Data-Driven Distributionally Robust Vehicle Balancing Using Dynamic Region Partitions*

Fei Miao
University of Pennsylvania
miaofei@seas.upenn.edu

Shuo Han
University of Pennsylvania
hanshuo@seas.upenn.edu

Abdeltawab M. Hendawi
University of Virginia
hendawi@virginia.edu

Mohamed E Khalefa
University of Alexandria
khalefa@alexu.edu.eg

John A. Stankovic
University of Virginia
stankovic@virginia.edu

George J. Pappas
University of Pennsylvania
pappasg@seas.upenn.edu

ABSTRACT

With the transformation to smarter cities and the development of technologies, a large amount of data is collected from sensors in real-time. This paradigm provides opportunities for improving transportation systems' performance by allocating vehicles towards mobility predicted demand proactively. However, how to deal with uncertainties in 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 an efficient algorithm for constructing uncertainty sets of random demand probability distributions, and leverage a quad-tree dynamic region partition method for better capturing the dynamic spatial-temporal properties of the uncertain demand. We then prove equivalent computationally tractable form for numerically solving the distributionally robust problem. We evaluate the performance of the data-driven vehicle balancing framework based on four years of taxi trip data for New York City. 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 partition method, compared with vehicle balancing solutions based on static region partitions without considering demand uncertainties. This is about 60 million miles or 8 million dollars 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;*

KEYWORDS

Distributionally Robust Vehicle Balancing, Dynamic Region Partition, Average Idle Distance, Uncertain Demand

ACM Reference format:
Fei Miao, Shuo Han, Abdeltawab M. Hendawi, Mohamed E Khalefa, John A. Stankovic, and George J. Pappas. 2017. Data-Driven Distributionally Robust Vehicle Balancing Using Dynamic Region Partitions. In *Proceedings of The 8th ACM/IEEE International Conference on Cyber-Physical Systems, Pittsburgh, PA USA, April 2017 (ICCPS 2017),* 11 pages.
DOI: http://dx.doi.org/10.1145/3055004.3055024

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 [26]. With the development of technologies, we are able to collect, store, and analyze a large amount of data efficiently [17]. Intelligent transportation system is one example, in which sensing data collected in real time provides us opportunities for understanding spatial-temporal human mobility patterns. For instance, traffic speed [2], travel time [3, 19], passengers' demand model of taxi network [25], and road transportation network efficiency [32] are inferred and measured.

Researchers have been working on various approaches to improve the performance of transportation systems. Resilience properties of dynamical networks are analyzed for distributed routing policies [7, 8]. Smart parking systems that allocate and reserve parking space for drivers [14], routing and motion planning problems for mobile systems [20, 33] 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 system-wide coordination perspective. Vehicle balancing methods reduce the number of vehicles needed to serve all passengers with mobility-on-demand systems [29, 35, 36] and bike-sharing systems [30], or reduce customers' waiting time [29, 36] and taxis' total idle distance [23] with the same number of empty vehicles. However, the limit knowledge we have about demand and mobility patterns [13] affect the performance of vehicle balancing strategies, and making real-time decisions under demand model uncertainties is still a challenging and unsolved task. Although robust optimal solution shows its advantage in worst-case scenarios compared with non-robust approaches [1, 21, 22], there is still trade-off between

*This work was supported by NSF CPS-1239152, NSF CNS-1239224,US Department of Transportation through the UTC program, and TerraSwarm.

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 ACM 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.
ICCPS 2017, Pittsburgh, PA USA
© 2017 ACM. 978-1-4503-4965-9/17/04…$$15.00
DOI: http://dx.doi.org/10.1145/3055004.3055024

the system's average performance and the worst-case performance with a probabilistic guarantee [24].

In this work, we integrate the process of gathering actionable information from data and designing decision-making objectives and constraints for vehicle balancing problems of ride-sharing service, to ensure real-time resource-allocating efficiency from the perspective of expected cost. It is difficult to obtain an explicit true probability distribution of the random demand purely based on data without prior knowledge, therefore, we minimize the expected vehicle balancing cost under a set of possible probability distributions of demand learned from data. Distributionally robust optimization techniques have been developed for minimizing expected cost under the worst-case probability distributions of random parameters for linear programming (LP), semi-definite programming (SDP) problems [10, 15], and linear controllers [28] in the literature. But there is no real-time distributionally robust vehicle balancing approach for complex transportation networks with uncertain demand probability distributions predicted from data yet.

