The Simulation of Dynamic Shared Taxi Routing Problem

Yinhu Wang, Chong Wei and Guixia Chen

ABSTRACT

In recent years, taxi sharing is becoming a popular transportation mode. Compared with the traditional operation mode, taxi sharing possesses some advantages. For example, it allows several passengers to share one taxi in the same route thus improving the utilization rate of taxi seats and reduces the taxi traveled miles. Besides, taxi sharing decreases the number of taxis needed, which efficiently relieve the traffic congestion and reduces the emission of carbon dioxide. In this paper, we consider the dynamic shared taxi routing problem (D-STRP) and develop a simulator to mimic the process of taxi sharing. To maximize the objective of the proposed problem, we define a parameterized taxi allocation policy to make a trade-off between the passenger wait and detour time and taxi route length. Furthermore, we propose the insertion algorithm to assign the passenger request to the best available at the lowest expense. The simulation results indicate that the parameterized taxi allocation policy has obvious impact on the objective of D-STRP. Through simulations, we can find the optimal taxi allocation policy in each trip scenario fast, which is of great significance for the taxi service provider.

INTRODUCTION

With the rapid growth of trip demand, the limited resource of taxi fleet cannot satisfy passenger requests for a ride, such that many people complaint against the
problem of getting a taxi, especially during the peak hours. At the same time, the low utilization of the taxi capacities increases the number of taxis needed and thus results in extra fuel consumption as well as traffic congestion.

Currently, taxi sharing has been proposed as a potential strategy to greatly mitigate the adverse impact of traditional taxi operating mode on urban traffic and environment. Taxi sharing means that a taxi can service several passengers in the same route with the constraints of time windows and taxi capacity. It can charge lower fare to each customer and thus reduce travel cost. Meanwhile, it increases taxi drivers’ revenue as it allows drivers to pick up multiple passengers in a route. From the perspective of society, taxi sharing also saves energy consumption and decreases the emission of carbon dioxide which contributes to global warming.

In this paper, we consider the dynamic shared taxi routing problem (D-STRP) in which each taxi is permitted to accept multiple passengers in the same route on the premise of meeting some constraints. In real life, routing decision is one of the most important problems[1]. Commonly, in many researches related to vehicle routing problem, a new passenger request is assigned to the best available taxi at the lowest cost of either taxi driven distance or passenger traveling time. However, in this work, we make a trade-off between the taxi driven distance and passenger traveling time during the process of taxi allocation and routing.

The main contributions of this paper can be summarized as follows: (1) based on the theories and research methods associated with vehicle routing problem, we formulate a shared taxi routing optimization problem; (2) we develop a taxi sharing simulator, which is able to handle large-scale request demand quickly; (3) we make a balanced combination of the taxi driven distance and passenger traveling time by introducing a parameter called policy parameter. Through abundant simulations where different allocation and routing policies obtained by adjusting the policy parameter are executed, we find the optimal policy to maximize the objective of D-STRP.

The remainder of this paper is organized as follows. Section 2 mainly discusses the related work in the field of vehicle routing problem. In Section 3, we give a detailed description of the shared taxi routing optimization problem and the taxi sharing simulator. Section 4 proposes the allocation policy and insertion algorithm which is applied to find the best service order of passenger requests. Subsequently, a series of experiments are described and the analysis of our results is provided in Section 5. Finally, Section 6 concludes the paper with a summary.

LITERATURE REVIEW

The shared taxi routing problem belongs to the category of the multi-vehicle DARP[2]. To our knowledge, as a generalization of vehicle routing problem, the DARP has been studied extensively in the literature. Cordeau and Laporte proposed a tabu search heuristic for the static multi-vehicle DARP, aiming to design a set of minimum cost vehicle routes capable of accommodating all
passengers[3]. Diana et al. presented a parallel regret insertion heuristic for the solution to large-scale DARP with the constraint of time windows[4]. Kirchler et al. proposed a granular tabu search algorithm for the static DARP modeled by a mixed integer linear programming formulation, with the objective of generating good solutions in a short amount of time [5]. Sophie N. Parragh et al. brought forward a variable neighborhood search-based heuristic for the static multi-vehicle DARP using three classes of neighborhoods[6]. In Parragh and Schmid, a hybrid column generation and large neighborhood search approach to the static DARP was presented and more than two-thirds computation time was saved meanwhile the solution quality was guaranteed[7]. Several variants of multi-vehicle DARP have also been studied. In Masmoudi et al., a hybrid genetic algorithm for the heterogeneous DARP (H-DARP) was proposed and the effectiveness was tested on the data of benchmark instances and newly generated larger instances[8]. Masmoudi et al. raised a new form of the DARP and proposed a mathematical programming formulation and three meta-heuristics. A large amount of experiments verified the effectiveness of the proposed algorithms for solving the problem[9].

