공학석사 학위논문

Order Dispatching in Ride-Pooling with Walking Points Search

대안 승하차지점 탐색을 통한 승차공유 서비스의 배차 알고리즘 개발

2023년 2월

서울대학교 대학원
공과대학 건설환경공학부

박 성 연

Order Dispatching in Ride-Pooling with Walking Points Search

지도 교수 김 동 규

이 논문을 공학석사 학위논문으로 제출함

2022년 12월

서울대학교 대학원

공과대학 건설환경공학부

박 성 연

박성연의 공학석사 학위논문을 인준함

2023년 1월

위 원 장 ______ 이 청 원 ______ (인)

부위원장 ______ 김 동 규 ______ (인)

위 원 ______ 고 승 영 ______ (인)

Abstract

Ride−pooling has significantly enhanced the system efficiency in current on−demand ride−sharing services. However, as the numbers of on−board passengers increase, more detours inevitably occur since it provides door−to−door service for everyone. To solve this problem, we focus on rider−participating dispatch by searching walking points, equivalent to alternative pick−up points from origins and alternative drop−off points from destinations. Based on the existing framework for large−scale ride−pooling, we develop our walking point search algorithm, which finds cost−minimizing alternatives. In addition, our approach enables the model to reflect the sensitivity of riders to given walking points by introducing the probability of riders' acceptance. We conduct a simulation with the Yellow Cap Taxi dataset in New York City to validate and compare with the base model, which does not include walking. The results show an increase from 69.56% to 77.84% in the service rate, an improvement of 18.2% in delay time, and 8.6% in in−vehicle time. With the increased service rate, the average travel times of vehicles are reduced by 1.5%, allowing drivers to spend more time rebalancing. Furthermore, we show that the effect of walking is maximized in high−demand areas during peak hours. This study demonstrates that walking can substantially enhance operational efficiency, mitigating the supply−demand imbalance with limited fleets. The proposed model can also be utilized in optimizing the meeting points for various high−capacity vehicles, such as on−demand shuttles.

Keyword : Ride−sharing, Ride−pooling, On−demand mobility, Walking to ride, Meeting points

Student Number : 2021−26822

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

The sharing economy has dramatically changed mobility industries, led by various ride-sharing companies, such as Uber and Lyft. The entire shared-mobility market accounted for $130 to $140 billion in global consumer spending in 2019 (pre-pandemic). Ride-sharing services accounted for the largest share, in the range of $120 billion to $130 billion, and they had massive growth from 2016 to 2019 during their number of trips almost tripled (Heineke et al., 2021).

Ride-sharing services, such as Uber Pool and Lyft Line, have expanded their coverage to ride-pooling to offer more rides with limited fleets and reduced fares. In ride-pooling, multiple on-demand requests are served by a single vehicle. Online matching of this system was done by solving assignment problems, constructing multiple trip sets in a batch, and assigning them to vehicles. These separated modules enable real-time dispatching in large-scale ride-sharing services (Alonso et al., 2017; Simonetto et al., 2019; Shah et al., 2020). In addition to these real-time serving algorithms, dynamic strategies have been developed to enhance the efficiency of the service. For instance, Uber uses a dynamic waiting mechanism to thicken the pool contributing to assigning the nearest passengers to the same trip. Dynamic waiting requires passengers to wait for a certain duration by dynamically adjusting the waiting time to join with other passengers with similar origins and destinations. This mechanism thickens the pool of eligible requests for matching, resulting in a higher matching rate (Yan et al., 2020).

When a region's demand is relatively higher than the demands of other regions, riders sometimes have difficulty in finding empty vehicles. However, some people implicitly know where to search for available vehicles if they are willing to walk some distance. Idle vehicles that would not be included in the radius of the previous searches can be found by walking to certain points. The problem is that most potential riders do not know where they should walk to attain rides. Therefore, suggesting optimized walking points can be informative for these riders, especially in high-demand hours and regions. It also benefits operators since it improves the utility rates of their fleets by shortening routes with less detour while serving more riders.

In current ride-pooling algorithms, when a vehicle serves more riders in a trip, the computational complexity grows exponentially, and more in-car delays occur. Therefore, it is essential to find ways to shorten travel detours to maintain the quality of service and improve the system's efficiency. In this aspect, optimizing the meeting points of vehicles and riders can be an efficient solution since it not only suggests locations where riders can walk but also minimizes the detours they will incur. Figure 1 shows an example of how walking reduces detours, thereby serving additional riders. To this end, we propose a ride-pooling framework with a walking points search algorithm that provides cost-minimizing pick-up and drop-off locations. We show that this algorithm can significantly reduce riders' delay times and achieve higher service rates.

