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@cs.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 balancing vehicles towards mobility demand predicted based on data. However, how to deal with uncertainties in demand probability distribution for improving the average system performance is still a challenging and unsolved task. In this work, we develop a data-driven distributionally robust vehicle balancing model to minimize the worst-case average cost considering spatial-temporally correlated demand probability distribution uncertainties. We design an efficient algorithm for constructing distributional uncertainty sets of random demand vectors, 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 demand-supply ratio error and average total idle driving distance are reduced by 28.6% and 10.05%, respectively, with a static grid region partition method. With the quad-tree dynamic region partition method, the average total idle distance is reduced by 20% more. This is about 60 million miles or 8 million dollars cost reduction annually for all taxis in NYC compared with vehicle balancing solutions based on static region partitions without considering demand distribution uncertainties.

Categories and Subject Descriptors

H.4 [Information System Application]: Miscellaneous; I.2.8 [Problem Solving, Control Methods, and Search]: Control Application, Transportation

Keywords

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

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 [25]. With the development of technologies, we are able to collect, store, and analyze a large amount of data efficiently [16]. 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, 18], passengers’ demand model of taxi network [24], and road transportation network efficiency [30] 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 [13], routing and motion planning problems for mobile systems [19, 31] 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 [27, 32, 33, 34] and bike-sharing systems [28], or reduce customers’ waiting time [27, 34] and taxis’ total idle distance [22] with the same number of
empty vehicles. However, the knowledge and assumptions about the demand model affect the performance of vehicle balancing strategies. A robust optimal solution shows its advantage in worst-case scenarios compared with non-robust approaches [1, 20, 21], but there is still trade-off between the system’s average performance and the worst-case performance with a probabilistic guarantee [22].

How to optimize the expected vehicle balancing cost or average performance of a large-scale vehicle system is still a challenging and unsolved task. It is difficult to obtain an explicit form of the true distribution of the random demand purely based on data in practice, and stochastic programming (SP) approaches that minimize the average cost with a specific distribution of the random demand is not appropriate to describe the ambiguity about demand uncertainties. Distributionally robust optimization techniques have been developed for optimal expected cost under the worst-case distribution of random parameters in the optimization literature [10, 14]. But it is not clear whether there is a computationally efficient approach for balancing vehicles in complex transportation networks, from the process of modeling spatial-temporally correlated demand based on real data, to a distributionally robust problem formulation and its evaluations. Hence, we construct a set of distributions that includes the true distribution of the random demand via a demand data set, and minimize the expected cost over the worst-case distribution in the set.

In this work, we design a data-driven distributionally robust dynamic vehicle balancing model under uncertainty about the distribution of demand. An efficient algorithm for constructing an uncertainty set of the distributions based on data is proposed, without assumptions about prior knowledge. We utilize the structure property of the distributional uncertainty set. 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 computationally tractable form of the 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 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 an efficient algorithm to construct a distributional set for large-scale spatial-temporally correlated demand data, and a quad-tree dynamic region partition process that is compatible with the distributional set.
- 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 demand-supply ratio error is reduced by 28.6%, 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 cost reduction annually for all taxis in NYC compared with non-robust balancing 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 dynamic distributionally robust vehicle balancing problem. The goal includes balancing vehicles for efficient service and reducing the total cost of balancing vehicles, such as vehicles’ total idle distance or the total number of vehicles sent to other regions or stations. By considering possible probability distributions of demand model 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 [27, 28, 33, 34] or aims to find a robust vehicle balancing solution for a single value (not a distribution) of worst-case demand [21, 22, 23, 24]. The generalization of the vehicle balancing problem formulation is also explained in Subsection 2.2. A list of parameters and variables in the problem formulation is shown in Table 5.

