Network Science-aided Online Vehicle Dispatch for Ride-hailing Platform

Jiaman Wu, Ruiting Wang, Jiaxi Huang

November 2021

Abstract

Ride-sharing helps to reduce carbon emissions as it reduces the number of private cars. Nowadays, more attention has been attached to the ride-sharing matching algorithm design since it determines the upper bound of the value of ride-sharing platforms in reducing carbon emission. In this project, network science is implemented as a fundamental tool to characterize and handle the uncertainty in sharing mobility of the City of Chicago, with a goal to approximate the optimal vehicle dispatch for ride-hailing platforms. We first explore the structure of the shareability network, then we identify important orders based on the exploration of shareability network structure. Then we assign different weights according to orders' importance and propose an online vehicle dispatch algorithm. At last, numerical studies show that network metrics are important to vehicle dispatch algorithm design. Our results indicate that under the framework of the shareability network, the rider-driver matching rate can be maximized with the help of the proposed online vehicle dispatch algorithm.

1 Introduction

In the global practice of reducing carbon emissions and promoting a more sustainable future, transportation decolonization plays a crucial role. Reducing the number of private cars and increasing the proportion of public transportation in urban areas are of great significance. But to what extent could we reduce the number of private cars? Vazifeh *et al.* addressed the question in [13] with the answer that only 30% of cars we have now are actually needed. However, this striking number is estimated under an idealized setting, while in real-life applications, much more constraints and uncertainty need to be considered. In this work, we use taxi trips dataset from the Chicago Data Portal, which collects around 197 million taxi rides within the City of Chicago over the year 2020, to conduct online dispatch algorithm.

1.1 Literature Review

The number of private cars burdens carbon emissions, while ride-sharing, as a kind of public transportation, provides an alternative solution for urban mobility. Ride-hailing platform, for example, increases mobility for car-free households and exerts long-term impact on the environment and energy consumption [11].

This prompts investigations into the potential of ride-hailing to be more efficient for optimal dispatch design. It has been shown that the optimal dispatch problem is NP-hard to solve. Cheng *et al.* proposed utility-aware ridesharing on road networks [5] and approximate this problem using a bilateral arrangement algorithm, greedy algorithm, and a grouping-based scheduling algorithm. Javier *et al.* conducted analysis on real-time high-capacity ride-sharing data which starts from a greedy assignment and improves it through a constrained optimization [1]. Results show that this model is able to quickly converge to the optimal assignment over time, considering tradeoffs between fleet size, waiting time, travel delay, etc.

But given uncertainties, in reality, it is also necessary to design online dispatch algorithms. Existing algorithms take tradeoffs between effectiveness and efficiency, either ignoring future demands to guarantee real-time or solving a complex combinatorial optimization to improve service rate. In recent works, researchers switched gears to the network sciences method. Linkages between urban studies and network sciences have dramatically increased over the years[6]. Manik *et al.* study the effects of the underlying street network topology on the viability of ride bundling in [10]. Xu *et al.* constructed a vehicle-shareability network in a min-cost flow to find the optimal dispatch in offline cases, and a multi-sample multi-network flow-based algorithm is proposed for online dispatching[14]. A case study in Chengdu conducted by Tu *et al.* also showed that the shareability-network-based ride-splitting trip identification algorithm increased both potential savings and the number of shared trips[12].

In contrast to previous works, we notice that for the formulation of a shareability network, it is also crucial to evaluate the significance of each trip. However, it remains unclear which trips are more critical than others and how they will affect the functionality of the network. In the meantime, network metrics are not fully explored in the online vehicle dispatch problem. Therefore, we seek to explore the structure of the shareability network. Then based on this, we combine network flow formulation and learning framework to improve the platform efficiency facing uncertainty.

1.2 Our Contributions

In this work, the goal is to maximize the matching rate of the ride-hailing platform under uncertainty of the demand side. Specifically, our contribution can be summarized as follows.

- Shareability Network Analysis: we visualize the network and compute the network metrics. To be specific, we include the degree of centrality, closeness, betweenness and provide analysis on network metrics.
- Critical Order Identification: we identify the importance of different orders using network metrics and design weight for the online dispatch algorithm.
- Network-aided Dispatch Algorithm Design: we design an online algorithm for the ride-hailing platform. Specifically, we first propose the offline algorithm, then adjust the weight of orders to embed future information in the current operation.

The remainder of our paper is organized as follows. Section 2 introduces the shareability network and provides visualization analysis. Section 3 illustrates the offline and online vehicle dispatch algorithm. In Section 4, we evaluate the performance of the proposed framework with real data, while in Section 5 we conclude this work and suggest some interesting future research directions.

1.3 Team Work Statement

Our work statement is as follows:

- Jiaman Wu: Offline benchmark design and re-optimization algorithm design. Vehicle-shareability network metrics exploration analysis. Code for offline, online algorithm and network metrics analysis.
- Ruiting Wang: Trip weight design and calculation for shareability network. Calculation of the marginal benefit of each trip.
- Jiaxi Huang: Chicago data processing and EDA. Ground truth calculation. Model prediction and evaluation. Network models fitting. Visualization.