To the best of our knowledge, only one paper addressed the optimization of walking points, and it only resulted in a 2% to 4% enhancement in request rejection rates (Fielbaum et al., 2021). In addition, few studies have focused on the effects of allowing the walking option or the probability of accepting suggested walking times (Stiglic et al., 2015). Since there is a trade−off between the system efficiency and riders' convenience, the objective function in the assignment problem should include riders' acceptance probability to reflect whether the suggested matching is likely to be accepted.

In this paper, we propose a dispatching algorithm with walking points search in ride−pooling. We incorporate rider−side flexibility to improve the service' s overall quality, unlike most existing literature on online dispatching focused on enhancing computational efficiency or the rebalancing method. Furthermore, by including the riders' utility model regarding walking time in the objective function, our framework presents a more practical assessment of willingness−to−ride of riders on optimized pick−up and drop−off points. We achieved an enhancement in the service rate that was more than double the enhancements in the previous literature on optimizing walking points. Our work can contribute to the current ride−sharing market and other types of high−capacity ride−sharing services where dynamic meeting points are needed.

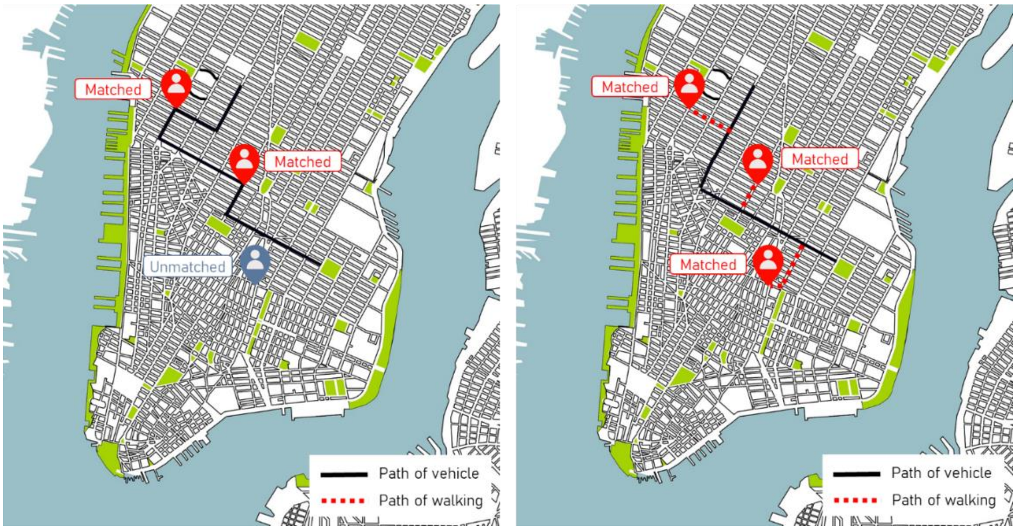

Figure 1. An illustration of ride−pooling with and without walking points

Chapter 2. Literature Review

Matching and assignment in ride-sharing, including both single-rider and multi-rider services, are based on various algorithms and objective functions. Since matching riders and drivers is constructed as a bipartite structure, mixed integer linear optimization and combinatorial optimization often are used (Hosni et al., 2014; Qian et al., 2017). In addition, dynamic programming also is applied to model the dynamics of complex systems (Yu et al., 2020; Duan et al., 2019). The integrated framework of combinatorial optimization with learning models refers to predicted information for non-myopic solutions (Shah et al., 2020; Zhang et al., 2019; Xu et al., 2018). Deep reinforcement learning also is used to represent interaction with the environment and to increase adaptability to rapidly changing environments (Wang et al., 2018; Al-Abbasi et al., 2019). Previous studies have used diverse objective functions for order dispatching in single-rider services, such as minimizing pick-up time, minimizing passenger request waiting time, maximizing quality of service, and maximizing total profit (Lee et al., 2004; Wong et al., 2006; Seow et al., 2010; Bertsimas et al., 2019). Unlike single-rider services, order dispatching in multi-rider services (ride-pooling) considers travel detours that occur by other riders served together. Studies on ride-pooling aim to minimize an increase in travel distance, total mileage driven with limited detours, passenger travel time, detour cost, and total travel miles (Simonetto et al., 2019; Qian et al., 2017; Ma et al., 2015; Pelzer et al., 2015; Jung et al., 2016).

