A Grid-Based Taxi Dispatch Method Based on Reinforcement Learning (Postprint)

Authors: He Shengxue

Date: 2018-05-02T00:00:00+00:00

Abstract

Highly informatized grid-based urban management can provide novel real-time dynamic passenger demand information and vehicle location information for taxi operation optimization. Seizing this opportunity, this paper proposes a reinforcement learning control method for grid-based real-time dynamic taxi dispatching to address the problems of high vacant driving rates and low driver-passenger matching rates in urban taxi services. By providing taxis with dynamic optimal routes for empty cruising, the new control method aims to enhance taxi service efficiency and reduce passenger waiting time. First, based on urban unit grids, the key issues in taxi dispatching are identified; second, a reinforcement learning model for dispatching is established with dynamic adjustment of empty cruise routes as the control mechanism; finally, a Q-learning algorithm for solving the model is presented, and the effectiveness of the new dispatching method is verified through a numerical example. The study demonstrates that the new method can effectively improve driver-passenger matching rates, increase total taxi operation revenue, reduce average passenger waiting time, and decrease total taxi vacant driving time.

Full Text

Grid-Based Taxi Dispatching Method Based on Reinforcement Learning

He Shengxue
(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Highly-informed grid-based city management can supply real-time passenger information and position information of taxis for taxi operation optimization. On this account, we propose a grid-based taxi dispatching dynamic control method based on reinforcement learning to solve the problem of high vacant taxi rates and low matching rates between taxis and passengers. By providing optimal cruising routes for vacant taxis, the new control method aims to improve the service level of taxis and lower passenger waiting times. Firstly, based on city grids, we clarify the key problem of taxi dispatching. Secondly, by using the adjustment of vacant taxi routes as the control action, we formulate the reinforcement learning model of taxi dispatching. Finally, we propose the corresponding Q-learning algorithm to solve the new model. Numerical examples demonstrate the effectiveness of the new dispatching method. The results show that the new method can not only increase the match rate between taxis and passengers and the total income of taxi service operations, but also reduce the average waiting time of passengers and the total travel time of vacant taxis.

Keywords: urban transportation; taxi dispatching; reinforcement learning; grid management; adaptive control

0 Introduction

The rapid development of the information age presents new opportunities and challenges for transportation science research. The widespread application of big data technology has transformed the empirical estimation of taxi travel demand into real-time dynamic network-based characteristic analysis and prediction. Advances in GPS and GIS provide management departments with more comprehensive information about taxi real-time positions and trajectories. Traditional street-hailing taxi services are gradually being replaced by various APP-based information platforms. These changes offer opportunities for taxi companies and relevant management departments to further optimize system operations. How to effectively utilize real-time vehicle location information and passenger travel demand distribution to achieve real-time dispatching optimization has become a primary technical challenge for taxi enterprises. Relying on grid-based city management, this paper proposes a dynamic taxi dispatching control method based on grid-based travel demand information updates and taxi route optimization.

Below is a brief review of relevant research on taxi dispatching. Reasonable taxi dispatching can effectively reduce costs generated by vacant cruising, sometimes by as much as 90% [?, ?]. Early research focused on reasonable pricing and overall fleet size, while current studies involve various aspects of taxi service systems. These mainly include network-scale taxi operation modeling [?, ?], stochastic passenger travel demand [?], taxi e-hailing systems [?, ?], taxi operation based on cellular networks [?], demand-responsive taxi services [?], and analysis of driver-passenger matching processes [?, ?, ?]. The common feature of these studies is the assumption that vacant taxi operations satisfy network equilibrium, travel demand information is known in advance, and taxi cruising speeds are given. Under equilibrium conditions, each vacant taxi chooses the nearest area with maximum potential revenue as its destination, and no taxi can obtain higher revenue by unilaterally changing its vacant cruising route. These studies seldom consider the impact of dynamically changing travel demand and real-time traffic conditions, making it difficult to meet the development needs of today' s highly information-based taxi market.