2 Vehicle-shareability Network Overview

2.1 Vehicle-shareability Network Formulation

We consider the trip dispatch problem for the vehicle-sharing platform, with three major participants: the platform, the drivers, and the passengers. The passengers send trip orders to the platform such that the platform could dispatch the drivers according to the collected order information.

To highlight the value of future information in vehicle dispatch, we assume the drivers will obey the platform's dispatch order, and the consumers will truthfully submit their order information without changing their mind afterwards. Specifically, suppose the platform collects a set $\mathcal{O}$ of N trip orders $O_i(t_i^o, t_i^d, o_i, d_i), i = 1, ..., N$, where t_i^o denotes the expected pick-up time; t_i^d denotes the expected drop-off time; o_i and d_i represent the trip origin and destination, respectively. The platform also has access to the location information of the drivers: p_i^t describes the position of the driver i at time t.

The vehicle-shareability network is represented by a directed graph $G(V, E)$, where V is the set of nodes and E is the set of links. Each node $v \in V$ corresponds to a trip $O_i \in \mathcal{O}$ and each link $(u, v) \in E$ denotes the connectivity between the nodes u and v, i.e., the connectivity between two trips corresponds to the connectivity between two nodes. In particular, two trips (O_i and O_j) are connected if there exists at least one driver, who can arrive at the pick-up location no later than the expected pick-up time of O_j after finishing O_i, i.e., $t_i^d + t_{ij} \leq t_j^o$. Note that t_{ij} denotes the traveling time between O_i and O_j. Since an edge means two trips can be consecutively served by a single driver, a path in the graph G corresponds to the trip orders that can be satisfied by a single driver.

One toy example for this shareability network is shown in Figure 1. The green and red routes represent two subsequent orders. The first order (the green route) starts at 9 am and ends at 10 am, while the second-order (the red route) starts at 11 am and ends at 12 pm. The yellow route indicates that it takes 30 min to travel from the drop-off location of the first order to the pickup location of the second order. Since the travel time is less than the gap time between two orders, we abstract orders 1 and 2 using squares. In other words, these two trips correspond to two nodes in the vehicle-shareability network. There is a link between these two nodes because they can be shared by one driver.

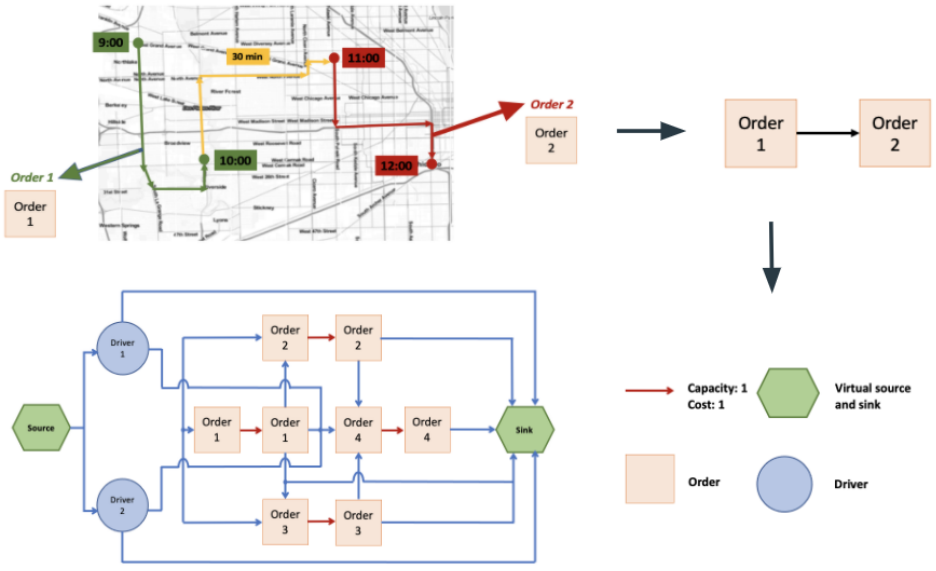


Figure 1: Toy Example for Shareability Network Flow Formulation

2.2 Vehicle-shareability Network Metrics Illustration

Taxi drivers usually prefer trips that lead them to busy areas with higher demand. In this heuristic human decision process, there is an implicit evaluation of the value of each trip. Some trips are viewed as more valuable for increasing drivers' exposure to more trips and orders. In order to maximize the matching rate of the ride-hailing platform under uncertainty of the demand side and supply side, it is essential to identify those valuable tips and design the weight of trips in a delicate matter in the application of the network-flow formulation, such that it is a more accurate representation of the complexity of the real case. This concept of node value is compatible with the concept of centrality in network science. The notion of centrality was first introduced in social-network analysis to determine the rank of each actor corresponding to their network position [7]. The importance of nodes in networks of different nature is evaluated from variable perspectives in literature, including degree centrality, closeness, betweenness, eigenvector centrality, current flow closeness, current flow betweenness, and Katz centrality [9], etc.

