Optimal Trip-Vehicle Dispatch with Multi-Type Requests

Extended Abstract

Taoan Huang
Tsinghua University
hta15@mails.tsinghua.edu.cn

Bohui Fang
Shanghai Jiaotong University
fangbohui@sjtu.edu.cn

Hoon Oh
Carnegie Mellon University
hooh@andrew.cmu.edu

Xiaohui Bei
Nanyang Technological University
xhbei@ntu.edu.sg

Fei Fang
Carnegie Mellon University
feifang@cmu.edu

ABSTRACT

In recent years, traditional transportation platforms which mainly provide real-time ride-hailing services have started to accept rides scheduled in advance. The presence of both ride requests posted in real time and scheduled rides leads to new challenges to the service providers in deciding which requests to accept and how to dispatch the vehicles in a dynamic and optimal way, which, to the best of our knowledge, have not been addressed by existing works. To fill the gap, we provide the following contributions: (i) a novel two-stage decision-making model where in the first stage, the system decides whether to accept requests scheduled in advance in an online fashion, and in the second stage, dispatches vehicles to on-demand ride requests in real time given the accepted scheduled requests as constraints; (ii) novel algorithms for both stages that take an estimated distribution of on-demand ride requests into account to handle both on-demand and scheduled requests.

KEYWORDS

Trip-vehicle dispatch; Ride-hailing; On-demand request; Scheduled request

ACM Reference Format:

Taoan Huang, Bohui Fang, Hoon Oh, Xiaohui Bei, and Fei Fang. 2019. Optimal Trip-Vehicle Dispatch with Multi-Type Requests. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

The fast development of position-tracking technology, the growth in the number of smartphones users, and the reduced cost in cellular communications have together led to a revolution in mobility and the emergence of on-demand ride-hailing transportation systems. These systems produce a tremendous positive societal impact by reducing pollution and congestion while increasing personal mobility. One key research problem faced by transportation service providers is how to dispatch drivers or vehicles to pick up riders. There is a rising interest in this problem from the aspect of multi-agent control and coordination [1, 7, 10], and it is closely related to other research problems in multi-agent systems, including dispatching agents for collection and delivery of goods [16], traffic control [4, 13], metro lines scheduling, and last-mile transportation[6].

In recent years, the emergence of ride-hailing platforms which match drivers and riders in real time, such as Uber, Curb, Shenzhou Zhuanche and ComfortDelGro, leads to increased efficiency in urban transportation. In addition to supporting real-time on-demand ride requests, these platforms start to allow the riders to schedule rides in advance, providing them more flexibility in planning their trips. Shuttle service and other services which traditionally rely on advance booking also start to accept on-demand requests in real time [3].

Accepting scheduled requests reduces the demand uncertainty and gives the service provider more time to prepare and optimize these trips. However, one may raise his/her concern that certain scheduled requests may prevent the assigned driver from serving more valuable on-demand requests, hurting the overall revenue of the driver and the service provider. So it is important, meanwhile challenging, for the service providers to design a system that decides whether to accept the scheduled requests and how to dispatch vehicles optimally. Note that at first glance, one might be tempted to simply treat all scheduled requests as regular on-demand requests when their pick-up time is due, and directly apply an existing dispatch algorithm for on-demand requests [1, 8, 11, 14, 17, 19]. However, the challenges are mainly brought by the fundamental difference between scheduled and on-demand requests: any scheduled request, once accepted by the platform, becomes a commitment that the platform *must* fulfill. Such practice may lead to failure in serving committed scheduled requests and therefore, hurt the credibility of the service provider and even its long-term sustainability, since many scheduled requests are for important purposes such as catching a flight or attending an important meeting. This presents to the design of the dispatch with new challenges, which, to the best of our knowledge, have not been addressed by existing works.