With the emergence of numerous taxi-hailing APPs, taxi dispatching has become increasingly important for improving driver-passenger matching rates [?]. Researchers have modeled the taxi dispatching optimization problem from the perspectives of the Vehicle Routing Problem (VRP) [?, ?] and the Bipartite Graph Matching Problem (BGMP) [?, ?, ?]. Generally, VRP-based studies assign a series of passengers to each taxi, while BGMP-based studies match taxis and passengers based on proximity principles. The common shortcoming of these studies is insufficient consideration of the randomness in taxi travel times and passenger travel demand, limiting their practical applicability.

To address these research gaps, this paper makes improvements in four aspects: a) By defining a grid-based block diagram of the city, we establish dynamic time distributions of travel rates and destination selection rates for each grid based on big data of passenger travel (obtainable from hailing APP platforms and actual surveys). These data are used to predict demand and determine vacant taxi routes during control method implementation. Actual travel demand can also be reflected through real system state variables and utilized by the dispatching method; b) To better reflect the impact of real-time traffic network conditions on taxi cruising speeds, the new control method can update arrival times to destinations using real-time traffic information and current taxi GPS positions at each control moment (by recalculating shortest paths), thereby capturing the stochastic characteristics of taxi cruising speeds and improving the reliability of existing dispatching systems; c) The new dispatching method is based on a simulation model of taxi system operation, enabling more detailed state evolution and more accurate control action selection. For example, precise description of each key time point in a passenger' s travel process; d) Based on reinforcement learning theory, the new control method can effectively handle online and offline optimization learning and adaptively adjust to various external changes, such as surges in travel demand.

1.1 Basic Parameters and Variables

Let K denote discrete time steps, i.e., the time interval for implementing dispatching control. Let k denote the time segment index, where $t_k = kT$ represents a time point for implementing dispatching control, and K is the time range for implementing dispatching control.

Let P denote the set of all passengers, with p representing a typical passenger. Let V denote the set of all taxis, with v representing a typical taxi. Let G denote the set of all unit grids, with g representing a typical grid. Let $\gamma_{g,h}$ denote the passenger generation rate in grid g during a time segment. Let $r_{g,h}$ denote the probability that a passenger generated in grid g chooses another grid h as their destination.

Let O_p denote the departure grid (origin) of passenger p. Let D_p denote the destination grid of passenger p. Let $t_{p,1}$ denote the generation moment of passenger p, i.e., when the passenger requests service via APP. Let $t_{p,2}$ denote the boarding moment of passenger p at their origin grid. Let $t_{p,3}$ denote the alighting moment when passenger p arrives at their destination. Let W_p denote the waiting time of passenger p. When a taxi satisfies their demand, $W_p = t_{p,2} - t_{p,1}$. Let R_p denote the travel distance between passenger p' s origin and destination. Let η_v denote the status of taxi v. When the taxi has a passenger, η_v takes value 1; otherwise, it is 0. Let g_v denote the marked grid of taxi v. When the taxi has a passenger, g_v takes value D_p; otherwise, it is the actual grid where taxi v is located. Let t_v denote the waiting activation time of taxi v, i.e., the time interval until the next dispatching action is implemented on that vehicle. When the taxi has a passenger, t_v denotes the travel time to grid D_p; otherwise, the value of t_v is 0.

Let G_g^n denote the set of grids that a taxi can reach from grid g within n time steps, i.e., the n-level neighborhood of grid g. Let P_g^k denote the set of passengers waiting for taxis in grid g at time k, with $|P_g^k|$ representing the total number of passengers in the set. Let V_g^k denote the set of vacant taxis in grid g at time k, with $|V_g^k|$ representing the total number of taxis in the set. Let $V_{g,k}^{k+1}$ denote the set of taxis carrying passengers and traveling toward grid g at time k, with $|V_{g,k}^{k+1}|$ representing the total number of taxis in the set.

Let c_0 denote the taxi' s starting fare. Let c denote the unit price per kilometer beyond the starting distance d_0. Let t_p denote the maximum acceptable waiting time for passengers at their origin.