Borgatti et al. compared different centrality measures in their works [2], [3], and raised that the popular application of centrality measures, when disregarding the implicit assumptions about how traffic flows through a network, could result in "wrong" interpretations. One example is that closeness centrality and betweenness centrality, defined by Freeman [8], [7] is based on geodesic paths, which assume flows moves along the shortest possible paths.

A variety of well-defined and comprehensively-studied network centrality metrics are taken into consideration, but designing trip weight for shareability networks requires more case-specific analysis, which is also the main contribution of this paper. The definition of a shareability network involves the notion of nodes and edges being somewhat peculiar from the traditional traffic network: nodes in this network are orders, which have attributed including geographically different start and end locations with respective time stamps; edges in this network represent drivers' accessibility of orders. In this project report, we present two different ways of applying network metrics into designing trip weight and solving the dispatching problem:

1) The preliminary study in Section 3.2.2, where we proposed several model-free and model-based functions to capture trip importance with existing centrality measures.

2) In the future work section, Section 5, we proposed a trip-importance index based on the "marginal benefit" of each trip in the batch-by-batch re-optimization model. Existing network centrality measures were calculated and used to evaluate the "marginal benefit" of each trip using a learning algorithm.

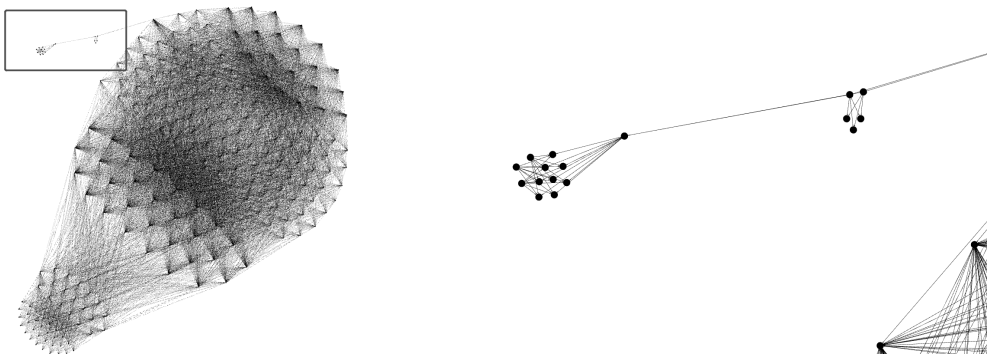
2.3 Vehicle-shareability Network Metrics Exploration Analysis

In this section, we aim to explore the network structure by analyzing the metrics of the network. We first visualize the shareability network of one day during January. Fig. 7 shows that there exist some critical orders so if we remove critical orders from the network, many orders cannot be served. This implies that some orders should be assigned with more weights than others during dispatch.

Fig. 3 plots five metrics of the shareability network in January. For each day, we compute metrics for all orders and show their distribution. We observe that network metrics show periodicity. For the Eigen centrality and Katz centrality, the outliers, as well as the interquartile range, follow a weekly periodicity. For degree centrality, in-degree centrality, and closeness, the distribution also follows a weekly periodicity. Besides, the in-degree centrality and closeness share similar distribution. This implies that we can predict metrics using historical data.