Specifically, in ride-pooling, pick-up and drop-off sequences should be considered to serve each of the riders while satisfying their constraints, and this makes the assignment of ride-pooling combinatorially harder (Yan et al., 2020). To deal with the large-scale ride-sharing services with higher capacities in real time, Alonso et al. proposed a highly-scalable, anytime optimal algorithm (Alonso et al., 2017). They built the framework by constructing feasible pairs of trips and vehicles from an existing concept of the shareability graph (Santi et al., 2014). Simonetto et al. enhanced computational efficiency based on the framework of Alonso et al. (Simonetto et al., 2019). They proposed a distributable optimized framework pointing out that current centralized systems are unsuitable for multiple ride-sharing operators in the market. This algorithm reduced computational time with the single-request assignment four times more. Liu et al. improved the computational efficiency of the framework in Alonso et al. (Liu et al., 2022). They proposed search space pruning techniques to reduce the computation time and input/output reduction techniques for parallelization, allocating requests with similar candidate vehicle sets to the same computing unit. Since these studies only consider current time steps, Shah et al. pointed out that optimizing the fixed objective function, which ignores its effects on future time steps, results in myopic solutions (Shah et al., 2020). They provided approximate dynamic programming updating value from the integer linear programming (ILP) based assignment. As a result, their proposed method served more requests during peak times and improved the state of the art by 16%.

Existing works on ride-sharing dispatching have been conducted given fixed locations of riders. However, a few works have addressed meeting points where riders are picked up. Stiglic et al.

investigated the benefits of meeting points where multiple riders can be picked−up or dropped−off within their acceptable distances (Stiglic et al., 2015). They achieved a significant increase in possible matches without any increase in the number of stops for vehicles. However, four randomly generated meeting points were used in each travel analysis zones, which limits the flexibility of riders and hinders the identification of the optimal meeting points. Fielbaum et al. designed frameworks for requesting users to walk towards/from actual origins and destinations (Fielbaum et al., 2021). By applying the proposed method to the real dataset, they showed that walking improved the system meaningfully, especially the number of rejections and vehicle−hours traveled. However, they allowed only a small proportion of walking by applying a fixed walking−time penalty, which resulted in only up to 4%p reduction in rejection rates in one−hour simulations with approximately 10,000 requests in Manhattan. Aivodji et al. proposed a decentralized architecture for computing privacy−preserving meeting points in ride−sharing (Aivodji et al., 2016). Their experiment results showed the feasibility of both the privacy of location information and utility levels.

While adjusting meeting points by walking in ride−sharing rarely has been addressed, walking in public transit has been addressed in numerous studies. Although there has been very little research that has explored riders' willingness to walk as a part of ride−sharing, the research on public transit passengers would provide a reasonable criterion. Studies on passengers' walking behavior to reach public transit can be utilized in ride−sharing with free−floating walking points in terms of modeling the relationship between walking distance and percentage of passengers, determining walking distance threshold, and reflecting modal difference.

Zhao et al. found that transit usage is reduced exponentially with walking distance to transit stops, so that the relationship could be modeled by a decay function (Zhao et al., 2003). In addition, walking distances greater than 800 m did not affect increasing accessibility in southeast Florida. Applying the walking time decay function in Zhao et al., Chia et al. revealed the variation in walking time among bus riders with different socioeconomic characteristics (Chia et al., 2016). The household travel survey in Brisbane, Australia showed that transit use drops drastically at 5 min and 10 min, 375 m and 750 m, with a mean walking speed of 75 m/min. Walking access also varies with travel mode. Ker et al. showed that, in the case of rail stations, passengers walk further than the conventionally-assumed 5 min and 10 min limits (Ian et al., 1998). Likewise, Weinstein et al. revealed that passengers walk more than 805 m, approximately 10 min, to railway stations, a much longer distance than they will walk to bus stops (Agrawal et al., 2008). Similarly, Daniels et al. found that passengers walk farther to access train stations than bus stations, suggesting that the difference in the supply of travel modes contributes to variability in walking distances (Daniels et al., 2013).

Although a few studies have taken into account riders' walking in the ride-sharing dispatching problem, some dynamic pricing and matching algorithms consider riders' sensitivity to price and waiting time or their preference on trip attributes. For example, Yan et al. jointly optimized dynamic pricing and dynamic waiting to mitigate price variability while considering the probability of riders' acceptance (Yan et al., 2020). hey estimated riders' request rate function with respect to surge multiplier and waiting time by calibrating parameters using UberX data. By maximizing welfare defined by these riders' request functions, they showed that the joint optimization of price and waiting time prevents high/volatile prices or waiting times. Prior to Yan et al., Castillo et al. demonstrated that dynamic pricing could prevent from falling into Wild Goose Chase zones, depleting idle vehicles on the street in high-demand areas or times, thus significantly disrupting service functioning (Castillo et al., 2017). This study defined gross utility in terms of riders' willingness to pay and wait when determining platform revenue to validate their theory. On the other hand, Qiu et al. specified a choice model with respect to the type of service, travel mode, travel time, and travel cost (Qiu et al., 2018). For a practical optimal pricing strategy, they used this behavior model reflecting riders' sensitivity towards price surges and reductions and showed superior performance in profit. As stated so far, many strategic frameworks for pricing or dispatching indicate that the final decision is directly affected by riders' preference or their likelihood to accept orders. Thus, riders' reluctance to walk, which might be higher than waiting still at a fixed point, should be considered to evaluate the model in realistic settings.