1.2 Grid-Based Taxi Dispatching Concept

[Figure 1: see original paper] shows a schematic diagram of a gridded city. Passenger p travels from origin O_p to destination D_p along the dotted path. Any passenger' s travel route is determined by the dispatching system through a shortest path search algorithm. There are two ways to define the 1-level neighborhood of grid g. One consists of the four grids immediately adjacent vertically and horizontally plus itself. The other further includes the four diagonally adjacent grids. The 2-level neighborhood of a grid is composed of the 1-level neighborhoods of all grids contained in its 1-level neighborhood. The first approach yields what is called a basic neighborhood, while the second yields an extended neighborhood. In the numerical analysis, this paper compares the impact of these two neighborhood construction methods on results.

Two passenger generation modes need to be distinguished. In practical applications, current passenger information is entirely provided by the corresponding APP platform, while future passengers are predicted from historical data. In the numerical analysis section, this paper simulates current passenger information and predicts future passenger information through given grid passenger generation rates $\gamma_{g,h}$ and destination distribution probabilities $r_{g,h}$. The specific operation can be divided into three steps. First, passengers for the current time period are randomly generated based on $\gamma_{g,h}$. Then, the destination of each new passenger is determined using the roulette wheel rule based on $r_{g,h}$. Finally, the passenger' s travel route is determined using a shortest path search algorithm.

In the passenger generation and driver-passenger matching process, system operation is essentially deterministic. The greatest challenge for the dispatching system is determining the cruising routes of vacant taxis after feasible matching is completed. In traditional street-hailing taxi services, drivers determine vacant cruising routes based on personal experience. However, this approach involves significant randomness, often resulting in too many taxis in some areas while passengers in other areas cannot be served promptly. In today' s highly information-based era, the APP-based nature of taxi services provides an opportunity to reconsider this problem from a system-wide perspective, i.e., optimizing vacant taxi cruising routes. The next section analyzes this problem in detail from a reinforcement learning perspective.

The core of taxi dispatching is determining the cruising routes of vacant vehicles, with the goal of reducing passengers' average waiting time, improving driver-passenger matching rates, reducing total vacant travel time, and increasing taxi operation revenue. This study uses the improvement of driver-passenger matching rates as the direct value assessment indicator for control actions. In the numerical analysis, we examine passengers' average waiting time, driver-passenger matching rates, total vacant travel time, and total taxi operation revenue.

2 Reinforcement Learning Model

The purpose of the reinforcement learning model is to determine the optimal policy that achieves certain objectives under given constraints. One objective of this study is to determine a reasonable relationship between vacant taxi cruising routes and grid state quantities to improve driver-passenger matching rates.

The reinforcement learning (RL) control model for the taxi dispatching system includes five main components: control actions, control information, state variables, state value functions, and control policies.

Assume the current moment is k. After feasible driver-passenger matching is completed in each grid, some areas may have passengers continuing to wait while other grids have vacant taxis. At this point, decisions on cruising routes for vacant taxis are needed. Let the 1-level neighborhood of the grid g where current vacant taxi v is located be G_g^1. Then any grid in G_g^1 can be chosen as the destination at moment $k+1$. Let $a_k \in A_k$ represent the control action at current moment k, and let the selected next destination grid be $g_{v,k+1}$. Then the grid vector composed of all $g_{v,k+1}$ constitutes the control action corresponding to current moment k, i.e., $a_k = \{g_{v,k+1} | v \in V_{g,k}^2\}$. Obviously, when set G_g^1 contains many elements, the feasible A_k will be very large. The purpose of reinforcement learning-based dispatching control is to select reasonable control actions at each control moment to achieve overall system optimization in space and time.

In system operation, random or uncertain factors accompanying the process are called control information of reinforcement learning. There are two main uncertainty factors in taxi dispatching. One is the randomness of passenger generation in unit grids; the other is the uncertainty of taxi travel time when carrying passengers. The former is determined by passenger generation rates $\gamma_{g,h}$ and destination distribution probabilities $r_{g,h}$ in the model, while the latter is reflected by the random waiting activation time t_v of taxis. For convenience of expression, these random factors are represented by an abstract vector ω_k. Vector ω_k is the control information required by the system.