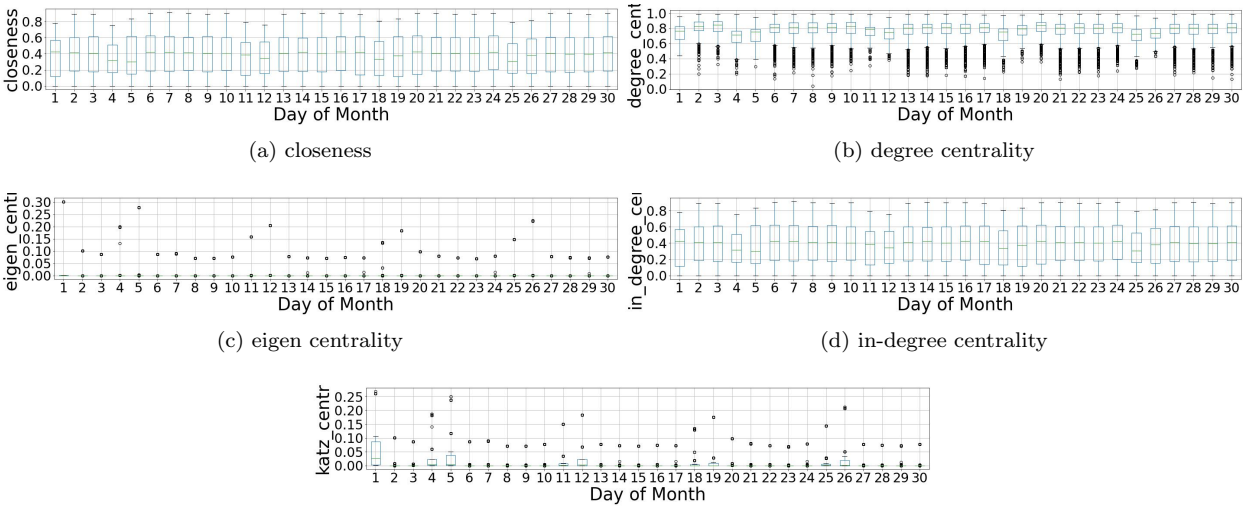
2.4 Network Models Comparison

We constructed a directed shareability network using empirical order data throughout January 2020 and visualized in Figure 4 and Figure 5. In the same plot, we also applied Small-Worlds(SW) Model, Barabasi-Albert(BA) Model, and Erdos-Renyi(ER) Model as comparisons. Among these three models, Barabasi-Albert model best captured the empirical data features.



(a) global (b) zoom

Figure 2: Network Structure of One Day in January.



(a) closeness (b) degree centrality

(c) eigen centrality (d) in-degree centrality

(e) katz centrality

Figure 3: Boxplot of Network Metrics during January.

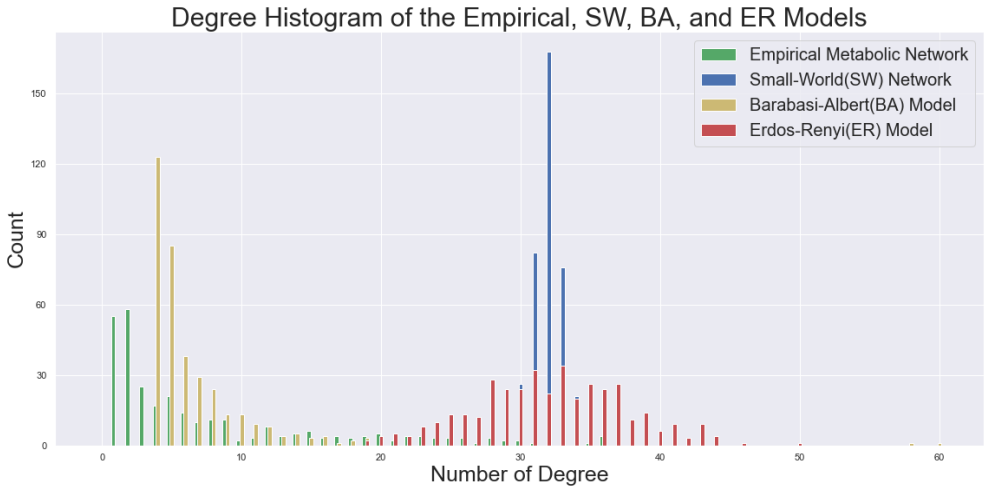


Figure 4: Degree Histogram of Order Shareability Network during January

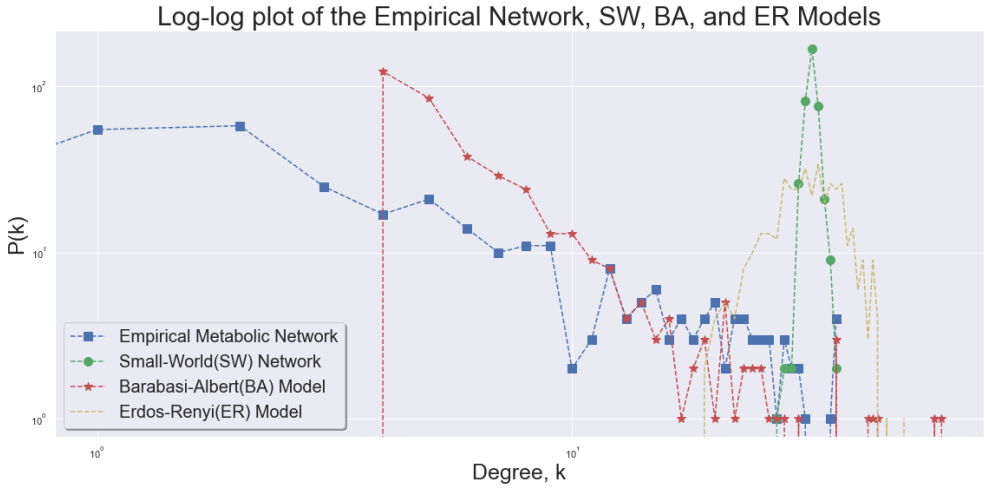


Figure 5: Log-log Degree Distribution of Order Shareability Network during January

