Data-Driven Distributionally Robust Vehicle Balancing Using Dynamic Region Partitions

FEI MIAO, University of Connecticut
XIMING CHEN, University of Pennsylvania
SHUO HAN, University of Illinois at Chicago
ABDELTAWAB M. HENDAWI, University of Virginia
MOHAMED E KHALEFA, University of Alexandria
JOHN A. STANKOVIC, University of Virginia
VICTOR PRECIADO, University of Pennsylvania
GEORGE J. 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. 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;

Additional Key Words and Phrases: Distributionally Robust Vehicle Balancing, Dynamic Region Partition, Average Idle Distance, Uncertain Demand Sets

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

Authors’ addresses: Fei Miao, University of Connecticut, fei.miao@uconn.edu; Ximing Chen, University of Pennsylvania, xmingch@seas.upenn.edu; Shuo Han, University of Illinois at Chicago, hanshuo@uic.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; Victor Preciado, University of Pennsylvania, preciado@seas.upenn.edu; George J. Pappas, University of Pennsylvania, 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 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.

© 2017 Association for Computing Machinery.
XXX-XXXX/2017/11-ART $15.00
https://doi.org/10.1145/nnnnnn.nnnnnnn

, Vol. 1, No. 1, Article . Publication date: November 2017.
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 [28]. With the development of technologies, we are able to collect, store, and analyze a large amount of data efficiently [3]. 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 [5], travel time [6, 21], passengers’ demand model of taxi network [27], and road transportation network efficiency [35] 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 [10, 11]. Smart parking systems that allocate and reserve parking space for drivers [17], routing and motion planning problems for mobile systems [22, 36] 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 [31, 40, 41] and bike-sharing systems [32, 33], or reduce customers’ waiting time [31, 41] and taxis’ total idle distance [25] with the same number of empty vehicles. However, the limit knowledge we have about demand and mobility patterns [16] 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 [4, 23, 24], there is still trade-off between the system’s average performance and the worst-case performance with a probabilistic guarantee [26].

In this work, we integrate the process of gathering actionable information from data and designing decision-making objectives and constraints for vehicle balancing problems, to ensure real-time resource-allocating efficiency from the perspective of expected cost of ride-sharing service. 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 [13, 18], and linear controllers [30] in the literature. But there is no approaches for real-time distributionally robust vehicle balancing over complex transportation networks, or algorithms to model uncertain demand probability distributions set from data yet.

We design a computationally tractable distributionally robust dynamic vehicle balancing method under uncertainties about the probability distributions of demand. Efficient algorithms for constructing an uncertainty set of the probability distributions based on data and different demand model are developed. 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 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 costs of
the distributionally robust vehicle balancing method, based on uncertainty sets constructed based on different region partition and demand model 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 distributionally 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 efficient algorithms to construct uncertainty sets of probability distributions based on different demand model. These sets better capture the spatial-temporal correlations of demand uncertainties based on data.
- We derive a computationally tractable form to numerically solve the distributionally robust problem. The problem is not a standard linear programming (LP) or semi-definite programming (SDP) that has already been proved a computationally tractable form in the literature.
- 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 [31, 33, 40, 41] or aims to find a robust vehicle balancing solution for a single value (not a probability distribution) of worst-case demand [24–26, 31] 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 $\tau$ 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
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 $r^k_j > 0$ 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 $\tau$ 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 $X^k_{ij} \geq 0$ 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:

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

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

Table 1. Parameters and variables of taxi dispatch problem (8).
DD Dynamic Region Distributionally Robust Vehicle Balancing

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 [13, 18]. In Section 3 we will design an algorithm of calculating the set $F$ such that $F^* \in F$ 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.

$$\begin{align*}
\min_{X^1} \quad & \max_{F \in F} \quad \mathbb{E} \left[ J(X^{1: \tau}, r_c) \right] \\
\text{s.t.} \quad & X^{1: \tau} \in D_c.
\end{align*}$$

(2)

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

$$S^k_i = \sum_{j=1}^{n} X^k_{ij} - \sum_{j=1}^{n} X^k_{ij} + V^k_j > 0, \quad k = 1, \ldots, \tau,$$

$$V^{k+1}_i = \sum_{j=1}^{n} P^k_{v,ij} S^k_j + \sum_{j=1}^{n} Q^k_{v,ij} O^k_j, \quad k = 1, \ldots, \tau - 1,$$

$$O^{k+1}_i = \sum_{j=1}^{n} P^k_{o,ij} S^k_j + \sum_{j=1}^{n} Q^k_{o,ij} O^k_j, \quad k = 1, \ldots, \tau - 1,$$