At each control time point, the control system needs to determine the optimal control action based on the system' s current state and relevant control information. State variables are a set of feature vectors describing the system' s state at a given moment. Comprehensively describing the taxi service system requires 刻画大量细节. However, this not only makes state variables overly cumbersome but also makes subsequent control action decisions burdened by numerous secondary factors. Therefore, this study takes grids as the object and mainly considers three features: the number of waiting passengers, the number of vacant taxis, and the number of taxis carrying passengers to the grid. After passenger generation and driver-passenger matching are completed, for any grid g, the number of waiting passengers $|P_g^{k+1}|$ and the number of vacant taxis $|V_g^{k+1}|$ cannot both be positive. This paper selects $|P_g^{k+1}|$ and $|V_g^{k+1}|$ as the state features of grid g at moment $k+1$. Integrating the state features of all grids at moment $k+1$ yields the system state variable $s_{k+1} = \{|P_g^{k+1}|, |V_g^{k+1}|, |V_{g,k}^{k+1}|\}$. All feasible system state variable values constitute the system state space set S.

A control policy refers to a decision function that determines control actions from system state variables, i.e., given any system state, a unique corresponding control action can be obtained according to the control policy. Represent any control policy as $\pi \in \Pi$, where Π is the set of all feasible policies. The functional form of the control policy can be expressed as $a = \pi(s)$, where $a \in A(s)$.

Once a control action is determined for any control stage, it will inevitably affect the system' s current and subsequent operations. Generally, the immediate impact of a control action is relatively easy to measure. For example, in this paper, once the destination of a vacant vehicle for the next moment is determined, the resulting increase in driver-passenger matches will change accordingly. Due to the constantly changing system operating environment, the long-term system impact of control actions is often difficult to estimate accurately. In reinforcement learning, the change in the system's optimization objective directly caused by a control action is called the reward of that action, with the corresponding reward function denoted as $\Delta((s_k, a_k, \omega_{k+1}))$.

The calculation of the reward function value involves four steps. First, complete all feasible driver-passenger matching in each grid at the current stage and determine a specific control action a_k. Second, based on a_k, update the number of vacant taxis in each grid at stage $k+1$. Third, implement all possible matching between waiting passengers and vehicles that have just become vacant in each grid. Finally, implement matching between remaining waiting passengers and vacant taxis in each grid, and record the total number of such matches, which is the reward function value corresponding to control action a_k. The basis for this calculation is that vehicles that have just completed passenger-carrying tasks are located in the grid corresponding to D_p, while other vacant taxis are obtained from vacant taxis of the previous stage after route control (i.e., the effect of control action a_k).

Reasonable control decisions must consider the impact of control actions on subsequent system states and the time discounting of such impacts. Therefore, we select the following optimization objective:

$$\pi^* = \arg\max_{\pi\in\Pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \Delta(s_k, \pi(s_k), \omega_{k+1})\right]$$

where γ is the discount factor, and the operator $\mathbb{E}$ represents the expectation of the variable. Directly solving optimization problem (3) is very difficult, so the usual approach is to solve its equivalent Bellman equation instead. To simplify formula expression, we replace the time label "k" of variables with subscript "k" . Let the state value when the system is in state s_k be $V(s_k)$. According to the Bellman equation, the optimal state value $V^*(s_k)$ satisfies:

$$V^*(s_k) = \max_{a_k\in A(s_k)} \mathbb{E}\left[\Delta(s_k, a_k, \omega_{k+1}) + \gamma V^*(s_{k+1})\right]$$

The following section designs a corresponding Q-learning algorithm to solve the above problem.

3 Q-Learning Algorithm

The Q-learning algorithm is a very effective algorithm for solving reinforcement learning models [?]. By converting the state value function to a Q-function, the optimal control action can be selected by directly comparing Q-function values during algorithm execution. Through iterative learning, when the approximate form of the Q-function approaches the actual Q-function, the selection of control actions becomes more effective. When using the state value function to determine the optimal control action, it is necessary to perform state transitions of the system for each optional action to determine the value function of subsequent states. This is not only cumbersome compared to the Q-learning algorithm but also involves considerable computation due to system state transitions. Therefore, this paper selects the Q-learning algorithm as the basic algorithm for solving the reinforcement learning model.

