A Detour Planning Algorithm for Ride-Pooling Vehicles with Available Seats

Pengbo Zhu*[1], Giancarlo Ferrari-Trecate[2], and Nikolas Geroliminis[1]

[1]École Polytechnique Fédérale de Lausanne, Urban Transport Systems Laboratory, 1015 Lausanne, Switzerland, pengbo.zhu@epfl.ch, nikolas.geroliminis@epfl.ch
[2]École Polytechnique Fédérale de Lausanne, Dependable Control and Decision Group, 1015 Lausanne, Switzerland, giancarlo.ferraritrecate@epfl.ch

Short summary

Ride-pooling services, such as UberPool and Lyft Shared Saver, enable a single vehicle to serve multiple passengers within a shared trip. This work introduces a novel detour planning algorithm designed for ride-pooling vehicles with available capacity. Instead of following the shortest path from the current passenger's origin to destination, the proposed method aims to design a path, which maximizes the likelihood of matching with additional passengers en route while respecting the time constraint for the current passenger. In contrast to traditional approaches that group passengers and vehicles based on predefined time windows, our algorithm provides immediate responses to passenger requests, dynamically considers real-time passenger demand, and coordinates partially occupied vehicles with other fleets. Approximated as an integer linear programming (ILP) problem, the method is computationally efficient and practical for real-world implementation. The effectiveness of the proposed approach is verified by simulation results built on the road network of Shenzhen, China, demonstrating significant improvements in service quality.

Keywords: Shared mobility; Traffic, network, and mobility management; Vehicle routing; Path planning; Operations research applications.

1 Introduction

Urban mobility demand is rising rapidly due to the expansion of metropolitan areas and increasing population. Compared to public transportation, the ride-sourcing service offers a more convenient and privacy-preserving mobility alternative, by efficiently connecting drivers and passengers through mobile internet. Moreover, as a part of the sharing economy, it enhances vehicle utilization Ke et al. (2020) and helps reduce air pollution. However, this expanding market also brings challenges, such as developing efficient car-passenger matching Pelzer et al. (2015), optimizing routing Alonso-Mora et al. (2017), and real-time fleet operations Wallar et al. (2019); Zhu et al. (2024).

In addition to the traditional service model, where one vehicle serves one ride, ride-sourcing providers are introducing ride-pooling services, such as UberPool and Lyft Shared Saver, which allow two or more passengers to share their trips in a single vehicle Storch et al. (2021); Ke et al. (2021). It improves urban mobility accessibility by utilizing available seats, not only increasing driver profits but also helping alleviate traffic congestion. What's more, passengers benefit from discounted trip costs when sharing rides Zhang et al. (2023), making it a win-win-win situation for the urban environment, drivers, and passengers.

One critical challenge in ride pooling services is the routing problem. Many studies have been conducted to find route planning for vehicles and passengers that optimizes a prescribed objective. For example, in order to minimize the total traveling distance, a large-scale taxi-searching algorithm was proposed to quickly match candidate taxis with passenger requests in Ma et al. (2013). An online ride-sharing system was designed in Cici et al. (2015) to maximize the number of matched requests, where the passengers send their requests for a ride in advance. Besides, Asghari et al. (2016) focused on maximizing the total revenue of the system, while Hosni et al. (2014) formulated a mixed-integer programming approach to maximize driver profit and minimize passenger travel time and costs. A unified approach for route planning was proposed in Tong et al. (2018) for shared mobility, which optimized the sequence of pickups and dropoffs while managing conflicting objectives. Most of these studies, however, assume static conditions where the travel requests are fully known in advance. Dynamic route planning is essential Furuhata et al. (2013) in practical

applications, due to the time-varying traffic conditions and the stochastic nature of passenger demand. To address this, Shah et al. (2020) introduced an Approximate Dynamic Programming method to enable real-time routing that can respond effectively to time-varying demand.