2.1 Problem Formulation

We consider vehicle balancing or re-balancing decisions in a time window of \( \tau \) discrete time slots, where \( k = 1, 2, \ldots, \tau \), and the effect of current decisions to the future re-balancing cost is involved. We assume that there are \( n \) regions (nodes) to be served at discrete time \( k \), with \( r^k_j \geq 0 \) as the predicted total amount of demand (number of passenges for a mobility-on-demand system) within region \( j \) during time slot \( k, j = 1, \ldots, n, k = 1, \ldots, \tau \). It is worth noting that the number of region in the network can change with time. We consider \( r^k \in \mathbb{R}^n \) as a random demand 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

\[
F_c = (r^1, r^2, \ldots, r^\tau), \quad n_c = \sum_{k=1}^{\tau} n^k.
\]

We assume that \( F_c \) is the true distribution function for 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 \(X^{1:\tau} = \{X^1, X^2, \ldots, X^\tau\}\). With an objective \(J(X^{1:\tau}, r_c)\) related to the random demand \(r_c\), a stochastic programming vehicle balancing is defined as:

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

However, in many application problems we only have limited knowledge about the true distribution \(F^*\). Moreover, problem (1) is computationally demanding, not suitable for a large-scale dynamic load balancing problem in smart cities in general. The knowledge of random demand \(r_c\) is restricted to a set of samples—historical or streaming demand data, and the samples are independently and randomly 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, 14]. In Section 3 we will describe the process of calculating the set \(\mathcal{F}\) such that \(F^* \in \mathcal{F}\) with a high probability. We then consider the following distributionally robust problem as a robust form of stochastic programming (1) to minimize the worst-case expected cost

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

In the rest of this section we will define concrete forms of objective function and constraints.

### 2.1.1 Service quality metric function \(J_E\)

We define \(V^k \in \mathbb{R}_+, O^k \in \mathbb{R}_+\) 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 recending 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 \(S^k_i > 0\) as the total amount of vehicles available within region \(i\) during time \(k\) with dispatch decision \(\{X^1, \ldots, X^k\}\)

\[
S^k_i = \sum_{j=1}^{n_k} X^k_{ji} - \sum_{j=1}^{n_k} X^k_{ij} + V^k_i, \quad k = 1, \ldots, \tau,
\]

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

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

(3)

where \(P^k_{v,ij}, P^k_{o,ij}, Q^k_{v,ij}, Q^k_{o,ij} \in \mathbb{R}^{n_k \times n_k+1}\) 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, \(O^k_{o,ij}\) 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 empty (occupied) at the beginning of time interval \((k + 1)\). The region transition matrices are learned from historical data, and satisfies

\[
\sum_{j=1}^{n_k} P^k_{v,ij} + P^k_{o,ij} = 1, \quad \sum_{j=1}^{n_k} Q^k_{v,ij} + Q^k_{o,ij} = 1.
\]

The service quality is related to the amount of supply provided to each region. In this work, we consider a service quality function \(J_E\)

\[
J_E(X^{1:\tau}, r^k) = \sum_{i=1}^{n_k} \left( \frac{a_{ik} r^k_{ij}}{(S^k_i)^{\alpha}} \right), \tag{4}
\]

where \(a_{ik} > 0, i = 1, \ldots, n_k, k = 1, \ldots, \tau\) are positive constants denoting region priorities, \(\alpha > 0\) is a parameter that is designed according to the service requirement. In particular, balancing the supply-demand ratio across the network is one type of service quality metric in power resource allocation [20], taxi dispatch [22] and autonomous mobility on demand systems [33], and bike sharing problems [23]. Hence, we would like to find a solution that minimizes the difference between the local and global demand-supply ratio for \(\tau\) time intervals

\[
\sum_{k=1}^{\tau} \sum_{i=1}^{n_k} \left| r^k_{ij} - \frac{\sum_{j=1}^{n_k} r^k_{ij}}{\sum_{j=1}^{n_k} S^k_j} \right|^2. \tag{5}
\]