In the algorithm, k represents the sequence number of the current stage, obviously corresponding to a control moment. m represents the sequence number of system simulation runs. K and M are the total number of stages and total number of simulation runs, respectively. The specific steps of the Q-algorithm for grid-based taxi dispatching optimization are as follows:

a) Initialization. For all system states s_0, provide approximate values of the value function $Q(s_0, a_0)$ and control actions $a_0 \in A(s_0)$, and initialize $k = 0$, $m = 1$, and parameters θ.

b) Select a sample path of random information ω.

c) Loop and iterate the following operations according to $k = 0, 1, \dots, K-1$:

   (a) Determine the control action using the ϵ-greedy rule. With probability ϵ, randomly select action a_k from the control action set $A(s_k)$. With probability $1-\epsilon$, select action a_k using the formula $a_k = \arg\max_{a\in A(s_k)} Q(s_k, a; \theta)$.

   (b) Sample the information ω_{k+1} and calculate the state variable s_{k+1}.

   (c) Calculate the reward $\Delta(s_k, a_k, \omega_{k+1})$.

   (d) Update parameters θ. The control action a_{k+1} can be determined by the ϵ-greedy rule. Then, based on the updated parameters θ, calculate the Q-factor according to:

$$\hat{q}_k = \Delta(s_k, a_k, \omega_{k+1}) + \gamma \max_{a'\in A(s_{k+1})} Q(s_{k+1}, a'; \theta)$$

and update:

$$Q(s_k, a_k; \theta) \leftarrow Q(s_k, a_k; \theta) + \alpha \left[\hat{q}_k - Q(s_k, a_k; \theta)\right]$$

where α is the learning rate.

d) If $m < M$, return to step b).

e) Output the final Q-factor $Q(s_k, a_k)$.

The purpose of the above algorithm can be seen as minimizing the function $z(\theta) = \frac{1}{2}\sum_{k=1}^{MK} \left(\hat{q}_k - Q(s_k, a_k; \theta)\right)^2$. The negative gradient direction of function $z(\theta)$ is $\nabla_\theta z(\theta) = -\sum_{k=1}^{MK} \left(\hat{q}_k - Q(s_k, a_k; \theta)\right) \nabla_\theta Q(s_k, a_k; \theta)$. According to classical nonlinear programming theory, within the feasible domain of variable θ, if the current θ does not correspond to a local or global minimum of the function value and the step size is sufficiently small, the value of the objective function $z(\theta)$ will descend along the negative gradient direction at the current variable θ. Therefore, the parameter λ in the algorithm is equivalent to the search step size along the feasible descent direction. A reasonable value for λ can be determined through trial and error.

The solution depends on the specific form of $Q(s, a)$, but currently no specific form exists for $Q(s, a)$. Therefore, it is necessary to use some approximate function $Q(s, a; \theta)$ with undetermined parameters θ. The most common approach is to use Artificial Neural Network (ANN) approximation techniques to complete this task. At this point, parameters θ represent the connection weights and node thresholds of the neural network [?, ?]. According to ANN theory, multi-layer ANNs can achieve unlimited approximation of various functions. Especially when the specific function form is unknown, the standardized unified structure of ANNs and effective parameter adjustment mechanisms provide a convenient and effective function approximation tool for research.