Different from conventional route planning which typically suggests the shortest path for drivers, the ride-pooling scenario allows a certain detour within the threshold in exchange for reduced fares for passengers. In B. Wang et al. (2019), it was claimed that a high-quality planning path crowded with potential passengers is more valuable than a low-quality path for both drivers and passengers. A detour-planning algorithm was also examined in crowdsourcing systems by Liao & Hsu (2013), where workers were encouraged to take detours to complete additional tasks for profit without exceeding their original deadlines. However, existing research lacks fine-grained detour path planning algorithms for ride-pooling services, especially in dynamic, real-time routing contexts.

In this work, we formulate the detour planning problem as an Arc Orienteering Problem (AOP) Souffriau et al. (2011), a routing category focused on determining a sequence of edges to visit that maximizes the total collected score or profit within a specified time or resource budget. More specifically, we address path planning for partially occupied taxis, designing detailed routes from their current positions to the current passenger's destination with the goal of maximizing the likelihood of picking up additional passengers en route. At the same time, this approach also facilitates coordination with other vehicles in the urban area. Rather than batching customers and vehicles based on a predetermined time window, our algorithm enables real-time responses to passenger requests.

The remainder of this paper is organized as follows: In Section 2, the motivation for the path-planning problem for partially occupied vehicles is provided. In Section 2, we model the matching probability considering two key factors, *Attractiveness* and *Repulsiveness*, related to the influence of other vehicles. We then approximate the objective function in linear form and formulate the path-planning task as an integer linear programming problem in Section 3. The proposed algorithm is tested on a ride-sharing simulator in Section 4. Finally, we conclude the work in Section 5.

2 PROBLEM STATEMENT AND MODELING
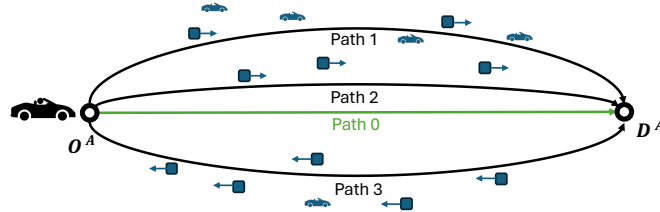
Motivation

Figure 1: Four different path choices for a partially occupied vehicle. The shortest path, labeled 'Path 0', is shown in green. Waiting passengers are indicated by square markers, with arrows pointing to their intended travel directions. The arc lengths represent the lengths of each path.

In a ride-sharing scenario, a taxi can serve multiple passengers simultaneously. Specifically, we focus on the common case where taxis have a capacity of two. It means they can serve up to two passengers with separate origins and destinations on one trip. A partially occupied vehicle has already picked up one passenger at their origin O^A and is en route to the passenger's destination D^A. There are multiple candidate paths as shown in Fig. 1. Path 0, the shortest route between O^A and D^A in terms of distance, has no waiting passengers along it. Thus, if the vehicle follows Path 0, it will complete a solo trip without picking up additional passengers. Path 3, although populated with waiting passengers, primarily has passengers traveling in the opposite direction of the current trip, from O^A to D^A, making it less viable for pooling. There are waiting passengers whose trips are compatible with the current one near Paths 1 and 2, but Path 1 has more competing vehicles for matching. Therefore, even though Path 2 involves a detour, making it longer than the shortest route, it is considered the most preferable option due to the greater likelihood of successfully

pooling another passenger along this path.

This example demonstrates that, in a ride-pooling context, the shortest path is not necessarily the optimal choice. Instead, an efficient path-planning algorithm for partially occupied vehicles is essential to improve operational efficiency and service quality. In this work, we focus on the path-planning problem of planning an efficient route from the origin to the destination of the first passenger, maximizing the likelihood of picking up an additional passenger en route.

Based on the observations above, an effective route should consider the following factors:

- **Attractiveness**: Passenger demand. By routing through areas with high concentrations of requests with similar travel directions, a vehicle increases its likelihood of matching with a second passenger.

- **Repulsiveness**: Competition from other vehicles. Coordinating these vehicles enables route planning that avoids oversupply or undersupply in urban areas.