Chapter 3. Methods

3.1. Preliminaries

In every batch, we consider a fixed number of vehicles, m, with capacity ν of vehicle set $V = \{v_1, v_2, \dots, v_m\}$ and n newly received requests set $R = \{r_1, r_2, \dots, r_n\}$. Each vehicle $v \in V$ consists of the current location updated at every iteration, assigned schedules, and on-board passengers. Each request $r \in R$ consists of origin o_r, destination d_r, maximum pick-up time δ_r^p, and maximum drop-off time δ_r^d determined by parameters of maximum waiting time, Ω, and maximum detour rate, Δ, respectively. Trips generated for vehicle v are denoted as T_v, which is a set of $T_{v,\Gamma}^k$ that contains a set of k new requests, Γ.

3.2. General Formulation of the Ride-Pooling Problem

For the general formulation, we chose the most extensively used framework for online dispatching in Alonso et al. and its implementation Li et al. (Alonso et al., 2017; Li et al., 2021). It separates the process of finding feasible schedules containing multiple trips from assigning the schedules generated for vehicles, which maximizes the objective function. A set of requests that can be served by a vehicle satisfying δ_r^p and δ_r^d denotesa feasible or possible schedule. In each iteration, for each vehicle v, we generated possible drop-off schedules for on-board passengers in v. Then, for newly received requests, including unassigned requests in the previous batch, we insert every $\{o_r, d_r\}$ into every location of its drop-off schedules checking feasibility. These generated feasible schedules of size 1 (one new request inserted) are saved as a set of trip T_v^1 with every schedule' s total travel durations, δ_t. A schedule of minimum duration is denoted as the best feasible schedule. Based on T_v^1, we generated T_v^2 containing 2 new requests. For every 2 size combinations of new requests, for example $\{r_1, r_2\}$, we insert $\{o_{r_2}, d_{r_2}\}$ into $T_{v,\{r_1\}}^1$ to check its feasibility. Then, these feasible sets of trips constitute $T_{v,\{r_1,r_2\}}^2$. As was done in the previous step, we build T_v^k based on T_v^{k-1}. However, if $T_{v,\Gamma\backslash r}^{k-1}$ does not exists for a request r, we do not build $T_{v,\Gamma}^k$ since the base schedule is already infeasible.

Each $T_{v,\Gamma}^k$ contains the best feasible trips for every set of requests, and they are considered when assigning trips to vehicles. Optimal assignment is obtained by solving the ILP problem, which is used to determine whether to assign a trip to a vehicle. The term x_{tv} is a binary variable, and it is 1 if a trip t is assigned to a vehicle v. The ILP formulation minimizes $c_{tv}x_{tv}$ following constraints, and each vehicle must serve less than one trip, and each request must be served by less than one vehicle. The problem formulation is described in Equation (1a) – (1d).

$$\text{Min} \quad \sum_{t\in T}\sum_{v\in V} c_{tv}x_{tv} \tag{1a}$$

s.t.

$$\sum_{t\in T} x_{tv} \leq 1, \qquad \forall v \in V \tag{1b}$$

$$\sum_{t\in T(r)} x_{tv} \leq 1, \qquad \forall r \in R \tag{1c}$$

$$x_{tv} \in \{0, 1\}, \qquad \forall t \in T, v \in V \tag{1d}$$

Idle vehicles are matched to unassigned requests to relocate vehicles to high-demand areas efficiently. Only one vehicle is assigned to one unassigned request to prevent assigning multiple vehicles to the same request. Vehicles not assigned any schedules, including rebalancing schedules and not traveling to pick-up requests, are denoted as empty vehicles.

