Learning augmented vehicle dispatching with slack times for high-capacity ride-pooling

Youngseo Kim[a*], Vindula Jayawardana[b], and Samitha Samaranayake[a]

[a] Civil and Environmental Engineering, Cornell University, Ithaca, United States
yk796@cornell.edu, samitha@cornell.edu

[b] Electrical Engineering and Computer Science, MIT, Cambridge, United States
vindula@mit.edu

[*] Corresponding author

Extended abstract submitted for presentation at the Conference in Emerging Technologies in Transportation Systems (TRC-30)
September 02-03, 2024, Crete, Greece

April 29, 2024

Keywords: High Capacity Mobility-on-Demand, Ride-Pooling, Request-Vehicle Matching Algorithm, Reinforcement Learning, Discrete Choice Modeling

1 INTRODUCTION

On-demand pooling services (e.g., UberPool, microtransit, paratransit) utilizing high-capacity vehicles have also received significant attention due to their ability to provide prompt and personalized transportation while having a high vehicle utilization rate. The development of request-vehicle matching algorithms becomes pivotal for ensuring the smooth functioning of these pooling services. We pinpoint a new decision-making challenge inherent in ride-pooling services, absent in ride-hailing counterparts. Unlike ride-hailing, ride-pooling vehicles can accommodate additional passengers en route, resulting in dynamic changes in the Estimated Time of Arrival (ETA) as depicted in fig. 1. Yet, for service reliability, operators must provide a cautious estimate of ETA when providing an offer, guaranteeing passengers' on-time arrival at their destinations. Apart from the complexities of accurate ETA estimation in congested roads, which falls outside this paper's scope, determining the optimal *slack time* for detour holds significance within the ride-pooling context. This dilemma entails a tradeoff: a tight slack time limits a vehicle's flexibility in incorporating detours for additional pickup, while a loose slack time reduces passengers' likelihood of accepting offers.

To the best of our knowledge, this is the first Mobility-on-Demand (MoD) matching algorithm capable of determining the optimal *slack time for detours*, which balances the ability to serve additional passengers en route with the loss in acceptance probabilities due to detours, and better realizes the concept of a pooled service. To this end, we propose a way to integrate discrete choice modeling and reinforcement learning, which is more comprehensive than the previous endeavor of incorporating endogenous demand. This integrated model quantifies the trade-off between the immediate opportunity of serving a passenger and the potential to serve additional passengers in the future. To calculate the anticipated future revenue, we design a Markov Decision Process (MDP) and solve it using reinforcement learning, as seen in previous literature (Xu *et al.*, 2018, Shah *et al.*, 2020). Moreover, to estimate the expected imminent revenue from matching, we utilize discrete choice modeling to gauge customers' acceptance probability of the matching offer. Our matching algorithm goes a step further than existing ones; beyond just selecting the best matching pairs, it also optimizes the slack time to ensure maximum efficiency.