(3)

where $P^k_v, P^k_o, Q^k_v, Q^k_o \in \mathbb{R}^{n \times n}$ are region transition matrices: $P^k_{v,ij}$ ($P^k_{o,ij}$) 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, $Q^k_{v,ij}$ ($Q^k_{o,ij}$) 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

$$\sum_{j=1}^{n} P^k_{v,ij} + P^k_{o,ij} = 1, \quad \sum_{j=1}^{n} Q^k_{v,ij} + Q^k_{o,ij} = 1.$$

Balancing the supply-demand ratio across the network is one type of service quality metric in power resource allocation [23], taxi dispatch [25] and autonomous mobility on demand systems [40]. Hence, we aim to minimize the difference between the local and global demand-supply ratio for $\tau$ time intervals

$$\sum_{k=1}^{\tau} \left| \sum_{i=1}^{n} \frac{r^k_i}{S^k_i} - \frac{\sum_{j=1}^{n} r^k_j}{\sum_{j=1}^{n} S^k_j} \right|.$$

(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 \( J_E \)

\[
J_E(X^{1:\tau}, r^k) = \sum_{k=1}^{\tau} \sum_{i=1}^{n_k} \left( \frac{a_{ik}r^k}{(S^k_i)^{\alpha}} \right),
\]

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) [24], and minimizing (5) means a balancing vehicle objective. With the definition of \( S^k_i \) 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 \( W^k_i \in \mathbb{R}^{n_k \times n_k} \) 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 \( W^k_{ij} \) is the approximated distance to drive from region \( i \) to region \( j \), the \textit{en route} idle distance is considered as the cost for allocating one empty vehicle. When \( W^k_{ij} = 1 \), 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

\[
J_D(X^k) = \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} X^k_{ij} W^k_{ij}.
\]

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. 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 [5], [1], the distance from region \( i \) to region \( j \) is \( \text{dist}_{ij} \). We denote a structural constraint matrix \( M^k \in \mathbb{R}^{n_k \times n_k} \), such that \( M^k_{ij} = 0 \) when \( \text{dist}_{ij} \leq m^k \), and \( M^k_{ij} = 1 \) otherwise. Then the following constraint

\[
X^k \circ M^k = 0, \quad X^k_{ij} \geq 0
\]

indicates a solution satisfies that \( X^k_{ij} = 0 \) for \( \text{dist}_{ij} > m^k \), \( i, j = 1, \ldots, n_k \). 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 \( \beta \) 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

\[
\min_{X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau}} \max_{F \in \mathcal{F}} \mathbb{E} \left[ \sum_{k=1}^{\tau} \left( J_D(X^k) + \beta \sum_{i=1}^{n_k} \frac{r^k_i}{(S^k_i)^{\alpha}} \right) \right]
\]

s.t. \( (3), (7) \),

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

, Vol. 1, No. 1, Article . Publication date: November 2017.
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 \( F \in F \).

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 [19]. Considering a balanced demand-supply ratio is considering to balance the average number of waiting customers intuitively. Robotic mobility-on-demand systems [37, 40] usually assume a queuing model to describe the passenger arrival rate at region \( i \) is \( \lambda^k_i \). When calculating the arrival rate for one time interval from historical data, \( \lambda^k_i \) 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 \( \lambda^k_i \) still exist when calculating this arrival rate \( \lambda^k_i \) 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 \( \lambda^k_i \) 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 [33]. 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 [8, 13]. 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 $\tau$ 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 $d_t$ as $\hat{r}_c(d_t, t)$, 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 $\hat{r}_c(d_1, t), \hat{r}_c(d_2, t), \ldots, \hat{r}_c(d_N, t)$ 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 $\mu_0$ and covariance $\Sigma_0$, test statistics $\gamma_1, \gamma_2$, with probability at least $1 - \alpha_h$, the random vector $r_c$ satisfies that [13]

$$H_0 : (\hat{r}_c - \mu_0)^T \Sigma_0^{-1} (\hat{r}_c - \mu_0) \leq \gamma_1, \quad (\hat{r}_c - \mu_0)(\hat{r}_c - \mu_0)^T \leq \gamma_2 \Sigma_0. \tag{9}$$

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—estimating the mean and covariance and calculating threshold values $\gamma_1$ and $\gamma_2$ such that (9) is an acceptable hypothesis by the data set. The problem of constructing $\mathcal{F}$ is formally defined as the following

**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 $\gamma_2^B$, such that with probability at least $1 - \alpha_h$ with respect to the samples the hypothesis testing (9) is acceptable. Then with probability at least $1 - \alpha_h$ the true distribution of $r_c$ is contained in the following distributional set $\mathcal{F}$

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

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 \[\hat{r}_c(t), \hat{\Sigma}_c(t), \hat{\Sigma}_c(t), \hat{Y}_1^B, \hat{Y}_2^B\] for \( r_c(t), t = 1, 2, \ldots, K \) of every time step, 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 \( \tau \) 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 \( t \).

