electronics

MDPI

Article

A Distributed Scheme for the Taxi Cruising Route Recommendation Problem Using a Graph Neural Network

Ying Li *,†, Yongsheng Huang *,†, Zhipeng Liu and Bin Zhang

School of Software, Northeastern University Shenyang, Shenyang 110167, China; 2101244@stu.neu.edu.cn (Z.L.); zhangbin@mail.neu.edu.cn (B.Z.)

* Correspondence: liying@mail.neu.edu.cn (Y.L.); 2371447@stu.neu.edu.cn (Y.H.); Tel.: +86-152-9791-7911 (Y.H.)

† These authors contributed equally to this work.

Abstract: Despite considerable research efforts being devoted to the taxi cruising route recommendation (TCRR) problem, existing studies still have some shortcomings. To begin with, the competition and collaboration between taxis are not sufficiently taken into account. Furthermore, the TCRR is heavily reliant on potential taxi demand, which is time-variant and difficult to accurately predict due to the underlying spatiotemporal correlation and dynamic traffic patterns. Moreover, the consideration of competition and cooperation among taxis increases the complexity of the TCRR problem, making conventional centralized algorithms computationally expensive. In this paper, we first formulate TCRR as a biobjective optimization problem to balance the collaboration and competition between taxis. Subsequently, we forecast short-term taxi demand using the proposed long-short-term-memory-based graph convolutional network (LSTM-GCN), which considers diverse factors such as road topology, points of interest (POIs), and multiple time-scale features. Lastly, we propose a distributed algorithm based on a Lagrange dual decomposition. The experimental and simulation results demonstrate that our TCRR scheme performs better than any other counterpart, (i) resulting in a 3% reduction in idle taxis per hour, (ii) performing four times faster than the centralized algorithms to obtain the optimal solution, and (iii) resulting in a 7% increase in average profit.

Keywords: TCRR; optimization; LSTM-GCN; distributed algorithm

check for updates

Citation: Li, Y.; Huang, Y.; Liu, Z.; Zhang, B. A Distributed Scheme for the Taxi Cruising Route Recommendation Problem Using a Graph Neural Network. *Electronics* **2024**, *13*, 574. https://doi.org/10.3390/electronics13030574

Academic Editor: Nikolay Hinov

Received: 27 November 2023
Revised: 14 January 2024
Accepted: 23 January 2024
Published: 31 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As an essential transportation mode in metropolitan areas, taxis offer a convenient and flexible way to travel in daily life. Over the past decades, several efforts have been dedicated to researching real-time taxi dispatch to help taxi drivers and passengers work well with each other. Notably, some techniques have been successfully implemented in real life, such as the ride-hailing applications Uber and Didi [1–3]. However, substantial amounts of fuel and time are consumed when taxis cruise on the road looking for passengers, without any offers from ride-hailing applications. Therefore, research on taxi cruising route recommendation (TCRR) has attracted considerable attention [4–7].

TCRR has several obvious benefits. On the one hand, taxi drivers can find passengers in a shorter amount of time by using an accurate recommendation rather than by randomly looking for them. The fuel consumption caused by aimless cruising can therefore be reduced. This can improve the service quality of taxis by decreasing passengers' waiting time and alleviating order cancellations caused by long delays on the way to the pick-up location.

However, TCRR is challenging since many problems should be addressed when employing TCRR. For instance, where can taxi drivers find potential passengers? Where is the destination? How do taxi drivers choose a passenger from multiple candidates? Which route should a taxi driver take to reach the pick-up location? Despite the many recently proposed solutions to problems in TCRR [1,5,8,9], they still have the following shortcomings.

Firstly, existing research mainly focuses on maximizing the individual profit of taxis with less attention being paid to competition and cooperation among taxis, which may affect an individual's profit [2,8,10]. In practice, competition rises between taxis at popular pick-up areas, when there is an oversupply. Some taxis have no passengers to pick up at these locations, which results in traffic jams and a shortage of taxis available in some areas of the city [11]. However, it is worth nothing that the maximization of global profit does not guarantee that each taxi driver will be able to increase their individual profit. Global and individual profit maximization impact each other. Therefore, efforts can be made to provide a suboptimal solution for a taxi driver who has multiple options but to provide the best solution for the taxi with a single option.

Secondly, to predict potential passengers, some studies employ statistical learning methods [4,12]. Others use techniques such as time series models or machine learning methods [4,13,14]. Despite the fact that many studies have been conducted on traffic prediction (e.g., traffic flow), using a graph neural network (GNN) [15–17] to study taxi demand still has potential. Compared to convolutional/recurrent neural networks (CNNs/RNNs) [3,18–20], a GNN has the nature of expressing a road network topology with its graph-structure-based input. Thus, for traffic prediction tasks like traffic flow prediction [21], GNN has been commonly employed. In addition, traffic flow prediction and taxi demand prediction are different. For instance, taxi demand is more correlated with time. According to a case study (further details on the data are illustrated in Section 5.1), traffic flow prediction methods cannot be directly applied to forecast taxi demand since the level of spatial correlation between taxi demand in different zones does not strictly follow the distance pattern. Additionally, taxi demand presents a different variation tendency under various time scales. As far as the authors know, the study that is most relevant to this paper was conducted by Xu et al. [22], who designed a spatiotemporal multigraph convolution network (ST-MGCN) to predict region-level taxi demands. However, multiple time-scale features were not considered in that study.

Thirdly, the methods developed in most research are centralized approaches. These methods are feasible for cruising route recommendations for an individual taxi, but they consume many computational resources when recommending routes for thousands of taxis in a whole road network. In particular, when considering both competition and collaboration among taxis, the situation becomes more complicated. Notably, in Section 3, TCRR is modeled as an analogue of a resource allocation problem that is NP-hard. Therefore, it is difficult to satisfy the timeliness requirement of TCRR and simultaneously achieve an optimal solution with a centralized strategy.

To address the above challenges, we formulated TCRR as a biobjective optimization problem, minimizing the unoccupied taxi count while maximizing global revenue. It should be noted that the suggested solution can be readily expanded to encompass a more general case of ride-sharing, wherein both taxis and private vehicles are taken into account. The first objective aims to increase the individual profit of taxis. Meanwhile, the lower the number of idle taxis, the greater the number of passengers served. The second objective is the result of the synergy considering both collaboration and competition among taxis. After providing a recommended pick-up zone to a taxi, the shortest route to that zone is recommended to idle taxis. The main contributions of this research are summarized as follows:

- We developed a long short-term memory graph convolutional network (LSTM-GCN) for predicting traffic demand. It formulates the input as graph structure data based on the graph model of the road network segmented by the proposed Girvan–Newman and speaker–listener label propagation algorithm (GN-SLPA). The predictor captures the spatiotemporal correlation and dynamic patterns of taxi demand with consideration for POI information, road topology, and multiple time-scale features.
- We designed a distributed Lagrange dual decomposition (LDD)-based algorithm to acquire the optimal solution to the TCRR problem, which can be simultaneously implemented at each taxi and server terminal client. Based on the theoretical complexity

analysis, the proposed distributed algorithm is less costly than centralized algorithms. This makes our TCRR scheme more scalable in large road networks.

- The experimental results show that our proposed LSTM-GCN has better performance than its counterparts. The simulation results show that this model has the best performance regarding average profit and occupancy rates, based on the recommendations obtained from our proposed method regarding (i) the reduction in idle taxis, (ii) the improvement in efficiency of finding the optimal solution, and (iii) the increase in the global profit of taxis.

This paper is structured as follows. The related studies are summarized in Section 2. The problem formulation is introduced in Section 3. GN-SLP, together with LSTM-GCN, the destination estimation, and the proposed distributed optimization algorithm are introduced in Section 4. We validate and discuss the performance of our proposed method in Section 5. Finally, in Section 6, we come to our conclusions.

2. Related Studies

This section first introduces some novel studies on TCRR, followed by an illustration of typical methods developed for taxi demand prediction and some advanced GNN-based techniques used for traffic prediction problems.

2.1. Related Studies on TCRR

Profit is the driving force for the growth of the taxi market. Thus, we broadly divide the research on TCRR into two categories: studies on short-term-profit-driven TCRR and studies on long-term-profit-driven TCRR. Short-term profit is the profit that a taxi driver earns in a single trip, which consists of two successive routes, namely the cruising route and the occupied route [10,11]. Correspondingly, long-term profit is the profit that a taxi driver earns from multiple trips.

Several studies have been carried out on short-term-profit-driven TCRR. For instance, Yuan et al. [12] presented a study recommending a cruising route to a taxi driver regarding the specific objective that the taxi driver wants to achieve in a single trip, such as maximizing revenue or minimizing the length of the cruising route. The authors explored potential passengers and estimated their destinations with the knowledge of the behaviors of taxis and passengers learned using a statistical learning method. The statistical learning method was also used in Qu et al.'s research [11] to estimate the probability of a pick-up event in each road segment. They maximized the expected revenue using two strategies, namely, a brute-force strategy and a recursive recommendation strategy. With the same objective, a Markov decision process (MDP) was applied in Zhou et al.'s research [10]. A system state was described as a parameter with the current location of a taxi. The travel decision made for the next trip was defined as an action. They proposed a rolling-horizon configuration strategy to obtain the optimal cruising route, which was formulated as a sequence of states and actions. In [8], Qu. et al. developed a probabilistic network to model pick-up probabilities in different road segments. By expressing the objective as the expected length of a cruising route, the authors developed a profitable grid assignment algorithm to determine the shortest route by taking into account the load balance among different pick-up locations. They also employed a MapReduce and KD tree data structure in the algorithm to improve the recommendation efficiency. Based on Coulomb's law, taxis and passengers were modeled as positive and negative charges in [1]. The force between two opposite charges was used to express the attraction between a taxi and a passenger, whereas the force between two positive (negative) charges was used to formulate the competition between two taxis (passengers). The best recommendation was the route towards a pick-up location with the maximal traffic force for a taxi.

In the matter of long-term-profit-based TCRR, Dong et al. [23] aimed to maximize the net revenue (revenue minus fuel cost) of a taxi within one and a half trips, where the half trip is the route from one drop-off location to the next pick-up location. They calculated the net revenue with a linear equation system and obtained the optimal solution

by utilizing a heuristic algorithm. Similar to [10], MDP was used in both [2,24]. Verma et al. attempted to maximize the expected revenue of a taxi over a week. They proposed a reinforcement-learning-based method to solve the problem, as well as a dynamic abstraction mechanism to enhance the efficiency of the basic learning mechanism. Separately, an MDP model in [2] was built based on the hypothesis that the arrivals of passengers and the geographical distributions of idle taxis followed the spatial and temporal Poisson distribution. Moreover, Yu et al. [2] applied the queuing theory to model the competition among taxis and passengers. An enhanced-value iteration algorithm was proposed to create the recommendation with which the expected revenue over the full working period of a day could be maximized.