4 Numerical Experiments

This chapter validates and analyzes the proposed method using a taxi dispatching area composed of 15 grids as shown in [Figure 2: see original paper]. The grid numbers and corresponding passenger generation rates $\gamma_{g,h}$ are marked in the figure. The length and width of each grid are both 1000 m. The discrete time step T is 100 s, and the total number of time segments K is set to 100. The average taxi speed is 36 km/hr (i.e., 10 m/s). The taxi starting fare c_0 is 14 yuan, the starting distance d_0 is 3 km, and the unit price per kilometer beyond the starting distance c is 2.5 yuan. The maximum acceptable waiting time for passengers at their origin t_p is set to 400 s. The total number of taxis operating in the network is assumed to be 30. The total number of simulations M is 300, parameters ϵ and discount factor γ are both set to 0.5, and the algorithm step size λ is set to 0.01. The size of ϵ determines the proportion of using existing experience versus random selection of control actions during the learning process. A larger ϵ value means a higher probability of random control action selection, which may require more iterations in the early stage of algorithm operation to achieve good system performance, but later optimal control action selection will be more effective. When ϵ is smaller, system performance is better in the early stage of algorithm operation, but more iterations are needed to achieve later system performance improvement [?]. As a temporal discount factor for objectives, the size of γ reflects the algorithm' s discounted evaluation of system performance over time [?]. The above choices for ϵ and γ are just one possibility among various options, and their reasonable values should be determined through trial calculations in practical applications. The selection of step size λ is also just an example, and reasonable values should be determined through trial calculations in actual algorithm applications. Generally, when Q-function values are large, the value of λ should be small. Otherwise, due to drastic changes in the objective function, the algorithm will be extremely unstable in the early execution stage [?]. It is assumed that the random error in travel time for a taxi traveling 1 km follows a normal distribution with mean 0 and standard deviation of 20 s.

To make subsequent charts clearer and more concise, the following symbolic variables are defined: TC represents the total number of driver-passenger matches;

NS represents the number of passengers lost due to not receiving service; Wt represents the average passenger waiting time (s); IN represents total taxi operation revenue (yuan); OT represents total taxi vacant travel time (s). NC denotes the taxi operation scenario without control; CBN denotes dispatching control based on basic neighborhoods; CEN denotes dispatching control based on extended neighborhoods.

[Figure 3: see original paper] and [Figure 4: see original paper] respectively show the changes in the number of driver-passenger matches TC and the number of lost passengers NS with increasing simulation runs. As the number of simulations increases, TC shows an upward trend, while NS shows a downward trend. The random fluctuations in specific values stem from the stochastic nature of the system itself and the randomness of control action selection by the ϵ-greedy mechanism in the algorithm.

[Figure 5: see original paper] shows that the average passenger waiting time Wt fluctuates randomly around 130 seconds as the number of simulations increases. This indicates that increasing the number of simulations cannot significantly improve passengers' average waiting time. However, this paper finds that using extended neighborhoods can reduce average waiting time, and the average waiting time under control conditions is lower than in the no-control scenario.

compares the optimization results under three different control scenarios. The data for control scenarios come from system operation results after 300 simulations, excluding the ϵ-greedy mechanism and using only the best control actions. It can be seen that compared with the no-control scenario, dispatching control can significantly improve system operation efficiency, while dispatching based on extended neighborhoods is superior to dispatching based on basic neighborhoods in multiple aspects. Since the direct optimization objective is to increase the number of driver-passenger matches, dispatching based on extended neighborhoods requires taxis to travel vacant for more time to achieve more matches. This can explain why in , the total taxi vacant travel time OT for CEN is greater than that for CBN.

[Figure 6: see original paper] shows that total operation revenue IN presents a clear increasing trend with the number of simulations. [Figure 7: see original paper] shows that total taxi vacant travel time OT presents a clear decreasing trend with the number of simulations. These changes correspond to the increasing trend of driver-passenger matches shown in [Figure 3: see original paper].

The computer program for solving the numerical examples was written in Java 1.8.0 and implemented in the NetBeans IDE 8.0.2 development environment, using a computer processor of Intel® Core i3-3120M CPU. The computation time to complete 300 system simulations under both control modes was 14 s, and the computation time for a single simulation was approximately 0.04 s.

5 Conclusion

Against the backdrop of grid-based city management, and targeting grid-based dynamic taxi travel demand data and grid-based taxi route planning, this paper proposes a reinforcement learning control method for network taxi dispatching. The new control method can effectively handle the system' s stochastic dynamic characteristics (including random travel times and travel demand), and achieve vacant taxi route dispatching through unsupervised adaptive reinforcement learning. By defining grid and grid neighborhood concepts, the method can effectively utilize taxi travel demand characteristics and demand prediction based on grid big data during implementation. The dynamic route adjustment process also makes it possible to use real-time vehicle positioning information. In terms of taxi service system operation performance, the new dispatching control method can not only increase the number of driver-passenger matches and reduce the risk of passenger loss but also increase total taxi operation revenue and reduce passengers' average waiting time.