In this paper, we study the trip-vehicle dispatch with both scheduled and real-time requests. First, we propose a novel two-stage decision-making model for this problem. In the first stage (Stage 1), the system is presented with a sequence of scheduled requests and needs to select which requests to accept and decide how to dispatch vehicles to the accepted requests in an online fashion. All scheduled requests are received before any of the on-demand requests. In the second stage (Stage 2), the system will receive on-demand requests on top of these scheduled requests in real-time. While ensuring all scheduled requests accepted in Stage 1 are satisfied, the system needs to dispatch vehicles to on-demand requests in real-time or suggest relocations of empty vehicles. While most work on trip-vehicle dispatch ignores uncertainty in demand and only

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

plans myopically based on received requests [1, 5, 8], in our work, we take a data-aware view and assume the platform knows the spatio-temporal distribution of the on-demand requests, which can be estimated from historical data. Recent work also emphasizes the value of data [2, 9, 12, 15, 18].

We propose new algorithms for both stages to handle both types of requests. We design best-fit algorithms to accept or reject scheduled requests in an online fashion to solve Stage 1. In addition, we build an online planning algorithm for Stage 2 to dispatches vehicles to on-demand ride requests in real time given the accepted scheduled requests as constraints.

2 METHODOLOGY

We consider a discrete-time, discrete-location model with single-capacity vehicles and impatient riders. We split the duration of one day into T discrete time steps. The set of time steps is $[T] = \{1, \ldots, T\}$. We also divide the whole map into a set of N different locations or regions $[N] = \{1, \ldots, N\}$.

We employ a two-stage model for processing the scheduled requests and on-demand requests. In Stage 1, the system receives a sequence of scheduled requests. A scheduled request is described by a tuple $r = (o_r, d_r, t_r, v_r)$, representing a requested ride from the origin o_r to the destination d_r that needs to start at exactly time t_r, and v_r represents the *value* the platform will receive for serving this request. We assume v_r is given to the system when the request is made, e.g., provided by the rider or an external pricing scheme. When a scheduled request arrives, the platform should either accept and assign it to a specific driver or reject it. The decision needs to be made immediately before the next request arrives. We assume that the scheduled requests for a specific day will all arrive before the day starts.

In Stage 2, the system starts to take real-time on-demand requests. An on-demand request is described by a tuple $r = (o_r, d_r, t_r, v_r)$ with the same meaning as scheduled requests. Note that these requests are received in real-time, i.e., request r will appear at time t_r. Upon receiving a set of on-demand requests R_t at time t, the platform needs to decide immediately for each request either to dispatch it to an available vehicle currently at location o_r, or to reject this request. During the processing of these on-demand requests, the platform also needs to ensure that *all* scheduled requests that it previously accepted must be served at their respective scheduled times and by their respective drivers. In Stage 2, we also allow for relocating a vehicle to location d when it is dispatched for no request.

The goal of the system is to maximize the total value of all accepted requests, including both scheduled and on-demand requests. We assume the distribution of real-time on-demand requests is known or can be estimated from historical data. However, due to the irregularity of scheduled requests, we do not make any prior assumption on the scheduled requests.

The system needs to deploy two algorithms for trip-vehicle dispatch, one for each stage. The decisions made in Stage 1 will serve as constraints in Stage 2, and the design of Stage 1 algorithm should be adaptive to the dispatch rule in Stage 2. In this work, we design novel algorithms for both stages to handle both on-demand and scheduled requests, which are built upon the Score function defined below.

Score Function. The Score function $\text{score}(t, l|r)$ represents the expected total value of trips a vehicle can serve under the optimal policy between time t and the start time of r given (i) it is located at l and is available to serve an on-demand request at time t; (iii) it is committed to serve r in the near future; (iv) it is the only vehicle in the system.

Solving Stage 1. To solve Stage 1, we design an efficient online request selection algorithm. Upon the arrival of each scheduled request r, for each vehicle that can serve r, in the algorithm, we estimate the expected value increment from this assignment with the help of the Score function and assign r to a vehicle or reject it deterministically. We also consider a variant of this algorithm by adding an additional random priority component to the value increment.