Noting that the literatures introduced briefly above mainly focus on the static multi-vehicle DARP, in the following discussion, we turn the direction to the dynamic multi-vehicle DARP. Xiang et al. proposed a fast heuristic algorithm for a dynamic DARP under time-dependent and stochastic environments[10]. Schilde et al. studied a special problem of servicing patients or elderly people and proposed four modifications of meta-heuristic solution approaches for this problem[11]. In Berbeglia et al., a hybrid approach, combining an exact constraint programming algorithm and a tabu search heuristic, was presented for the dynamic DARP in which ride request arrives one by one in real time[12].

As stated above, the multi-vehicle DARP have received much attention over the past years and many researches on the static and dynamic version of this problem have been done and achieved great progress. However, the work closely related to taxi sharing problem is scarce. Hosni et al.[2] argued that taxi sharing plays a prominent role in improving taxi service standard and mitigating traffic congestion as well as reducing the impact on environment and then formulated the taxi sharing problem as a mixed integer program utilizing a Lagrangian decomposition approach.

PROBLEM DESCRIPTION

In this section, we mainly present a formulation of the shared taxi routing optimization problem and demonstrate the dynamic taxi sharing simulator in detail.

Dynamic shared taxi routing problem

The dynamic shared taxi routing problem (D-STRP) studied in this paper is defined as follows. Consider a virtual road network $G = \{V, A, D\}$, in which $V = \{1,$
2, \ldots, n} is the set of vertices, \( A = \{(i, j) : i, j \in V, i \neq j\} \) is the set of edges linking the nodes, and \( D = \{d_{ij} : i, j \in V, i \neq j\} \) are the distances between adjacent nodes \( i \) and \( j \).

The passenger requests occur according to a poisson process with arrival rate \( \lambda \).

Following the notations shown in Table 1, each request \( r \) contains two nodes, origin node \( r_o \) and destination node \( r_d \). In addition, a time window is also imposed on the stop node such as the time window \([e_{ro}, l_{ro}]\) associated with origin node. Each taxi \( k \) starts from the initial location \( p_k^0 \) and has a limited capacity \( Q \).

Figure 1 illustrates the modifications in taxi routes. Next we depict the taxi sharing process briefly. When a new request occurs, the information of the ride request is revealed. Based on the information, we make a matching between the taxi and the request. The detailed assignment algorithm is discussed in Section 4.

Generally, a new request is assigned to taxi \( k \) and then the service order of requests on taxi \( k \) maybe change correspondingly. The new request must be rejected in the situation where there are always some constraints triggered after trying to assign the request into all available taxis.

![Figure 1. Modification in route after a new request is assigned to a taxi.](image)

The objective of this problem is to maximize the total profit during the period of taxi operation and the profit can be formulated as follows:

\[
\text{profit} = \left[ \sum_{k=1}^{K} \sum_{r=1}^{R} (\zeta + \omega d_{ro, rd} - \tau (t_r^d - t_r^p - t_{ro, rd})) q_k^r - \delta \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \right]
\]

while satisfying the following constraints and assumptions which are referred to Liet al. and Fagnant[13].

Each passenger can only be served by at most one taxi;

The amount of passengers on board cannot exceed the taxi capacity \( Q \);

Time windows of passengers to be served must be respected;

The pick-up node must be visited prior to the drop-off node;
The wait time of a passenger at origin node cannot be beyond WT; 
A traveler’s excess travel time resulted by detour is not allowed to exceed DT.