3 Network-aided Dispatch Algorithm Design

3.1 Dispatch Algorithm with Perfect Information

A flow network is a directed graph $G = (V, E)$ with a source vertex $s \in V$ and a sink vertex $t \in V$, where each $(u, v) \in E$ has capacity $c(u, v) > 0$, flow $f(u, v) \geq 0$ and cost $a(u, v)$. The cost of sending this flow along an edge (u, v) is $f(u, v) \cdot a(u, v)$. The problem requires an amount of flow d, i.e., demand, to be sent from source s to sink t. The definition of the problem is to minimize the total cost of the flow over all edges:

$$\sum_{(u,v)\in E} a(u, v) \cdot f(u, v) \tag{1}$$

with capacity constraints, skew symmetry constraints, flow conservation constraints and required flow constraints as follows:

$$\begin{aligned} s.t. \quad & f(u, v) \leq c(u, v) && (2)\\ & f(u, v) = -f(v, u) && (3)\\ & \sum_{w\in V} f(u, w) = 0 \text{ for all } u \neq s, t && (4)\\ & \sum_{w\in V} f(s, w) = d \text{ and } \sum_{w\in V} f(w, t) = d. && (5) \end{aligned}$$

For our problem, we consider a directed graph $G(V, E)$. The node-set V contains three kinds of node sets. Each trip order can be represented by two nodes, i.e., the origin node and the destination node. We group all the trip orders in set $\mathcal{O}$. The driver node-set $\mathcal{C}$ represents all available drivers. Besides these sets, we need a virtual source $s \in V$ and a virtual sink $t \in V$. There are four kinds of edges in the graph. Each trip node has an internal link to represent the trip's origin and destination and we denote the set of all internal links as $\mathcal{I}$. Each driver node has a directed link to the virtual source and sinks while each trip node has a directed link to the virtual sink. Each edge (u, v) represents the connectivity between two trips (drivers and trips).

Denote the flow on each edge (u, v) by $f(u, v)$, and we can correspondingly define the flow capacity $c(u, v)$ and the unit cost $a(u, v)$ for the flow. In addition, we require uniform capacity for all links,

$$c(u, v) = 1, \quad \forall u, v \in V, \tag{6}$$

and we require the binary cost for each edge,

$$a(u, v) = \begin{cases} 0 & (u, v) \notin \mathcal{I} \\ 1 & (u, v) \in \mathcal{I}. \end{cases} \tag{7}$$

The demand d is the amount of flow that needs to be sent from the source to the sink, i.e., the number of available drivers. The source and sink nodes s and v enforce the demand dispatch. To enforce a demand of 2, we can simply inject a flow of 2 into s. The solution to the maximum cost flow corresponds to the optimal dispatch. This allows us to formulate the optimal offline dispatch problem as follows:

$$\begin{aligned} \max \quad & \textstyle\sum_{(u,v)\in E} a(u, v) f(u, v) \\ \text{s.t.} \quad & f(u, v) \leq c(u, v), \quad \forall (u, w) \in E, \\ & f(u, v) = -f(v, u), \quad \forall (u, w) \in E, \\ & \textstyle\sum_{w\in V} f(u, w) = 0, \quad \forall u \neq s, t \\ & \textstyle\sum_{w\in V} f(s, w) = d, \quad \sum_{w\in V} f(w, t) = d, \\ & \textstyle\sum_{w\in V} f(u, w) \leq 1, \quad \forall u \in \mathcal{O}, \mathcal{C}, \\ & f(v, u) \in \{0, 1\}, \quad \forall (u, w) \in E. \end{aligned} \tag{8}$$

The objective function aims to maximize the matching rate, i.e., the number of passenger-driver couples. From the definition of network flow, the constraints can ensure that each order can only be served by one driver. All drivers will be considered in the optimization problem. Each driver can only serve one order at the same time. At each time, a driver can do binary choice, serve or not serve an order. In this case, we can solve the maximum-cost flow problem using network simplex. We term this procedure as an offline benchmark algorithm (OBA).

3.2 Dispatch Algorithm with Imperfect Information

3.2.1 Re-optimization Framework

In the real world, perfect information within the whole operation time is usually not available. In this section, we provide a metrics-based re-optimization framework, which constantly applies OBA as new demand information becomes available. This solution is used to decide the vehicles' actions in an online fashion.

Specifically, the whole dispatch period is denoted as T. To model the vehicle dispatching over time, we divide T into k smaller time batches $\{T_1, ..., T_k\}$. We only know the orders within each batch T_i, at the beginning of batch i. Then for each batch, we constantly apply the OBA to match orders and drivers. At the end of the batch i, we update the positions of all drivers and receive new orders' information in batch $i+1$. We term this algorithm as the re-optimization benchmark algorithm (ROBA).

3.2.2 Weight-adjusted Re-optimization Framework

The key concept of our metrics-based online algorithm consists of two parts, respectively the learning-based metrics prediction and the metrics-based weight optimization.

