	−9.47%
$t = 30$ m, $\gamma^B_{\delta,2}$	1.20	1.09	−9.17%
$t = 20$ m, $\gamma^B_{\delta,1}$	0.0103	0.0105	1.94%
$t = 20$ m, $\gamma^B_{\delta,2}$	1.29	1.36	5.43%
$t = 15$ m, $\gamma^B_{\delta,1}$	0.0106	0.0108	1.89%
$t = 15$ m, $\gamma^B_{\delta,2}$	1.31	1.38	5.34%

Change Rate is calculated via $(V_{Quad-Tree} - V_{Grid})/V_{Grid}$, where $V_{\{\cdot\}}$ means the values in the corresponding column.

reduction compared with grid-region partition, ARIMA demand prediction without considering model uncertainties.

5.4 Neural Network–based STDN Prediction and Distributional Uncertainty Sets

We also evaluate our algorithm using a recently developed neural network–based demand prediction algorithm STDN [46]. The STDN model utilizes the novel 2D local convolutional layers, LSTM units, and an attention mechanism on the sequential data. The STDN builds upon the Deep Multi-View Spatio-Temporal Network [47], which introduces the local convolutional layer. Using STDN to predict demand, The Root Mean Square Error is 21.94% and the MAPE is 15.5%. We show the performance of our proposed distributionally robust vehicle balancing algorithm and quad-tree dynamic region partition method based on STDN in this subsection.

We compare the average total idle distance with distributionally robust dispatch solutions by solving Equation (19) (equivalent form of Equation (8)), based on equal-area grid region partition and quad-tree region partition methods in Table 10. For a fixed time interval of 1 hour, quad-tree region partition method can reduce average total idle distance by 10.80%, and for 30-minute intervals, the reduction rate is 16.69% compared with grid region partition method. When we use the grid method for region partitioning, compared with vehicle dispatch decisions not considering

Table 10. Comparison of Average Total Idle Distance (Weekdays 5 pm–8 pm) with Distributionally Robust Dispatch Solutions by Solving (19) (Equivalent form of Equation (8)), Demand Predicted by the STDN Model

Region division	Grid	Quad-tree	change rate
$t = 1$ h	6.76×10^4	6.03×10^4	−10.80%
$t = 30$ m	6.17×10^4	5.14×10^4	−16.69%

the demand prediction error by STDN, the average total idle driving distance is reduced by 6.59%, though the STDN model is more accurate than most earlier demand prediction model (the idle distance reduction rate of the distributionally robust compared with non-robust solutions is 10.05%, when using grid method for region partition and the average requests number during each unit time as the demand model [29], equivalent to the passenger arrival rate of a queueing model in each unit time [50, 51]). Hence, using a data-sensitive quad-tree method from the beginning for region partition and a distributional set better captures the spatial-temporal correlated uncertainties of demand, and helps to reduce the total idle distances, even the demand prediction model is relatively accurate. They together provide a 23.28% mileage reduction compared with grid-region partition and STDN demand prediction without considering model uncertainties.

6 CONCLUSION

Vehicle balancing strategies coordinate vehicles to fairly serve customers from a systemwide perspective, and reduce total idle distance to serve the same number of customers compared with strategies without balancing. However, the uncertain probability distribution of demand predicted from data affects the performance of solutions and has not been considered by previous work. In this article, we design a data-driven distributionally robust vehicle balancing method to minimize the worst-case average cost under uncertainties about the probability distribution of demand. Then we design efficient algorithms to construct a set of distributions given a spatial-temporal demand dataset and different demand prediction models, and leverage a quad-tree dynamic region partition method to better capture the dynamic properties of the random demand. We prove an equivalent computationally tractable form of the distributionally robust problem under the constructed distributional set. Evaluations show that for a prediction model of using the average of historical value, the average demand–supply ratio mismatch error is reduced by 28.6%, and the average total idle driving distance is reduced by 10.05% compared with non-robust solutions, if using the distributionally robust algorithm desgined in this work. With quad-tree dynamic region partitions, the average total idle distance is reduced by 20% more. For a more accurate time-series model, the average total idle distance is reduced by 7.68% by considering demand prediction uncertainties with static grid region partition, and is reduced by 19.60% more with the quad-tree dynamic region partition. In the future, we will design hierarchical vehicle balancing strategies for heterogeneous vehicle networks.