In the following subsection, a matching probability model is developed to quantify the likelihood of a partially occupied vehicle picking up a second passenger en route to the current passenger's destination. This model is used as the foundation for our path-planning algorithm, which aims to maximize this likelihood by considering the aforementioned two factors.

Trip-specific Matching Probability Modeling

The road network is represented by a graph $G = (Q, E, \omega)$, where Q is the set of nodes (road intersections), E is the set of edges (road segments) and ω is the length of each road segment accordingly. A passenger request can be described through a pair $(O^A, D^A) \in Q \times Q$, where O^A and D^A represent the origin and destination nodes, respectively, with $O^A \neq D^A$.

Inspired by the matching function considered in Buchholz (2021) and the empirical law of pooling-matching probability found in Ke et al. (2021), we formulate the probability of a match for a partially occupied vehicle at node i, given the current trip (O^A, D^A), as follows

$$p_{node}^t(i, O^A, D^A) = 1 - \zeta exp\left(-\frac{\lambda_{Att}^t(i, O^A, D^A)}{\eta n^t(i)}\right), \tag{1}$$

where $\zeta \in (0, 1]$ is a parameter related to the matching radius d_m^t and the distributions of passengers and vehicles. The matching radius defines the area within which the vehicle searches for potential passengers. η is a region-specific positive parameter related to the difficulty of searching available vehicles, influenced by factors such as the network structure.

Some studies have been devoted to modeling the matching probability at the network level (see Yan et al. (2020); Ke et al. (2021)). However, these models are not suitable for our focus on trip-specific pool-matching probabilities. To account for pooling constraints, such as acceptable detour distances for individual passengers, we define the expected compatible passenger demand at node i for a partially occupied vehicle currently on trip (O^A, D^A) as

$$\lambda_{Att}^t(i, O^A, D^A) = \sum_{j \in Q} \lambda^t(i, j) \frac{\mathcal{L}_s(O^A, D^A) + \mathcal{L}_s(i, j)}{2\mathcal{L}_p(O^A, D^A, i, j)}, \tag{2}$$

where $\lambda^t(i, j)$ is the Poisson arrival rate of customers requesting trips from node i to node j at time t, estimated from historical data. $\mathcal{L}_s(a, b)$ is a function that returns the traveling distance if a vehicle travels directly from node a to b following the shortest path calculated by Floyd-Warshall Algorithm Floyd (1962). $\mathcal{L}_p(O^A, D^A, i, j)$ returns the minimal combined traveling distance when passenger A pools a ride with another passenger requesting a trip from i to j.

In this work, we consider two possible shared traveling sequences for ride-pooling: $O^A \rightarrow O^B \rightarrow D^A \rightarrow D^B$ (first pick-up, first drop-off) and $O^A \rightarrow O^B \rightarrow D^B \rightarrow D^A$ (first pick-up, last drop-off). Accordingly, $\mathcal{L}_p(O^A, D^A, i, j)$ is defined as:

$$\begin{aligned} \mathcal{L}_p(O^A, D^A, i, j) = \min\{&\mathcal{L}_s(O^A, i) + \mathcal{L}_s(i, j) + \mathcal{L}_s(j, D^A), \\ &\mathcal{L}_s(O^A, i) + \mathcal{L}_s(i, D^A) + \mathcal{L}_s(D^A, j)\} \end{aligned} \tag{3}$$

Compared with the traditional formulation $\lambda^t(i) = \sum_{j \in Q} \lambda^t(i, j)$ describing the Poisson arrival at node i without considering pooling constraints (as used in Buchholz (2021)), our formulation (2) includes the ratio $\frac{\mathcal{L}_s(i,j) + \mathcal{L}_s(O^A, D^A)}{2\mathcal{L}_p(O^A, D^A, i, j)} \in (0, 1]$. This ratio serves as a weighting factor that quantifies the efficiency of pooling the two trips (O^A, D^A) and (i, j). If its value is close to 1, it indicates

that the combined trip does not significantly increase the total travel distance compared to both passengers traveling separately, making the potential match more attractive; while a lower ratio suggests the pooling would involve a substantial detour. For example, when many trips from node i are heading in the opposite direction of trip (O^A, D^A), this results in a smaller ratio, decreasing the attractiveness of node i. Therefore, the entire expression in (2) quantifies the attractiveness of node i for picking up a second passenger compatible with the current trip.