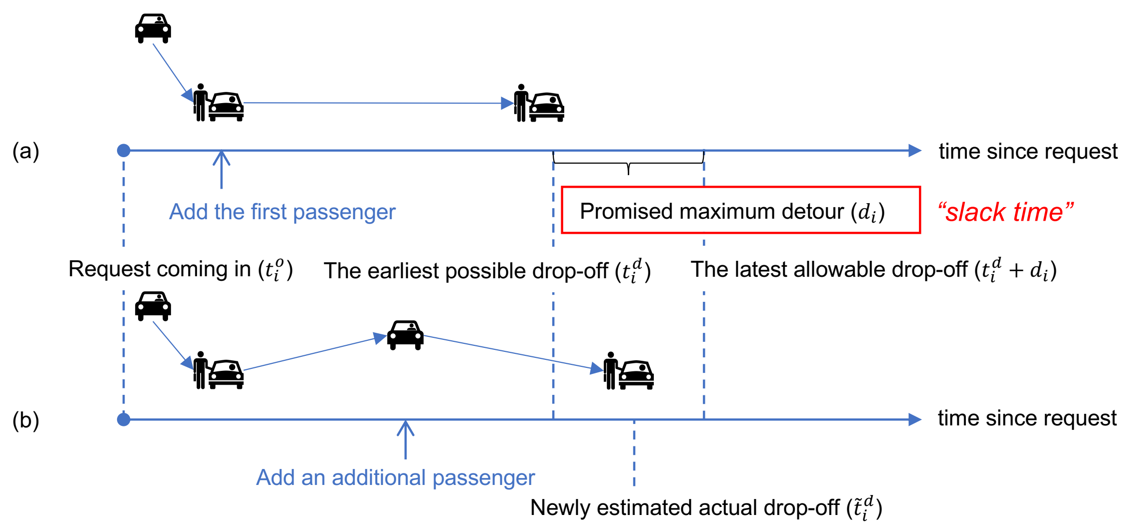


Figure 1 – *A passenger's perspective is depicted by: (a) the initial assignment of a vehicle, where the passenger receives an offer with an ETA that includes a buffer for possible detours, and (b) adjustments to the ETA when additional passengers are picked up en route, while still guaranteeing that the arrival time stays within the previously promised boundary.*

2 METHODOLOGY

Choosing the best action (i.e., assignment) for each agent (i.e., vehicle) may often violate system-level feasibility (i.e., assigning more than two vehicles to a request or vice versa). We utilize Request-Vehicle (RV)-Integer Linear Programming (ILP) to find the best assignment for centrally coordinated agents to ensure feasible matching. In this context, a centralized planner collects information on vehicle statuses and makes dispatching decisions for all vehicles.

We start by constructing RV graph as depicted in fig. 2. At each batch t, a centralized system collects an eligible request set $\hat{\mathcal{R}}_t$ and an available vehicle set $\hat{\mathcal{V}}_t$. The eligible request set $\hat{\mathcal{R}}_t$ consists of requests that have been recorded in the system up to the current batch t and have waited for assignment for a duration less than a predefined threshold. This graph consists of all possible edges between $i \in \hat{\mathcal{R}}_t$ and $j \in \hat{\mathcal{V}}_t$, following a similar approach to (Xu *et al.*, 2018, Shah *et al.*, 2020). Expanding upon the prior literature, we expand the RV graph to incorporate $|\mathcal{D}|$ copies of edges (e.g., $\mathcal{D} = \{0 \text{ min}, 15 \text{ min}\}$), where each copy corresponds an element of the set. Hence, the constructed RV graph contains $|\mathcal{R}| + |\mathcal{V}|$ nodes and $|D| \times |E|$ edges. The presence of edges $e_{ijd} \in E$ is determined by solving routing problems with time windows.

After creating the RV graph, the next step is to formulate a bipartite matching problem using ILP. The decision variable, denoted by e_{ijd}, specifies the pairs to be matched. If a vehicle j decides to serve a request i with a given slack time for detour d, then $e_{ijd} = 1$; otherwise, $e_{ijd} = 0$.

$$
\begin{aligned}
\underset{e_{ijd}}{\text{argmax}} \quad & \sum_{d \in \mathcal{D}} \sum_{i \in \hat{\mathcal{R}}_t} \sum_{j \in \hat{\mathcal{V}}_t} W(i,j,d) e_{ijd} \\
\text{s.t.} \quad & \sum_{d \in \mathcal{D}} \sum_{i \in \hat{\mathcal{R}}_t} e_{ijd} \leq 1, \forall j \in \hat{\mathcal{V}}_t \\
& \sum_{d \in \mathcal{D}} \sum_{j \in \hat{\mathcal{V}}_t} e_{ijd} \leq 1, \forall i \in \hat{\mathcal{R}}_t \\
& e_{ijd} \in \{0,1\}
\end{aligned}
\tag{1}
$$

The objective function maximizes the total value of matchings. Constraints ensure that each