A APPENDIX

A.1 Proof of Theorem 4.1

Proof. We have $\frac{a_{ik}}{(S_i^k)^\alpha} > 0$ and $r_c \geqslant 0$ by the definitions of J_E in Equation (5) and the demand model, then for any vector $y \in \mathbb{R}^{n_c}$, $y = [y_1^1, y_2^1, \ldots, y_1^\tau, y_2^\tau, \ldots, y_{n^\tau}^\tau]^T$ that satisfies $0 < \frac{a_{ik}}{(S_i^k)^\alpha} \leqslant y_i^k$, we also have $0 \leqslant \sum_{k=1}^{\tau} \sum_{i=1}^{n^k} \frac{a_{ik} r_i^k}{(S_i^k)^\alpha} \leqslant y^T r_c$, and the second inequality strictly holds when all

$\frac{a_{ik}r_i^k}{(S_i^k)^\alpha} = y_i^k$, for $i = 1, \ldots, n^k$, $k = 1, \ldots, \tau$. The constraints of problem (8) are independent of r_c, hence, for any r_c, the minimization problem

$$\min_{X^k} \quad \beta \sum_{k=1}^{\tau} \sum_{i=1}^{n^k} \frac{a_{ik} r_i^k}{(S_i^k)^\alpha} + \sum_{k=1}^{\tau} J_D(X^k), \qquad \text{s.t} \quad X^{[1,\tau]}, S^{[1,\tau]}, V^{[2,\tau]}, O^{[2,\tau]} \in \mathcal{D}_c$$

is equivalent to

$$\begin{aligned} \min_{X^k} \quad & \beta y^T r_c + \sum_{k=1}^{\tau} J_D(X^k) \\ \text{s.t.} \quad & \frac{a_{ik}}{(S_i^k)^\alpha} \leqslant y_i^k, \ y \in \mathbb{R}^{n_c}, \\ & y = [y_1^1, y_2^1, \ldots, y_1^\tau, y_2^\tau, \ldots, y_{n^\tau}^\tau]^T, \quad X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau} \in \mathcal{D}_c. \end{aligned} \tag{20}$$

In this proof, we use the objective function of problem (20). In particular, only the part of $y^T r_c$ is related to r_c, and we first consider the following maximization problem:

$$\max_{r_c \sim F, F \in \mathcal{F}} \mathbb{E}[y^T r_c]. \tag{21}$$

By the definition of problem (8) and problem (20), only the objective function includes the random vector r_c, and is concave of r_c, convex of X^k for $k = 1, \ldots, \tau$. The distributional set $\mathcal{F}$ constructed by Algorithm 1, the domain of y, $X^{1:\tau}$, $S^{1:\tau}$, $V^{2:\tau}$, and $O^{2:\tau}$ are convex, closed, and bounded sets. Hence, problem (21) satisfies the conditions of Lemma 1 in Reference [14], and the maximum expectation value of $y^T r_c$ for any possible $r_c \sim F$, where $F \in \mathcal{F}$ equals the optimal value of the problem,

$$\begin{aligned} \min_{Q,q,v,t} \quad & v + t \\ \text{s.t.} \quad & v \geqslant y^T r_c - r_c^T Q r_c - r_c^T q, \quad \forall r_c \in [\hat{r}_{c,l}, \hat{r}_{c,h}], \quad Q \succeq 0 \\ & t \geqslant (\gamma_2^B \hat{\Sigma}_c + \hat{r}_c \hat{r}_c^T) \cdot Q + \hat{r}_c^T q + \sqrt{\gamma_1^B} \|\hat{\Sigma}_c^{1/2}(q + 2Q\hat{r}_c)\|_2. \end{aligned} \tag{22}$$