| TABLE I. PARAMETERS AND VARIABLES FOR THE D-STRP. |
|-----------------|------------------|
| $R$             | Number of requests |
| $K$             | Number of taxis   |
| $T$             | The maximum duration of taxi service |
| $t_r$           | The time point when request $r$ occurs |
| $r_o$           | The origin of request $r$ |
| $r_d$           | The destination of request $r$ |
| $\sigma_r$      | The customer numbers of request $r$ |
| $t^p_r$         | The pick-up time point of request $r$ |
| $t^d_r$         | The drop-off time point of request $r$ |
| $[e_{ro}, l_{ro}]$ | Time windows for pick-up node of request $r$ |
| $[e_{rd}, l_{rd}]$ | Time windows for drop-off node of request $r$ |
| $p^0_k$         | The initial location of taxi $k$ |
| $p^t_k$         | The location of taxi $k$ at time point $t$ |
| $Q$             | The maximum passengers one taxi can accommodate |
| $L_k$           | The amount of passengers on taxi $k$ after visiting pick-up node or drop-off node |
| $d_{i,j}$       | The shortest distance between node $i$ and node $j$ along the shortest path |
| $t_{i,j}$       | The shortest travel time between node $i$ and node $j$ |
| $Z$             | Initial fare charging for delivering a passenger |
| $\Omega$       | Average cost per kilometer for delivering a request |
| $T$             | Penalty cost per minute for excess ride time during delivering |
| $q^r_k$         | Binary variables equal to 1 if taxi $k$ serve request $r$ |
| $\lambda_{i,j}^k$ | Binary variables equal to 1 if taxi $k$ goes directly from node $i$ to node $j$ |
| $\Delta$        | Average oil cost per kilometer |
| $WT$            | Maximum passenger wait time at origin node |
| $DT$            | Maximum passenger detour time during ride |

**A Taxi Sharing simulator**

To simulate the taxi sharing process, a shared-taxi simulator is encoded in Microsoft Visual C++. Next, we introduce them in detail below.
The trip generation module produces taxi requests arriving randomly according to a poisson process. A taxi request contains occurrence time, origin, destination, the shortest travel distance and the number of passengers. The origin node and destination node are selected uniformly from the node set $V$.

The main function of taxi location module is to update the position of each taxi. Let $p_k^t = (i, j, d)$ denotes the location of taxi $k$ at time $t$. An illustration of location updating process is shown in Figure 2. The box drawn by dashed line represents the old taxi location $p_k^{old} = (1,2, d_1)$ and the box drawn by solid line indicates the newest location of taxi when a new request occurs. During the time interval between the last request and the new request, the distance traveled of the taxi is $d_3$. Based on the traveling route and vehicle miles traveled, the taxi location can be updated to $p_k^{new} = (2,3, d_2)$.

The request assignment module is deciding whether the new request can be accepted and the assignment of the accepted request. Considering a new request reappears, this module tries to assign the request to all taxis sequentially. It is obvious that the route of taxi $k$ changes as the request $r$ is insert into taxi $k$. Therefore, the total cost to serve all in-vehicle passengers and the prospective request also alters. Based on the assignment algorithm proposed in Section 4, the module determines the best available taxi and the optimum route to serve their quest with the minimum cost.

The final module is state update module. Taxis’ states change dynamically. When a taxi reaches a stop node, the status of the taxi has to be updated, i.e., the amount of passengers in vehicle. In addition, the origin node and destination node are inserted into the route of the appointed taxi, therefore the route of that taxi must be modified accordingly.

**TAXI ALLOCATION POLICY AND TAXI ROUTING**

In the taxi sharing system, when a new request appears, the dispatcher must make a decision about whether the request is accepted or not. If the request is accepted, we should choose the best attainable taxi to serve the passenger at the least cost. This process mainly involves taxi allocation policy and taxi routing. Next, we discuss these two aspects in detail.

![Figure 2. The process of taxi location update.](image-url)
Taxi Allocation Policy

As stated before, we make a trade-off by introducing a parameter $\mu (0 \leq \mu \leq 1)$ between the taxi route length and passenger wait and detour time. For a given route $\Omega$ of taxi $k$, let $d(\Omega_{old})$ denote the route length in time after finishing all current in-vehicle passengers and $t(\Omega_{old})$ be the total wait and detour time of all in-vehicle passengers. Assuming that a new request is assigned to taxi $k$, the new route length and total wait and detour time are denoted as $d(\Omega_{new})$ and $t(\Omega_{new})$ respectively. We consider a parameterized allocation policy $\psi(\mu)$ in equation (2):

$$\psi(\mu) = \mu(t(\Omega_{new}) - t(\Omega_{old})) + (1 - \mu)(d(\Omega_{new}) - d(\Omega_{old}))$$ (2)

When $\mu$ is close to 0, it means that allocation policy mainly take the route length into consideration. On the contrary, the value of $\mu$ approximate to 1 indicates the allocation policy is sensitive to the increase in the passenger wait and detour time.