We design a computationally tractable distributionally robust dynamic vehicle balancing method under uncertainties about the probability distributions of demand. An efficient algorithm for constructing an uncertainty set of the probability distributions based on data is proposed by utilizing 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 prove an equivalent convex optimization form of the non LP or SDP form of distributionally robust vehicle balancing problem, and guarantee both average performance of the system and computational tractability. Finally, we evaluate the average vehicle balancing costs of the distributionally robust solutions based on real data.

The contributions of this work are

- We take explicitly the ambiguity of demand probability distribution into account when minimizing vehicle balancing cost. We design a data-driven dynamic distributionlly robust vehicle balancing model to optimize the expected cost over the worst-case distribution of demand, and analyze its applications in taxi dispatch, autonomous mobility-on-demand and bike balancing. Previous vehicle balancing work either focuses on one specific probability distribution or aims to find a robust solution for a single value of worst-case demand.
- For the first time, we design a quad-tree dynamic region partition method and an efficient algorithm to construct an uncertainty set of probability distributions that better captures the spatial-temporal correlations of demand uncertainties based on data.
- We derive a computationally tractable form to numerically solve the distributionally robust problem.
- 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, and show that 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.

The rest of the paper is organized as follows. The distributionally robust vehicle balancing problem is proposed in Section 2. An efficient algorithm for constructing distributional uncertainty sets based on spatial-temporal demand data 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 data set 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 regions. By considering possible probability distributions of demand predicted from data, we take explicitly the ambiguity of demand probability distributions to guarantee the average system performance. Previous work either assumes an explicit demand distribution [29, 30, 35, 36] or aims to find a robust vehicle balancing solution for a single value (not a probability distribution) of worst-case demand [22–24, 29] for static spatial region partitions. The generalization of the vehicle balancing problem formulation in this work is also explained in Subsection 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 is calculated in a receding horizon process, and at time t, empty vehicles are allocated towards demand with time index $(t, t+1, \ldots, t+\tau-1)$ respectively. 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 rebalancing solution, and 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 arrive at its dispatched region, a local controller will assign the vehicle to pick up a passenger existing in this region's request queue according to greedy algorithms (e.g., shortest path). 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. Examples of receding horizon resource allocation applications include economic power dispatch [21], taxi dispatch [23], autonomous mobility-on-demand [36], etc.

2.1 Problem Formulation

We assume that 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. Each region j has $r_j^k \geqslant 0$ predicted total amount of demand (e.g., number of passengers for a mobility-on-demand system) during time k, where $j = 1, \ldots, n^k$, $k = 1, \ldots, \tau$. We consider $r^k \in \mathbb{R}^{n^k}$ 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 sequences 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., $r_c \sim F^*$.

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 vacant vehicles sent from region i to region j (or node i to node j) at time k according to 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 a cost function 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^*. Moreover, problem (1) is computationally demanding, not suitable for a large-scale dynamic supply balancing problem in smart cities in general. 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 learning and data-driven problems [10, 15]. In Section 3 we will design an algorithm of calculating the set $\mathcal{F}$ such that $F^* \in \mathcal{F}$ with a high probability. We then consider the following distributionally robust problem to minimize the worst-case expected cost as a robust form of problem (1). In the rest of this section we will define concrete forms of objective function and constraints.

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

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 at 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 $\{X^1, \ldots, X^k\}$

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

where $P_v^k, P_o^k, Q_v^k, Q_v^k \in \mathbb{R}^{n^k \times n^{k+1}}$ are region transition matrices: $P_{v,ji}^k$ ($P_{o,ji}^k$) describe 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, $Q_{v,ji}^k$ ($Q_{o,ji}^k$) describe 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 region transition matrices are learned from historical data, and satisfy

$$\sum_{j=1}^{n^k} P_{v,ij}^k + P_{o,ij}^k = 1, \quad \sum_{j=1}^{n^k} Q_{v,ij}^k + Q_{o,ij}^k = 1.$$

Balancing the supply-demand ratio across the network is one type of service quality metric in power resource allocation [21], taxi dispatch [23] and autonomous mobility on demand systems [35]. Hence, we aim to minimize the 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 random parameters r^k, not computationally tractable as an objective function for (2). Hence, in this work, we consider a service quality function J_E