To capture order information in the near future, we assign weights $w(u,v), (u,v) \in \mathcal{I}$ for all orders in current batch, which correspond to the cost $a(u,v), (u,v) \in \mathcal{I}$ in our network formulation. The optimization goal is to maximize the overall weight of the matrix, the weight $w(u,v), (u,v) \in \mathcal{I}$ needs to be carefully designed so that it includes the key information of the network. The question we aim at answering in weight design is, which trips increase the drivers' chance of getting a subsequent trip? The hidden assumption is that some trips are more important, or even crucial in increasing the supply and demand match rate and we hope to identify those trips and assign them with higher weight, with our metric-based online algorithm.

The assigned weights $w(u,v), (u,v) \in \mathcal{I}$ for all orders in current batch are correspond to the cost $a(u,v), (u,v) \in \mathcal{I}$ in our network formulation. Serving orders with higher weights can bring a higher matching rate in the whole operation period.

The weight design is based on the metrics of each trip $O_i \in \mathcal{O}$. Due to the characteristics of the specific network built based on the data set, certain metrics cannot be calculated. The considered metrics for weight design include degree centrality, in-degree centrality, out-degree centrality, closeness, Katz centrality, and eigenvector centrality.

These metrics evaluate the importance of nodes from various aspects. By nature, we would like to adhere more importance and higher weight to those trips with higher to all centrality-related metrics. Different combinations of these metrics for the weight design are presented in the following functions. A trial-and-error approach is adopted to determine the final weight design.

$$w_{i1}(u,v) = m_{clo}(O_i) m_{cen}(O_i) \tag{9}$$

$$w_{i2}(u,v) = \alpha m_{clo}(O_i) + (1-\alpha) m_{cen}(O_i) \tag{10}$$

Where $m_{clo}(O_i)$ refers to the closeness of trip O_i and the $m_{cen}(O_i)$ represents the centrality-related metrics of trip O_i. Specifically, $m_{cen}(O_i)$ can be one of the centrality-related metrics among degree centrality, in-degree centrality, out-degree centrality, Katz centrality, and eigenvector centrality, or the average of all centrality-related metrics. A set of different values of α has been tried from range 0.1 to 0.9 with an interval of 0.2. In all, 36 different weight designs have been tried out on a selected data set.

With weight $w_i(u, v)$, we construct graph $G(V, E)$ for orders and drivers in the batch. Then we use the OBA to obtain the optimal dispatch for the batch. We directly apply for the dispatch order and update the drivers' information (i.e., location at the end of T_i) for the dispatch in the next time slot T_{i+1}. Note that such a re-optimization method enjoys remarkable performance in practice [4], and in some special cases, has provably asymptotic optimal performance [15]. We term this algorithm as a learning-aided re-optimization algorithm (LROA).

4 Numerical Studies

4.1 Data Description

Our taxi trips dataset was downloaded from the Chicago Data Portal, which collects around 197 million taxi rides within the City of Chicago over the year 2020. To preserve records with both valid census tract ID and locations, rides that contain NaNs were dropped. Among the 23 features provided in this dataset, a few columns appeal to our interest. Here are the features we will mainly focus on in this project:

Table 1: Features of Interest and Definitions

Feature	Definition
Taxi ID	A unique identifier for the taxi.
Pickup Location	Census-tract-level longitudes and latitudes where trip starts.
Dropoff Location	Census-tract-level longitudes and latitudes where trip ends.
Pickup Timestamp	Trip starts timestamp rounded to the nearest 15 minutes.
Dropoff Timestamp	Trip ends timestamp rounded to the nearest 15 minutes.
Pickup Census Tract	Census Tract ID where the trip started.
Pickup Census Tract	Census Tract ID where the trip ended.
Trip Duration	Unit in seconds.
Trip Distance	Unit in miles.

4.2 Network Metrics Prediction Analysis

4.2.1 Visualization of Apparent Metrics

We select 4 features from the taxi trips dataset, which are: trip distance, trip speed, trip duration, and the number of trips per hour. This step is to sketch the shapes of overall metric distributions. According to Figure 1, we notice long-tailed property in the distributions of trip distance and trip duration, which indicates most rides happen below certain temporal and spatial thresholds, while only a small number of trips are long in time and distance. Also, the distribution of trip speed shapes like normal and centers around 9 mph, despite some erroneous zeros that might be misreported. As for the daily pattern of trips, the majority of trips happen during the daytime and peak around 18 PM. This population-level data pattern delineates the typical traveling habits related to working, schooling, and returning home.