When \(a_{ik} = 1, i = 1, \ldots, n_k, k = 1, \ldots, \tau, \alpha > 0\) is a parameter close to 0, function \(J_E(X^{1:\tau}, r^k)\) defined as (4) is a surrogate function of (5) as explained in [21], and \(J_E\) defines a balancing vehicle objective. With the definition of \(S^k_i\) as (5), \(J_E\) 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 (4), we also consider minimizing 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 \(W^k_{ij}\) is the approximated distance to drive from region \(i\) to region \(j\), the \(en\) route 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

\[
J_D(X^k) = \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} X^k_{ij} W^k_{ij}. \tag{6}
\]
The distance every vehicle can travel is bounded, because of the speed limit during time \(k\) and traffic conditions—during congestion hours, the distance each vehicle can go to pick up a passenger should be shorter than normal hours. Assume that the idle distance upper bound for a vehicle at time \(k\) is \(m^k\), provided by traffic speed monitors and forecasting models \([2, 29]\). 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 \(dist_{ij} \leq m^k\), and \(M^k_{ij} = 1\) otherwise. Then the following constraint
\[
X^k \circ M^k = 0, \quad X^k_{ij} \geq 0
\] (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 \(\beta\) of two objectives \(J_D\) in \([6]\) and \(J_E\) in \([4]\). With constraints \((5)\) and \((7)\), we consider the following distributionally robust vehicle balancing problem under uncertain probability distributions of random demand
\[
\min_{X^{1:\tau}, S^{1:\tau}, V^{2:\tau}, O^{2:\tau} \in \mathcal{F}^\tau} \max_{\tau_{1:k}} \mathbb{E}\left[ \sum_{k=1}^{\tau} \left( J_D(X^k) + \beta \sum_{i=1}^{n^k} \left( S_i^k \right) \alpha \right) \right]
\text{ s.t. } (3), (7),
\] (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 \(F \in \mathcal{F}\). 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 \(t\), we solve \((8)\) with demand model of index \((t, t+1, \ldots, t+\tau-1)\) respectively. Only the solution of \(k = 1\) for \(t\) is implemented, while the solutions for remaining time slots are not materialized. 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, for the current time window. Examples of receding horizon resource allocation applications include economic power dispatch \([20]\), taxi dispatch \([22]\), and autonomous mobility-on-demand \([34]\), etc.

### 2.2 Discussions about 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 \([15]\). Considering a balanced demand-supply ratio is considering to balance the average number of waiting customers intuitively. Robotic mobility-on-demand systems \([32, 53]\) usually assume a queuing model to describe the process of vehicles serving passengers and the passenger arrival rate at region \(i\) is \(\lambda_i^k\). When calculating the arrival rate for one time interval from historical data, \(\lambda_i^k\) equals the total number of requests appearing in one time interval, or \(r_i^k\) in this work. Mean and covariance of the estimation of \(\lambda_i^k\) still exist when calculating this arrival rate \(\lambda_i^k\) via data. Hence, when a mobility-on-demand system can be described by a queuing model, solving problem \((8)\) provides a solution for balancing vehicles for \(\lambda_i^k\) in a range instead of a deterministic value. Therefore, we do not restrict the demand model to satisfy a specific distribution and we reduce the dependency of the average performance of solutions to the accuracy of demand model.

Similarly, bicycle balancing and re-balancing problems also require that the demand-supply ratio of each station is restricted inside a range in order to provide a certain level of service satisfaction \([28]\). While adjusting the range of demand-supply ratio or supply-demand ratio back and forth is computationally expansive, when we find a feasible solution \((8)\), the demand-supply ratio of each region should 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 \(O_{il}^k\) as the number of vehicles with capacity \(C_l\) before dispatch at region \(i\), and \(X_{l,ij}^k\) as the number of vehicles that should go from region \(i\) to region \(j\). Then the total number of available seats or supply is \(S_i^k = \sum_{l=1}^{4} C_l \left( O_{il}^k + \sum_{j=1}^{n} X_{l,ij}^k - \sum_{j=1}^{n} X_{l,j}^k \right)\). With this number \(S_i^k\), objective function \(J_E\) defined as \((4)\) is still concave in \(r^k\), convex in \(X_{l,j}^k\), \(l = 1, 2, 3, 4\). 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 \((4)\) is generalizable to consider 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 algorithm for constructing the distributional set $\mathcal{F}$ of 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 via a bootstrap method [6] and hypothesis testing 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 $\tau$ time slots (the demand prediction and vehicle balancing time length) of one day, we leverage the structure property of the covariance matrix of the random demand vector to develop an efficient distributional set construction algorithm. Furthermore, to reflect the spatial-temporal dynamic properties of the demand model and index regions efficiently, we build our distributional set based on a dynamic space partition method.