Hence, we first analytically find the optimal value of problem (22). Note that the first constraint about v is equivalent to $v \geqslant f(r_c^*, y)$, where $f(r_c^*, y)$ is the optimal value of the following problem:

$$\max_{r_c} \quad y^T r_c - r_c^T Q r_c - r_c^T q, \qquad \text{s.t.} \quad \hat{r}_{c,l} \leqslant r_c \leqslant \hat{r}_{c,h}. \tag{23}$$

For a positive semi-definite Q, problem (23) is convex. The Lagrangian of (23) under the constraint $y_1^+, y_1^- \geqslant 0$ is $\mathcal{L}(r_c, y_1^+, y_1^-) = y^T r_c - r_c^T Q r_c - r_c^T q + (y_1^+ - y_1^-)^T r_c - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}$. When $Q \succeq 0$, the inverse of matrix Q is convex, the supreme value of the Lagrangian is calculated via taking the partial derivative over r_c, let $\nabla_{r_c} \mathcal{L} = 0$, $y_1 = y_1^+ - y_1^-$, $y_1^+, y_1^- \geqslant 0$, and a convex constraint

$$\sup_{r_c} \mathcal{L}(r_c, y_1^+, y_1^-) = \frac{1}{4}(q - y - y_1)^T Q^{-1}(q - y - y_1) - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}.$$

Then the first inequality constraint of problem (22) for any $\hat{r}_{c,l} \leqslant r_c \leqslant \hat{r}_{c,h}$ is equivalent to

$$v \geqslant \frac{1}{4}(q - y - y_1)^T Q^{-1}(q - y - y_1) - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}.$$

By Schur complement, the above constraint is

$$\begin{bmatrix} v + (y_1^+)^T \hat{r}_{c,l} - (y_1^-)^T \hat{r}_{c,h} & \frac{1}{2}(q - y - y_1)^T \\ \frac{1}{2}(q - y - y_1) & Q \end{bmatrix} \succeq 0.$$

Together with other constraints, the equivalent convex optimization form of problem (8) is problem (19). □

REFERENCES

[1] B. L. Smith, B. M. Williams, and R. K. Oswald. 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transport. Res. C: Emerg. Technol.* 10, 4 (2002), 303–321.

[2] S. S. Jones, R. S. Evans, T. L. Allen, A. Thomas, P. J. Haug, S. J. Welch, and G. L. Snow. 2009. A multivariate time series approach to modeling and forecasting demand in the emergency department. *J. Biomed. Inf.* 42, 1 (2009), 123–139.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Fut. Gener. Comput. Syst.* 29, 7 (2013), 1645–1660.

[4] S. Ali, A. A. Maciejewski, H. J. Siegel, and Jong-Kook Kim. 2004. Measuring the robustness of a resource allocation. *IEEE Trans. Parallel Distrib. Syst.* 15, 7 (2004), 630–641.

[5] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc. Natl. Acad. Sci. U.S.A.* 114, 3 (2017), 462–467.

[6] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fathi, M. Xu, M. M. Dhanya, N. Mitrovic, and P. Jaillet. 2014. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Trans. Intell. Transport. Syst.* 15, 2 (2014), 797–804.

[7] Rajesh Krishna Balan, Khoa Xuan Nguyen, and Lingxiao Jiang. 2011. Real-time trip information service for a large taxi fleet. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys'11)*. 99–112.

[8] A. Ben-Tal and A. Nemirovski. 1998. Robust convex optimization. *Math. Operat. Res.* 23, 4 (1998), 769–805.

[9] Dimitris Bertsimas, Vishal Bupta, and Nathan Kallus. 2018. Data-driven robust optimization. *Math. Program.* 167 (2018), 235–292.

[10] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, New York, NY.

[11] Efron Bradley. 1979. Bootstrap methods: Another look at the jackknife. *Ann. Stat.* 7, 1–26 (1979).

[12] Ximing Chen, Fei Miao, George Pappas, and Victor Preciado. [n.d.]. Hierarchical data-driven vehicle dispatch and ride-sharing. In *Proceedings of the IEEE 56th Conference on Decision and Control*.