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

where $a_{ik} > 0, i = 1, \ldots, n^k, k = 1, \ldots, \tau$ are positive constants denoting region priorities, $\alpha > 0$ is a power parameter that is designed according to the service requirement. In particular, When $a_{ik} = 1$, $i = 1, \ldots, n^k$, $k = 1, \ldots, \tau$, $\alpha > 0$ is a close to 0, the objective function (5) approximates the objective (4) [22], and minimizing (5) means a balancing vehicle objective. With the definition of S_i^k as (3), 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 service quality function (5), we also consider minimizing the costs (such as idle distance) by sending vacant vehicles according to X^k. Given a spatial network structure during time k, we define $W^k \in \mathbb{R}^{n^k \times n^k}$ as the weight matrix that describes the cost of sending one vehicle among regions for time k according to the 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

Parameters of (8)	Description
n^k and τ	the number of regions at time k and model predicting time horizon
$r_c \in \mathbb{R}^{n_c} \sim F^*, F^* \in \mathcal{F}$	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 region 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

Table 1: Parameters and variables of taxi dispatch problem (8)**.**

should be shorter than normal hours. 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 [2], [31], the distance from region i to region j is $dist_{ij}$. We denote a structural constraint matrix $M^k \in \mathbb{R}^{n^k \times n^k}$, such that $M^k_{ij} = 0$ when $dist_{ij} \leqslant m^k$, and $M^k_{ij} = 1$ otherwise. Then the following constraint

$$X^k \circ M^k = 0, \quad X^k_{ij} \geqslant 0 \tag{7}$$

indicates a solution satisfies that $X^k_{ij} = 0$ for $dist_{ij} > m^k$, $i, j = 1, \ldots, n^k$. Here $\circ$ means Schur or entry-wise product. Both $J_D(X^k)$ in (6) and constraint (7) are linear of X^k.

We aim to balance vehicles with minimum idle distance, and define a weight parameter β of two objectives J_D in (6) and J_E in (5). With constraints (3) and (7), we consider the following distributionally robust vehicle balancing problem under uncertain probability distributions of random demand

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

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 directly related to the random demand r^k, balancing cost J_D and all the constraints are linear of the variables and not functions of r^k. Hence, we only need to find an equivalent convex form for J_E under $F \in \mathcal{F}$.

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 [16]. Considering a balanced demand-supply ratio is considering to balance the average number of waiting customers intuitively. Robotic mobility-on-demand systems [34, 35] usually assume a queuing model to describe the passenger arrival rate at region i is λ^k_i. When calculating the arrival rate for one time interval from historical data, λ^k_i equals the total number of requests appearing in one time interval, or r^k_i in this work. Mean and covariance of the estimation of λ^k_i still exist when calculating this arrival rate λ^k_i 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^k_i 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 in order to provide a certain level of service satisfaction [30]. While adjusting the range of demand-supply ratio or supply-demand ratio back and forth is computationally expansive, when we find a feasible solution of (8), the demand-supply ratio of each region should not be far away from the global demand-supply ratio, and fall 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 (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 $O^k_{l,i}$ as the number of vehicles with capacity C_l before dispatch at region i, and $X^k_{l,ij}$ as the number of vehicles that should go from region i to region j. Then the total number of available seats or supply is $S^k_i = \sum_{l=1}^{4} C_l \left(O^k_{l,i} + \sum_{j=1}^{n^k} X^k_{l,ji} - \sum_{j=1}^{n^k} X^k_{l,ij} \right)$. With this number S^k_i, objective function J_E defined as (5) is still concave in r^k, convex in X^k_l, $l = 1, 2, 3, 4$. The balancing cost function (6), constraints about region transition (3) and idle distance bound (7) can be modified accordingly and still be convex of decision variables. Under this scenario, with a modified definition of total supply at each region, the vehicle balancing model (8) is generalizable for carpooling or heterogeneous capacity vehicle balancing problems. With periodically re-balancing vehicles every hour or 30-minutes, a lower level matching between passengers and vehicles within each region will assign one vehicle to several requests according to its capacity. A hierarchical carpooling framework with higher layer distributionally robust vehicle balancing and a lower layer routing or matching process is a venue for future work.