In the above literature, we can observe that only a few studies have considered either competition [1,2] or collaboration [8] among taxis. In addition, the approaches to obtaining an optimal solution in many studies are based on centralized algorithms with a high consumption of computational resources. To improve the recommendation efficiency, distributed frameworks like an MAS (multiagent system) and distributed numerical optimization [25,26] are alternatives. Indeed, distributed frameworks have already been applied in transportation research such as dynamic routing [27], congestion management [28], and intelligent traffic control [29]. With the development of 5G, the minimum latency of communication enables a mobile distributed computing model to have the benefit of an improved system efficiency. Therefore, we modeled the TCRR as a biobjective optimization problem and obtained a solution with the proposed distributed Lagrange dual-decomposition-based algorithm.

2.2. Taxi Demand Prediction

A critical problem that should be solved in TCRR is how to find potential passengers. It is usually researched as an independent problem and has received extensive attention over past decades. GNNs have gained increasing popularity and have been empirically proven to perform better than CNNs or RNNs in time series prediction. In the following, we first present some typical GNN-based predictors, followed by the application of a GNN in taxi demand prediction.

Regarding GNN-based predictors, Yu et al. [30] introduced a spatial–temporal graph convolutional network (STGCN) for predicting vital indicators such as speed and volume. The STGCN consisted of multiple spatial–temporal Conv-blocks, each containing a GNN layer and a CNN layer to capture the spatial–temporal features of the traffic. Li et al. [31] utilized a proposed diffusion convolutional recurrent neural network to forecast traffic flow. Specifically, they employed bidirectional random walks on the graph model of the road network to capture spatial correlation and used an encoder–decoder architecture with scheduled sampling to capture temporal correlation. Guo et al. [32] proposed an attention-based spatial–temporal graph convolutional network (ASTGCN) for traffic flow prediction. ASTGCN comprised three subnetworks, each modeling traffic patterns at different time scales (recent, day, and week). In each subnetwork, a spatial–temporal attention mechanism, along with spatial–temporal convolutional layers, was integrated to capture the spatiotemporal correlation among traffic. In [33], Zhao et al. developed a temporal graph convolutional network that combined graph convolution and gated recurrent unit techniques to predict traffic flow. The graph convolution was employed to explore spatial correlation based on road network topology, while the gated recurrent unit was utilized to learn temporal correlation that varied with the time of the day.

The use of a GNN in predicting taxi demand has been explored in previous studies. For example, Geng et al. [22] introduced a spatiotemporal multigraph convolution network (ST-MGCN) to forecast ride-hailing demand. This approach leveraged multigraph convolutional layers and dynamic weight-based RNN layers to capture the spatiotemporal correlations between taxi demands. In this paper, we propose an LSTM-GCN model for taxi demand prediction. Unlike the ST-MGCN, our model takes into account both

the spatiotemporal correlations and dynamic patterns of taxi demands across different time scales.

3. Problem Formulation

We focused on short-term-profit-driven TCRR since long-term profit is affected by multiple uncertainties in a traffic network, which increases the inaccuracy of the recommendation. As shown in Figure 1, we divided the problem into two main phases. In the first phase, our objective is to define the mathematical problem using the available idle taxis in the time slot t and the demand for passengers in $t+1$ predicted using the proposed GNN-based model. In real-life scenarios, the matching process between an idle taxi and a passenger depends on various factors. To simplify the model, this paper considered the most significant factors that had been commonly examined in previous studies, such as the distance between the pick-up and drop-off locations (OD pair), along with the fare and cost associated with the taxi trip. It is easy to extend our problem formulation to handle more complicated scenarios. In the second phase, we propose a distributed algorithm that aims to efficiently find the optimal solution to the mathematical problem mentioned above. This optimal solution corresponds to the best matching between each OD pair. By utilizing this solution, we are able to determine the most suitable cruising route for each taxi, which is then assigned to the respective taxi driver.

We start by modeling the road network as an undirected graph $G = \{U, E\}$, with its vertex set U and undirected edge set E. We define each vertex $u \in U$ as a link of a one-way road segment bounded by two adjacent road intersections. There is an edge $e \in E$ between u_1 and u_2, denoted by $e = (u_1, u_2)$ if u_1 and u_2 are physically connected. Suppose a taxi driver picks up and drops off a passenger on the side of the link; a route r is defined as a sequence of vertices $u_1, u_2, \ldots, u_{l_r+1}$ connected by the edges $e_1, e_2, \ldots, e_{l_r}$. We divide G into a set of disjoint zones $Z = \{z_1, z_2, \ldots, z_{|Z|}\}$, where each zone is a subgraph of G that includes a set of links clustered by the POI distribution. We then forecast the taxi demand in G by predicting the taxi demand in each zone to avoid data sparsity problems caused by limited samples in some links. More details on the graph segmentation are illustrated in Section 4.2.

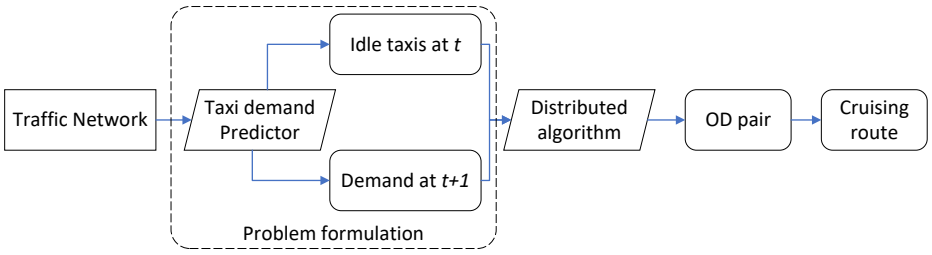


Figure 1. The framework of cruising route recommendation.

We divide a day into a group of successive time slices evenly, e.g., 5 min. The idle taxis accumulated in zone $z_i \in Z$ at the end of time interval t is $\mathcal{N}_{z_i,t}$. The set of idle taxis over the whole road network in t is $\mathcal{N}_t = \{\mathcal{N}_{z_i,t} | z_i \in Z\}$. Suppose taxi demand in $\forall z_i \in Z$ in time slice $t+1$ is $y_{z_i,t+1}$. We then formulate $\boldsymbol{y}_{t+1} = \{y_{z_i,t+1} | z_i \in Z\}$ as the taxi demand of the whole road network in $t+1$. The way to predict $\boldsymbol{y}_{t+1}$ is illustrated in Section 4.3. By introducing a set of binary variables $\boldsymbol{x} = \{x_{nm} | n \in N_t, m \in M_{t+1}\}$, where $N_t = \sum_{z_i \in Z} |\mathcal{N}_{z_i,t}|$ and $M_{t+1} = \sum_{z_i \in Z} |y_{z_i,t+1}|$, we model the TCRR problem $\mathbf{P}$ as follows:

$$\mathbf{P}: \quad \{\min\ f_1(\boldsymbol{x}), \max\ f_2(\boldsymbol{x})\} \tag{1}$$

$$s.t. \quad x_{nm} \in \{0,1\}, \forall x_{nm} \in N_t, m \in M_{t+1} \tag{2}$$

$$\sum_{m \in M_{t+1}} x_{nm} \in \{0,1\}, \forall n \in N_t \tag{3}$$

$$\sum_{n \in N_t} x_{nm} \in \{0,1\}, \forall m \in M_{t+1} \tag{4}$$

$$\sum_{m \in M_{t+1}} R(x_{nm}) \geqslant 0, \forall n \in N_t \tag{5}$$

In **P**, constraint (2) indicates that x_{nm} is a binary variable. $x_{nm} = 1$, if the nth idle taxi is recommended to cruise on a route to pick up the mth passenger from a zone, and $x_{nm} = 0$ otherwise. Ridesharing is not allowed due to constraints (3) and (4). In constraint (5), $\sum_{m \in M_{t+1}} R(x_{nm})$ is the expected profit for the nth taxi. In particular, $R(x_{nm})$ is formulated as

$$R(x_{nm}) = x_{nm}(fare_{nm} - cost_{nm} - \epsilon_{thr}). \tag{6}$$

Evidently, $\sum_{m \in M_{t+1}} R(x_{nm}) = 0$ if there is no cruising route recommended to the nth taxi. Otherwise, the expected profit is determined by the fare ($fare_{nm}$) and cost ($cost_{nm}$) generated over a single trip, $r_{nm} = \{r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm}\}$, where $r^{\mathcal{C}}_{nm}$ and $r^{\mathcal{O}}_{nm}$ are denoted as the cruising route and the occupied route, respectively. As $r^{\mathcal{C}}_{nm}$ and $r^{\mathcal{O}}_{mn}$ are unknown, we regard $r^{\mathcal{C}}_{nm}$ and $r^{\mathcal{O}}_{mn}$ as two discrete random variables with the probability distributions illustrated in Section 4.1. We set a threshold ϵ_{thr} to satisfy the condition that the minimum revenue earned by the nth taxi is greater than ϵ_{thr}. In this paper, ϵ_{thr} is a constant value.

There are two goals in **P**, where $f_1(\boldsymbol{x}_t)$ and $f_2(\boldsymbol{x})$ represent the number and global revenue of idle taxis, formulated by

$$f_1(\boldsymbol{x}) = N_t - \sum_{n \in N_t} \sum_{m \in M_{t+1}} x_{nm}. \tag{7}$$

$$f_2(\boldsymbol{x}) = \sum_{n \in N_t} \sum_{m \in M_{t+1}} R(x_{nm}). \tag{8}$$

It is evident that $f_1(\boldsymbol{x}) \geqslant 0$ is subject to the constraint (3). The optimal solution for **P** is $\boldsymbol{x}^*$, which minimizes $f_1(\boldsymbol{x})$ and maximizes $f_2(\boldsymbol{x})$. Using $\boldsymbol{x}^*$, a recommended route can be determined as the shortest path that the idle taxi takes towards the assigned zone (alternative path generalization rules, such as the shortest-time criterion, can also be employed to substitute the shortest-path-based approach). To obtain $\boldsymbol{x}^*$, we employ the weighted summation method [34] to transform the biobjective problem into a single objective:

$$\begin{aligned} \mathbf{P}: \quad & \min\ \frac{w_1}{sf_1} f_1(\boldsymbol{x}) - \frac{w_2}{sf_2} f_2(\boldsymbol{x}) \\ s.t. \quad & (2), (3), (4), (5) \end{aligned} \tag{9}$$

where the scale factors sf_1, sf_2 are employed to normalize the biobjective, and $w_1, w_2 > 0$ are the weights with $w_1 + w_2 = 1$. In Section 4.4, we propose a distributed algorithm to solve **P**.

4. Solutions

Given the formulation of the above problem, it becomes clear that the key to obtaining x^* relies on the estimation of $r^{\mathcal{C}}_{nm}$, $r^{\mathcal{O}}_{nm}$, and Z, the traffic demand predictor, and the optimization algorithms. These solutions are specifically discussed in the subsequent sections.