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 \( \bar{r} \) 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 \( \bar{r}_c(1) \), store it as \( \hat{\Sigma}_{[1, n^1]} \); and every time when rolling the time horizon from \( t \) to \( t + 1 \), only calculate the covariance matrix entries between \( \tau \) pairs of \( (r^{t+\tau-k}, r^{t+\tau}) \), \( k = 0, \ldots, \tau - 1 \) and store the result as

\[
\hat{\Sigma}_{[n^{[1, t+\tau-1]}, n^{[1, t+\tau-k]}, n^{[1, t+\tau-k+1]}]} = \hat{\Sigma}_{[n^{[1, t+\tau-k]}, n^{[1, t+\tau-k+1]}]} = \text{cov}(r^{t+\tau-k}, r^{t+\tau}),
\]

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 \( \hat{\Sigma} \) 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.
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 \(S(I_p)\) 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 \(\alpha_k < 1\), the number of bootstrap time \(N_B \in \mathbb{Z}_+\).

\[
\text{for } j = 1, \ldots, N_B \text{ do } \\
\quad \text{Re-sample } S^j(I_p) = \{\hat{r}(d_1, I_p), \ldots, \hat{r}(d_{N}, I_p)\} \text{ from } S(I_p) \text{ with replacement, calculate the mean } \hat{r}^j(I_p) \text{ and covariance } \hat{\Sigma}^j(I_p). \\
\text{end for }
\]

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

\[
\hat{r}(I_p) = \frac{1}{N} \sum_{j=1}^{N} \hat{r}^j(I_p), \quad \hat{\Sigma}(I_p) = \frac{1}{N} \sum_{j=1}^{N} \hat{\Sigma}^j(I_p), \\
\hat{r}_i(t, I_p) = \min \hat{r}_i(d, I_p), \quad \hat{r}_i(t, I_p) = \max \hat{r}_i(d, I_p), \text{ for all samples } \hat{r}(d, I_p) \text{ in the subset } S(I_p).
\]

**3. Bootstrapping \(y_1^j\) and \(y_2^j\) for each time index \(t\)**

\[
\text{for each subset } S(t, I_p) \text{ do } \\
\quad \text{for } j = 1, \ldots, N_B \text{ do } \\
\quad\quad (1) \text{ Get the mean and covariance vector for the } j\text{-th re-sampled set, } \hat{r}_c(t, I_p), \hat{\Sigma}_c(t, I_p), \hat{r}_c(t, I_p), \hat{\Sigma}^j_c(t, I_p) \text{ as (12).} \\
\quad\quad (2) \text{ Get } y_1^j(t, I_p) \text{ and } y_2^j(t, I_p) \text{ by (13) and (14).} \\
\quad \text{end for }
\]

Get the \([N_B(1 - \alpha_k)]\)-th largest value of \(y_1^j(t, I_p)\) and \(y_2^j(t, I_p)\) for each index \(t, I_p\), respectively.

\[
\text{end for }
\]

**Output:** Distributionally uncertainty sets (10).

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

\[
\hat{r}_c(t, I_p) = \hat{r}_c[n[t, r-1], n[t, r-1]](I_p), \quad \hat{\Sigma}_c(t, I_p) = \hat{\Sigma}^j_c[n[t, r-1], n[t, r-1]](I_p). \tag{12}
\]

For the \(j\)-th re-sampled subset \(S^j(t, I_p)\), the mean and covariance matrices are \(E[r_c] = \hat{r}_c(t, I_p)\) and \(E[r_c^T] = \hat{\Sigma}_c(t, I_p)\), respectively. For step 3(2), according to the definition of \(T\) in (10), we get \(y_1^j(t, I_p)\) by the following equation

\[
y_1^j(t, I_p) = [\hat{r}_c(t, I_p) - \hat{r}_c(t, I_p)]^T \hat{\Sigma}_c^{-1}(t, I_p)[\hat{r}_c(t, I_p) - \hat{r}_c(t, I_p)]. \tag{13}
\]

According to definition (10), the left part of the inequality related to \(y_2^j\) satisfies that

\[
E[(r_c - \hat{r}_c)(r_c - \hat{r}_c)^T] = E[r_c r_c^T] - \hat{r}_c E[r_c^T] - E[r_c] \hat{r}_c^T + \hat{r}_c \hat{r}_c^T = \Sigma_c - \hat{r}_c \hat{r}_c^T.
\]

Then we get \(y_2^j\) for index \((t, I_p)\) by solving the following convex optimization problem

\[
\min_{y_2} \quad y_2 \\
\text{s.t.} \quad \hat{\Sigma}_c(t, I_p) - [(r_c(t, I_p)](r_c(t, I_p)]^T \leq y_2 \hat{\Sigma}_c(t, I_p) \tag{14}
\]