3 EFFICIENT DISTRIBUTIONAL SET CONSTRUCTION ALGORITHM

We design an efficient algorithm 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. While theoretical bound of the distributional set is too conservative in practice, empirical estimates according to confidence regions of hypothesis testings are acceptable in portfolio management problems [5, 10]. However, vehicle trip or trajectory data is 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 of r_c to develop an efficient construction algorithm for set $\mathcal{F}$. 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. For each t, samples from N days $\tilde{r}_c(d_1, t), \tilde{r}_c(d_2, t), \ldots, \tilde{r}_c(d_N, t)$ are independent.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 a hypothesis testing H_0 given a data set of r_c: given mean μ_0 and covariance Σ_0, test statistics γ_1, γ_2, with probability at least $1 - \alpha_h$, the random vector

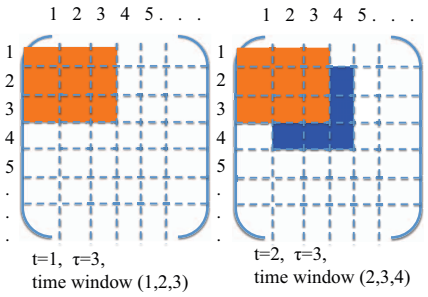

Figure 1: The process of calculating $\hat{\Sigma} \in \mathbb{R}^{n\times n}$, $n = \sum_{t=1}^{K} n^t$ when receding time horizon. When index moves from $t = 1$ to $t = 2$, only entries in matrix $\hat{\Sigma}$ shown in blue are new and necessary for calculating $\hat{\Sigma}_c(t), t = 2$.

r_c satisfies that [10]

$$\begin{aligned} H_0 :&(\tilde{r}_c - \mu_0)^T \Sigma_0^{-1} (\tilde{r}_c - \mu_0) \leqslant \gamma_1, \\ &(\tilde{r}_c - \mu_0)(\tilde{r}_c - \mu_0)^T \preceq \gamma_2 \Sigma_0. \end{aligned} \tag{9}$$

Without prior knowledge about the support, the true mean, covariance, constructing set $\mathcal{F}$ based on data is an inverse process of a hypothesis testing—calculating threshold values such that (9) is an acceptable hypothesis by the data set. The problem of constructing $\mathcal{F}$ is defined as:

Definition 3.1. **Problem 1.** Given a dataset of r_c, find the values of $\hat{r}_{c,l}, \hat{r}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, \gamma_1^B$ and γ_2^B, 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 $\mathcal{F}$

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

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 [6] estimations of $\hat{r}_{c,l}, \hat{r}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, \gamma_1^B, \gamma_2^B$ for every $r_c(t)$, $t = 1, 2, \ldots, K$, that makes H_0 in (9) acceptable and consistent with the data.

3.2 Reducing Computational Complexity

Because $\mathcal{F}(t)$ is a function of time index t, the dimension of $\hat{r}_c, \hat{\Sigma}_c$ is decided by the number of dynamic regions and prediction horizon, which can be large for spatial-temporal transportation data collected in smart cities. 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 during time $(t+1, t+2, \ldots, t+\tau-1)$. Hence, instead of always repeating the process of calculating a mean and covariance value for τ time slots together 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 up the corresponding components needed to construct $\hat{r}_c(t)$ and $\hat{\Sigma}_c(t)$ for each index t.

$\tilde{r}_c(d_l, t, I_p)$	one sample of $r_c(t)$ according to sub-dataset I_p, records of 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_c
$\hat{r}_{c,l}$, $\hat{r}_{c,h}$	the estimated lower and upper bound of vector r_c
γ_1^B, γ_2^B	the bootstrapped thresholds for accepting hypothesis testing (9)
α_h	significance level of a hypothesis testing

Table 2: Parameters of Algorithm 1.

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 component for each index t, the process is: 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

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

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_1-th to the b_2-th rows and b_2-th to the b_1-th 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 data set is the arriving time of each customer at different service nodes of a network, then the total number of customer appeared 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 data set as subsets first.

For step 3(1), the process of picking components from the mean and covariance matrices of the whole day demand is

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

For the j-th 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 (10), we get $\gamma_1^j(t, I_p)$ by the following equation

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

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

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

ALGORITHM 1: Algorithm for constructing distributional sets

Input: A data set of spatial-temporal demand

1. Demand aggregating and sample set partition

Partition space, aggregate demand of each region for each time t, cluster demand vector samples according to categorical information I_p, and denote $\mathcal{S}(I_p)$, $\mathcal{S}(t, I_p)$ as a sample set of the whole day demand and demand at time t of category I_p, $p = 1, \ldots, P$, respectively.