3.1 Distributional Set Formulation

To describe the demand changing trend at different time of one day, we denote one sample of vector $r_c(t) = (r_{c,1}^t, r_{c,2}^t, \ldots, r_{c,\tau}^t)$ at date $d_t$ as $\tilde{r}_c(d_t, t)$, a vector of demand at each region for time $\{t, t+1, \ldots, t+\tau-1\}$, $t = 1, \ldots, K$. The distributional set for $r_c$ is $\mathcal{F}(t)$. 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. Hence, we aim to construct a distributional set $\mathcal{F}(t)$ that describes possible distribution $r_c(t)$ based on the support, mean and covariance values of a random vector of a given data set for each $t$. For notational convenience, we omit $t$ for the following problem definition when there is no confusion. Possible distributions of a random vector $r_c$ is related to a hypothesis testing $H_0$ given a set of $r_c$: given mean $\mu_c$ and covariance $\Sigma_c$, test statistics $\gamma_1$, $\gamma_2$, with probability at least $1 - \alpha_h$, the random vector $r_c$ satisfies that

$$H_0: (\tilde{r}_c - \mu_c)^T \Sigma_c^{-1} (\tilde{r}_c - \mu_c) \leq \gamma_1,$$

$$\tilde{r}_c - \mu_c)^T \Sigma_c^{-1} (\tilde{r}_c - \mu_c) \leq \gamma_2 \Sigma_c.$$  \hspace{1cm} (9)

Delage and Ye propose a model of distributional set and prove a confidence region for the mean and the covariance matrix of a random vector [10]. 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. Using bootstrap process [6] for calculating the support (range), mean and covariance values, the problem of constructing a distributional set is defined as:

**Problem 1.** Given a dataset of $r_c$, find the values of $\tilde{r}_c, \hat{\Sigma}_c, \gamma_1^B$ and $\gamma_2^B$, 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 $\mathcal{F}$

$$\mathcal{F}(\tilde{r}_{c,1}, \tilde{r}_{c,h}, \hat{\Sigma}_c, \gamma_1^B, \gamma_2^B) = \{(E[r_c] - \tilde{r}_c)\Sigma_c^{-1}(E[r_c] - \tilde{r}_c) \leq \gamma_1^B,$$

$$E[(r_c - \tilde{r}_c)(r_c - \tilde{r}_c)^T] \leq \gamma_2^B \Sigma_c, \ r_c \in [\tilde{r}_{c,1}, \tilde{r}_{c,h}] \}$$

where $\text{supp}(r_c) \subset [\tilde{r}_{c,1}, \tilde{r}_{c,h}]$ is the support of $r_c$, $\tilde{r}_{c,1}$ and $\tilde{r}_{c,h}$ is the lower bound and upper bound of each of component of the demand vector, respectively.

We then design Algorithm [1] (a list of parameters in Appendix Table 6) to calculate the bootstrapped estimations of $\tilde{r}_{c,1}, \tilde{r}_{c,h}, \hat{\Sigma}_c, \gamma_1^B, \gamma_2^B$ 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

The computational cost of constructing a distributional set with bootstrap method for spatial-temporal data considered in this work is higher than that of the return model of financial assets [5, 10]. This is because $\mathcal{F}(t)$ is a function of time index $t$, the dimension of $\tilde{r}_c, \hat{\Sigma}_c$ is decided by the number of dynamic regions and prediction horizon, which can be large for applications rising in smart cities, such as taxi or autonomous driving car balancing problems and bicycle re-balancing problems.