### 3.3 Constructing Uncertainty Sets for a General Demand Prediction Model

Besides directly using the mean of concatenated demand vector \(r_c(t, I_p)\) for each index \((t, I_p)\) as the predicted demand, methods that predict demand for time \(t\) based on the latest observation of
time \( t - 1, t - 2, \ldots \) or with streaming data has also been applied in areas such as transportation network [27, 38], power network [23, 34] and health care systems [2]. More complicated models can be more accurate than the average value prediction. It is critical to develop a uncertainty set constructing algorithm for general demand modeling techniques, and explore the effects of considering uncertainties to improve the ride sharing service. In this subsection, we first design a process of constructing distributional uncertainty sets for a general demand prediction model, and then 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 \( f_{r} : O_{[t-1-L-t-1]} \rightarrow \mathbb{R}^{T_{u}} \) 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 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 \( \delta_{c}(t) \in \mathbb{R}^{T_{u}} \) 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, 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 be either purely historical data of demand at each day of time \( t \) such as what used to calculate the average demand of each time \( t \) in Algorithm 1. It can also include streaming demand data of the same day before time \( t \).

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

\[
  \tilde{\delta}_{c}(t) = \tilde{r}_{c}(t) - r_{c}(t). \tag{16}
\]

With a subset of training data \( S_{tr}(t) = \{\tilde{r}_{c}(t), \tilde{O}_{t-1}\} \) that includes both observations \( \tilde{O}_{t-1} \) till 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 \( S_{te}(t) = \{\tilde{r}_{c}(t), \tilde{O}_{t-1}\} \), according to (15), we have a set \( S_{t}(t) = \{\tilde{r}_{c}(t)\} \) as samples of estimated or predicted demand and a set \( S_{\delta}(t) = \{\tilde{\delta}_{c}(t)\} \) as samples of residuals \( \delta_{c}(t) \). We also have the corresponding mean and covariance values for the residuals in set \( S_{\delta}(t) \).

We consider each \( \tilde{\delta}_{c}(t) \in S_{\delta} \) as one sample of the random residual vector \( \delta_{c}(t) \). Since \( \hat{r}_{c}(t) \) is a deterministic vector for time index \( t \), the following equations hold (the time index \( t \) is omitted for notation convenience):

\[
  \mathbb{E}[r_{c}] - \hat{r}_{c} = \mathbb{E}[\delta_{c}], \quad \mathbb{E}[(r_{c} - \hat{r}_{c})(r_{c} - \hat{r}_{c})^{T}] = \mathbb{E}(\delta_{c}\delta_{c}^{T}), \quad \hat{r}_{c} + \delta_{c,l} = \hat{r}_{c} + \delta_{c,h}, \quad \hat{r}_{c} + \delta_{c,h} = \hat{r}_{c} + \delta_{c,h},
\]

\[
  r_{c} - \mathbb{E}[r_{c}] = \hat{r}_{c} + \delta_{c} - (\hat{r}_{c} + \mathbb{E}[\delta_{c}]), \quad \mathbb{E}[(r_{c} - \mathbb{E}[r_{c}])(r_{c} - \mathbb{E}[r_{c}])^{T}] = \Sigma_{\delta}, \quad \hat{\Sigma}_{c} = \hat{\Sigma}_{\delta}, \tag{17}
\]

where \( \Sigma_{c} \) and \( \Sigma_{\delta} \) are the unknown true covariance of \( r_{c} \) and \( \delta_{c} \) respectively, \( \hat{\Sigma}_{c} \) and \( \hat{\Sigma}_{\delta} \) are the estimated matrices for \( \Sigma_{c} \) and \( \Sigma_{\delta} \) respectively, \( \delta_{c,l} \) 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 \( \delta_{c} \).

Publication date: November 2017.
A grid file [29] 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 [24, 25, 40]. 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 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).

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 bound, mean and covariance values of the residual $\delta_c$

**Definition 3.2. Problem 2.** Given a dataset of $r_c$, for a prediction method $f_r$, find the values of $\hat{\delta}_{c,t}, \hat{\delta}_{c,h}, \hat{r}_c, \hat{\Sigma}_c, Y^B_1, Y^B_2$ 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 $\mathcal{F}$

$$\mathcal{F}(\delta_{c,t}, \delta_{c,h}, \hat{r}_c, \hat{\Sigma}_c, Y^B_1, Y^B_2) = \{r_c \in [\hat{r}_c + \delta_{c,t}, \hat{r}_c + \delta_{c,h}] : (E[\delta_c])^T \hat{\Sigma}_c^{-1} E[\delta_c] \leq Y^B_1, \ E(\delta_c \delta_c^T) \leq Y^B_2 \hat{\Sigma}_c \} \quad (18)$$

The Algorithm 1 considers to construct an uncertainty set of the concatenated demand vector $r_c$, and the estimated demand $\hat{r}_c(t)$ for each index $t$ is the average value of bootstrapped samples. For a general modeling method $f_r$, we design the following Algorithm 2 to build an uncertainty set of $r_c$ based on repeated estimations of residual $\delta_c$.