In addition, the repulsiveness due to the competition from other vehicles is considered in our model Eq. (1). Specifically, $n^t(i)$ represents the number of available vehicles surrounding node i within a matching radius d_m^t at time t, calculated as:

$$n^t(i) = n_{empty}^t(i) + n_{drop}^t(i) + 0.5 n_{partial}^t(i), \tag{4}$$

where captures the effective number of available vehicles near node i, accounting for different levels of availability. Empty vehicles, represented by $n_{empty}^t(i)$, are fully available for new passenger requests. Vehicles in the process of dropping off their final passenger, $n_{drop}^t(i)$, will soon be available once they reach their destinations. Partially occupied vehicles, denoted by $n_{partial}^t(i)$, are assigned a multiplier of 0.5, as they can only accept one additional passenger, i.e., half of their capacity.

Likelihood of getting a second passenger on edge

Assuming that the start and end nodes contribute equally to a potential match on a road segment, the probability of a match event occurring on the edge $(i,j) \in E$, starting at node i and ending at j, is defined as:

$$p_{edge}^t(i, j, O^A, D^A) = (p_{node}^t(i, O^A, D^A) + p_{node}^t(j, O^A, D^A))/2. \tag{5}$$

The travel time on edge (i,j) can be estimated using measurement methods such as infrared loop detector, historical data, or real-time vehicle GPS trajectories Y. Wang et al. (2014), denoted as $\hat{T}(i,j)$. By assuming that match opportunities occur independently over time, the probability of not obtaining a match during the travel time $\hat{T}(i,j)$ is $(1 - p_{edge}^t(i, j, O^A, D^A))^{\hat{T}(i,j)}$. Therefore, the probability of a partially occupied vehicle picking up a second passenger while traveling along edge (i,j) is given by

$$\mathcal{P}_{ij}^t = 1 - (1 - p_{edge}^t(i, j, O^A, D^A))^{\hat{T}(i,j)}. \tag{6}$$

3 En route Path-planning Algorithm

Objective Function

A path is planned for a partially occupied vehicle from a given start node (origin of current passenger O^A) to an end node (destination of current passenger D^A) that maximizes the likelihood of picking up a second passenger en route. We define the decision variable x_{ij} as follows:

$$x_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \in E \text{ is part of the path,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The overall probability of not picking up any passenger along the planned route, which includes a sequence of edges, can be determined by multiplying the probability of not picking up a second passenger on each individual edge. Thus, our objective can be expressed as

$$\max_{x_{ij}} \left(J = 1 - \prod_{(i,j) \in E} \left(1 - \mathcal{P}_{ij}^t \cdot x_{ij}\right) \right) \tag{8}$$

To simplify the objective function, we apply a logarithmic transformation to reformulate the maximization problem which gives:

$$\begin{aligned} \max_{x_{ij}} J &= \min_{x_{ij}} \left(\log \left(1 - J\right) \right) \\ &= \min_{x_{ij}} \left(\log \left(\prod_{(i,j) \in E} \left(1 - \mathcal{P}_{ij}^t \cdot x_{ij}\right) \right) \right) \\ &= \min_{x_{ij}} \left(\sum_{(i,j) \in E} \log \left(1 - \mathcal{P}_{ij}^t \cdot x_{ij}\right) \right). \end{aligned} \tag{9}$$

Given small values of $\mathcal{P}_{ij}^t \cdot x_{ij}$, we can use the approximation $\log(1 - y) \approx -y$ when y is close to zero. Therefore, the objective function is approximated to a linear form as follows

$$\max_{x_{ij}} \left(\sum_{(i,j) \in E} \mathcal{P}_{ij}^t \cdot x_{ij} \right). \tag{10}$$