4.1. Estimation of $r^{\mathcal{C}}_{nm}$ and $r^{\mathcal{O}}_{nm}$

In real-world scenarios, there can be various paths that connect two nodes. To simplify the problem, we consider a scenario where a taxi aims to travel along the route with the smallest distance. This assumption is also commonly adopted in numerous research studies, such as [35–37]. Hereafter, given $x_{nm} = 1$, that is, the nth taxi is recommended to pick up the mth potential passenger from zone z_i, we then have a group of possible cruising routes for the nth taxi, denoted by $\mathcal{C}^{z_i}_{nm}$, which has a size of $|z_i|$. The probability that a taxi travels along $r^{\mathcal{C}}_{nm}$ in $t+1$ is represented as $p(r^{\mathcal{C}}_{nm}|z_i, t+1), r^{\mathcal{C}}_{nm} \in \mathcal{C}^{z_i}_{nm}$. To estimate $p(r^{\mathcal{C}}_{nm}|z_i, t+1)$, we assume that $p(r^{\mathcal{C}}_{nm}|z_i, t+1)$ follows a uniform distribution, $p(r^{\mathcal{C}}_{nm}|z_i, t+1) = \frac{1}{|\mathcal{C}^{z_i}_{nm}|} = \frac{1}{|z_i|}, r^{\mathcal{C}}_{nm} \in \mathcal{C}^{z_i}_{nm}$. To put it differently, the likelihood of a passenger appearing in any link of z_i is equal. This assumption is justifiable as we divide the road network into zones based on the criterion that all links within a zone encircle the same POI. To further support the validity of this assumption, we impose a constraint on the size of each zone, setting it to a small threshold such as 120, as applied in our experiment. Since the zones are determined after the graph segmentation, we rewrite $p(r^{\mathcal{C}}_{nm}|z_i, t+1)$ as $p(r^{\mathcal{C}}_{nm}|z_i)$.

Unlike $r^{\mathcal{C}}_{nm}$, the link where the taxi drops off the mth passenger can fall into any zone. It increases the uncertainties of $r^{\mathcal{O}}_{nm}$, which in turn increases the estimation difficulties for the probability distribution of $r^{\mathcal{O}}_{nm}$. We first define $p(z_j|z_i, t+1)$ as the probability that a passenger travels from origin z_i to destination z_j in $t+1$. Moreover, we denote $p(r^{\mathcal{O}}_{nm}|z_j, t+1)$ as the probability that the passenger is dropped off at z_j in $t+1$. We assume that $p(z_j|z_i, t+1)$ and $p(r^{\mathcal{O}}_{nm}|z_j, t+1)$ are independent from each other. The probability distribution of $r^{\mathcal{O}}_{nm}$, denoted by $p(r^{\mathcal{O}}_{nm}|z_j, z_i, t+1)$, is then estimated by

$$p(r^{\mathcal{O}}_{nm}|z_j, z_i, t+1) = p(r^{\mathcal{O}}_{nm}|z_j, t+1)p(z_j|z_i, t+1). \tag{10}$$

Suppose that $p(r^{\mathcal{O}}_{nm}|z_j, t+1)$ follows the uniform distribution. We then rewrite $p(r^{\mathcal{O}}_{nm}|z_j, t+1)$ as $p(r^{\mathcal{O}}_{nm}|z_j)$, estimated by $p(r^{\mathcal{O}}_{nm}|z_j) = \frac{1}{|z_j|}, r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}$, where $\mathcal{O}^{z_j}_{nm}$ is the set of candidate occupied routes. Moreover, assuming that $p(z_j|z_i, t+1)$ is constant everywhere is unreasonable since the OD (origin–destination) pair has a close relationship with the behavior pattern of the passenger, which also varies with the time of day. For instance, in the morning rush hours, the destination of a passenger in a residential area is more likely a zone like the CBD (central business district). On the contrary, a passenger most likely intends to travel to a residential area from the CBD in the evening rush hours. Thus, we estimate $p(z_j|z_i, t+1)$ with a statistical learning method that is widely used in many studies [2,12]. More precisely, suppose we have data for the past Ω days. We use $\#_d(z_i \to z_j|t+1)$ to represent the number of taxis that travel from z_i to z_j in $t+1$ during the dth day, where d falls in the range $\{1, 2, \ldots, \Omega\}$. Thus, $p(z_j|z_i, t+1)$ is estimated as follows:

$$p(z_j|z_i, t+1) = \frac{\sum_{d\in\Omega} \#_d(z_i \to z_j|t+1)}{\sum_{z_j\in Z}\sum_{d\in\Omega} \#_d(z_i \to z_j|t+1)}, z_i, z_j \in Z. \tag{11}$$

With $p(r^{\mathcal{C}}_{nm}|z_i)$ and $p(r^{\mathcal{O}}_{nm}|z_j, z_i, t+1)$, we calculate $fare_{nm}$ and $cost_{nm}$ in the following way. First, the cost over r_{nm} is formulated by

$$cost(r_{nm}) = c_{fc} \cdot (t_{r^{\mathcal{C}}_{nm}} + t_{r^{\mathcal{O}}_{nm}}), \tag{12}$$

where c_{fc} is the fuel cost per kilometer, and t_r is the travel time on r. t_r can be estimated by the travel distance of r divided by the average travel speed according to the annual

transportation report of the major cities in China (https://report.amap.com/download.do, accessed on 13 January 2024). $cost_{nm}$ is the expectation of $cost(r_{mn})$, expressed as

$$\begin{aligned} cost_{nm} &= \mathbb{E}(cost(r_{mn})) \\ &= c_{fc} \cdot \Big(\sum_{r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}} p(r^{\mathcal{C}}_{nm}|z_i) t_{r^{\mathcal{C}}_{nm}} + \sum_{z_j \in Z} p(z_j|z_i, t+1) \sum_{r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}} p(r^{\mathcal{O}}_{nm}|z_j) t_{r^{\mathcal{O}}_{nm}} \Big). \end{aligned} \tag{13}$$

$fare(r_{nm})$ is calculated by

$$fare(r_{nm}) = c_{fk} \cdot t_{r^{\mathcal{O}}_{nm}}, \tag{14}$$

where c_{fk} is the fare per kilometer and $fare_{nm}$ is the expectation of $fare(r_{mn}|r^{\mathcal{O}}_{nm})$, expressed as

$$fare_{nm} = \mathbb{E}(fare(r_{mn}|r^{\mathcal{O}}_{nm})) = c_{fk} \cdot \sum_{z_j \in Z} p(z_j|z_i, t+1) \sum_{r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}} p(r^{\mathcal{O}}_{nm}|z_j) t_{r^{\mathcal{O}}_{nm}}. \tag{15}$$

4.2. Estimation of Z

The segmentation of urban areas based on their geographic location has advantages for analyzing human mobility patterns [38], which in turn affects the variation in taxi demand. Existing methods for segmenting road networks are roughly divided into two classes: uniform grid partition methods, such as the regular grid layout method proposed by Yao et al. [18,39], and nonuniform grid partition methods, such as the quadtree-based method developed by Steed et al. [40]. The main difference between these two methods is the shape of the resulting zones after segmentation. The zones in the first class are regular, resembling hexagons, while the zones in the second class are irregular.

In contrast to the methods mentioned earlier, our aim is to partition the graph model of a road network, denoted as G, into multiple subgraphs. Each subgraph represents a zone that comprises a group of spatially connected links surrounding a specific POI. We base this segmentation process on the observation that taxis tend to cruise in the links near POIs that are relevant to daily human activities. For instance, at or near locations such as airports and shopping malls, there usually exists a high taxi demand. To achieve this, we initially assign labels to POIs, as shown in Table 1, where POIs with the same labels exhibit similar characteristics. For instance, "Chang'an University" and "XiDian University" are labeled as "SCH1" and "SCH2", respectively. We propose the GN-SLPA algorithm to classify the links into different zones, with each zone labeled by the identifier of the POI that is surrounded by the majority of links in that zone. The segmentation procedure involves the following steps.

Step 1: We obtain the K-nearest POIs associated with every u_i in set U. Additionally, each u_i in U has a memory that is used to store the identifiers of the POIs propagated from other vertices during each iteration. The memory of each vertex u_i in U is initialized with the identifier of one of its own K-nearest POIs.

Step 2: Assuming that u_i represents the listener, the first-order neighbors of u_i are considered as speakers who are responsible for selecting a POI identifier from their memory. This selection process is performed using a random strategy that takes into account the frequency of occurrence of the POIs in their memory. Subsequently, the selected POI identifier is transmitted to u_i. Upon receiving the set of POI identifiers from its neighbors, u_i stores the identifier with the highest frequency in its memory.

Step 3: We repeat **Step 2** for $\mathcal{I}_{\mathcal{G}}$ iterations until the number of zones reaches convergence. At this point, we can determine the occurrence frequency of the different POI identifiers in the memory of u_i. If the frequency of a POI identifier is below the threshold $\epsilon_{overlap}$, it is eliminated. Vertices that share a common POI identifier are then grouped into a zone and labeled with that POI identifier. It is important to note that the SLPA (for more information on the SLPA, please refer to [41]) is an overlapping algorithm, meaning

that multiple communities may contain the same vertex. To avoid overlapping, we set $\epsilon_{overlap} \geqslant 0.5$ [42].

Step 4: Suppose that we have a large zone with a size (the number of links) greater than ϵ_{size}; we use the GN algorithm to divide this large zone into multiple smaller zones, each with a size smaller than ϵ_{size}. The main idea behind the GN algorithm is as follows: Firstly, compute the betweenness (the betweenness associated with a vertex $u_i \in U$ is defined as the number of shortest paths between pairs of nodes that pass through u_i.) of all existing edges in the network. Secondly, we remove the edge with the highest betweenness and recompute the betweenness of all the edges. Finally, we repeat the above two steps until all edges are removed.

We impose a limit on the size of the zone to avoid the scenario in which a passenger is picked up and dropped off in the same zone ($z_i = z_j$). Since the conditional probability $p(z_j z_i, t+1)$ in Equation (15) becomes a constant value of 1 when $z_i = z_j$, the value of $R(x_{nm} = 1)$ is solely determined by $p(r^{\mathcal{O}}_{nm} z_j, t+1)$, which fails to capture the travel patterns of the passengers. In addition, larger zone sizes result in a larger number of links, leading to an increase in the proposed algorithm complexity. The algorithm complexity is primarily dependent on the SLPA, which has a time complexity of $O(\mathcal{I}_{\mathcal{G}} E)$. However, since it is an offline algorithm that only needs to be executed once, this complexity is manageable.

Table 1. The set of identifiers of POIs.