**3.3.2 Time series model prediction.** One example of model (15) is vector time series model applied for spatial-temporal data. Time series is a widely applied method for processing streaming data of predicting demand in resource allocation problems [2, 23, 27]. To involve both spatial and temporal correlations of demand, we use a vector autoregressive integrated moving average (ARIMA) model [39] model in this work, and show constructing a distributional uncertainty set by Algorithm 2 improves the efficiency of vehicle sharing service in data experiments in Section 5. We denote an order $(p, q)$ ARIMA model for the multivariate time series $r_c(t)$ as the following

$$r_c(t) = \theta_0 + \Pi_1 r_c(t-1) + \Pi_2 r_c(t-2) + \cdots + \Pi_p r_c(t-p) + \epsilon_t - 3ta_1\epsilon_{t-1} - \theta_2\epsilon_{t-2} - \cdots - \theta_q\epsilon_{t-q}. \quad (19)$$

Here $r_c(t-l)$ is the $l$-th lag of $r_c(t)$, $\theta_0$ is a constant vector of intercepts with the same dimension of $r_c(t)$, $\Pi_i (i = 1, 2, \ldots, p)$ and $\theta_j (j = 0, 1, 2, \ldots, q)$ are model parameters with appropriate dimensions, respectively; $p$ and $q$ are integers and are often referred to as orders of the model, $\epsilon_t$ are random errors. If $q = 0$, then (19) is an autoregressive (AR) model of order $p$. When $p = 0$, the model reduces to an moving average (MA) model of order $q$. We fit parameters $\theta_j, \Pi_i$ of (19) to data by least mean square error estimation, and the estimation residual is $\hat{\delta}_c(t) = r_c(t) - \hat{r}_c(t)$ that covers the random error components. By estimating the distributional uncertainty of $\hat{\delta}_c(t)$ via Algorithm 2 and time series prediction model, we describe how the true demand can deviate from our prediction through repeated data experiments.

**3.4 Dynamic Space Partitioning**

A grid file [29] 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 [24, 25, 40]. 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 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).
ALGORITHM 2: Algorithm for constructing distributional sets for a general prediction method

Input: A dataset of spatial-temporal demand

1. Demand aggregating and sample set partition

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

2. Estimate the parameters of prediction function $f_r$ (Ot-1) for all time steps $t$ in one day.

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

Re-sample $S(I_p) = \{\tilde{r}(d_1, I_p), \ldots, \tilde{r}(d_N, I_p)\}$ from $S(I_p)$ with replacement, calculate the estimation residual set $S(I_p) = \{\hat{\delta}(d_1, I_p)\}$ of all samples based on prediction function $f_r$, where

\[\hat{\delta}(d_1, I_p) = \tilde{r}(d_1, I_p) - r(d_1, I_p)\]

then the mean $\bar{\delta}(I_p)$, covariance $\bar{\delta}^T(I_p)$, and

\[\mathbb{E}[\hat{\delta}, \hat{\delta}^T(I_p)] = \frac{N}{N} \sum_{i=1}^{N} \hat{\delta}(d_N, I_p)(\hat{\delta}(d_N, I_p))^T \text{ for residual for all time steps.} \]

end for

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

\[\mathbb{E}[\hat{\delta}](I_p) = \frac{1}{B} \sum_{j=1}^{B} \hat{\delta}(I_p), \bar{\delta}(I_p) = \frac{1}{B} \sum_{j=1}^{B} \delta(I_p), \]

\[\hat{\delta}_j(I_p) = min(\hat{\delta}_d(I_p), \hat{\delta}_h(I_p) = max(\hat{\delta}_d(I_p), \hat{\delta}_h(I_p), \hat{\delta}_l(I_p), \hat{\delta}_r(I_p)) \text{ for all samples } \hat{\delta}(I_p) \text{ in the subset } S(I_p). \]

3. Bootstrapping $y_{\delta, 1}^B$ and $y_{\delta, 2}^B$ for each time index $t$

for $t = 1, \ldots, K$ do

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

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

Calculate $y_{\delta, 1}^B(t, I_p) = argmin_{y_1}(\hat{\delta}(I_p))^T(\bar{\delta}(I_p))^{-1}\hat{\delta}(I_p)$, and

$y_{\delta, 2}^B(t, I_p) = argmin_{y_2}(\hat{\delta}, \hat{\delta}^T(I_p)) < y_2 \bar{\delta}(I_p).$

end for

Get the $[N_B(1 - \alpha_b)]$-th largest value of $y_{\delta, 1}^B(t, I_p)$ and $y_{\delta, 2}^B(t, I_p)$, $j = 1, \ldots, N_B$, as $y_{\delta, 1}^B(t, I_p)$ and

$y_{\delta, 2}^B(t, I_p)$, respectively.

end for

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

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 [15].