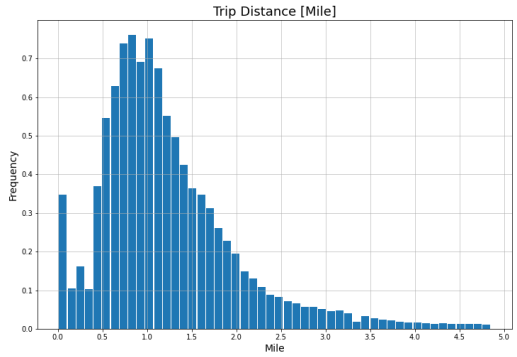
4.2.2 Model Hyper-parameters and Learning Accuracy

Our neural network architecture is adjusted by two key hyper-parameters: hidden layer sizes and initial learning rate. We designed tests to evaluate 5 different levels of each hyper-parameter using taxi trips data from January 2020. For each objective metric, we calculated mean absolute error (MAE), mean squared error (MSE), and root means square error (RMSE) respectively. A trial with the lowest errors is considered to be the optimal set of hyper-parameter for that specific metric. Here, errors are normalized using the difference between max and min of metric ground truth (max(y)-min(y)). We ran 175 trials in total to reduce model errors. As result, we listed the sets of hyper-parameters with the best performance in Table 2. Using the best set of hyperparameters, the predictions and ground truths for each metric are plotted in Figure 5.

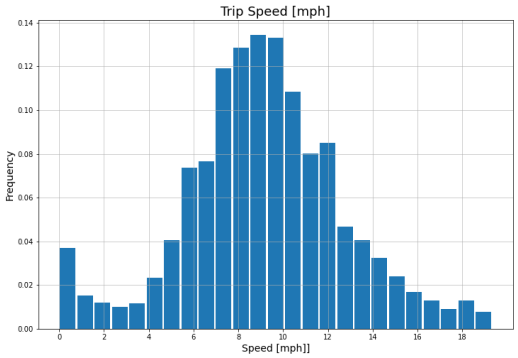
4.3 Network-aided Dispatch Algorithm Evaluation

4.3.1 Dispatch Algorithm Weight Design

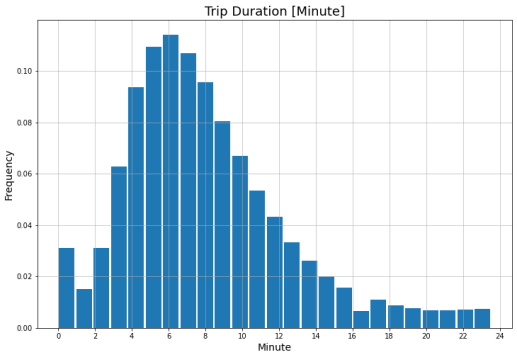
Grid-search method was applied to the weight design function listed, (9) and (10) in Section 3.2. The parameter setting of this study includes 800 drivers, 4672 orders in 4-hour morning peak of a day, maximum



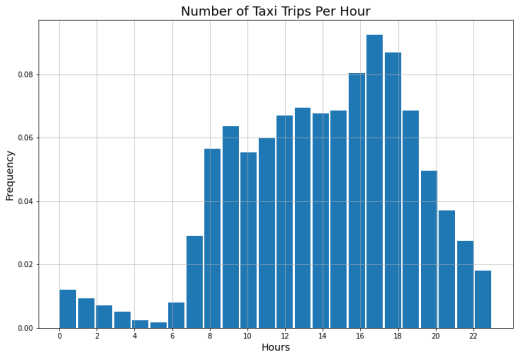
(a) Trip Distance Distribution



(b) Trip Speed Distribution



(c) Trip Duration Distribution



(d) Hourly Trips Distribution

Figure 6: Distributions of Key Apparent Features.

travel-without-passenger time of driver being 60 minutes, and a batch size of 60 minutes. Among all the combinations, we found that (11) has the best performance in dispatching vehicles to improve the matching rate. See the appendix for a detailed performance comparison of different weight designs.

$$w_i(u,v) = 0.3m_{clo}(O_i) + 0.7m_{cen}(O_i) \tag{11}$$

4.3.2 Network-aided Dispatch Algorithm Robustness Analysis

To further verify the robustness of the newly designed network-aided dispatch algorithm with weight design based on network metrics, different parameters setting, which represents different real-world scenarios has been tested. Table 3 shows the matching rates of both the re-optimization algorithm and the newly designed network-aided dispatch algorithm. It can be seen from the table that, while driver distribution, batch size, and allowed void time are at variance, the performance of the newly designed network-aided dispatch algorithm remains stable. The matching rate of the new algorithm proposed is always higher than it of the baseline re-optimization algorithm, from 0.2% to 1.1%.