This research can be expanded in multiple aspects, including considering different categories of taxis, passenger priority levels, different grid division methods, and real-time impact analysis of actual road traffic conditions. Meanwhile, the effectiveness of the research method needs further improvement through empirical analysis.

References

[1] Zhu Jincheng, Shuai Bin, Sun Chaoyuan, et al. Urban taxi congestion pricing model and algorithm under fixed rate [J]. *Computer Application Research*, 2013, 30(8): 2288-2291.

[2] Zhan X, Qian X, Ukkusuri S V. A graph-based approach to measuring the efficiency of an urban taxi service system [J]. *IEEE Trans on Intelligent Transportation Systems*, 2016, 17(9): 2479-2489.

[3] Hu Jihua, Huang Ze, Deng Jun. Research on spatiotemporal accessibility of Guangzhou taxis considering travel direction [J]. *Computer Application Research*, 2014, 31(2): 454-456.

[4] Jung J, Chow J Y, Jayakrishnan R, et al. Stochastic dynamic itinerary interception refueling location problem with queue delay for electric taxi charging stations [J]. *Transportation Research Part C*, 2014, 40(1): 123–142.

[5] Sayarshad H R, Chow J Y. Survey and empirical evaluation of non-homogeneous arrival process models with taxi data [J]. *Journal of Advanced Transportation*, 2016, 50(7): 1275–1294.

[6] Zhang W, Ukkusuri S V. Optimal fleet size and fare setting in emerging taxi markets with stochastic demand [J]. *Computer-Aided Civil and Infrastructure Engineering*, 2016, 31(9): 647-660.

[7] He F, Shen Z M. Modeling taxi services with smart phone-based e-hailing applications [J]. *Transportation Research Part C*, 2015, 58(1): 93–106.

[8] Wang X, He F, Yang H, et al. Pricing strategies for a taxi-hailing platform [J]. *Transportation Research Part E*, 2016, 93(2): 212–231.

[9] Wong R, Szeto W, Wong S. A cell-based logit-opportunity taxi customer-search model [J]. *Transportation Research Part C*, 2014, 48(1): 84–96.

[10] Amirgholy M, Gonzales E J. Demand responsive transit systems with time-dependent demand: user equilibrium, system optimum, and management strategy [J]. *Transportation Research Part B*, 2016, 92(2): 234–252.

[11] Yang T, Yang H, Wong S C. Taxi services with search frictions and congestion externalities [J]. *Journal of Advanced Transportation*, 2014, 48(6): 575–587.

[12] Zha L, Yin Y, Yang H. Economic analysis of ride-sourcing markets [J]. *Transportation Research Part C*, 2016, 71(2): 249–266.

[13] Nie Y M. How can industry survive tide of ride-sourcing? Evidence from Shen Zhen, China [J]. *Transportation Research Part C*, 2017, 79(2): 242–256.

[14] Pillac V, Gendreau M, Gueret C, et al. A review of dynamic vehicle routing problems [J]. *European Journal of Operations Research*, 2013, 225(1): 1–11.

[15] Hosni H, Naoum-Sawaya J, Artail H. The shared-taxi problem: formulation and solution methods [J]. *Transportation Research Part B*, 2014, 70(3): 303–318.

[16] Jung J, Jayakrishnan R, Park J Y. Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing [J]. *Computer-Aided Civil and Infrastructure Engineering*, 2016, 31(4): 275–291.

[17] Agatz N, Erera A, Savelsbergh M, et al. Optimization for dynamic ride-sharing: a review [J]. *European Journal of Operations Research*, 2012, 223(2): 295–303.

[18] Nourinejad M, Roorda M J. Agent based model for dynamic ride sharing [J]. *Transportation Research Part C*, 2016, 64(1): 117–132.

[19] Warren B P. *Approximate Dynamic Programming-Solving the Curses of Dimensionality* [M]. Hoboken: Wiley, 2011.

[20] Dimitri P B, John N T, John T. *Neuro-Dynamic Programming* [M]. Nashua: Athena Scientific, 1996.

Note: Figure translations are in progress. See original paper for figures.