Integer Linear Programming (ILP) for Partially Occupied Vehicle Path-Planning

The en-route path planning problem can be formulated as the following ILP problem:

$$\max_{x_{ij}} \quad \sum_{(i,j) \in E} \mathcal{P}_{ij}^t \cdot x_{ij}, \tag{11a}$$

subject to

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in E, \tag{11b}$$

$$\sum_{j:(O^A, j) \in E} x_{O^A, j} = 1, \tag{11c}$$

$$\sum_{i:(i, D^A) \in E} x_{i, D^A} = 1, \tag{11d}$$

$$\sum_{j:(i,j) \in E} x_{ij} - \sum_{k:(k,i) \in E} x_{ki} = 0, \forall i \in N \setminus \{O^A, D^A\}, \tag{11e}$$

$$\sum_{j:(i,j) \in E} x_{ij} \leq 1, \quad \sum_{k:(k,i) \in E} x_{ki} \leq 1, \quad \forall i \in N, \tag{11f}$$

$$\sum_{(i,j) \in E} \omega_{ij} \cdot x_{ij} \leq \alpha \cdot \mathcal{L}(O^A, D^A), \quad \alpha \geq 1. \tag{11g}$$

where $\mathcal{P}_{ij}^t$ is the probability of getting a second passenger while traveling along edge (i,j) at time t, as described in Eq. (6), and ω_{ij} represents the length of edge (i,j). Constraint (11b) enforces the binary nature of the decision variables. Constraints (11c) and (11d) guarantee that the route begins at node O^A (the origin of the current passenger) and ends at node D^A (the destination of the current passenger). Constraint (11e) ensures flow continuity at all nodes except the start and end nodes, maintaining a continuous path. Constraint (11f) ensures that each node is visited at most once, preventing cycles in the path. Finally, the total length of the planned route is limited in (11g) to be no more than α times the length of the shortest path $\mathcal{L}_s(O^A, D^A)$, i.e., the maximum detour distance. Solving this ILP problem provides the optimal set of edges, which maximizes the likelihood of picking up an additional passenger along it while adhering to constraints. The route $r_{current}$ is then constructed by sequentially connecting these selected edges, forming a continuous path from the origin O^A to the destination D^A.

4 CASE STUDY

Matching scheme

A basic matching policy is implemented in the simulation, following a "first-come, first-served" approach. When a passenger request is issued, the platform searches for the nearest empty vehicles to the passenger's origin. If the closest empty vehicle can reach the passenger within a predetermined waiting time threshold, T_w, the request is assigned to this vehicle. If no empty vehicle is available and the passenger is open to ride-pooling, the platform then searches for the nearest partially occupied vehicle whose current passenger is also willing to share the ride. A match will happen if the detour distance for each passenger does not exceed α times the length of their shortest path. In this simulation, cancellations after matching are not permitted. However, if no available vehicles are found within a maximum period, T_m—the duration the request remains in the matching pool—the passenger will cancel the request.

Experimental setup

The proposed algorithm was tested using a shared-mobility simulator Beojone & Geroliminis (2021) based on the urban road network of the Luohu and Futian districts in Shenzhen, China, including 1,858 intersections and 2,013 bidirectional road segments. We consider a three-hour testing simulation, where 400 requests per hour are issued in the first and third hours, i.e., low-demand hours and 800 requests per hour are issued in the second hour, i.e., the high-demand hour. The fleet size is 100, and their initial locations are randomly generated and evenly distributed over the road network. The passenger detour distance threshold is set to $\alpha = 1.2$, allowing detours up to 20% longer than the shortest path. The passenger waiting and matching time threshold are $T_w = 5$min and $T_m = 1$min, respectively. For our proposed method, In our case study, we selected the parameter configuration that maximizes the answer rate. Uniform values of $\zeta = 1$ and $\eta = 0.001$ are applied across all nodes, since our matching scheme is operated through a centralized cloud platform. For the sake of brevity, we assume a constant traveling speed when vehicles travel along a planned route. The travel speed is given by a macroscopic fundamental diagram (MFD, see Beojone & Geroliminis (2021)), which describes the relationship between the accumulation of vehicles (i.e., the total number of vehicles) and the space-mean speed v^t. Then, the travel time for edge (i, j) at time t, given its length ω_{ij} and speed v^t, can be calculated as $\hat{T}(i, j) = \frac{\omega_{ij}}{v^t}$. The corresponding matching radius is calculated by $d_m^t = T_w \cdot v^t$ km.