The quad-tree [15] 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: $(x_{min}, y_{min}, (x_{mid}, y_{mid}))$, $(x_{mid}, y_{mid}, (x_{max}, y_{max}))$, $(x_{mid}, y_{min}, (x_{mid}, y_{max}))$, and $(x_{min}, y_{mid}, (x_{max}, y_{max}))$, 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 [20]). 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 \((\text{latitude}, \text{longitude}, \text{time} – \text{interval})\). The time dimension is divided into fixed time intervals to provide a fair comparison with the grid structure, and the \((\text{latitude}, \text{longitude})\) 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 \(J_E(X^{1:t}, r_c)\) 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 \([7, 8, 12]\). 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 \(J_E(r, X^{1:t})\) 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

\[
\begin{align*}
\text{min.} \quad & \beta(v + t) + \sum_{k=1}^{r} J_D(X^k) \\
\text{s.t.} \quad & \begin{bmatrix}
    v + (y_1^+)^T \hat{r}_{c,1} - (y_1^-)^T \hat{r}_{c,h} \\
    \frac{1}{2}(q - y - y_1)^T \hat{r}_{c,h}
\end{bmatrix} \geq 0 \\
& t \geq (y_2^B \hat{x}_2 + \hat{r}_c \hat{r}_c^T) \cdot Q + \hat{r}_c^T q + \sqrt{\frac{1}{y_1^+}} \| \hat{\Sigma}_c^{1/2} (q + 2Q \hat{r}_c) \|_2 \\
& \frac{a_{ik}}{(s_i^k)^{a}} \leq y_i^+ - y_i^- \\
& y_1 = [y_1^+, y_2^+, \ldots, y_r^+, y_2^-, \ldots, y_r^-]^T \\
& y_1^+, y_2^-, \ldots, y_r^- \geq 0, \quad Q \geq 0 \\
& X^{1:t}, S^{1:t}, V^{2:t}, O^{2:t} \in D_c.
\end{align*}
\]
### 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) [14]. 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 and Algorithm 2, solve (20), the equivalent convex optimization form of problem (8) to get vehicle balancing solutions across regions. Region is partitioned by either static equal-area grid or dynamic quad-tree method, demand is predicted by either directly use the average value of historical data or ARIMA model (19). 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 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 [40], or bike sharing [32], 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 [40]. 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 $\gamma_1^B$ and $\gamma_2^B$ 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 $\gamma_1^B$ and $\gamma_2^B$ with different values of sample number $N_B$ and the dimension of $r_c (\tau n)$ in Table 4. When the value of $N_B$ is increased, values of $\gamma_1^B$ and $\gamma_2^B$ are reduced, which means the volume of the distributional set is smaller. For a large enough $N_B$, the value of $\tau n$ does not affect $\gamma_1^B$ and $\gamma_2^B$ much.
Table 4. Comparing thresholds $\gamma^B_1$ and $\gamma^B_2$ for different $N_B$ and dimensions of $r_c$

<table>
<thead>
<tr>
<th>$N_B$</th>
<th>$n$</th>
<th>$\tau$</th>
<th>$\gamma^B_1$</th>
<th>$\gamma^B_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>50</td>
<td>2</td>
<td>0.739</td>
<td>5.24</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>2</td>
<td>0.368</td>
<td>2.47</td>
</tr>
<tr>
<td>1000</td>
<td>50</td>
<td>3</td>
<td>0.013</td>
<td>1.56</td>
</tr>
<tr>
<td>5000</td>
<td>50</td>
<td>6</td>
<td>0.012</td>
<td>1.49</td>
</tr>
</tbody>
</table>

Fig. 2. The average cost of cross-validation tests for the distributionally robust solutions via solving (20) ("DRO" line), two types of uncertainty sets of the robust solutions (lines SOC and Box) and non-robust solutions.

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 [26], and the non-robust method with the average requests number during each unit time as the demand model [25] (equivalent to the passenger arrival rate of a queueing model in each unit time [40, 41]). 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 region partition method is 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 [26] is close to the average cost of the distributionally robust solutions of (20). 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 $\epsilon$ 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 $\epsilon$. The average cost of solutions of (20) 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 [26] provides a 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 $\gamma_{B_{1}}$ and $\gamma_{B_{2}}$ 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 $\alpha_{h} = 0.1$.