2. Bootstrapping mean and covariance matrix

A significance level $0 < \alpha_h < 1$, the number of bootstrap time $N_B \in \mathbb{Z}_+$.

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

Re-sample $\mathcal{S}^j(I_p) = \{\tilde{r}(d_1, I_p), \ldots, \tilde{r}(d_N, I_p)\}$ from $\mathcal{S}(I_p)$ with replacement, calculate the 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 (11).

end for

Get the bootstrapped mean covariance, and support of the whole day demand vector ($i = 1, \ldots, Kn$)

$\hat{r}(I_p) = \frac{1}{B}\sum_{j=1}^{B} \bar{r}^j(I_p)$, $\hat{\Sigma}(I_p) = \frac{1}{B}\sum_{j=1}^{B} \bar{\Sigma}^j(I_p)$,

$\hat{r}_{i,l}(I_p) = min_d \tilde{r}_i(d, I_p)$, $\hat{r}_{i,h}(I_p) = max_d \tilde{r}_i(d, I_p)$, for all samples $\tilde{r}(d, I_p)$ in the subset $\mathcal{S}(I_p)$.

3. Bootstrapping γ_1^B and γ_2^B for each time index t

for each subset $\mathcal{S}(t, I_p)$ **do**

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

(1) Get the mean and covariance vector for the j-th re-sampled set, $\hat{r}_c(t, I_p)$, $\hat{\Sigma}_c(t, I_p)$, $\bar{r}_c^j(t, I_p)$, $\bar{\Sigma}_c^j(t, I_p)$ as (12).

(2). Get $\gamma_1^j(t, I_p)$ and $\gamma_2^j(t, I_p)$ by (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.

end for

Output: Distributionally uncertainty sets (10).

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

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

3.3 Dynamic Space Partitioning

A grid file [27] 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 level [22, 23, 35]. However, when we capture the reality of spatial and spatial-temporal vehicle balancing problems like the taxi requests we address in this paper,

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 downtown 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 [12].

The quad-tree [12] is known as 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 which are within its boundaries. If the number of items exceeds the threshold, the node splits. The quad-tree is unbalanced, but it has good support for skewed data. Practically, real-world spatial data sets 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 [18]). 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 in handling data sparseness which occurs 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 data set. 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 time 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 the experiments we use various values of time intervals to show the effect of fixed time interval partitioning on the quality of the modeling process, or the uncertainty of the distribution function 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 linear programming (LP) or a semi-definite programming (SDP) problem examined by previous work [4, 5, 9]. Hence, the form of $J_D(X^k)$ keeps the same and the process of deriving a standard convex optimization problem that equivalent to problem (8) is mainly to analyze the $J_E(r^k, X^{1:\tau})$ part, as shown in the following theorem.

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

$$
\begin{aligned}
\textit{min.}\quad & \beta(v+t) + \sum_{k=1}^{\tau} J_D(X^k) \\
\textit{s.t.}\quad & \begin{bmatrix} v + (y_1^+)^T \hat{r}_{c,l} - (y_1^-)^T \hat{r}_{c,h} & \frac{1}{2}(q - y - y_1)^T \\ \frac{1}{2}(q - y - y_1) & Q \end{bmatrix} \succeq 0 \\
& t \geqslant (\gamma_2^B \hat{\Sigma}_c + \hat{r}_c \hat{r}_c^T) \cdot Q + \hat{r}_c^T q \\
& \qquad + \sqrt{\gamma_1^B} \|\hat{\Sigma}_c^{1/2}(q + 2Q\hat{r}_c)\|_2 \\
& \frac{a_{ik}}{(S_i^k)^\alpha} \leqslant y_i^k,\ y = [y_1^1, y_2^1, \ldots, y_1^\tau, y_2^\tau, \ldots, y_{n^\tau}^\tau]^T, \\
& y_1 = y_1^+ - y_1^-, \quad y_1^+, y_1^-, y \geqslant 0, \quad Q \succeq 0 \\
& X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau} \in \mathcal{D}_c.
\end{aligned}
\tag{15}
$$

Proof. See Appendix 7.1.

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 (15), 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) [11]. 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, as summarized in Table 3. We construct distributional uncertainty sets according to Algorithm 1, solve (15), the equivalent convex optimization form of problem (8) to get vehicle balancing solutions across regions. After reaching the dispatched regions, we assume that drivers pick up the nearest passenger, and add this inside region idle distance to the across-region idle distance of all taxis for calculating the total idle distance. We use taxi operational data for experiments because this data set is public, contains

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

Table 3: New York city data used in this evaluation section.