However, the mean and covariance matrices for $t, t+1, \ldots, t + \tau$ have overlapping components: for instance, $\tilde{r}_c(t)$ and $\tilde{r}_c(t+1)$ 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 $\mathcal{F}(t), t = 1, 2, \ldots, K$ 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 $\tilde{r}_c(t)$ and $\hat{\Sigma}_c(t)$ for each index $t$.

Specifically, we define the whole day demand vector as $r = (r^1, r^2, \ldots, r^K) \in \mathbb{R}^n, n = \sum_{t=1}^{K} n_t$, i.e., a concatenated demand vector for each time slot of one day. And we denote
\( \hat{r} \) as the estimated mean of the random vector \( r \). To get all covariance component for each random vector, the process is: at \( t = 1 \), calculate the covariance of \( r_c(1) \), store it as \( \Sigma_{[1, \tau+1]} \), 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
\[
\Sigma_{[n[1, \tau+1]-n[1, \tau-k], n[1, \tau+\tau-k+1]]} = \Sigma_{[n[1, \tau+\tau-k], n[1, \tau+\tau-k+1]]}
\]
(11)
\[
= \text{cov}(r^{t+\tau-k}, r^{t+\tau}),
\]
where \( n[1, \tau+1] = \sum_{j=1}^{\tau+1} n_j \), the subscript \([b_1 : b_2 : b] \) means entries from the \( b_1 \)-th to the \( b_2 \)-th rows and \( b \)-th to the \( b_1 \)-th columns of matrix \( \Sigma \). This process of calculating \( \Sigma \) is explained in Figure 1.

Algorithm 1 Algorithm for constructing distributional sets

Input: A data set of spatial-temporal demand

1. Demand aggregating and sample set partition
Dynamically partition space, aggregate demand of each region for each time 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. Cluster \( S(S(t)) \) according to categorical information \( I_p \) and denote \( S(I_p) \subset S(S(t, I_p)) \subset S(t) \).\( p = 1, \ldots, P \) as clusters.

2. Bootstrapping mean and covariance matrix
Initialization: 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 \( S^j(I_p) = \{\hat{r}(d_1, I_p), \ldots, \hat{r}(d_N, I_p)\} \) from \( S(I_p) \) with replacement. Calculate the mean \( \hat{\Sigma}(I_p) \) and covariance \( \hat{\Sigma}(I_p) \) of the whole day demand vector of set \( S^j(I_p) \) as (11).
   End for
Get the bootstrapped mean covariance, and support of the whole day demand vector \( (1, \ldots, K_n) \),
\[
\hat{r}(I_p) = \frac{1}{B} \sum_{j=1}^{B} \hat{r}(I_p), \hat{\Sigma}(I_p) = \frac{1}{B} \sum_{j=1}^{B} \hat{\Sigma}(I_p),
\]
\[
\tilde{r}_{i,t}(I_p) = \min_{d} \hat{r}(d, I_p), \tilde{r}_{i,h}(I_p) = \max_{d} \hat{r}(d, I_p), \text{ for all } \hat{r}(d, I_p) \text{ in the subset } S(I_p).
\]
3. Bootstrapping \( \gamma_1 \) and \( \gamma_2 \) for each time \( t \) for each subset \( S^j(I_p) \) do
   (1) Get the mean and covariance vector for time index \( t \) of the bootstrapped estimation, and the \( j \)-th re-sample, from the mean and covariance matrix of the whole day demand vector in step 2: \( \tilde{r}_c(t, I_p), \hat{\Sigma}(t, I_p), \hat{\Sigma}(t, I_p), \tilde{\Sigma}(t, I_p) \).
   (2) Get values of \( \gamma_1(t, I_p) \) and \( \gamma_2(t, I_p) \) according to (12) and (13), respectively.
   End for
(3) Get the \( N_B(1 - \alpha_h) \)-th largest value of \( \gamma_1(t, I_p) \) and \( \gamma_2(t, I_p) \) as \( \gamma_1^B(t, I_p) \) and \( \gamma_2^B(t, I_p) \), respectively.
End for
Output: Distributionally uncertainty sets \( \{\hat{r}(t, I_p) \}_{t=1, \ldots, \tau} \).