Abbreviation	Significance	Abbreviation	Significance
WA	Working area	SCH	School
RA	Residential area	EA	Entertainment area
SPM	Supermarket	HOS	Hospital
HT	Hotel	BT	Bus terminal
AP	Airport	SQ	Square
TS	Train station	PARK	Park
SM	Shopping mall	OTS	Others

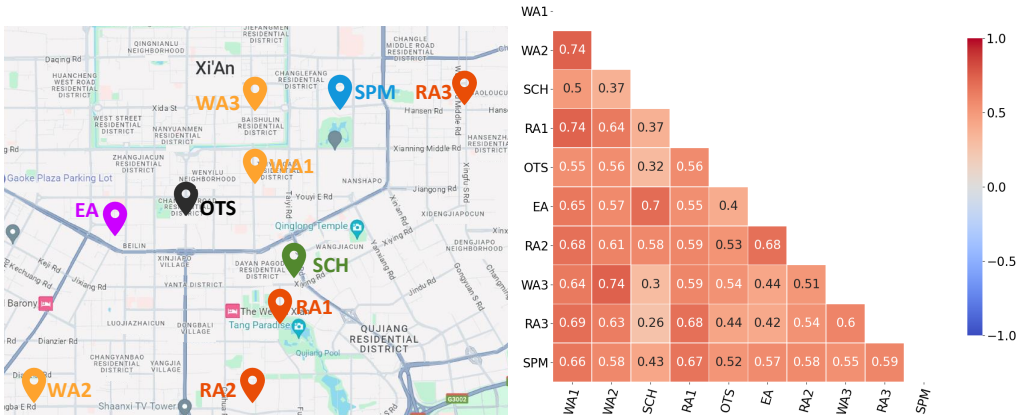
4.3. LSTM-GCN Model for Traffic Demand Prediction

As mentioned in Section 2, several studies have focused on incorporating spatiotemporal correlation to enhance the accuracy of taxi demand prediction. For capturing this correlation, many studies utilize black-box approaches, such as incorporating multiple spatial and temporal convolutional layers into a deep neural network. However, these approaches do not result in a clear understanding of how the spatiotemporal correlation contributes to the prediction results, leading to the empirical configuration of hyperparameters, such as the number of hidden layers. Another approach is based on time series models, such as ARIMA or its variants. These models consider that taxi demands exhibit a spatiotemporal correlation within specific spatial and temporal lags, which are estimated using a spatial autocorrelation function (ACF) and a spatial–partial ACF. Unfortunately, time series models are parametric approaches, which are advantageous in capturing regular variations but have limitations in capturing irregular variations.

To develop an interpretative model with the ability to capture spatiotemporal correlations and regular and irregular variations in taxi demand, we first analyzed factors that had a potential impact on the dynamics of taxi demand based on trajectory data (more details on the data are illustrated in Section 5). Aside from a widely accepted conclusion that taxi demand has different dynamic patterns under different time scales, e.g., per hour and per day, we also observed that the spatiotemporal correlation between taxi demand in the two zones did not follow any distance pattern; it depended on the similarity between the features of the two zones, as well as the frequency of human movement between the two zones.

To provide a clearer illustration of the aforementioned observation, let us examine the results obtained from analyzing the taxi demand data of ten zones after conducting a graph segmentation. These zones possessed distinct characteristics, including three WAs, three

RAs, one SCH, one OTS, one SPM, and one EA. Some of these zones were adjacent to each other, while others were not. We computed the Pearson correlation coefficients (depicted in Figure 2b) for the taxi demand across these zones over a span of 96 time intervals in a day, with each interval lasting fifteen minutes. From our analysis, we observed a strong correlation (0.74) in taxi demand between WA1 and WA2, despite their considerable distance from each other. Similar patterns could be observed between zones WA2 and WA3, as well as between RA1 and RA3. On the other hand, even though RA1 and SCH were in close proximity, their correlation was remarkably low (0.37). This discrepancy could be attributed to the fact that certain zones with similar characteristics exhibited similar levels and dynamic patterns of taxi demand. Furthermore, we also identified a significant spatiotemporal correlation between zones with different characteristics, such as the Pearson coefficient of 0.64 between RA1 and WA2. This phenomenon typically occurred between zones that experienced frequent or periodic human mobility throughout the day. For example, during the morning rush hours (between 7:00 am and 9:00 am), individuals traveled from an RA to a WA, and then intended to return to the WA during the evening rush hours (between 5:00 pm and 7:00 pm). The observation in Figure 2 highlights the challenge of directly applying existing traffic predictors to forecast taxi demand. This is because these predictors rely on capturing the spatiotemporal correlation based on road topology and traffic mobility patterns [43]. However, taxi demand does not follow any specific distance pattern; it heavily depends on the mobility of drivers.



(**a**) The centers of ten zones (**b**) Pearson correlation

Figure 2. Pearson correlation of taxi demand from ten zones over 96 time intervals within one day.

For handling the above challenges, we propose a taxi demand predictor that utilizes the graph convolutional network (GCN) model. This predictor considers the correlation between taxi demands in different zones over short and long time periods, as well as the evolving patterns of taxi demand at various time scales. Specifically, our proposed predictor incorporates an attention-enhanced long short-term memory (LSTM) mechanism developed by Si et al. [44]. As depicted in Figure 3, the input of our graph neural network (GNN) model follows a graph-based structure denoted $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. In this structure, each vertex in $\mathcal{V}$ represents a zone, and an edge exists between two zones, $z_{i_1} \in Z$ and $z_{i_2} \in Z$, if one of the following conditions is met: (1) there is at least one edge connecting $u_1 \in z_{i_1}$ and $u_2 \in z_{i_2}$ or (2) the Pearson correlation between z_{i_1} and z_{i_2} exceeds 0.6 (indicating a strong correlation). Compared to other methods that construct a graph structure based on the physical distance between zones or links, our graph structure offers two advantages: (i) it can capture the spatiotemporal correlation between two zones even if they do not follow any specific distance pattern and (ii) it eliminates the need for stacking multiple convolutional layers to explore the deep spatial correlation between zones. A signal on the graph $\mathcal{G}$ represents the demand for taxis throughout the entire road network in the time interval t, denoted $\boldsymbol{y}_t$.

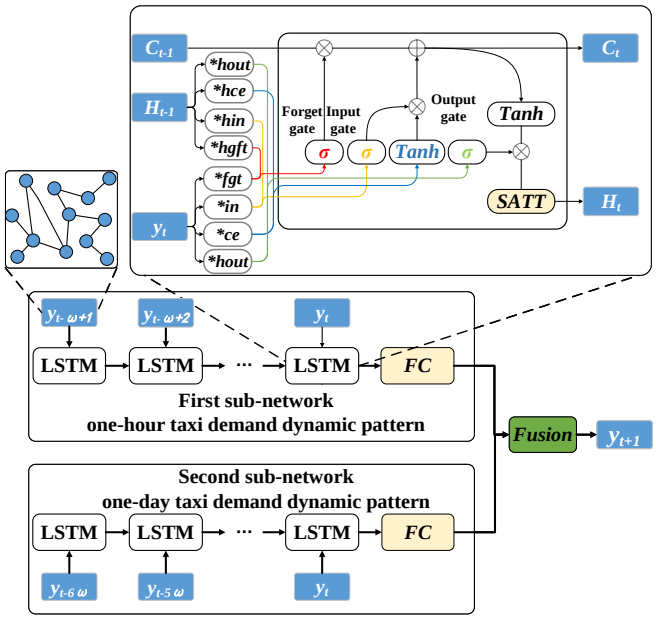


Figure 3. Overview of the LSTM-GCN (* denotes convolution).

There are two subnetworks in our model. They are used to capture different dynamic patterns of taxi demands on time scales of an hour and a day. Inspired by the application of LSTM, a language model to remember relevant context information within short and long time periods, in our model, we employ LSTM in each subnetwork to capture the spatiotemporal dependencies between two zones over short and long time periods. In the first subnetwork, we set $\omega = \left\lceil \frac{24hours}{\tau} \right\rceil$ in the LSTM components where τ is the length of a time slice. The inputs of all LSTM components are the set $\{\boldsymbol{y}_{t-\omega+1}, \boldsymbol{y}_{t-\omega+2}, \ldots, \boldsymbol{y}_t\}$. In each LSTM component, there are three gates, namely the input gate, the forget gate, and the output gate. The functions of these three gates are based on graph convolution using the following expression:

$$\begin{aligned} \boldsymbol{i}_t &= \sigma(\boldsymbol{y}_t * \kappa_{in} + \boldsymbol{H}_{t-1} * \kappa_{hin} + \boldsymbol{b}_{in}) \\ \boldsymbol{f}_t &= \sigma(\boldsymbol{y}_t * \kappa_{fgt} + \boldsymbol{H}_{t-1} * \kappa_{hfgt} + \boldsymbol{b}_{fgt}) \\ \boldsymbol{o}_t &= \sigma(\boldsymbol{y}_t * \kappa_{out} + \boldsymbol{H}_{t-1} * \kappa_{hout} + \boldsymbol{b}_{out}), \end{aligned} \tag{16}$$

where $*$ is the graph convolution operation, $\sigma(\cdot)$ represents the sigmoid function, and the $\boldsymbol{b}$'s denote the biases. With the Chebyshev polynomial kernel [45], $y *_{\mathcal{G}} \kappa$ in (16) can be represented as

$$\boldsymbol{y}_t *_{\mathcal{G}} \kappa = Q(\sum_{j=0}^{\mathcal{I}_\kappa} \alpha_j \Lambda^j) Q^T \boldsymbol{y}_t = \sum_{j=0}^{\mathcal{I}_\kappa} \alpha_j L^j \boldsymbol{y}_t$$

In the proposed LSTM component, the hidden state ($\boldsymbol{H}_t$) is calculated in the following way:

$$\begin{aligned} \boldsymbol{u}_t &= \tanh(\boldsymbol{y}_t *_{\mathcal{G}} \kappa_{ce} + \boldsymbol{H}_{t-1} *_{\mathcal{G}} \kappa_{hce} + \boldsymbol{b}_{ce}) \\ \boldsymbol{C}_t &= \boldsymbol{f}_t \odot \boldsymbol{C}_t + \boldsymbol{i}_t \odot \boldsymbol{u}_t \\ \hat{\boldsymbol{H}}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{C}_t) \\ \boldsymbol{H}_t &= f_{att}(\hat{\boldsymbol{H}}_t), \end{aligned} \tag{17}$$

In (17), $\boldsymbol{u}_t$ is the modulated input, $\boldsymbol{C}_t$ is the cell state, and $\hat{\boldsymbol{H}}_t$ is an intermediate hidden state. $f_{att}(\cdot)$ is designed on the basis of the spatial attention mechanism in Feng et al.'s work [46], and formulated as

$$f_{att}(\hat{\boldsymbol{H}}_t) = \gamma \hat{\boldsymbol{H}}_t, \tag{18}$$

where γ is a $|Z| \times |Z|$ matrix, and each element γ_{ij} measures the similarity between $\hat{\boldsymbol{H}}_{t,i}$ and $\hat{\boldsymbol{H}}_{t,j}$. In particular, γ_{ij} is estimated as:

$$\gamma_{ij} = \gamma_{ji} = \frac{\exp(\hat{\boldsymbol{H}}_{t,i} W_{ij} \hat{\boldsymbol{H}}_{t,j})}{\sum_{z \in |Z|} \exp(\hat{\boldsymbol{H}}_{t,i} W_{iz} \hat{\boldsymbol{H}}_{t,z})}, \tag{19}$$

where W_{ij} denotes the (i, j)th element in W. For simplicity, we use the adjacent matrix of $\mathcal{G}$ as a weight matrix. We apply the spatial attention scheme to measure the correlation among taxi demands in all zones. Finally, we use the active function ReLU (rectified linear unit) in the FC (fully connected layer).