3.3. Walking Points Search Algorithm

To determine walking points, which are equivalent to pick-up points from actual origins and drop-off points from actual destinations, we modified the module to find feasible schedules. Initially, we define neighborhood nodes of o_r and d_r as o_r' and d_r', which are nodes within maximum walking radius Λ from o_r and d_r. With this precomputed set of neighborhood nodes, we iterate the process of finding feasible schedules replacing o_r and d_r into $o_r' \in O_r'$ and $d_r' \in D_r'$. As described in Table 1, we replace o_r into o_r' only when sequence i where o_r is picked is feasible in base schedule. If this is satisfied, o_r' is inserted at i instead of o_r. To reduce the computational complexity, only when total travel duration δ_t is minimum, (o_r', d_r) is added to a set of possible pick-up and drop-off pairs of request r when assigned to vehicle v within trip t, $P_{r,v}^t$. Feasible trip t with replaced origin also is added to the feasible trip table of size 1, $T_{v,\Gamma}^1$. Also, only when this is the case, d_r' is searched, and (o_r', d_r') is added to $P_{r,v}^t$ when δ_t is at its minimum value. Thus, d_r' is only searched when the minimum δ_t of o_r' is discovered. After finishing this iteration for insertion location of r, (i, j), we conduct the same iteration, fixing the insertion location and replacing o_r and d_r.

When generating $k > 1$ size of trips, we utilize $P_{r,v}^{t}$ to make combinations of multiple requests with their previously discovered pick-up and drop-off points. We insert these alternative pairs $(o'_r, d'_r) \in P_{r,v}^{t}$ while iterating possible insertion locations (i,j). And same as building $k = 1$ size of trip, only the trips with the minimum δ_t are added to $T_{v,\Gamma}^{k}$. Algorithm for generating k > 1 feasible schedules in detail is described in Table 2

TABLE 1. Algorithm for generation of k = 1 feasible schedules with walking points

Algorithm 1: Generation of $k = 1$ feasible schedules with walking points
$T_{v,R}^{1} = \emptyset\ \forall v \in V$; O'_r = *get_neighborhood_nodes* (o_r); D'_r = *get_neighborhood_nodes* (d_r);
for each vehicle $v \in V$ **do**
for request in $r \in R$ **do**
$\Gamma = \{r\}$; $P_{r,v} = \{(o_r, d_r)\}$
[Insert request's pick-up & drop-off points]
for (i,j) ∈ possible_insertion_locations **do**
if new_schedule exists **then**
[Check feasibility of new pick-up points]
for $o'_r \in O'_r$ **do**
if new_pickup_schedule exists **then**
if new_pickup_cost < min_cost **then**
Add (o'_r, d_r) to $P_{r,v}$
Add new_pickup_schedule to $T_{v,\Gamma}^{1}$
[Check feasibility of new drop-off points]
for $d'_r \in D'_r$ **do**
if new_dropoff_schedule **then**
if new_dropoff_cost < min_cost **then**
Add (o'_r, d'_r) to $P_{r,v}$
Add new_dropoff_schedule to $T_{v,\Gamma}^{1}$

TABLE 2. Algorithm for generation of k > 1 feasible schedules with walking points

Algorithm 2: Generation of $k > 1$ feasible schedules with walking points

```
for v ∈ V do
   while k > 1 do
      for t ∈ T_{v,Γ}^{k} do
         for r ∈ R/Γ do
            Add r to Γ
            if T_{v,Γ/r}^{k} is empty break
            for (o'_r, d'_r) ∈ P_{r,v} do
               [Insert request's pick-up & drop-off points]
               for (i,j) ∈ possible_insertion_locations do
                  if new_schedule exists then
                     if new_schedule_cost < min_cost then
                        Add new_ schedule to T_{v,Γ}^{k+1}
                        k = k + 1
```

3.4. Rider Acceptance Probability with Walking Distance

In this section, we define the objective function that reflects the decreasing probability of rider acceptance with increasing walking time. It takes into account the stochastic behavior of riders that affect the total expected revenue. By including the reduction in the expected revenue depending on riders' acceptance probability, we analyze the realistic effects of the walking points suggestion. We assume that the riders' acceptance probability for the suggested walking points is determined by walking time.