Results and Analysis

To compare the service efficiency and fleet utilization, we consider the following performance metrics: **Answer rate**, which is the proportion of successfully completed orders relative to all passenger requests; **Av. waiting time**, which is the average time passengers spend waiting from the moment they issue a request until they are picked up, calculated as the total waiting time divided by the number of successfully completed orders; **No. of shared orders**, which is the total number of orders that are shared with another passenger; **Total shared distance**, measuring the cumulative distance traveled for shared rides; **Total empty distance**, which measures the distance traveled by vehicles when they are empty and cruising freely without passengers.

Table 1: Performance matrices

	Answer rate rate (%)	Av. waiting (time s)	No. of shared orders	Total shared dist (km)	Total empty dist (km)
Proposed	81.4	151.3	656	865.6	1772.8
Shortest	80.0	153.8	614	771.9	1845.1
No share	72.2	162.3	0	-	1917.0

We compare our method with two other approaches. The first is the shortest path policy, referred to as "Shortest" in Table 1. In this approach, after picking up the first passenger, a taxi travels to the passenger's destination using the shortest possible route in terms of travel distance, without considering traffic conditions. In both our proposed method and the "Shortest" policy, we assume that all taxis have a capacity of two and that all passengers are willing to share their trip. In contrast, the "No Share" method does not allow ride-pooling, meaning all passengers travel alone without sharing their trips with others.

With a capacity of two, ride-pooling methods, i.e., "Proposed" and "Shortest", serve a higher number of requests with reduced passenger waiting times, as shown in Table 1. Compared to the "Shortest" policy, our method results in a greater number of shared trips and a longer total shared distance, suggesting that it effectively plans routes to help taxis match with additional passengers for shared rides. Moreover, our proposed method shows a reduced total empty travel distance, indicating higher vehicle utilization.

Fig. 2 provides a snapshot from the simulator, illustrating how our proposed method plans the path for a taxi whose first passenger travels from node O^A to node D^A. The planned route maintains distance from empty vehicles and those dropping off passengers, avoiding overlap with the planned routes of other partially occupied vehicles. Compared to the shortest path, our planned routes pass through edges of high attractiveness, as indicated by the warmer colors of the paths.

The two planned paths shown in Fig. 2a and Fig. 2b do not exceed the 20% detour distance threshold. Notably, the planned path during the low-demand hour is longer (10.39 km) than that during the high-demand hour (10.29 km). It aligns with our expectations that, for the same
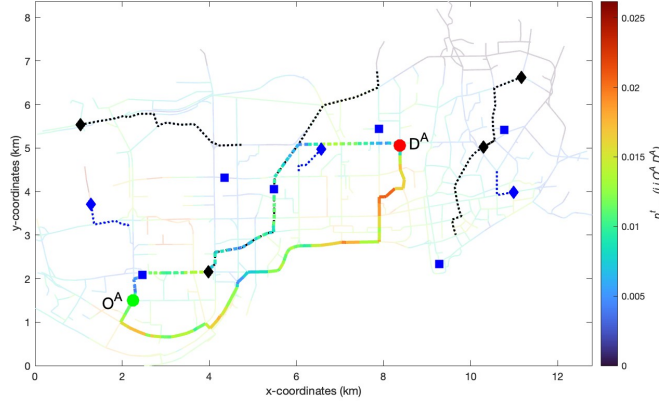
(a) Low-demand hour. Length of planned path: 10.39 km.
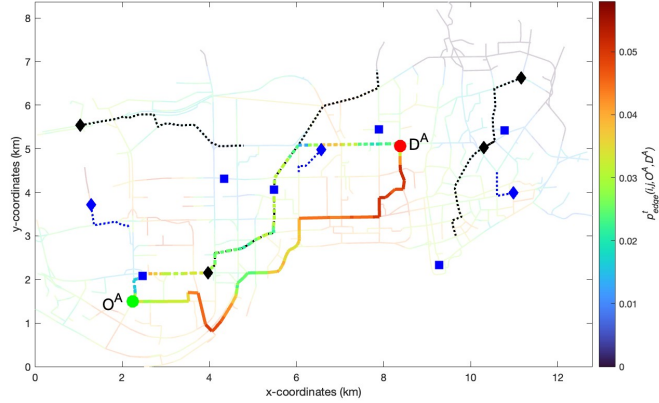
(b) High-demand hour. Length of planned path: 10.29 km.