		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
$N_B = 5000$	$n = 50, \tau = 6$	0.012	1.49

Table 4: Comparing thresholds γ_1^B and γ_2^B for different N_B and dimensions of r_c

information about peoples' mobility pattern, and we show the advantage of vehicle service provided according to our framework by bridging the gap between demand data to a balanced supply. The application of our framework does not need to be restricted to taxis, it can be autonomous mobility-on-demand systems [36], or bike sharing, depending on what kind of demand data is 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 [36]. 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. 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. For a large enough N_B, the value of τn does not affect γ_1^B and γ_2^B much.

5.1 Performance of Distributionally Robust Solutions

To compare the average performance of different methods, we use the idea of cross-validation from machine learning. All data is separated as a training subset for constructing the uncertain distribution set and a testing subset for comparing the true vehicle balancing costs for each time of testing. We compare three vehicle balancing methods, include the distributionally robust framework (8), the robust method of [24], and the non-robust method with the average

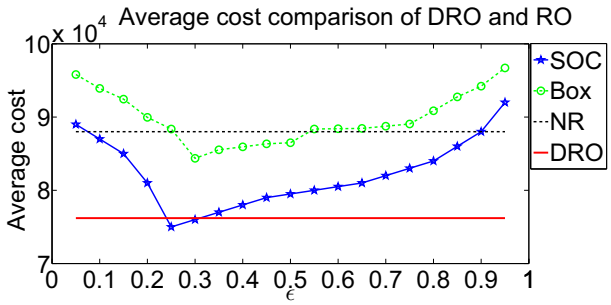

Figure 2: The average cost of cross-validation tests for the distributionally robust solutions via solving (15) ("DRO" line), two types of uncertainty sets of the robust solutions (lines SOC and Box) and non-robust solutions.

requests number during each unit time as the demand model [23] (equivalent to the passenger arrival rate of a queueing model in each unit time [35, 36]). 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 r^k from the data set, we use the demand-supply ratio mismatch error (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 5pm-8pm. The minimum average cost of a second-order-cone (SOC) robust solution [24] is close to the average cost of the distributionally robust solutions of (15). They both use the 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 (15) 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 (8) directly provides a better guarantee for the average performance under uncertain demand, and the SOC robust method designed in [24] provides a

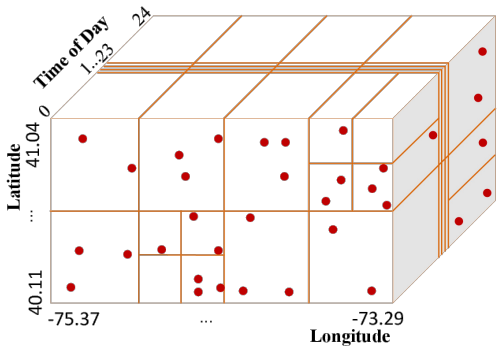

Figure 3: One-hour Interval Quad-Tree for Taxi Pickups

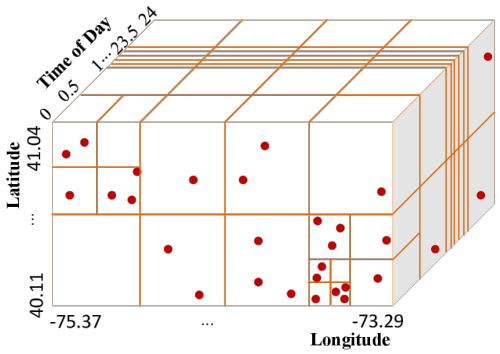

Figure 4: Half-hour Interval Quad-Tree for Taxi Pickups

probabilistic guarantee for the worst-case performance at a single point of the demand space.

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 one-hour intervals. Figure 4 gives a snapshot for the quad-tree partitions when we change the time dimension to be 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 5-8pm 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. When the values are smaller, the volume of the uncertainty set is smaller. 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 $\alpha_h = 0.1$.

	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	0.120	0.84%
t = 20 m, γ_2^B	1.41	1.48	4.96%
t = 15 m, γ_1^B	0.0120	0.123	2.50%
t = 15 m, γ_2^B	1.40	1.50	7.14%

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

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

Table 6: Comparison of average total idle distance (weekdays 5pm-8pm) with distributionally robust dispatch solutions by solving (15) (equivalent form of (8)).