We use the existing rider utility model with respect to waiting time and surge multiplier of price from Yan et al. (Yan et al., 2020), which is estimated based on UberX data that contain whether the rider accepted trips when providing the price and the waiting time of a trip. In this estimated model, the rider's utility of a trip is defined as $u(p,w)+\epsilon$, where ϵ is a Gumbel distributed random variable, as shown in **Equation (2a)**. Then, the rider's acceptance probability $\tau(w)$ is assumed to follow a logit form, as shown in **Equation (2b)**. The coefficients we use for η and β were 1.643 and −0.6693, respectively, as estimated in Yan et al. (Yan et al., 2020). However, since riders are expected to be more reluctant to walk than to wait, we use a higher value for δ than was used in Yan et al. (Yan et al., 2020). The value is set to −0.199, which was estimated empirically through the distribution of walking time of bus riders in Chia et al. (Chia et al., 2016). The surge multiplier is set to the default value of 1.0 since we do not consider the sensitivity to price. Considering that the revealed fixed hour-of-day effects $\{\kappa_i\}$ were between 0.05 and 0.2 in the existing work (Yan et al., 2020), we scale κ_i to be proportional to the demand within range from 0.05 to 0.2. **Figure 2** shows the acceptance probability with walking time in minutes when the hour of day effect κ_i is minimum (κ_4= 0.061) during 04:00–05:00 and maximum (κ_{19} = 0.211) during 19:00–20:00. The acceptance probability starts near 0.7 when the walking time is 0, and it decreases to 0.3 when the walking time is 10 minutes. In these two cases, the average difference of acceptance probability is 3.4%p.

$$u(p,w) = \eta + \beta \cdot p + \delta \cdot w + \sum_{i=1}^{23} \kappa_i \cdot I(h = i) \tag{2a}$$

$$\tau(w) = \frac{exp\big(u(p,w)\big)}{1 + exp(u(p,w)} \tag{2b}$$

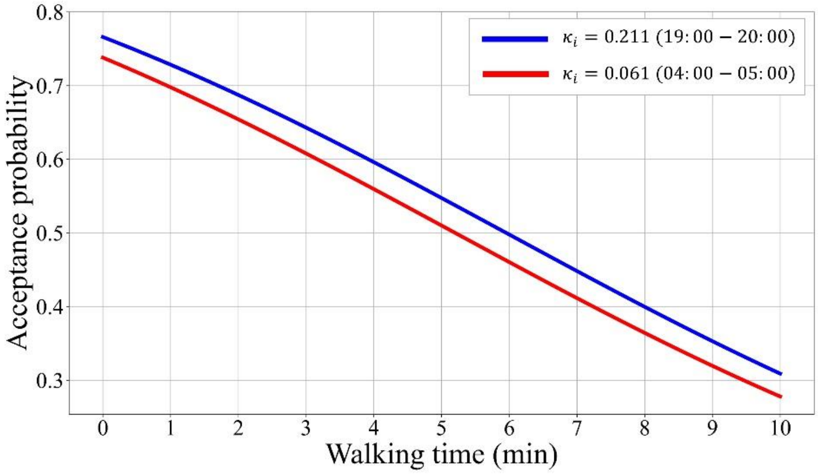

Figure 2 Acceptance probability to walking time with hour-of-day effects

Then, we define our objective function, p_{tv}, as maximizing the expected revenue and minimizing the cost, as in **Equation (3a)**. The expected revenue, e_{tv}, is determined by riders' acceptance probability and the fare that is proportional to the shortest travel distance, as shown in **Equation (3b)**. In the case of walking, replaced pick-up and drop-off points are used to calculate the shortest travel distance. Cost function c_{tv} is travel delay time d_r obtained by subtracting the request time t_r^q and the shortest travel duration s_r from drop-off time t_r^d, as in **Equation (3c)**. Thus, it includes the travel detour time and the waiting time of riders. By including the travel delay, the objective function works to increase both the profit and the operational efficiency.

$$p_{tv} = e_{tv} - c_{tv} \tag{3a}$$

$$e_{tv} = \sum_{r \in t} \tau_r(w_r) \cdot f_r(d) \tag{3b}$$

$$c_{tv} = \sum_{r \in t} (t_r^d - t_r^q) - s_r \tag{3c}$$

Chapter 4. Results

4.1. Simulation Settings

This section shows how we simulated the proposed walking points search model (WSM) to compare the performance with the original model without walking. We used the public dataset of New York City taxis on May 25, 2016 (404,310 requests). This dataset contains pick-up and drop-off points with the times that riders requested taxis. The road network of New York City consists of 4,091 nodes and 9,452 edges, with the mean travel time and shortest travel distance precomputed for routing. At the beginning of the simulation, vehicles are distributed uniformly in vehicle stations. For simulation parameters, the maximum waiting time of riders (including walking time) is set to 5 minutes, the maximum travel detour (in-vehicle time to shortest travel time) is set to 1.3, and the maximum walking time is set to 5 minutes (461 m in distance) for pick-up and drop-off points, respectively, assuming that people walk at 5 km/hr (Browning et al., 2006). The walking time used to calculate riders' acceptance probability is the sum of the walking time at the origin and destination. In addition, we fix the capacity parameters of vehicles to 4, and the batch period is 30 seconds. Unassigned requests to vehicles in previous batches are assigned requests in the next batch under their maximum waiting time constraints. The main simulation time is between 1:00 and 23:00 (22 hours), and we use 30% (113,126 requests) of the randomly selected requests among the real datasets with a fleet size of 500 vehicles.