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 \( p^k, k \in \{1, 2, \ldots, \tau \} \) according to the start time and GPS coordinate of the pick-up position of each trip. 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 a vector \( r^t \). When categorical information such as high demand season or low demand season of one year, different weather conditions or a combination of different contexts is available, indexed as \( I_p, p = 1, \ldots, P \), we cluster the dataset as subsets first. For step 3(1), the process of picking components from the mean and covariance matrices of the whole day demand is
\[
\hat{r}_c(t, I_p) = \hat{r}_{[n[1, \tau+1]-n[1, \tau-k], n[1, \tau+\tau-k+1]]}(I_p),
\]
\[
\hat{\Sigma}_{\gamma}(t, I_p) = \hat{\Sigma}_{[n[1, \tau+1]-n[1, \tau-k], n[1, \tau+\tau-k+1]]}(I_p).
\]
For the \( j \)-th re-sampled subset \( S^j(I_p) \), the mean and covariance matrices are \( \mathbb{E}[r_c] = \hat{r}_c(t, I_p) \) and \( \mathbb{E}[r_c^2] = \hat{\Sigma}_{\gamma}(t, I_p) \), respectively. For step 3(2), according to the definition of \( F \) in (10), we get \( \gamma_1(t, I_p) \) by the following equation
\[
\gamma_1(t, I_p) = [\hat{r}_c(t, I_p) - \hat{r}_c(t, I_p)]^\top \hat{\Sigma}_{\gamma}^{-1}(t, I_p)[\hat{r}_c(t, I_p) - \hat{r}_c(t, I_p)].
\]
(12)
According to definition (10), the left part of the inequality related to \( \gamma_2 \) satisfies that
\[
\mathbb{E}[(r_c - \hat{r}_c)(r_c - \hat{r}_c)^\top] = \mathbb{E}[r_c^2] - \mathbb{E}[r_c^2] + \mathbb{E}[r_c^2] + \mathbb{E}[r_c^2]^\top = \hat{\Sigma}_{\gamma} - \hat{r}_c \hat{r}_c^\top.
\]
Then we get \( \gamma_2 \) for index \( (t, I_p) \) by solving the following convex optimization problem
\[
\min_{\gamma_2} \gamma_2
\]
\[
s.t. \hat{\Sigma}_{\gamma}(t, I_p) - [\hat{r}_c(t, I_p)] \hat{r}_c(t, I_p)]^\top \leq \gamma_2 \hat{\Sigma}_{\gamma}(t, I_p)
\]
(13)

3.3 Dynamic Space Partitioning

A grid file [26] 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 bounded. 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 [21][22][33]. 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).

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 $(x_{\text{min}}, y_{\text{min}})$, $(x_{\text{max}}, y_{\text{max}})$ is spatially divided into adjacent disjoint nodes: $(x_{\text{min}}, y_{\text{min}})$, $(x_{\text{mid}}, y_{\text{mid}})$, $(x_{\text{mid}}, y_{\text{max}})$, $(x_{\text{max}}, y_{\text{min}})$, $(x_{\text{max}}, y_{\text{mid}})$, and $(x_{\text{min}}, y_{\text{max}})$, where $x_{\text{mid}} = a_{\gamma}(x_{\text{min}}, x_{\text{max}})$ and $y_{\text{mid}} = a_{\gamma}(y_{\text{min}}, y_{\text{max}})$. 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 [17]). 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 $\mathcal{J}_E(X^{1:}\tau, r_c)$ 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 semidefinite programming (SDP) problem examined by previous work [4, 5, 9]. Hence, the form of $\mathcal{J}_D(X^k)$ keeps the same and the process of deriving a standard convex optimization problem that equivalent to problem (8) is mainly to analyze the $\mathcal{J}_E(r^k, X^{1:}\tau)$ part, as shown in the following theorem.