The second subnetwork is similar to the first one. We have seven LSTM components with the set of inputs $\{\boldsymbol{y}_{t-6\omega}, \boldsymbol{y}_{t-5\omega}, \ldots, \boldsymbol{y}_t\}$. Finally, in the fusion operation, the information from two subnetworks is aggregated as follows:

$$\boldsymbol{y}_{t+1} = W_{hr} \boldsymbol{y}_{hr,t+1} + W_{day} \boldsymbol{y}_{day,t+1}, \tag{20}$$

where W_{hr} and W_{day} are the learning parameters, and $\boldsymbol{y}_{hr,t+1}$ and $\boldsymbol{y}_{day,t+1}$ are the taxi demand in $t+1$ predicted via the two subnetworks. We use the gradient descent method to learn all parameters in the LSTM-GCN.

4.4. Distributed Algorithm

The multiobjective-based TCRR can be solved using various heuristic and numerical optimization algorithms [47,48]. However, traditional centralized methods face a challenge in terms of computational complexity, which increases with the values of N_t and M_{t+1} related to the length of a time slice. The longer the time slice, the larger the values of N_t and M_{t+1}. In practical scenarios, the TCRR algorithm is implemented at regular intervals, resulting in a significant number of idle taxis and potential recommended cruising routes. For instance, if we consider a case where $N_t = 100$ and $M_{t+1} = 1000$, the computational complexity becomes $O(100 \times 1000)$. Consequently, a substantial quantity of resources is required to solve **P**. For addressing this issue, we propose a distributed LDD algorithm.

The distributed LDD algorithm is presented in Algorithm 1, where a client represents a taxi (an application on a smartphone or other devices installed in each taxi) and the server represents the terminal server at the taxi operation company. On the client side (lines 1 to 6), we calculate $R(x_{nm} = 1), \forall m \in M_{t+1}$ for each client $n \in N_t$ in order to filter the candidate pick-up locations where $R(x_{nm} = 1) < \epsilon_{rev}$. This ensures that constraint (2) is satisfied. The available pick-up locations for the nth taxi are denoted by M^n_{t+1}. Additionally, we relax (5) by allowing $0 \leqslant x_{nm} \leqslant 1$ in order to transform **P** into a continuous convex problem. It should be noted that this relaxation is unlikely to significantly affect the solution, as explained later in this section. Based on the above operations, we redefine **P** as follows:

$$\mathbf{P}: \quad \min \ \frac{w_1}{sf_1} f_1(\boldsymbol{x}) - \frac{w_2}{sf_2} f_2(\boldsymbol{x}) \tag{21}$$

$$s.t. \quad \sum_{m \in M^n_{t+1}} x_{nm} \leqslant 1, \forall n \in N_t \tag{22}$$

$$\sum_{n \in N_t} x_{nm} \leqslant 1, \forall m \in M^n_{t+1} \tag{23}$$

$$0 \leqslant x_{nm} \leqslant 1, \forall n \in N_t, m \in M_{t+1} \tag{24}$$

Algorithm 1 Distributed algorithm.

Client
Initialization: $M_{t+1}^n \leftarrow M_{t+1}$
1 **for** $\forall m \in M_{t+1}$
2 Calculate $R(x_{nm} = 1)$ based on (13) and (15)
3 **if** $R(x_{nm} = 1) < \epsilon_{rev}$
4 $M_{t+1}^n = M_{t+1}^n - m$
5 **endif**
6 **endfor**
7 receive $\mu^{(k)}$ from terminal
8 **if MSG_TERMINATE = FALSE**
9 $x_{nm^*}^{(k)} = 1, m^* = \underset{m \in M_{t+1}^n}{\text{argmax}}\{Q_{x_{nm}=1}\}$
10 send $x_{nm^*}^{(k)} = 1$ to terminal
11 **else**
12 $x_{nm^*}^{(k-1)}$ is the solution
13 **endif**
Server
Initialization: $\mu^{(0)}, k = 0$, **MSG_TERMINATE = FALSE**
1 send $\mu^{(k)}$ to client
2 $k \leftarrow k + 1$
3 receive $\boldsymbol{x}^{(k)}$ from client
4 Update $\mu_m^{(k)}, \forall m \in M_{t+1}^n$ with (31)
5 **if** $\mu^{(k)}$ converges **or** $k = \mathcal{I}_{\mathcal{D}}$
6 **MSG_TERMINATE=TRUE**
7 **endif**
8 send **MSG_TERMINATE** to client

We introduce M_{t+1}^n Lagrange multipliers $\mu = \{\mu_m | \mu_m \geqslant 0, \forall m \in M_{t+1}^n\}$. By substituting μ and constraint (4) into the objective, we rewrite the Lagrangian of **P**, $\mathcal{L}(\boldsymbol{x}, \mu)$ as the one in (26). With $\mathcal{L}(\boldsymbol{x}_t, \mu)$, the dual problem of **P**, $\mathcal{D}(\mathbf{P})$ can be reformulated as

$$\begin{aligned} \mathcal{D}(\mathbf{P}): \quad & \max_{\mu} \ \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \mu) \\ s.t. \quad & (22), (24) \end{aligned} \tag{25}$$

Suppose that the optimal solutions of **P** and $\mathcal{D}(\mathbf{P})$ are $\boldsymbol{x}^*$ and $\bar{\boldsymbol{x}}^*$, respectively. As **P** is a convex problem, the strong duality holds. Thus, we have $\boldsymbol{x}^* = \bar{\boldsymbol{x}}^*$. To find $\bar{\boldsymbol{x}}^*$, the pseudocodes from line 7 to 13 at the client and those from line 1 to 8 at the server are iteratively implemented. More precisely, in the kth ($k \geqslant 1$) iteration, $x_{nm} \in \boldsymbol{x}^{(k)}$ is updated by the client n using $\mu^{(k-1)}$ sent by the terminal (lines 7–10 of the client). When $k = 1$, the value of $\boldsymbol{x}^{(1)}$ is updated using the initial values of μ, which are denoted as $\mu^{(0)}$. Once the termination message (**MSG_TERMINATE=TRUE**) is received, the optimal solution for the nth taxi is obtained from line 11 to 13 of the client. The server updates $\mu^{(k)}$ using $\boldsymbol{x}_t^{(k)}$ on lines 3 and 4 on the client side.

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \mu) &= \frac{w_1}{sf_1} f_1(\boldsymbol{x}) - \frac{w_2}{sf_2} f_2(\boldsymbol{x}) + \sum_{m \in M_{t+1}^n} \mu_m \Big(\sum_{n \in N_t} x_{nm} - 1 \Big) \\ &= \frac{w_1}{sf_1} \Big(N_t - \sum_{n \in N_t} \sum_{m \in M_{t+1}^n} x_{nm} \Big) - \frac{w_2}{sf_2} \sum_{n \in N_t} \sum_{m \in M_{t+1}^n} R(x_{nm}) + \sum_{m \in M_{t+1}^n} \mu_m \sum_{n \in N_t} x_{nm} - \sum_{m \in M_{t+1}^n} \mu_m \\ &= w_1 N_t - \left(\sum_{n \in N_t} \sum_{m \in M_{t+1}^n} \Big(\frac{w_1}{sf_1} - \mu_m \Big) x_{nm} + \frac{w_2}{sf_2} R(x_{nm}) \right) - \sum_{m \in M_{t+1}^n} \mu_m. \end{aligned} \tag{26}$$

The algorithm terminates when $\mu^{(k)}$ converges or the maximal iterations ($\mathcal{I}_{\mathcal{D}}$) are achieved (lines 5 to 7 on the side of the terminal server). In our paper, the algorithm converges when $\mathcal{I}_{\mathcal{D}}$ is equal to 25. Lastly, **MSG_TERMINATE** is sent to each client. The

process related to the updates of $\boldsymbol{x}_t^{(k)}$ and $\mu^{(k)}$ is shown in the following. First, we define a Lagrange dual function $g(\mu^{(k)})$ as

$$\begin{array}{rcl} g(\mu^{(k)}) & = & \min\limits_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \mu^{(k)}) \\ & s.t. & (22), (24) \end{array} \tag{27}$$

and a separate $g(\mu^{(k)})$ in terms of each taxi as follows:

$$g(\mu^{(k)}) = -\sum_{n \in N_t} g_n(\mu^{(k)}) + \min_{\boldsymbol{x}} (\frac{w_1}{sf_1} N_t - \sum_{m \in M_{t+1}^n} \mu_m^{(k)}), \tag{28}$$

where $g_n(\mu^{(k)})$ is in the form of

$$\begin{array}{c} g_n(\mu^{(k)}) = \max\limits_{x_{nm}} \sum_{m \in M_{t+1}^n} (\frac{w_1}{sf_1} - \mu_m^{(k)}) x_{nm} + \frac{w_2}{sf_2} R(x_{nm}) \\ s.t. \, (22), (24) \end{array} \tag{29}$$

Clearly, the optimal solution of $g(\mu^{(k)})$ for a fixed $\mu^{(k)}$ can be obtained by solving $g_n(\mu^{(k)})$ for all $n \in N_t$. This is because the term $\min\limits_{\boldsymbol{x}} (\frac{w_1}{sf_1} N_t - \sum_{m \in M_{t+1}^n} \mu_m^{(k)})$ in Equation (28) is constant. We define $Q_{x_{nm}=1}^{(k)} = \frac{w_1}{sf_1} - \mu_m^{(k)} + \frac{w_2}{sf_2} R(x_{nm})$. Then, by fixing $\mu^{(k)}$, the optimal solution to $g_n(\mu^{(k)})$ is $x_{nm^*} = 1$ with m^* expressed as

$$m^* = \underset{m \in M_{t+1}^n}{\operatorname{argmax}} \{Q_{x_{nm}=1}^{(k)}\}, \forall n \in N_t \tag{30}$$

It is important to acknowledge that there may exist several values of m for which $Q_{x_{nm}=1}^{(k)}$ is the highest. In such situations, we choose one of these values randomly and designate it as m^*.