Figure 2: Illustration of the paths for a partially occupied vehicle traveling from the origin O^A of its first passenger to the destination D^A under different demand levels: (a) low-demand hour and (b) high-demand hour. A heat map is superimposed over the road network, representing the matching probability $p^t_{edge}(i, j, O^A, D^A)$ of each edge (i, j). The shortest path between O^A and D^A is depicted as a dashed line, with edges colored according to the matching probability. Its length is 8.83 km. The routes planned by our proposed method are shown as solid colored lines. Empty vehicles are indicated by blue square-shaped markers. The current positions and planned routes of other partially occupied vehicles are represented by black diamond-shaped markers and black dotted lines. Vehicles dropping off their last passengers are shown as blue diamonds.

origin and destination, less detour distance is needed during high-demand hours due to the higher probability of finding a second passenger. In contrast, during low-demand hours, the vehicle must explore longer distances to find additional passengers. This demonstrates that our method can adapt effectively to varying demand levels.

5 CONCLUSION

This paper presents a novel en-route path planning algorithm for partially occupied vehicles in ride-pooling systems. Instead of following the shortest path, our proposed method plans an efficient route that guides vehicles through high-demand areas while coordinating with other vehicles to maximize the likelihood of picking up a second passenger to share the current ride. We verify the performance of our method using an agent-based simulation of the urban network of Shenzhen, China, demonstrating that it enhances service quality and improves vehicle utilization.

The current work estimates passenger arrival rate using historical data. Future efforts will incorporate online prediction of stochastic demand. Accurate prediction of traveling speeds on each edge is also critical to be investigated.

ACKNOWLEDGEMENTS

REFERENCES

Alonso-Mora, J., Wallar, A., & Rus, D. (2017). Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *2017 ieee/rsj international conference on intelligent robots and systems (iros)* (p. 3583-3590). doi: 10.1109/IROS.2017.8206203

Asghari, M., Deng, D., Shahabi, C., Demiryurek, U., & Li, Y. (2016). Price-aware real-time ride-sharing at scale: an auction-based approach. In *Proceedings of the 24th acm sigspatial international conference on advances in geographic information systems.* New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/2996913.2996974` doi: 10.1145/2996913.2996974

Beojone, C. V., & Geroliminis, N. (2021). On the inefficiency of ride-sourcing services towards urban congestion. *Transportation Research Part C: Emerging Technologies*, *124*, 102890. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0968090X20307907` doi: https://doi.org/10.1016/j.trc.2020.102890

Buchholz, N. (2021, 09). Spatial Equilibrium, Search Frictions, and Dynamic Efficiency in the Taxi Industry. *The Review of Economic Studies*, *89*(2), 556-591. Retrieved from `https://doi.org/10.1093/restud/rdab050` doi: 10.1093/restud/rdab050

Cici, B., Markopoulou, A., & Laoutaris, N. (2015). Designing an on-line ride-sharing system. In *Proceedings of the 23rd sigspatial international conference on advances in geographic information systems.* New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/2820783.2820850` doi: 10.1145/2820783.2820850

Floyd, R. W. (1962, June). Algorithm 97: Shortest path. *Commun. ACM*, *5*(6), 345. Retrieved from `https://doi.org/10.1145/367766.368168` doi: 10.1145/367766.368168

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., & Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, *57*, 28-46. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0191261513001483` doi: https://doi.org/10.1016/j.trb.2013.08.012