THEOREM 1. The distributionally robust resource allocation problem [8] with a distribution set [10] is equivalent to the following convex optimization form

$$\begin{align*}
\min & \quad \beta(v + t) + \sum_{k=1}^{T} \mathcal{J}_D(X^k) \\
\text{s.t.} & \quad \left[ v + \frac{1}{2}(y_1^+)^T \bar{r}_{c,h} - (y_1^-)^T \bar{r}_{c,h} \right] + \frac{1}{2} (q - y_1)^T Q (q - y_1) \geq 0 \\
& \quad t \geq \left( \gamma B \Sigma_c + \bar{r}_c \bar{r}_c^T \right) Q + \bar{r}_c Q \\
& \quad + \sqrt{\gamma B ||\Sigma_c^{1/2}(q + 2Q\bar{r}_c)||_2} \\
& \quad a_{ik} \left( \frac{S_k}{y_k^{\alpha}} \right)^\alpha \leq y_k^k, y = [y_1^1, y_2^1, \ldots, y_1^1, y_2^1, \ldots, y_n^1]^T, \\
& \quad y_1 = y_1^+ - y_1^-, \quad y_1, y_2, y \geq 0, \quad Q \geq 0 \\
& \quad X^{1:}, S^{1:}, V^{2:}, O^{2:} \in \mathcal{D}_c.
\end{align*}$$

Proof. See Appendix 7.2

Specifically, with the constraints of problem [8] to represent the constraint $X^{1:}, S^{1:}, V^{2:}, O^{2:} \in \mathcal{D}_c$ in (14), 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 1. We construct distributional uncertainty sets according to Algorithm 1 solve (14), 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 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 system, or bike sharing, depending on what kind of demand data is available.

**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 variable should be included for modeling the 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 (14) 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 (23) provides a probabilistic guarantee for the worst-case performance at a single point of the demand space.

**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. For every method we compare, the distributionally robust framework (8), the robust balancing model of (23), and the non-robust vehicle balancing model of (14), we applying the optimal solution of each method to balance taxis under all testing samples of $r_c$. The minimum average cost of a second-order-cone (SOC) robust solution (23) is close to the average cost of the distributionally robust solutions of (14). 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 (14) 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 (23) 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 the area of New York city. 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

<table>
<thead>
<tr>
<th>Taxi Trip Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collecting Period</td>
</tr>
<tr>
<td>01/01/2010-12/31/2013</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Trip Information</th>
<th>Time Resolution</th>
<th>Trip Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start and end points</td>
<td>Second</td>
<td>GPS coordinates</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1: New York city data used in this evaluation section.**

<table>
<thead>
<tr>
<th>$N_B = 10$</th>
<th>$n = 50, \tau = 2$</th>
<th>$\gamma_1^B$</th>
<th>$\gamma_2^B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_B = 100$</td>
<td>$n = 50, \tau = 2$</td>
<td>0.739</td>
<td>5.24</td>
</tr>
<tr>
<td>$N_B = 1000$</td>
<td>$n = 50, \tau = 3$</td>
<td>0.013</td>
<td>1.56</td>
</tr>
<tr>
<td>$N_B = 5000$</td>
<td>$n = 50, \tau = 6$</td>
<td>0.012</td>
<td>1.49</td>
</tr>
</tbody>
</table>

**Table 2: Comparing thresholds $\gamma_1^B$ and $\gamma_2^B$ for different $N_B$ and dimensions of $r_c$.**

![Graph showing the average cost comparison of DRO and RO](image-url)
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. In the experiments, we fixed the time interval as 2 hours down to 15 minutes as shown in Table 3, and get different partitions on (longitude, latitude) dimensions. We then use demand vectors after these partitions to calculate the uncertain distributional sets, to show the effect of time-interval length on the quality of the modeling.

Table 3 shows the comparison of $\gamma_1^B$ and $\gamma_2^B$ 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.