Solving Stage 2. The problem in Stage 2 can be viewed as a Markov Decision Process (MDP). A dispatch algorithm in Stage 2 implicitly represents a policy for the MDP and we aim to design an algorithm that induces an optimal or near-optimal policy. When there is only one vehicle in the system, we design a polynomial-time algorithm that induces an optimal policy with the aid of the Score function. It first calculates the Score function for all available actions and chooses the action with the highest expected value the vehicle could gain. For the multiple-vehicle case, we extend the algorithm for the single-vehicle case by sequentially dispatching available vehicles and updating a virtual demand distribution.

3 CONCLUSION AND DISCUSSION

In this paper, we focused on the problem of trip-vehicle dispatch with the presence of scheduled and on-demand request. We proposed a novel two-stage model and novel algorithms with on-demand request distribution taken into account for both stages.

We consider a multi-stage model with discretized time, discretized location as well as impatient requests, which is already a challenging problem when taking into account multi-type requests and the demand distribution. Our model and solution approaches can be further extended for problems with relaxed assumptions. For example, our work can be applied to problems with patient requests, which can be treated as duplicated requests when there is only one driver. We can also allow for the case where scheduled requests are made a shorter time in advance. Furthermore, one can integrate our study with work on last-mile routing to handle the actual road network. We defer the further investigation to future work.

REFERENCES

[1] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* 114, 3 (2017), 462–467.
[2] Javier Alonso-Mora, Alex Wallar, and Daniela Rus. 2017. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on.* IEEE, 3583–3590.
[3] Apple Inc. 2018. App Store Preview, Supershuttle. https://itunes.apple.com/us/app/supershuttle/id376771013?mt=8. (2018).
[4] Ana LC Bazzan and Camelia Chira. 2015. A Hybrid Evolutionary and Multiagent Reinforcement Learning Approach to Accelerate the Computation of Traffic

Assignment. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1723–1724.

[5] Dimitris Bertsimas, Patrick Jaillet, and Sébastien Martin. 2018. Online Vehicle Routing: The Edge of Optimization in Large-Scale Applications. (2018).

[6] Shih-Fen Cheng, Duc Thien Nguyen, and Hoong Chuin Lau. 2014. Mechanisms for arranging ride sharing and fare splitting for last-mile travel demands. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1505–1506.

[7] Brian J Coltin and Manuela Veloso. 2013. Towards ridesharing with passenger transfers. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1299–1300.

[8] Der-Horng Lee, Hao Wang, Ruey Cheu, and Siew Teo. 2004. Taxi dispatch system based on current demands and real-time traffic conditions. *Transportation Research Record: Journal of the Transportation Research Board* 1882 (2004), 193–200.

[9] Meghna Lowalekar, Pradeep Varakantham, and Patrick Jaillet. 2018. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence* 261 (2018), 71–112.

[10] Hongyao Ma, Fei Fang, and David C Parkes. 2018. Spatio-Temporal Pricing for Ridesharing Platforms. *arXiv preprint arXiv:1801.04015* (2018).

[11] Shuo Ma, Yu Zheng, and Ouri Wolfson. 2013. T-share: A large-scale dynamic taxi ridesharing service. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 410–421.

[12] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393–1402.

[13] Deepika Pathania and Kamalakar Karlapalem. 2015. Social network driven traffic decongestion using near time forecasting. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1761–1762.

[14] Kiam Tian Seow, Nam Hai Dang, and Der-Horng Lee. 2010. A collaborative multiagent taxi-dispatch system. *IEEE Transactions on Automation Science and Engineering* 7, 3 (2010), 607–616.

[15] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. 2017. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1653–1662.

[16] Chao Wang, Somchaya Liemhetcharat, and Kian Hsiang Low. 2016. Multi-agent continuous transportation with online balanced partitioning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1303–1304.

[17] Zhe Xu, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, and Jieping Ye. 2018. 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*. ACM, 905–913.

[18] Lingyu Zhang, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, and Jieping Ye. 2017. A taxi order dispatch model based on combinatorial optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2151–2159.

[19] Rick Zhang and Marco Pavone. 2016. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research* 35, 1-3 (2016), 186–203.