Table 2: Optimal Hyperparameter Settings for Metrics

Metrics	Hidden Layer Sizes	Learning Rate	NMAE	NMSE	NRMSE
closeness	(64, 16)	0.01	5.166	30.035	5.777
current flow centr	(64, 16)	1.00E-06	18.138	107.506	18.260
degree centr	(128, 16)	1.00E-05	7.707	51.956	7.799
eigen centr	(128, 32)	0.01	17.037	54.022	26.533
in degree centr	(64, 8)	0.01	5.145	30.077	5.781
katz centr	(64, 8)	0.01	17.679	57.016	27.288
out degree centr	(64, 8)	0.0001	4.298	23.761	4.909

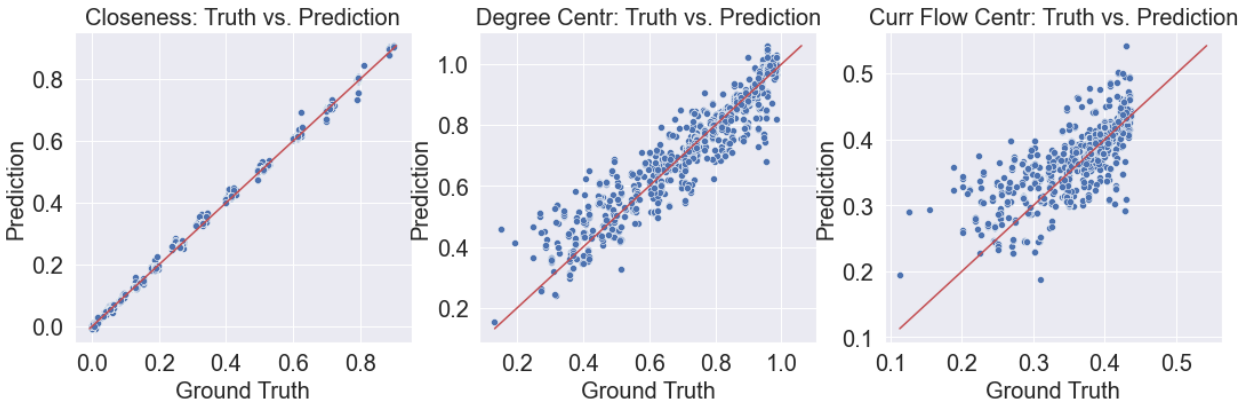


Figure 7: Scatter Plots for Closeness, Degree Centrality, and Current Flow Centrality

Table 3: Algorithm Matching Rate (MR) under Different Parameter Settings

Driver Distribution	Batch Size	Travel W/O Passengers	ROBA MR	LROA MR
norm	30	60	7.6%	7.8%
norm	60	60	30.6%	31.8%
norm	120	60	36.9%	37.2%
log	30	60	7.5%	7.9%
log	60	60	30.7%	31.8%
log	120	60	36.9%	37.2%
half	30	60	7.5%	7.9%
half	60	60	30.8%	31.8%
half	120	60	36.8%	37.2%

5 Conclusions and Future Work

To relieve the burden of the transportation sector and improve platform efficiency, we seek to reveal the value of information in vehicle dispatch. We formulate the problem in the network flow framework to maximize the matching rate, then we explore the structure of the shareability network. Based on the analysis of the shareability network, we design a learning-based algorithm. Numerical studies show that the network metrics are very important to the vehicle dispatch algorithm.

The current weight design method disregards the embedding implication in network centrality measures. With the different nature of shareability networks, this interpretation may lead to an undesired result. In this optimization problem, our ultimate goal is to maximize the matching rate of order dispatching. We define the marginal benefit of each trip being the negative effect brought to the baseline network matching rate by removing a trip (a vertex) from the network. The baseline network matching rate is captured by the ROBA. Due to the computational complexity of the marginal benefit of each trip, which brings difficulty to online and in time dispatching, we proposed using existing network centrality measures to evaluate the "marginal benefit" of each trip with a learning algorithm.

References

[1] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[2] Stephen Borgatti and Martin Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28:466–484, 10 2006.

[3] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[4] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, NY, 2013.

[5] Peng Cheng, Hao Xin, and Lei Chen. Utility-aware ridesharing on road networks. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1197–1210, 2017.

[6] Ben Derudder and Zachary Neal. Uncovering links between urban studies and network science. *Networks and Spatial Economics*, 18(3):441–446, 2018.

[7] Linton Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 03 1977.

[8] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978.

[9] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, March 1953.

[10] Debsankha Manik and Nora Molkenthin. Topology dependence of on-demand ride-sharing. *Applied Network Science*, 5(1):1–16, 2020.

[11] Alejandro Tirachini. Ride-hailing, travel behaviour and sustainable mobility: an international review. *Transportation*, 47(4):2011–2047, 2020.

[12] Meiting Tu, Ye Li, Wenxiang Li, Minchao Tu, Olivier Orfila, and Dominique Gruyer. Improving ridesplitting services using optimization procedures on a shareability network: A case study of chengdu. *Technological Forecasting and Social Change*, 149:119733, 2019.

[13] Mohammed M Vazifeh, Paolo Santi, Giovanni Resta, Steven H Strogatz, and Carlo Ratti. Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, 557(7706):534–538, 2018.

[14] Yuhang Xu, Wanyuan Wang, Guangwei Xiong, Xiang Liu, Weiwei Wu, and Kai Liu. Network-flow-based efficient vehicle dispatch for city-scale ride-hailing systems. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[15] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. The power of predictions in online control. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1994–2004, Vancouver, Cananda, 2020. Curran Associates, Inc.