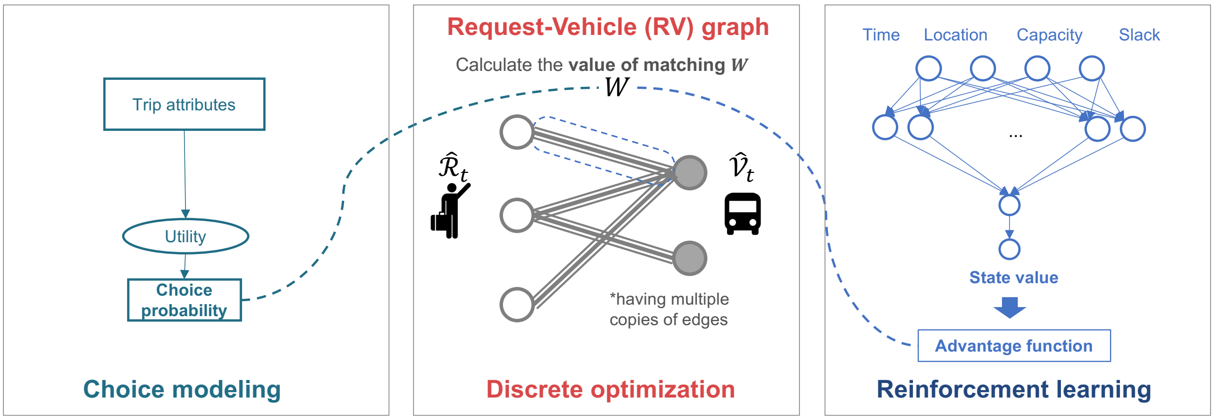


Figure 2 – *Integration of discrete choice modeling and reinforcement learning for assessing the value of matching; Utilizing immediate acceptance probabilities and future values of vehicles to estimate matching valuations*

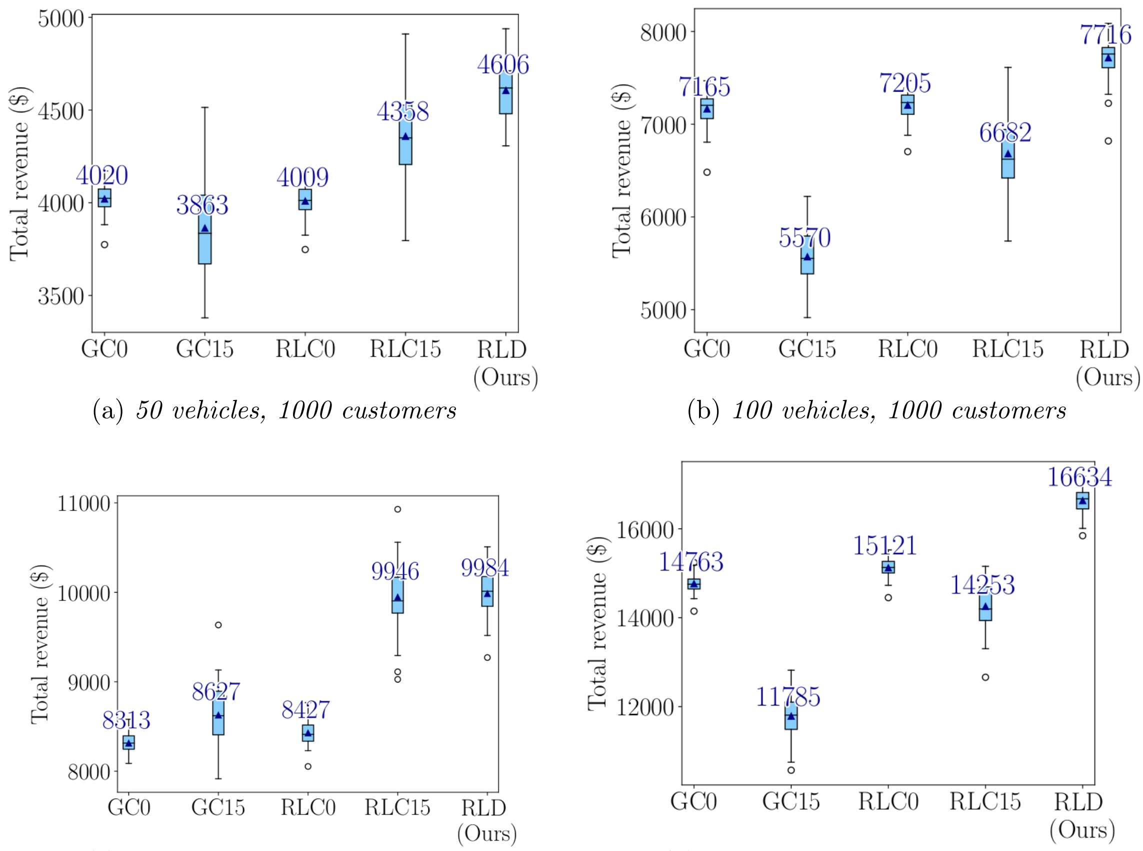
vehicle can serve at most one request per batch, and that each request can be assigned to at most one vehicle per batch. These constraints are in place to maintain system-level feasibility and to facilitate globally coordinated decisions among vehicles and customers.