After finding $x_{nm^*}^{(k)}$s, for $\forall n \in N_t$, we obtain $\boldsymbol{x}^{(k)}$ with $\mu_m^{(k)}s$ and update $\mu_m^{(k+1)}$ for each $m \in M_{t+1}^n$ by the following gradient descent method [49]:

$$\mu_m^{(k+1)} = \left[\mu_m^{(k)} - \beta(k) \left(1 - \sum_{n \in N_t} x_{nm} \right) \right]^+. \tag{31}$$

As $\mu_m \geqslant 0$, the operator $[\cdot]^+$ in (31) indicates the operation to obtain the maximum between $\mu_m^{(k)} - \beta(k)(1 - \sum_{n \in N_t} x_{nm}), \forall m \in M_{t+1}^n$, and 0. Furthermore, we define $\beta(k) = \frac{\lambda_{\mathcal{D}}}{k}$ for the purpose of controlling the convergence speed of μ.

On the side of the client, the estimation of M_{t+1}^n has a complexity of $O(M_{t+1})$. With the quick-sort algorithm, the update of $x_{nm^*}^k$ has a complexity of $O(\mathcal{I}_{\mathcal{D}} M_{t+1}^n log M_{t+1}^n)$. On the side of the terminal server, the complexity depends on the update of μ, which is $O(\mathcal{I}_{\mathcal{D}} M_{t+1})$. As the algorithm is implemented simultaneously on both the client and the server in each iteration, the overall complexity of Algorithm 1 is $O((1 + log M_{t+1}) \mathcal{I}_{\mathcal{D}} M_{t+1})$.

4.5. Cruising Route Generalization

Based on $\boldsymbol{x}^*$, we exploit Algorithm 2 to estimate the cruising route for each idle taxi in t. More precisely, given an idle taxi n assigned to z_i, we select a link from $z_i - \bar{z}_i$ that is nearest to the position of taxi n using the pseudocodes from line 8 to 14. Particularly, $\bar{z}_i$ includes the links matched to other idle taxis recommended to travel toward z_i. Thus, $z_i - \bar{z}_i$ is the set of links without assignment. With $z_i - \bar{z}_i$, we can avoid the idle taxis with the same recommended zone traveling to the same link. For the case when the number of idle taxis assigned to z_i is larger than the size of z_i, there must be idle taxis traveling to the same link. We implement the pseudocodes from line 5 to 7 to achieve this goal.

Algorithm 2 Cruising route recommendation algorithm in t.

Input: x^*
for $z_i \in Z$
$\bar{z}_i \leftarrow \emptyset$
  for $n \in N_{z_i}$
  $dist = +\infty$, $v^* = \emptyset$
  if $z_i - \bar{z}_i = \emptyset$
    $\bar{z}_i \leftarrow \emptyset$
  endif
    for $v \in z_i - \bar{z}_i$
      if $dist_{v,v_n} < dist$
        $dist = dist_{v,v_n}$, $v^* = v$
      endif
    endfor
  the shortest route from v_n to v^* is recommended to taxi n
  $\bar{z}_i \leftarrow v^*$
  endfor
endfor

The number of zones, the sizes of the zones, and the number of idle taxis assigned to each zone determine the complexity of Algorithm 2. In this paper, we limited the size of a zone $|z_i|, \forall z_i \in Z$ to below 120. Although $|Z|$ is over 1900 zones after segmentation (Section 5.1), the number of zones assigned to idle taxis in t is much smaller than $|Z|$. In addition, the idle taxis in each assigned zone is also much smaller than N_t. Therefore, it is not costly to implement Algorithm 2 at the terminal.

5. Results

The results include two groups. In the first group, we present the result of the graph segmentation and predict the taxi demand with the proposed LSTM-GCN model using the data collected from Xi'an, China. The second group presents the simulation results used to validate our proposed TCRR strategy. The experiment and the simulation were coded in Python 3.6, configured in the platform with an eight-core 3.5 GHz Intel Core i7-7000X CPUs, 64 GB RAM, and NVIDIA GeForce GTX 1080 Ti.

5.1. Result of Taxi Demand Prediction

In Xi'an, China, we obtained 30,549 links from the road network that spans the entire city. Additionally, we collected GPS trajectories from more than 11,000 taxicabs while maintaining anonymity. The data collection process yielded over 2.7 billion raw data records, which spanned a period of 90 days from 1 September 2016 to 29 November 2016. These records included the time stamp indicating when the data were reported, the geographical coordinates, the instantaneous travel speed, and the travel state. The travel state could take one of three values: stop, cruising, or occupied.

We implemented map matching since the GPS trajectories of the collected data usually fell into incorrect links. Due to the limitation of the data, we only had historical records of taxi demand inferred from the states of taxis through their trajectories. In other words, the taxi demands we collected were the orders that had been accepted by taxis. In practice, the real taxi demands were greater than the records collected from the data because of the limited usage of taxis in some zones, especially during peak hours. However, as all the counterparts were implemented based on the same set of data, we could ignore the impact caused by the incompleteness of the taxi demand data.

In the GN-SLPA, we set the arguments $\epsilon_{overlap}$ and ϵ_{size} to 0.5 and 120, respectively. From Figure 4 [50], we can observe that the number of zones segmented with the SLPA converged to 1430 after 1200 iterations. Based on this observation, we set $\mathcal{I}_\mathcal{G}$ to 1200. Figure 5 [50] shows that we had 1945 zones after segmentation using the GN-SLPA. The zone with the maximal size had 113 links. Compared with the SLPA, the size of each zone was well controlled.

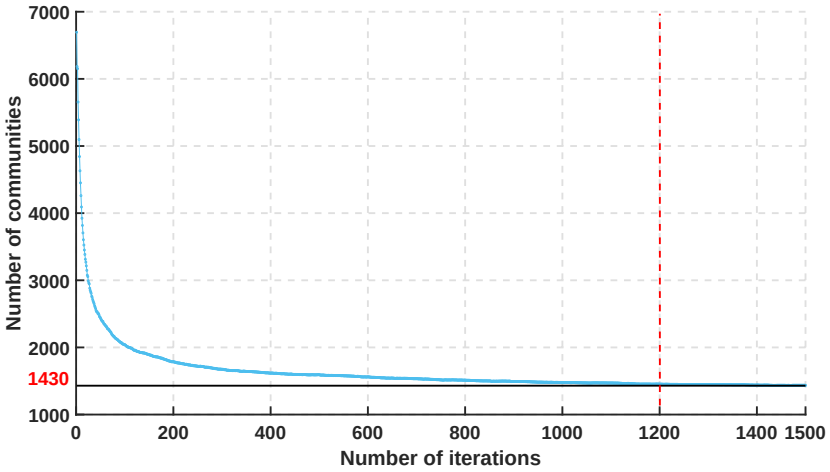


Figure 4. Number of zones varying with the increase in iterations [50].

We grouped a day into 288 time slices, denoted $\mathcal{T} = \{t_i | i = 0, 1, 2, \ldots, 287\}$, where each time slice had a length of $\tau = 5$ min. In our paper, we set the average travel speed to 26.18 km/h. Given $\tau = 5$ min, the average travel distance of a taxi was 2.18 km, which was long enough for a taxi to travel from one zone to another. Additionally, we considered the fact that there were few observations from 0:00 am to 5:00 am and from 9:00 pm to midnight in our data. Therefore, we only implemented the experiment over 15 h, from 6:00 am to 9:00 pm, and did the same for the simulation described in the next section.

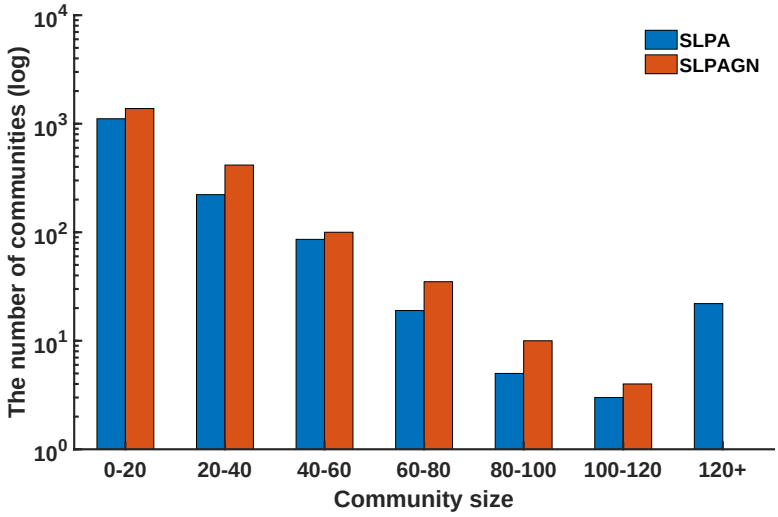


Figure 5. Number of communities based on the SLPA in different iterations [50].

With this, we acquired 210 (15×12) LSTM sequences in the first subneural network, and 7 LSTM sequences in the second subneural network. To implement the graph convolution in each LSTM sequence, we set $\mathcal{I}_\kappa = 3$. Indeed, a large value of $\mathcal{I}_\kappa$ can enhance the accuracy of predicting the consumption of many computational resources. The input of the model had a graph structure based on the graph segmentation shown in Figure 5 [50]. After the model training, we implemented a single-step prediction. With a batch size of 32 and a learning rate of 0.0001, we used ten-week, one-week, and one-week data, respectively, to train, validate, and test the model.

We first compared our predictor to the models without considering the spatiotemporal correlation and multiple time-scale features.

- Historical average (HA) model: this model takes the average value of taxi demand over the past ten time intervals as the final prediction result.

- ARIMA model: commonly used for forecasting traffic, this is a classical time series model.
- Vector autoregressive (VAR) model: this is another typical time series model.

The metric to measure the performance was based on the root-mean-square error (RMSE), formulated by

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\frac{1}{|Z|}\sum_{z_i\in Z}(y_{z_i,t}-\widehat{y}_{z_i,t})^2}, \quad (32)$$

where $\widehat{y}_{z_i,t}$ is the predicted taxi demand in zone z_i in t. From Figure 6, it can be seen that our proposed method performed best. It outperformed the ARIMA, VAR, and HA models on all the tested days. On the one hand, compared with time series models like ARIMA and VAR, our predictor could capture both regular and irregular dynamic taxi demand patterns. On the other hand, the counterparts did not consider the spatial correlation in taxi demand. It further validates our analysis that the patterns hidden in drivers' behaviors are not simply captured using a linear method, as demonstrated in Figure 2b.