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 $\gamma_{B_{1}}$ and $\gamma_{B_{2}}$ 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 short, predicting demand based on the quad tree method is worse than that based on the simple equal-area grid


<table>
<thead>
<tr>
<th>t = 2 h, γ_1^B</th>
<th>Grid</th>
<th>Quad-Tree</th>
<th>Change Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>t = 2 h, γ_2^B</td>
<td>0.016</td>
<td>0.021</td>
<td>31.25%</td>
</tr>
<tr>
<td>t = 1 h, γ_2^B</td>
<td>1.73</td>
<td>2.05</td>
<td>18.50%</td>
</tr>
<tr>
<td>t = 1 h, γ_1^B</td>
<td>0.0130</td>
<td>0.0110</td>
<td>−15.38%</td>
</tr>
<tr>
<td>t = 50 m, γ_1^B</td>
<td>1.56</td>
<td>1.35</td>
<td>−13.46%</td>
</tr>
<tr>
<td>t = 50 m, γ_2^B</td>
<td>0.0128</td>
<td>0.0107</td>
<td>−16.41%</td>
</tr>
<tr>
<td>t = 40 m, γ_1^B</td>
<td>1.53</td>
<td>1.32</td>
<td>−13.73%</td>
</tr>
<tr>
<td>t = 40 m, γ_2^B</td>
<td>0.0125</td>
<td>0.0102</td>
<td>−18.40%</td>
</tr>
<tr>
<td>t = 30 m, γ_1^B</td>
<td>1.49</td>
<td>1.28</td>
<td>−15.44%</td>
</tr>
<tr>
<td>t = 30 m, γ_2^B</td>
<td>0.0121</td>
<td>0.0095</td>
<td>−21.49%</td>
</tr>
<tr>
<td>t = 20 m, γ_1^B</td>
<td>1.46</td>
<td>1.12</td>
<td>−17.12%</td>
</tr>
<tr>
<td>t = 20 m, γ_2^B</td>
<td>0.0119</td>
<td>0.120</td>
<td>0.84%</td>
</tr>
<tr>
<td>t = 15 m, γ_1^B</td>
<td>1.41</td>
<td>1.40</td>
<td>4.96%</td>
</tr>
<tr>
<td>t = 15 m, γ_2^B</td>
<td>0.0120</td>
<td>0.123</td>
<td>2.50%</td>
</tr>
</tbody>
</table>

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_{i-1} means the values in the corresponding column.

<table>
<thead>
<tr>
<th>Region division</th>
<th>Grid</th>
<th>Quad-tree</th>
<th>change rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>t = 1h</td>
<td>7.63 × 10^4</td>
<td>6.62 × 10^4</td>
<td>13.1%</td>
</tr>
<tr>
<td>t = 30m</td>
<td>6.84 × 10^4</td>
<td>5.47 × 10^4</td>
<td>20.0%</td>
</tr>
</tbody>
</table>

Table 6. Comparison of average total idle distance (weekdays 5pm-8pm) with distributionally robust dispatch solutions by solving (20) (equivalent form of (8)).

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 (20) (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.

### 5.3 Time series demand prediction and distributional uncertainty sets

In this subsection, we show the demand prediction error at different time of one day using the ARIMA time series model (19), 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. In the previous experiments, demand at each time t is predicted directly as the average value of historical demand at time t, and the uncertainty set is built according to the bootstrapped mean and covariance estimation in Algorithm (1). Though
Fig. 5. Heatmap 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 error rate at each hour is from 7% to 22%.

Fig. 7. Comparison of the true demand and demand predicted by ARIMA model in region 24 in one day. The error rate at each hour is from 6% to 20%.

Fig. 8. Comparison of the true demand and demand predicted by ARIMA model in region 42 in one day. The error rate at each hour is from 4% to 35%.

Arima model or similar time series models in the literature such as [27] provides a relatively more accurate model, still there exist random errors and considering prediction uncertainties benefit the vehicle dispatch decisions.

We first compare the true demand and predicted demand via ARIMA model (19) for different time of weekdays in Figures 6, 7, and 8. Figure 5 shows a static equal-area grid region partition for Manhattan area 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. The prediction is more accurate than directly.

When demand is predicted by (19) and uncertainty sets are constructed by Algorithm 2, Table 7 shows the comparison of $\gamma_{B,1}$ and $\gamma_{B,2}$ 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. For other parameters of the experiments, we use $\tau = 4$ for weekdays’ 5 – 8 pm, $N_s = 1000$, 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 $\gamma_{B,1}$ and $\gamma_{B,2}$ values when we use the quad tree partition method, according to the results when $t = 50$ m,
Quad-Tree

Change Rate with distributionally robust dispatch solutions by solving (20) (equivalent form of (8)), based on \( \gamma = t \) (20) (the idle distance reduction rate of the distributionally robust and non-robust solutions is 10.05%).

<table>
<thead>
<tr>
<th>Region division</th>
<th>Grid</th>
<th>Quad-Tree</th>
<th>Change Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t = 1 h )</td>
<td>( 7.15 \times 10^4 )</td>
<td>( 6.36 \times 10^4 )</td>
<td>11.05%</td>
</tr>
<tr>
<td>( t = 30 m )</td>
<td>( 6.58 \times 10^4 )</td>
<td>( 5.29 \times 10^4 )</td>
<td>19.60%</td>
</tr>
</tbody>
</table>

Table 7. Demand predicted by ARIMA model: comparison of average total idle distance (weekdays 5pm-8pm) with distributionally robust dispatch solutions by solving (20) (equivalent form of (8)).