According to the results of t = 2 h and t = 1 h shown in Table 5 for weekdays' demand data from 5pm to 8pm, we conclude that the granularity of time also affects demand prediction accuracy. 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 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 (15) (equivalent form of (8)), based on equal-area grid region partition and quad-tree region partition methods. For a fixed time interval of 1 hour, quad-tree region partition method can reduce average total idle distance by 13.1%, and for a fixed 30-minutes interval, the reduction rate is 20%. This is about a 30% or 60 million miles reduction of total idle distance or 8 million 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.

6 CONCLUSION

Vehicle balancing strategies coordinate vehicles to fairly serve customers from a system-wide 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 paper, 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 an efficient algorithm to construct a distributional set given a spatial-temporal demand data set, 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 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. With quad-tree dynamic region partitions, the average total idle distance is reduced by 20% more. In the future, we will design hierarchical vehicle balancing strategies for heterogeneous vehicle networks.

REFERENCES

[1] S. Ali, A. Maciejewski, H. Siegel, and J.-K. Kim. Measuring the robustness of a resource allocation. *IEEE Transactions on Parallel and Distributed Systems*, 15(7):630–641, July 2004.

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

[3] R. K. Balan, K. X. Nguyen, and L. Jiang. Real-time trip information service for a large taxi fleet. In *Proceedings of the 9th MobiSys*, pages 99–112, 2011.

[4] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[5] D. Bertsimas, V. Bupta, and N. Kallus. Data-driven robust optimization. *Operations Research*, (arXiv: 1401.0212), 2015.

[6] E. Bradley. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1-26), 1979.

[7] G. Como, K. Savla, D. Acemoglu, M. A. Dahleh, and E. Frazzoli. Robust distributed routing in dynamical networks–part i: Locally responsive policies and weak resilience. *IEEE TAC*, 58(2):317–332, Feb 2013.

[8] G. Como, K. Savla, D. Acemoglu, M. A. Dahleh, and E. Frazzoli. Robust distributed routing in dynamical networks–part ii: Strong resilience, equilibrium selection and cascaded failures. *IEEE TAC*, 58(2):333–348, Feb 2013.

[9] F. A. Cuzzola, J. C. Geromel, and M. Morari. An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183–1189, 2002.

[10] E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[11] B. Donovan and D. B. Work. Using coarse gps data to quantify city-scale transportation system resilience to extreme events. In *Transportation Research Board Annual Meeting*, July 2015.

[12] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.

[13] R. Ganti, M. Srivatsa, and T. Abdelzaher. 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, pages 166–175, 2014.

[14] Y. Geng and C. Cassandras. New "smart parking" system based on resource allocation and reservations. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1129–1139, 2014.

[15] J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-part-1):902–917, 2010.

[16] D. Gross. *Fundamentals of queueing theory*. John Wiley & Sons, 2008.

[17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645 – 1660, 2013.

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

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

[20] L. Kiet, K. Walid, and B. Alexandre. On learning how players learn: Estimation of learning dynamics in the routing game. In *Proceedings of the 7th ICCPS*, pages 1–10, Los Alamitos, CA, USA, 2016. IEEE Computer Society.

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

[22] F. Miao, S. Han, S. Lin, and G. J. Pappas. Robust taxi dispatch under model uncertainties. In *54th CDC*, pages 2816–2821, 2015.

[23] F. Miao, S. Han, S. Lin, J. A. Stankovic, H. Huang, D. Zhang, S. Munir, T. He, and G. J. Pappas. Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach. *IEEE Transactions on Automation Science and Engineering*, 13:463–478, April 2016.

[24] F. Miao, S. Han, S. Lin, Q. Wang, J. Stankovic, A. Hendawi, D. Zhang, T. He, and G. J. Pappas. Data-driven robust taxi dispatch under demand uncertainties. *submitted, preprint can be found at http://www.seas.upenn.edu/ miaofei/taxi-journal.pdf*.

[25] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, Sept 2013.

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

[27] J. Nievergelt, H. Hinterberger, and K. C. Sevcik. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems (TODS)*, 9(1):38–71, 1984.

[28] B. P. G. V. Parys, D. Kuhn, P. J. Goulart, and M. Morari. Distributionally robust control of constrained stochastic systems. *IEEE Transactions on Automatic Control*, 61(2):430–442, Feb 2016.