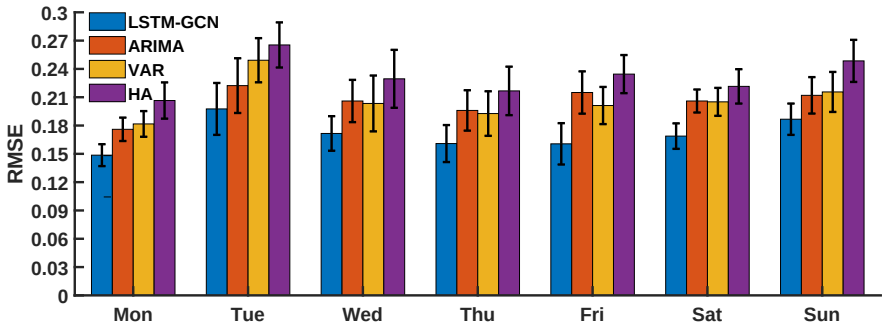


Figure 6. Comparison between the LSTM-GCN, ARIMA, VAR, and HA.

We also compared LSTM-GCN with the following two models in which spatiotemporal correlation was considered.

- Spatiotemporal GCN (STGCN) model: This model was developed by Guo et al. [32] to predict traffic flow. We slightly modified the model and employed it for taxi demand prediction. In the STGCN, we set the kernel size along the temporal dimension to three. There were 16 graph convolution kernels in each graph convolution layer and 32 convolution kernels in each temporal convolution layer. Unlike the model in [32], where the dynamics of traffic flow in the time scale of one week were considered, in our paper, only one-hour and one-day dynamics of taxi demand were considered.
- LSTM-GCN without considering multiple time-scale features (LSTM-GCN$^-$): this model only considered the one-hour dynamics of taxi demand.

With ARIMA as the baseline, we can see that all the models that considered spatiotemporal correlation performed better than ARIMA in Figure 7. This further validates the impact of spatiotemporal correlation on accurate taxi demand prediction. Comparing the LSTM-GCN with the STGCN, our method achieved its best performance in most cases, except for those on Wed. and Fri. This is attributed to the fact that the STGCN employed a GCN and stacked multiple spatial–temporal Conv-blocks where each block consisted of multiple GNN layers to capture spatiotemporal correlation. In addition, the STGCN also considered multiple time-scale features. However, as discussed in Section 4.3, the hyperparameters like the number of GNN layers were set empirically and, by nature, followed the distance pattern. The analysis in Figure 2 reveals that the distribution of taxi demand is closely connected to drivers' behaviors, rather than being influenced by the physical distance pattern. As a result, the true potential of the STGCN cannot be adequately demonstrated. Moreover, the efficiency of the STGCN is constrained by hyperparameters like the number of graph convolution layers. Additionally, the STGCN, together with the LSTM-GCN$^-$, does not consider multiple time-scale features. The limitation mentioned above has a

detrimental impact on the precise prediction of taxi demand. This is because there is a distinct fluctuation in the demand for taxis between working days and weekends, taking into account the mobility of passengers. Therefore, both STGCN and LSTM-GCN^{-} had worse performance than our method on most tested days.

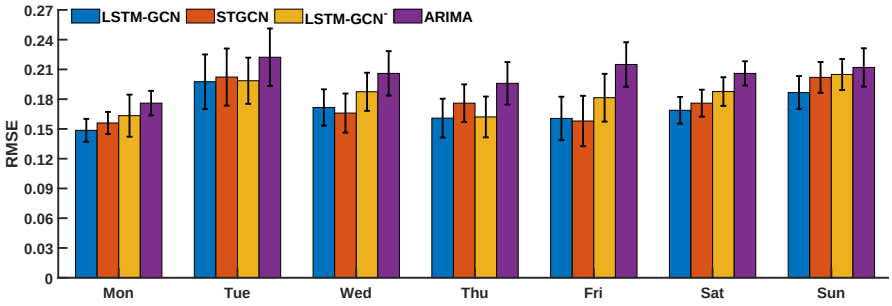


Figure 7. Comparison between the LSTM-GCN, STGCN, and LSTM-GCN^{-}.

5.2. Results of Simulation

We validated the proposed distributed LDD algorithm in a simulated environment based on the road network. We allocated 8000 taxis in the whole road network, following the principle that taxi drivers prefer to stay in zones with high demand for taxis. The number of taxis in each zone was determined by

$$N_{z_i,t_0} = \left[N_{t_0} \times \frac{y_{z_i,t_0}}{\boldsymbol{y}_{z,t_0}} \right], \tag{33}$$

where t_0 is the first time slice between 5:00 am and 5:05 am, and $[\cdot]$ is the round operation. In each time slice t_i, a taxi dispatch algorithm, e.g., the KM (Kuhn–Munkres) algorithm [51], was implemented based on the taxi demand data collected on 27 November 2016. The TCRR system collected all the taxis states, with which we obtained data on idle taxis without any orders assigned at the end of t_i. From this, we first predicted the taxi demand in t_{i+1} by the proposed LSTM-GCN model. Then, we implemented the distributed LDD-based algorithm and recommended cruising routes for idle taxis. The above procedure was iteratively implemented every 5 min until 9:00 pm. We compared our method to the following four counterparts:

- Empirical method: this method recommends an idle taxi with a route toward the nearest zone in which there is the highest taxi demand, estimated by the statistic learning method [4].
- MDP-based method: This is a model based on Yu et al.'s model [2], with a slight modification. The model focuses on long-term profit, with the assumption that passengers' arrival rate in each link follows a one-dimensional space–time Poisson process distribution. In this paper, we configured the arrival rate based on the historical data of taxi demand. Additionally, another important parameter for the MPD method is taxi density, which was also configured based on the historical data.
- Single-object-formulated model: Either objective $f_1(\boldsymbol{x})$ or objective $f_2(\boldsymbol{x})$ is considered in the problem formulation of TCRR. The same taxi demand prediction and distributed LDD algorithm are used to obtain the solution to each single-object-formulated problem. For the ease of description, we named these two single-object-formulated models $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$, respectively.

To evaluate the performance of different schemes, we employed the unit profit and occupancy rate over the road network as the metrics. The unit profit was calculated as the average profit earned by a taxi in an hour. The occupancy rate was estimated as the percentage of taxis that pick up passengers per hour. The parameters of our distributed algorithm are shown in Table 2.

Table 2. The configuration of parameters in the distributed LDD-based algorithm.

Parameter	Value
f	$8, r^{\mathcal{O}} \leqslant 2$ km $8 + 2(r^{\mathcal{O}} - 2), r^{\mathcal{O}} > 2$ km
c	0.64
Average speed	24.5 km/h during morning peak hours 26.2 km/h during off-peak hours 24.2 km/h during evening peak hours
$\mathcal{I}_{\mathcal{D}}$	30
$\lambda_{\mathcal{D}}$	0.5
ϵ_{thr}	8
w_1 w_2	0.5
sf_1 sf_2	Up-bound of $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ based on $x_{nm} = 1, \forall n \in N_t, \forall n \in M_{t+1}$

Table 3 presents the average profit and occupancy rates over three hours, which were selected from the morning peak hours, off-peak hours, and evening peak hours through observations of the taxi demand variation in the real data. A more accurate classification for both peak and off-peak hours can be obtained using clustering algorithms like K-nearest neighbors, but this is not the focus of this study. We also present the mean values of two performance metrics using the simulated results for each hour between 6:00 am and 9:00 pm. In Table 3, we can observe that our method had the best performance of all the models regarding the average profit and occupancy rate over one day. Particularly, our method had a 7% (mean) and 4% (mean) improvement, with approximately 400 more taxis receiving these benefits, compared to the empirical method. Although the average profit and occupancy rate values obtained using the MDP-based approach were close to ours, Figure 8 shows that much more time (approximately 2.5 to 4 more time than our scheme) was consumed when using the MDP. This demonstrates that implementing the cruising recommendation algorithm in a distributed manner can significantly reduce the computational complexity. In real-life scenarios, the distributed algorithm is more practical as it aligns with the drivers' natural behavior of cruising and picking up passengers based on their own experiences. This enhances the robustness of the solution. In addition, the assumption of the one-dimensional-space Poisson process distribution in the MDP-based method has the limitation of requiring an accurate estimation of taxi demand in real life.

Table 3. The performance of our proposed distributed scheme and that of its counterparts.

Method	Average Profit				Occupancy Rate			
	7:00–8:00 Peak	14:00–15:00 Off-Peak	18:00–19:00 Peak	Mean	7:00–8:00 Peak	14:00–15:00 Off-Peak	18:00–19:00 Peak	Mean
Empirical method	1.87	1.84	2.12	1.98	0.69	0.70	0.73	0.71
MDP-based method	1.92	1.93	2.23	2.08	0.72	0.73	0.76	0.74
$f_1(\boldsymbol{x})$-based method	1.93	1.93	2.24	2.05	0.74	0.75	0.76	0.74
$f_2(\boldsymbol{x})$-based method	1.97	1.96	2.25	2.07	0.73	0.74	0.75	0.73
Distributed method	1.97	2.01	2.31	2.12	0.73	0.75	0.78	0.75

From the comparison between our method and $f_1(\boldsymbol{x})/f_2(\boldsymbol{x})$, it is obvious that the optimal global profit can be achieved even when taxis keep idle for a long time. For instance, between 6:00 pm and 7:00 pm, the occupancy rate of $f_1(\boldsymbol{x})$ was larger than that of $f_2(\boldsymbol{x})$, but the average profit of $f_1(\boldsymbol{x})$ was lower than that of $f_2(\boldsymbol{x})$. This further validates our intuition that the two objectives, global profit maximization and individual profit maximization, conflict with each other. Encouragingly, our method solves this problem by making a trade-off between these two objectives through the synergy of the competition and collaboration among taxis.

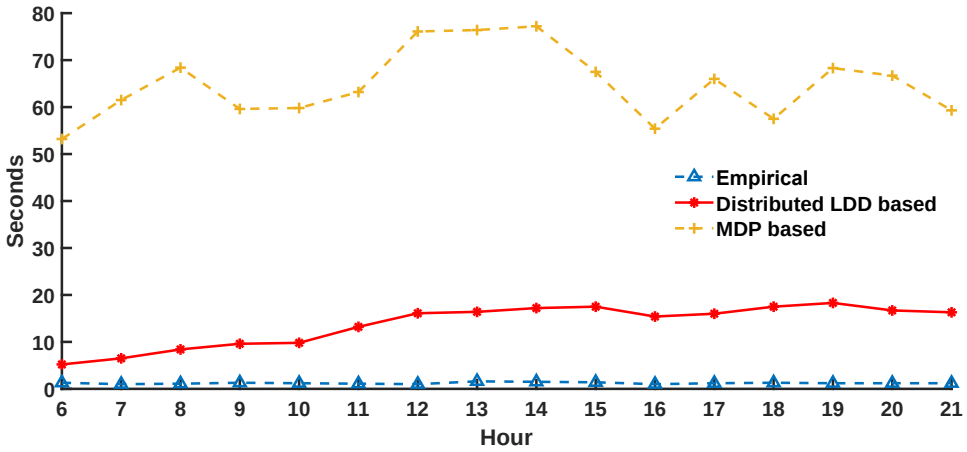


Figure 8. Running time of different TCRR strategies at each hour.