[13] Francesco A. Cuzzola, Jose C. Geromel, and Manfred Morari. 2002. An improved approach for constrained robust model predictive control. *Automatica* 38, 7 (2002), 1183–1189.

[14] Erick Delage and Yinyu Ye. 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operat. Res.* 58, 3 (2010), 595–612.

[15] Brian Donovan and Daniel B. Work. 2015. Using coarse GPS data to quantify city-scale transportation system resilience to extreme events. In *Proceedings of the Transportation Research Board Annual Meeting*.

[16] Raphael A. Finkel and Jon Louis Bentley. 1974. Quad trees a data structure for retrieval on composite keys. *Acta Inf.* 4, 1 (1974), 1–9.

[17] Raghu Ganti, Mudhakar Srivatsa, and Tarek Abdelzaher. 2014. On limits of travel time predictions: Insights from a new york city case study. In *Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems (ICDCS'14)*. 166–175.

[18] Yanfeng Geng and C. G. Cassandras. 2014. New "Smart parking" system based on resource allocation and reservations. *IEEE Trans. Intell. Transport. Syst.* 14, 3 (2014), 1129–1139.

[19] Joel Goh and Melvyn Sim. 2010. Distributionally robust optimization and its tractable approximations. *Operat. Res.* 58, 4-part-1 (2010), 902–917.

[20] Donald Gross. 2008. *Fundamentals of Queueing Theory*. John Wiley & Sons.

[21] Antonin Guttman. 1984. R-trees: A dynamic index structure for spatial searching. *SIGMOD Rec.* 14, 2 (June 1984), 47–57.

[22] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen. 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transport. Res. C* 18, 4 (2010), 568–583.

[23] Lam Kiet, Krichene Walid, and Bayen Alexandre. 2016. On learning how players learn: Estimation of learning dynamics in the routing game. In *Proceedings of the 7th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'16)*. 1–10.

[24] X. Li, M. Li, Y. Gong, X. Zhang, and J. Yin. 2016. T-DesP: Destination prediction based on big trajectory data. *IEEE Trans. Intell. Transport. Syst.* 17, 8 (August 2016), 2344–2354. DOI: https://doi.org/10.1109/TITS.2016.2518685

[25] Lingbo Liu, Zhilin Qiu, Guanbin Li, Qing Shun Wang, Wanli Ouyang, and Liang Lin. 2019. Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Trans. Intell. Transport. Syst.* 20, 10 (Oct. 2019), 3875–3887.

[26] A. Lorca and A. Sun. 2015. Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind. In *Proceedings of the Power Energy Society General Meeting*. 1–1.

[27] J. Lv, Q. Li, Q. Sun, and X. Wang. 2018. T-CONV: A convolutional neural network for multi-scale taxi trajectory prediction. In *Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp'18)*. 82–89. DOI: https://doi.org/10.1109/BigComp.2018.00021

[28] F. Miao, S. Han, S. Lin, and G. J. Pappas. 2015. Robust taxi dispatch under model uncertainties. In *Proceedings of the 54th IEEE Conference on Decision and Control (CDC'15)*. 2816–2821.

[29] Fei Miao, Shuo Han, Shan Lin, John A. Stankovic, Hua Huang, Desheng Zhang, Sirajum Munir, Tian He, and George J. Pappas. 2016. Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach. *IEEE Trans. Autom. Sci. Eng.* 13, 2 (April 2016), 463–478.

[30] F. Miao, S. Han, S. Lin, Q. Wang, J. A. Stankovic, A. Hendawi, D. Zhang, T. He, and G. J. Pappas. 2019. Data-driven robust taxi dispatch under demand uncertainties. *IEEE Trans. Contr. Syst. Technol.* 27, 1 (2019), 175–191.

[31] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. 2013. Predicting taxi-passenger demand using streaming data. *IEEE Trans. Intell. Transport. Syst.* 14, 3 (Sept. 2013), 1393–1402.