[29] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus. Robotic load balancing for mobility-on-demand systems. *Int. J. Rob. Res.*, 31(7):839–854, June 2012.

[30] J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. *To appear, European Journal of Operational Research*, 2016.

[31] B. L. Smith, B. M. Williams, and R. K. Oswald. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4):303 – 321, 2002.

[32] H. Terelius and K. H. Johansson. An efficiency measure for road transportation networks with application to two case studies. In *CDC*, pages 5149–5155, 2015.

[33] J. Tumova, S. Karaman, C. Belta, and D. Rus. Least-violating planning in road networks from temporal logic specifications. In *Proceedings of the 7th ICCPS*, pages 17:1–17:9, 2016.

[34] C. Yuan, J. Thai, and A. M. Bayen. Zubers against zlyfts apocalypse: An analysis framework for dos attacks on mobility-as-a-service systems. In *Proceedings of the 7th ICCPS*, 2016.

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

[36] R. Zhang, F. Rossi, and M. Pavone. Model predictive control of autonomous mobility-on-demand systems. In *ICRA*, 2016.

7 APPENDIX

7.1 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 (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

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

is equivalent to

$$\begin{aligned} \min_{X^k} \quad & \beta y^T r_c + \sum_{k=1}^{\tau} J_D(X^k) \\ \text{s.t.} \quad & \frac{a_{ik}}{(S_i^k)^\alpha} \leqslant y_i^k,\ y \in \mathbb{R}^{n_c}, \\ & y = [y_1^1, y_2^1, \ldots, y_1^\tau, y_2^\tau, \ldots, y_{n^\tau}^\tau]^T, \\ & X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau} \in \mathcal{D}_c \end{aligned} \tag{16}$$

In this proof, we use the objective function of problem (16). In particular, only the part of $y^T r_c$ is related to r_c, and we first consider the following maximization problem

$$\max_{r_c \sim F, F \in \mathcal{F}} \mathbb{E}[y^T r_c] \tag{17}$$

By the definition of problem (8) and problem (16), 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 (17) satisfies the conditions of Lemma 1 in [10], 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

$$\begin{aligned} \min_{Q,q,v,t} \quad & v + t \\ \text{s.t.} \quad & v \geqslant y^T r_c - r_c^T Q r_c - r_c^T q, \quad \forall r_c \in [\hat{r}_{c,l}, \hat{r}_{c,h}] \\ & t \geqslant (\gamma_2^B \hat{\Sigma}_c + \hat{r}_c \hat{r}_c^T) \cdot Q + \hat{r}_c^T q \\ & \qquad + \sqrt{\gamma_1^B} \|\hat{\Sigma}_c^{1/2} (q + 2Q\hat{r}_c)\|_2 \\ & Q \succeq 0. \end{aligned} \tag{18}$$

Hence, we first analytically find the optimal value of problem (18). 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

$$\begin{aligned} \max_{r_c} \quad & y^T r_c - r_c^T Q r_c - r_c^T q \\ \text{s.t.} \quad & \hat{r}_{c,l} \leqslant r_c \leqslant \hat{r}_{c,h}. \end{aligned} \tag{19}$$

For a positive semi-definite Q, the optimal solution of problem (19) exists. The Lagrangian of (19) under the constraint $y_1^+, y_1^- \geqslant 0$ is

$$\begin{aligned} \mathcal{L}(r_c, y_1^+, y_1^-) =& y^T r_c - r_c^T Q r_c - r_c^T q + (y_1^+ - y_1^-)^T r_c \\ & - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}. \end{aligned}$$

When $Q \succeq 0$, the supreme value of the Lagrangian is calculated via taking the partial derivative over r_c, let $\Delta_{r_c} \mathcal{L} = 0$, and

$$\begin{aligned} \sup_{r_c} \mathcal{L}(r_c, y_1^+, y_1^-) =& \frac{1}{4}(q - y - y_1)^T Q^{-1} (q - y - y_1) \\ & - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}, \\ y_1 =& y_1^+ - y_1^-, \quad y_1^+, y_1^- \geqslant 0. \end{aligned}$$

Then the first inequality constraint of problem (18) for any $\hat{r}_{c,l} \leqslant r_c \leqslant \hat{r}_{c,h}$ is equivalent to

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

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 (15). □