Table 4 is a comparison of average total idle distance with distributionally robust dispatch solutions by solving (14) (equivalent form of (8)).

In Table 4, we compare the average total idle distance with distributionally robust dispatch solutions by solving (14) (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 with the distributionally robust vehicle balancing solutions, 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 Tables of Parameters and Variables

A list of parameters and variables of the distributionally robust vehicle balancing problem (8) is shown in Table 5 and a list of parameters of Algorithm 1 is shown in Table 6.

7.2 Proof of Theorem 1

**Proof.** We have \( \frac{a_{ik}}{(S_i^k)^\alpha} > 0 \) and \( r_c > 0 \) by the definitions of \( J_E \) in (4) and the demand model, then for any vector \( y \in \mathbb{R}^{n_c} \), \( y = [y_1^1, y_1^2, \ldots, y_1^n, y_2^1, \ldots, y_\tau^n]^T \) that satisfies \( 0 < \frac{a_{ik}}{(S_i^k)^\alpha} \leq y_i^k \), we also have

\[
0 \leq \sum_{k=1}^\tau \sum_{i=1}^{n_c} \frac{a_{ik} y_i^k}{(S_i^k)^\alpha} \leq y^T r_c,
\]

and the second inequality strictly holds when all \( \frac{a_{ik} y_i^k}{(S_i^k)^\alpha} = y_i^k \) for \( i = 1, \ldots, n_c \), \( k = 1, \ldots, \tau \). The constraints of problem (8) are independent of \( r_c \), hence, for any \( r_c \), the minimization problem

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

is equivalent to

\[
\begin{align*}
\min_{X^k} \quad & \beta y^T r_c + \sum_{k=1}^\tau J_D(X^k) \\
\text{s.t.} \quad & y = [y_1^1, y_1^2, \ldots, y_1^n, y_2^1, \ldots, y_\tau^n]^T, \\
& x^{1:\tau}, s^{1:\tau}, v^{2:\tau}, o^{2:\tau} \in D_c
\end{align*}
\]

In this proof, we use the objective function of problem (15). In particular, only the part of \( y^T r_c \) is related to \( r_c \), and we first consider the following maximization problem

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

By the definition of problem (8) and problem (15), only the objective function includes the random vector \( r_c \), and is concave of \( X^k \) for \( k = 1, \ldots, \tau \). The distributional set \( \mathcal{F} \) constructed by Algorithm 1 in the domain of \( y \), \( x^{1:\tau}, s^{1:\tau}, v^{2:\tau}, o^{2:\tau} \) are convex, closed, and bounded sets. Hence, problem (16) 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

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

(17)

Hence, we first analytically find the optimal value of problem (17). 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

$$\begin{align*}
\max_{r_c} & \quad y^T r_c - r_c^T Q r_c - r_c^T q \\
\text{s.t.} & \quad \hat{r}_{c,l} \leq r_c \leq \hat{r}_{c,h}.
\end{align*}$$

(18)

For a positive semi-definite $Q$, the optimal solution of problem (18) exists. The Lagrangian of (18) under the constraint $y_1^+, y_1^- \geq 0$ is

$$L(r_c, y_1^+, y_1^-) = y^T r_c - r_c^T Q r_c - r_c^T q + (y_1^+ - y_1^-)^T r_c$$

$$\quad - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}.$$ 

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

$$\sup_{r_c} L(r_c, y_1^+, y_1^-) = \frac{1}{4} (q - y - y_1) Q^{-1} (q - y - y_1)$$

$$\quad - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h},$$

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

Then the first inequality constraint of problem (17) for any $\hat{r}_{c,l} \leq r_c \leq \hat{r}_{c,h}$ is equivalent to

$$v \geq \frac{1}{4} (q - y - y_1) Q^{-1} (q - y - y_1)$$

$$\quad - (y_1^+)^T \hat{r}_{c,l} + (y_1^-)^T \hat{r}_{c,h}.$$ 

By Schur complement, the above constraint is

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

Together with other constraints, the equivalent convex optimization form of problem (8) is problem (14).