[32] Abood Mourad, Jakob Puchinger, and Chengbin Chu. 2019. A survey of models and algorithms for optimizing shared mobility. *Transport. Res. B: Methodological* 123 (May 2019), 323–346. DOI: https://doi.org/10.1016/j.trb.2019.02.003

[33] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris. 2011. Smarter cities and their innovation challenges. *Computer* 44, 6 (2011), 32–39.

[34] Jürg Nievergelt, Hans Hinterberger, and Kenneth C Sevcik. 1984. The grid file: An adaptable, symmetric multikey file structure. *ACM Trans. Database Syst.* 9, 1 (1984), 38–71.

[35] B. P. G. Van Parys, D. Kuhn, P. J. Goulart, and M. Morari. 2016. Distributionally robust control of constrained stochastic systems. *IEEE Trans. Automat. Control* 61, 2 (2016), 430–442.

[36] Marco Pavone, Stephen L Smith, Emilio Frazzoli, and Daniela Rus. 2012. Robotic load balancing for mobility-on-demand systems. *Int. J. Rob. Res.* 31, 7 (June 2012), 839–854.

[37] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari. 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE Trans. Intell. Transport. Syst.* 15, 4 (Aug. 2014), 1567–1578. DOI: https://doi.org/10.1109/TITS.2014.2303986

[38] Jasper Schuijbroek, Robert Hampshire, and Willem-Jan van Hoeve. 2017. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research* 257, 3 (March 2017), 992–1004.

[39] Wen Shen, Vahan Babushkin, Zeyar Aung, and Wei Lee Woon. 2013. An ensemble model for day-ahead electricity demand time series forecasting. In *Proceedings of the 4th International Conference on Future Energy Systems (e-Energy'13)*. 51–62.

[40] K. Spieser, S. Samaranayake, W. Gruel, and E. Frazzoli. 2016. Shared vehicle mobility-on-demand systems: A fleet operators guide to rebalancing empty vehicles. In *Proceedings of the Transportation Research Board 95th Annual Meeting*, Vol. 5. 100–111.

[41] Håkan Terelius and Karl Henrik Johansson. 2015. An efficiency measure for road transportation networks with application to two case studies. In *Proceedings of the IEEE Conference on Decision and Control (CDC'15)*. 5149–5155.

[42] Jana Tumova, Sertac Karaman, Calin Belta, and Daniela Rus. 2016. Least-violating planning in road networks from temporal logic specifications. In *Proceedings of the 7th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'16)*. Article 17, 9 pages.

[43] A. Wallar, M. Van Der Zee, J. Alonso-Mora, and D. Rus. 2018. Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'18)*. 4539–4546.

[44] S. Weikl and K. Bogenberger. 2013. Relocation strategies and algorithms for free-floating car sharing systems. *IEEE Intell. Transport. Syst. Mag.* 5, 4 (2013), 100–111.

[45] H. White and J. Racine. 2001. Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates. *IEEE Trans. Neur. Netw.* 12, 4 (July 2001), 657–673. DOI: https://doi.org/10.1109/72.935080

[46] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. *arXiv:1803.01254*. Retrieved from https://arxiv.org/abs/1803.01254.

[47] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Annual Conference in Artificial Intelligence (AAAI'18)*.

[48] Chenyang Yuan, Jérôme Thai, and Alexandre M. Bayen. 2016. ZUbers against ZLyfts apocalypse: An analysis framework for DoS attacks on mobility-as-a-service systems. In *Proceedings of the 7th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'16)*.

[49] Desheng Zhang, Tian He, Shan Lin, S. Munir, and J. A. Stankovic. 2014. Dmodel: Online taxicab demand model from big sensor data in a roving sensor network. In *Proceedings of the 2014 IEEE International Conference on Big Data (BigData'14)*. 152–159.

[50] Rick Zhang and Marco Pavone. 2016. Control of robotic mobility-on-demand systems. *Int. J. Rob. Res.* 35, 1–3 (Jan. 2016), 186–203.

[51] Rick Zhang, Federico Rossi, and Marco Pavone. 2016. Model predictive control of autonomous mobility-on-demand systems. In *Proceedings of the International Conference on Robotics and Automation (ICRA'16)*.

Received January 2020; revised July 2020; accepted August 2020