3 RESULTS & DISCUSSION

To validate our approach, extensive experiments were conducted within the Manhattan network using real-world data. We acquire the taxi demand data in Manhattan, New York, for January 4-5th 7-11 am, 2016. We conduct experiments in various scenarios to reflect different supply-demand balances. First, we split the entire dataset into training and testing sets with a ratio of 7:3. We train the Neural Network (NN) over 1,000 training episodes. Each episode corresponds to a simulation day with randomly sampled demand data. The training process continues until it reaches a predefined maximum number of episodes. Once the training is complete, we conduct testing over 50 days using the trained value function approximated with NN. We trained the model with a scenario of 20 vehicles and 600 customers, and tested it on a larger scale in order to achieve desirable scalability, which is often referred to as zero-shot transfer in the Reinforcement learning (RL) community. The fleet size varies among 50, 100, and 200, and the number of customers varies between 1000 and 2000. Figure 3 displays the box plots for total revenue generated by all the baselines and our proposed algorithm across all scenarios. Note that the sources of stochasticity include fluctuations in daily demand and the choice of travelers. Extended experiments are detailed in our full paper (Kim *et al.*, 2024).

The observed trends of revenue are aligned with our expectations, showing that revenue increases with higher demand (as indicated by the number of customers) and with higher supply (as indicated by the number of vehicles). All trained models exhibit positive growth in revenue, indicating that the training process is successful. Specifically, Reinforcement Learning approach with constant slack time with 0 minute (RLC0) outperforms Greedy online matching approach with constant slack time with 0 minute (GC0), and Reinforcement Learning approach with constant slack time with 15 minutes (RLC15) surpasses Greedy online matching approach with constant slack time with 15 minutes (GC15). Most notably, our algorithm consistently surpasses all baseline models in expected revenue across all scenarios.

Our algorithm demonstrates desirable scalability, handling up to 200 vehicles and 2,000 customers effectively through zero-shot transfer. In all scenarios, Reinforcement Learning approach with *dynamic* slack time (RLD) consistently shows a positive percentage increase over the pre-trained models (GC0 and GC15). This is because RLD strikes a better balance between the



(a) *50 vehicles, 1000 customers*

(b) *100 vehicles, 1000 customers*

(c) *100 vehicles, 2000 customers*

(d) *200 vehicles, 2000 customers*

Figure 3 – *Trained with 20 vehicles and 600 customers, then transferred to large scale*

immediate probability of a passenger accepting an offer and the future potential to accommodate more passengers en route. Conversely, RLC15 fails to outperform the pre-trained models. In sub-figure (b), the revenue generated by RLC15 amounts to $6,682, which falls short of GC0's $7,165. Similarly, in sub-figure (d), RLC15's revenue of $14,253 is lower compared to GC0's $14,763. Furthermore, the size of the error bars for RLD is consistently smaller than those for RLC15, suggesting that our adaptive maximum detour strategy reduces the stochasticity of customer rejections. Hence, with its superior and consistent performance across all test scenarios, RLD holds significant real-world value, especially since its zero-shot transfer capability eliminates the need for retraining the model from scratch in response to frequent changes in demand and supply.

REFERENCES

Kim, Youngseo, Jayawardana, Vindula, & Samaranayake, Samitha. 2024. Learning-Augmented Vehicle Dispatching with Slack Times for High-Capacity Ride-Pooling. *Available at SSRN 4801437.*

Shah, Sanket, Lowalekar, Meghna, & Varakantham, Pradeep. 2020. Neural approximate dynamic programming for on-demand ride-pooling. *Pages 507–515 of: Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34.

Xu, Zhe, Li, Zhixin, Guan, Qingwen, Zhang, Dingshui, Li, Qiang, Nan, Junxiao, Liu, Chunyang, Bian, Wei, & Ye, Jieping. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. *Pages 905–913 of: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*