\( t = 40 \text{ m} \), and \( t = 30 \text{ m} \). However, when the length of one time instant is too long such as \( t = 2 \text{ h} \) and \( t = 1 \text{ h} \), or too short, such as \( t = 20 \text{ m} \) and \( t = 15 \text{ m} \), predicting demand based on the quad tree method is worse than that based on the simple equal-area grid partition. The values of \( y_{\delta,1}^B \) and \( y_{\delta,2}^B \) 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 model (19), we compare the average total idle distance with distributionally robust dispatch solutions by solving (20) (equivalent form of (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 grid method for region partitioning, compared with vehicle dispatch decisions not considering the demand prediction error by model (19), 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 and non-robust solutions is 10.05%). 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% mileage reduction compared with grid-region partition, ARIMA demand prediction without considering model uncertainties.

5.4 Carpool

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

7 APPENDIX

7.1 Proof of Theorem 4.1

Proof. We have \(-\frac{a_{ik}}{(S_k^T)^{\alpha}} > 0\) and \(r_c \geq 0\) by the definitions of \(J_r\) in (5) and the demand model, then for any vector \(y \in \mathbb{R}^{n_r}, y = [y^1_1, y^1_2, \ldots, y^1_{r}, y^2_1, \ldots, y^k_{r}]^T\) that satisfies \(0 < \frac{a_{ik}}{(S_k^T)^{\alpha}} \leq y^k_i\), we also have

\[
0 \leq \sum_{k=1}^{\tau} \sum_{i=1}^{n_k} a_{ik} r^k_i (S_k^T)^{\alpha} \leq y^T r_c,
\]

and the second inequality strictly holds when all \(\frac{a_{ik}}{(S_k^T)^{\alpha}} = y^k_i\), 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

\[
\min_{X^k} \beta \sum_{k=1}^{\tau} \sum_{i=1}^{n_k} a_{ik} r^k_i (S_k^T)^{\alpha} + \sum_{k=1}^{\tau} J_D(X^k)
\]

s.t. \(X^{[1, \tau]}, S^{[1, \tau]}, V^{[2, \tau]}, O^{[2, \tau]} \in D_c\)

is equivalent to

\[
\min_{X^k} \beta y^T r_c + \sum_{k=1}^{\tau} J_D(X^k)
\]

s.t. \(\frac{a_{ik}}{(S_k^T)^{\alpha}} \leq y^k_i, y \in \mathbb{R}^{n_r}, \) (21)

\[y = [y^1_1, y^1_2, \ldots, y^1_{r}, y^2_1, \ldots, y^k_{r}]^T,\]

\[X^{[1, \tau]}, S^{[1, \tau]}, V^{[2, \tau]}, O^{[2, \tau]} \in D_c\]

In this proof, we use the objective function of problem (21). In particular, only the part of \(y^T r_c\) is related to \(r_c\), and we first consider the following maximization problem

\[
\max_{r_c \sim F, \hat{F} \in \mathcal{F}} \mathbb{E}[y^T r_c]
\] (22)

By the definition of problem (8) and problem (21), 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 \(y, X^{1:r}, S^{1:r}, V^{2:r}\), and \(O^{2:r}\) are convex, closed, and bounded sets. Hence, problem (22) satisfies the conditions of Lemma 1 in [13], 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

\[
\min_{Q, q, v, t} ~ v + t
\]

s.t. \(v \geq y^T r_c - r^T_c Q r_c - R^T q, \forall r_c \in [\hat{r}_c, l, \hat{r}_c, h]\)

\[t \geq (y^B \hat{S}_c + \hat{r}_c \hat{r}_c^T) \cdot Q + \hat{r}_c^T q + \sqrt{y^B \hat{S}_c^{1/2}} (q + 2Q \hat{r}_c) \|_2\]

\[Q \geq 0\] (23)

Hence, we first analytically find the optimal value of problem (23). Note that the first constraint about \(v\) is equivalent to \(v \geq f(r^*_c, y)\), where \(f(r^*_c, y)\) is the optimal value of the following problem

\[
\max_{r_c} ~ y^T r_c - r^T_c Q r_c - R^T q
\]

s.t. \(\hat{r}_{c, l} \leq r_c \leq \hat{r}_{c, h}\).


For a positive semi-definite $Q$, the optimal solution of problem (24) exists. The Lagrangian of (24) under the constraint $y_1^+, y_1^- \geq 0$ is

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

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

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

$$y_1 = y_1^+ - y_1^-, \quad y_1^+, y_1^- \geq 0.$$

Then the first inequality constraint of problem (23) for any $\hat{r}_{c,l} \leq r_c \leq \hat{r}_{c,h}$ is equivalent to

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

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 (20). \qed