Taxi Routing

In this work, we propose the insertion algorithm to assign a new passenger. We denote the incremental cost function of a taxi route as $f(\cdot)$. Considering the parameterized taxi allocation policy, the cost function of inserting a new passenger into the taxi $k$ is defined in equation (3) as follows:

$$\Delta f(\Omega_{new}, \Omega_{old}, \mu) = \mu f(t(\Omega_{new}) - t(\Omega_{old})) + (1 - \mu)f(d(\Omega_{new}) - d(\Omega_{old}))$$ (3)

Where the function represents the incremental cost of taxi $k$ containing the new request. The function consists of two parts: (i) the incremental wait and detour time; (ii) the incremental route length.

In order to maximize the objective, the insertion algorithm aims to minimize the incremental cost function. When a new request appears, the simulator visits all the taxi and obtains their current route. Utilizing the insertion algorithm, the origin node and destination node are inserted into all possible position of each existing route. Each insertion can return an incremental cost using the equation (3). After testing all the possible position, we obtain the best available taxi to accept the new request and update the taxi routing to serve the passenger with the minimum cost.

SIMULATION EXPERIMENTS

To evaluate the performance of the proposed parameterized allocation policy in the optimization of shared taxi routing problem, we set several scenarios with different request arrival rate and run several experiments. In this section, we mainly discuss simulation settings and simulation results.
Simulation Setting

We establish a virtual road network including 2500 nodes and 9800 links. The number of taxis is 60 and these taxis can accommodate no more than four passengers at the same time. As for the demand for taxis, we set four different demand levels and the request arrival rates are 0.06 requests/s, 0.09 requests/s, 0.12 requests/s and 0.15 requests/s respectively. The passenger requests arrive according to a poisson process. The maximum wait time is limited to 10 minutes and the detour time is not allowed to exceed 15% of the travel time along the shortest path. The total simulation time is set to three hours.

Simulation Results

We establish four different demand scenarios with diverse arrival rates and run simulations for these scenarios. It is worth noticing that we define $\psi(1)$ as the benchmark policy, namely, $\psi(1)$ just considers the wait and detour time of passengers when allocating taxis to serve passengers.

Figure 3 indicates that executing different policies has diverse impact on the total profit. Besides, with the change in the request arrival rate, the best taxi allocation policy $\psi(\mu^*)$ is not fixed. Table II shows the best allocation policy in each scenario. In scenario 1, the request arrival rate is low and the resource of the taxi fleet is able to meet the ride demand, such that improving the service level by considering the passenger wait and detour time more is beneficial to enhance the total profit. With the increase in the arrival rate of passenger, the service level has to be limited because that taxis try to accept more passengers and thus increase the wait and detour time. However, when the ride demand exceeds the taxi supply too much, many requests are rejected, in order to maximize the profit, it is necessary to pay more attention to the service level again.

The Figure 4 illustrates the passenger wait and detour time based on the optimal allocation policy and the benchmark policy ($\mu = 1$) in the case where the request arrival rate $\lambda$ is equal to 0.12 requests/s. Shown in the figure, the average wait and detour time resulted by optimal allocation policy is shorter than the average wait and detour time based on the benchmark policy.

<table>
<thead>
<tr>
<th>Allocation policy</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit with BP(RMB)</td>
<td>19377.03</td>
<td>23387.22</td>
<td>25711.06</td>
<td>28197.78</td>
</tr>
<tr>
<td>Profit with OP(RMB)</td>
<td>20372.82</td>
<td>25544.99</td>
<td>27529.33</td>
<td>29594.70</td>
</tr>
<tr>
<td>Increased rate(%)</td>
<td>5.13</td>
<td>9.22</td>
<td>7.07</td>
<td>5.44</td>
</tr>
<tr>
<td>The value of $\mu^*$</td>
<td>0.7</td>
<td>0.6</td>
<td>0.4</td>
<td>0.8</td>
</tr>
</tbody>
</table>
CONCLUSIONS AND FUTURE RESEARCH

In this paper, we study the dynamic shared taxi routing problem where we aims to maximize the total profit of taxi service provider. We define a parameterized policy where a parameter $\mu$ is introduced and makes a balanced combination between the taxi route length and the passenger wait and detour time. Utilizing the taxi sharing simulator, we run several simulation experiments. From the simulation results, we find that the total profit varies with the change in the value of $\mu$. Besides, the best allocation policy $\psi(\mu^*)$ also changes with the
alteration of request arrival rate. This observation is of great significance for the taxi sharing. Based on the current research conclusions, we will devote our mind to the dynamic decision of taxi allocation according to the environment state. In addition, we will focus on the artificial intelligence algorithm for the vehicle routing problems.

REFERENCES