4.2. Simulation Results

4.2.1. Comparison with the base model

Table 3 shows performance metrics for the base model and WSM. With 500 vehicles, WSM serves 77.84% of requests, which is an 8.28%p improvement. The increased number of requests directly lead to an increase in the total revenue of $ 63,182. Average delay time (sum of waiting time and detour time) and average in−vehicle time decreased by 18.2% (44 sec) and 8.6% (68 sec), respectively. Even though average waiting time increases slightly, i.e., by 7.5% (12 sec), over the base model, walking shortens the total travel time of riders (from request time to drop−off time) significantly, which is attributed to the reduction of the time in the vehicle. The average travel time of each vehicle is decreased by 1,093 seconds (18.21 min) while serving more riders during the entire simulation time. Furthermore, less time is wasted in empty travel, i.e., with not being assigned any schedules and not serving any riders, showing a decrease of 12.8% (382 sec). Due to reduced detours, they rather spend more time rebalancing themselves to where there are unserved requests. By leading riders to walk, a small increase in waiting time contributes to shorter travel time and higher utility rates of vehicles, which enables the vehicles to serve more riders.

TABLE 3 Simulation Results of Performance Metrics

		WSM	Base
Serviced requests		88,054 (77.84 %)	78,691 (69.56 %)
Total revenue		$ 814,047	$ 750,865
Riders	Avg. waiting time (including walking time)	172 s	160 s
	Avg. delay time	258 s	302 s
	Avg. in−vehicle time	719 s	787 s
Vehicles	Avg. travel time	68,003 s	69,096 s
	Avg. empty travel time	2,583 s	2,965 s
	Avg. rebalancing travel time	5,937 s	4,796 s

Figure 3 describes the hourly distribution of these metrics to compare the difference with varying demands. **Figure 3** shows the total requests that occurred and the serviced (completed) requests. As shown in this figure, the gap of serviced requests, indicated by the shaded area, becomes larger when the demand is high. When the demand is low, around 17:00 and during the night, enough vehicles exist to serve requests for both models. The hourly differences are better described in **Figure 4**. The service rate of the base model around 9:00 drops to 62.86%, which is 10.87% lower than the service rate of WSM. Around 10:00, when the soaring demands at 19:00 are maintained, the base model serves only 50% of total requests, while WSM serves 60% of total requests. Similarly, around 3:00, the service rates of the base model and WSM show a gap of 12.62%. As demands increase or remain higher than the supply during a certain period, more near−capacity vehicles accumulate on the streets, worsening the service rate. Showing a larger performance gap, in this case, indicates that adjusted acceptance probability proportional to

demand has an impact on controlling the over−demand.

Figure 5 represents the waiting times of both models and the walking time of WSM. The waiting time of WSM is higher than the base model throughout the day, and the time riders spend walking also is more than the waiting time of the base model, usually within 20 seconds. Around 15:00, as above, which shows the maximum gap in the service rate, the walking time of WSM catches up with its waiting time. In addition, as walking time increases to deal with the high demands, waiting time increases, but less time is spent waiting at the pick−up points. However, the improvement in the delay time indicates that this slight increase in waiting time is canceled out, as depicted in **Figure 6**. After 7:00, when the number of requests exceeds 2,000 (four times more than fleet size 500), the delay time of WSM remains 30 to 60 seconds lower than the base model.