6. Conclusions

In this study, we formulated the TCRR as a biobjective optimization problem of the short-term profit of taxis. Unlike most existing studies that only focus on the individual profit of taxis, without much consideration for the cooperation and competition between taxis, the biobjective optimization attempts to employ a trade-off between the individual and the global profit, which is more reasonable in real life. To obtain the optimal solution, we developed a method that integrated a taxi demand predictor, namely the LSTM-GCN, and a distributed algorithm based on LDD. The LSTM-GCN has a lightweight structure but can capture the spatiotemporal correlation and the multiple time-scale features of taxi demand. The proposed distributed algorithm was implemented at each taxi and terminal server, with a reduction in running time: our model used four times less computation time than centralized approaches. The experimental results showed the good performance of the LSTM-GCN model compared with methods like time series models, e.g., ARIMA. Additionally, the simulation results verified that our method had a 7% and 3% improvement in average profit and occupancy rate over one day compared to an empirical-based approach. The experimental results, along with the simulation results, validated the novelty of our method from the following perspectives: (i) a reduction in idle taxis, (ii) an improvement in efficiency to obtain the optimal solution, (iii) an alleviation of benefit conflict, and (iv) an increase in the global profit of taxis.

In the future, we intend to extend our research into cases where the long-term profit of taxis is considered. Different from short-term profit, considering long-term profit is more challenging since it has more uncertainties in the variation in taxi demand and the mobility of taxis. Furthermore, the taxi dispatch and TCRR strategies interact with each other. We will also provide insights into the interactions between the taxi dispatch and TCRR strategies in our future research.

Author Contributions: Conceptualization, Y.L. and B.Z.; methodology, Y.L.; programming, Y.H. and Z.L.; validation, Y.L. and Y.H.; formal analysis, Y.L. and B.Z.; writing—Y.L. and Y.H.; writing—review and editing, B.Z.; supervision, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the project "Research on efficient supply and demand optimization of ride-hailing under multimodal scene changes in intelligent transportation systems" authorized by the Natural Science Foundation of Liaoning Province with grant number 02110076523001.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lai, Y.; Lv, Z.; Li, K.C.; Liao, M. Urban Traffic Coulomb's Law: A New Approach for Taxi Route Recommendation. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20* , 3024–3037. [CrossRef]
2. Yu, X.; Gao, S.; Hu, X.; Park, H. A Markov decision process approach to vacant taxi routing with e-hailing. *Transp. Res. Part B Methodol.* **2019**, *121*, 114–134. [CrossRef]
3. Moreira-Matias, L.; Gama, J.; Ferreira, M.; Mendes-Moreira, J.; Damas, L. Predicting taxi–passenger demand using streaming data. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1393–1402. [CrossRef]
4. Yuan, J.; Zheng, Y.; Zhang, L.; Xie, X.; Sun, G. Where to find my next passenger. In Proceedings of the 13th International Conference on Ubiquitous Computing, Beijing, China, 17–21 September 2011; pp. 109–118.
5. Wang, R.; Chow, C.Y.; Lyu, Y.; Lee, V.C.; Kwong, S.; Li, Y.; Zeng, J. Taxirec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 585–598. [CrossRef]
6. Wan, X.; Ghazzai, H.; Massoud, Y. A generic data-driven recommendation system for large-scale regular and ride-hailing taxi services. *Electronics* **2020**, *9*, 648. [CrossRef]
7. Garg, N.; Ranu, S. Route recommendations for idle taxi drivers: Find me the shortest route to a customer! In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1425–1434.
8. Qu, B.; Yang, W.; Cui, G.; Wang, X. Profitable Taxi Travel Route Recommendation Based on Big Taxi Trajectory Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 653–668. [CrossRef]
9. Rossi, A.; Barlacchi, G.; Bianchini, M.; Lepri, B. Modelling taxi drivers' behaviour for the next destination prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2980–2989. [CrossRef]
10. Zhou, X.; Rong, H.; Yang, C.; Zhang, Q.; Khezerlou, A.V.; Zheng, H.; Shafiq, M.Z.; Liu, A.X. Optimizing Taxi Driver Profit Efficiency: A Spatial Network-based Markov Decision Process Approach. *IEEE Trans. Big Data* **2018**, *6*, 145–158. [CrossRef]
11. Qu, M.; Zhu, H.; Liu, J.; Liu, G.; Xiong, H. A cost-effective recommender system for taxi drivers. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 45–54.
12. Yuan, N.J.; Zheng, Y.; Zhang, L.; Xie, X. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 2390–2403. [CrossRef]
13. Rodrigues, F.; Markou, I.; Pereira, F.C. Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. *Inf. Fusion* **2019**, *49*, 120–129. [CrossRef]
14. Ke, J.; Zheng, H.; Yang, H.; Chen, X.M. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 591–608. [CrossRef]
15. Sharma, A.; Sharma, A.; Nikashina, P.; Gavrilenko, V.; Tselykh, A.; Bozhenyuk, A.; Masud, M.; Meshref, H. A graph neural network (GNN)-based approach for real-time estimation of traffic speed in sustainable smart cities. *Sustainability* **2023**, *15*, 11893. [CrossRef]
16. Oluwasanmi, A.; Aftab, M.U.; Qin, Z.; Sarfraz, M.S.; Yu, Y.; Rauf, H.T. Multi-head spatiotemporal attention graph convolutional network for traffic prediction. *Sensors* **2023**, *23*, 3836. [CrossRef]
17. Chen, Z.; Zhao, B.; Wang, Y.; Duan, Z.; Zhao, X. Multitask learning and GCN-based taxi demand prediction for a traffic road network. *Sensors* **2020**, *20*, 3776. [CrossRef] [PubMed]
18. Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; Li, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
19. Xu, J.; Rahmatizadeh, R.; Boloni, L.; Turgut, D. Real-time prediction of taxi demand using recurrent neural networks. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2572–2581. [CrossRef]
20. Cao, Y.; Liu, L.; Dong, Y. Convolutional Long Short-Term Memory Two-Dimensional Bidirectional Graph Convolutional Network for Taxi Demand Prediction. *Sustainability* **2023**, *15*, 7903. [CrossRef]
21. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
22. Geng, X.; Li, Y.; Wang, L.; Zhang, L.; Yang, Q.; Ye, J.; Liu, Y. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019;
23. Dong, H.; Zhang, X.; Dong, Y.; Chen, C.; Rao, F. Recommend a profitable cruising route for taxi drivers. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 2003–2008.
24. Verma, T.; Varakantham, P.; Kraus, S.; Lau, H.C. Augmenting decisions of taxi drivers through reinforcement learning for improving revenues. In Proceedings of the Thirty-Third International Conference on Automated Planning and Scheduling, Prague, Czech Republic, 8–13 July 2023; pp. 18–23.
25. Chen, B.; Cheng, H.H. A review of the applications of agent technology in traffic and transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 485–497. [CrossRef]

26. Boostanimehr, H.; Bhargava, V.K. Unified and distributed QoS-driven cell association algorithms in heterogeneous networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 1650–1662. [CrossRef]
27. Weyns, D.; Holvoet, T.; Helleboogh, A. Anticipatory vehicle routing using delegate multi-agent systems. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Bellevue, WA, USA, 30 September–3 October 2007; pp. 87–93.
28. Logi, F.; Ritchie, S.G. A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 507–527. [CrossRef]
29. Gokulan, B.P.; Srinivasan, D. Distributed geometric fuzzy multiagent urban traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 714–727. [CrossRef]
30. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
31. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
32. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27–28 January 2019; Volume 33, pp. 922–929.
33. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
34. Marler, R.T.; Arora, J.S. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **2004**, *26*, 369–395. [CrossRef]
35. Duan, P.; Mao, G.; Kang, J.; Huang, B. Estimation of Link Travel Time Distribution With Limited Traffic Detectors. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3730–3743. [CrossRef]
36. Wang, H.; Tang, X.; Kuo, Y.H.; Kifer, D.; Li, Z. A simple baseline for travel time estimation using large-scale trip data. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–22. [CrossRef]
37. Gentili, M.; Mirchandani, P.B. Review of optimal sensor location models for travel time estimation. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 74–96. [CrossRef]
38. Yuan, N.J.; Zheng, Y.; Xie, X. *Segmentation of Urban Areas Using Road Networks*; MSR-TR-2012–65. Microsoft Research Technical Report; Microsoft Corporation: Redmond, WA, USA, 2012.
39. Ventresque, A.; Bragard, Q.; Liu, E.S.; Nowak, D.; Murphy, L.; Theodoropoulos, G.; Liu, Q. SParTSim: A space partitioning guided by road network for distributed traffic simulations. In Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, Dublin, Ireland, 25–27 October 2012; pp. 202–209.
40. Steed, A.; Abou-Haidar, R. Partitioning crowded virtual environments. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, Osaka, Japan, 1–3 October 2003; pp. 7–14.
41. Xie, J.; Szymanski, B.K. Towards linear time overlapping community detection in social networks. In Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining—Volume Part II, Kuala Lumpur, Malaysia, 29 May–1 June 2012; pp. 25–36.
42. Harenberg, S.; Bello, G.; Gjeltema, L.; Ranshous, S.; Harlalka, J.; Seay, R.; Padmanabhan, K.; Samatova, N. Community detection in large-scale networks: A survey and empirical evaluation. *Wiley Interdiscip. Rev. Comput. Stat.* **2014**, *6*, 426–439. [CrossRef]
43. Duan, P.; Mao, G.; Liang, W.; Zhang, D. A Unified Spatio-Temporal Model for Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 3212–3223. [CrossRef]
44. Si, C.; Chen, W.; Wang, W.; Wang, L.; Tan, T. An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition. *arXiv* **2019**, arXiv:1902.09130.
45. Hong, J.H.; Liu, X. The optimal pricing for green ride services in the ride-sharing economy. *Transp. Res. Part D Transp. Environ.* **2022**, *104*, 103205. [CrossRef]
46. Feng, X.; Guo, J.; Qin, B.; Liu, T.; Liu, Y. Effective Deep Memory Networks for Distant Supervised Relation Extraction. In Proceedings of the 26th International Joint Conference on Artificial Intelligence Main Track, Melbourne, Australia, 19–25 August 2017; pp. 4002–4008.
47. Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Eng.* **2018**, *5*, 1502242. [CrossRef]
48. Afshari, H.; Hare, W.; Tesfamariam, S. Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. *Appl. Soft Comput.* **2019**, *83*, 105631. [CrossRef]
49. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
50. Duan, P. Modeling, Analysis and Application of Big Traffic Data for Intelligent Transportation Systems. Ph.D. Thesis, University of Technology Sydney, Sydney, Australia, 2019.
51. Xu, Z.; Li, Z.; Guan, Q.; Zhang, D.; Li, Q.; Nan, J.; Liu, C.; Bian, W.; Ye, J. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 905–913.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.