Hosni, H., Naoum-Sawaya, J., & Artail, H. (2014). The shared-taxi problem: Formulation and solution methods. *Transportation Research Part B: Methodological*, *70*, 303-318. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0191261514001659` doi: https://doi.org/10.1016/j.trb.2014.09.011

Ke, J., Yang, H., & Zheng, Z. (2020). On ride-pooling and traffic congestion. *Transportation Research Part B: Methodological*, *142*, 213-231. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0191261520304094` doi: https://doi.org/10.1016/j.trb.2020.10.003

Ke, J., Zheng, Z., Yang, H., & Ye, J. (2021). Data-driven analysis on matching probability, routing distance and detour distance in ride-pooling services. *Transportation Research Part C: Emerging Technologies*, *124*, 102922. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0968090X20308214` doi: https://doi.org/10.1016/j.trc.2020.102922

Liao, C.-C., & Hsu, C.-H. (2013). A detour planning algorithm in crowdsourcing systems for multimedia content gathering. In *Proceedings of the 5th workshop on mobile video* (p. 55–60). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/2457413.2457426` doi: 10.1145/2457413.2457426

Ma, S., Zheng, Y., & Wolfson, O. (2013). T-share: A large-scale dynamic taxi ridesharing service. In *2013 ieee 29th international conference on data engineering (icde)* (p. 410-421). doi: 10.1109/ICDE.2013.6544843

Pelzer, D., Xiao, J., Zehe, D., Lees, M. H., Knoll, A. C., & Aydt, H. (2015). A partition-based match making algorithm for dynamic ridesharing. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2587-2598. doi: 10.1109/TITS.2015.2413453

Shah, S., Lowalekar, M., & Varakantham, P. (2020). Neural approximate dynamic programming for on-demand ride-pooling. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 507-515. doi: 10.1609/aaai.v34i01.5388

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2011). The planning of cycle trips in the province of east flanders. *Omega*, *39*(2), 209-213. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0305048310000496` doi: https://doi.org/10.1016/j.omega.2010.05.001

Storch, D.-M., Timme, M., & Schröder, M. (2021, 06). Incentive-driven transition to high ride-sharing adoption. *Nature Communications*, *12*. doi: 10.1038/s41467-021-23287-6

Tong, Y., Zeng, Y., Zhou, Z., Chen, L., Ye, J., & Xu, K. (2018, July). A unified approach to route planning for shared mobility. *Proc. VLDB Endow.*, *11*(11), 1633–1646. Retrieved from `https://doi.org/10.14778/3236187.3236211` doi: 10.14778/3236187.3236211

Wallar, A., Alonso-Mora, J., & Rus, D. (2019). Optimizing vehicle distributions and fleet sizes for shared mobility-on-demand. In *2019 international conference on robotics and automation (icra)* (p. 3853-3859). doi: 10.1109/ICRA.2019.8793685

Wang, B., Zhu, R., Zhang, S., Zhao, Z., Yang, X., & Wang, G. (2019). Ppvf: A novel framework for supporting path planning over carpooling. *IEEE Access*, *7*, 10627-10643. doi: 10.1109/ACCESS.2019.2891570

Wang, Y., Zheng, Y., & Xue, Y. (2014). Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (p. 25–34). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/2623330.2623656` doi: 10.1145/2623330.2623656

Yan, C., Zhu, H., Korolko, N., & Woodard, D. (2020). Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics (NRL)*, *67*(8), 705-724. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.21872` doi: https://doi.org/10.1002/nav.21872

Zhang, W., Jacquillat, A., Wang, K., & Wang, S. (2023). Routing optimization with vehicle–customer coordination. *Management Science*, *69*(11), 6876-6897. Retrieved from `https://doi.org/10.1287/mnsc.2023.4739` doi: 10.1287/mnsc.2023.4739

Zhu, P., Ferrari-Trecate, G., & Geroliminis, N. (2024). Hierarchical control for vehicle repositioning in autonomous mobility-on-demand systems. *IEEE Transactions on Control Systems Technology*, 1-14. doi: 10.1109/TCST.2024.3448300