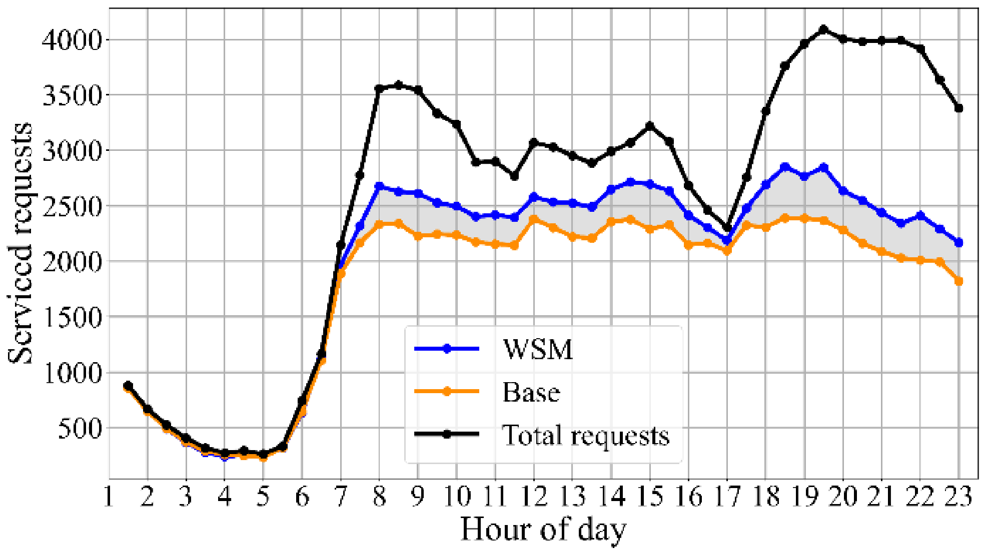

Figure 3 Hourly distribution of (a) total/serviced requests

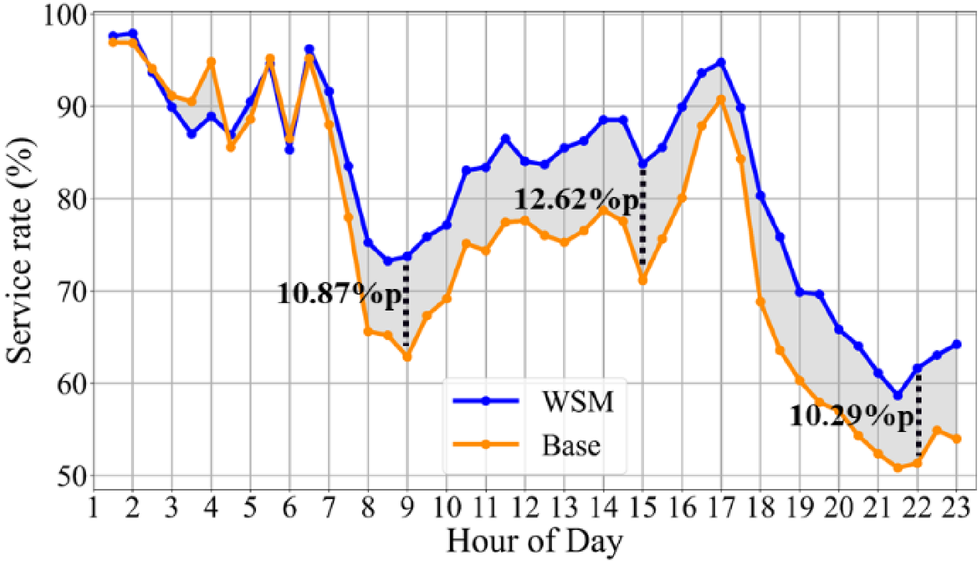

Figure 4 Hourly distribution of (b) service rate

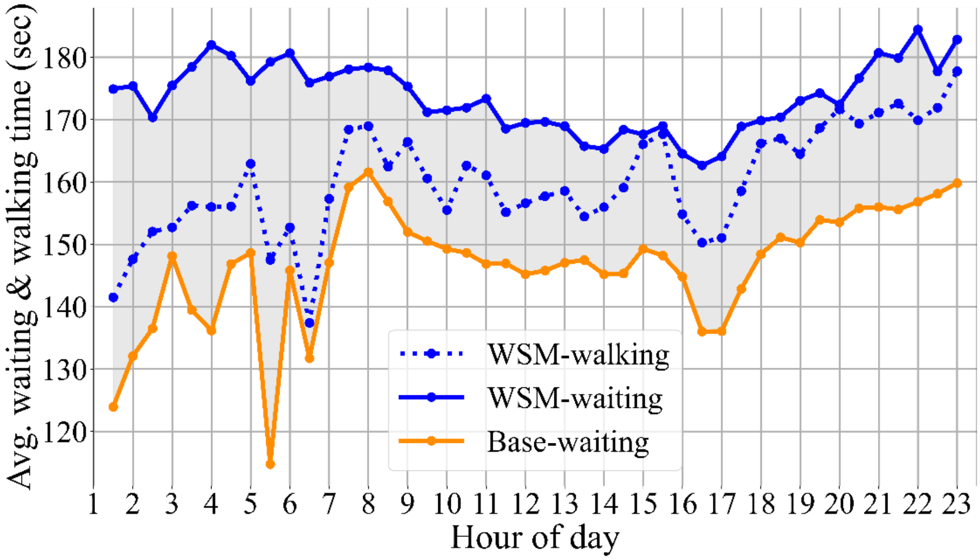

Figure 5 Hourly distribution of (c) waiting/waking time

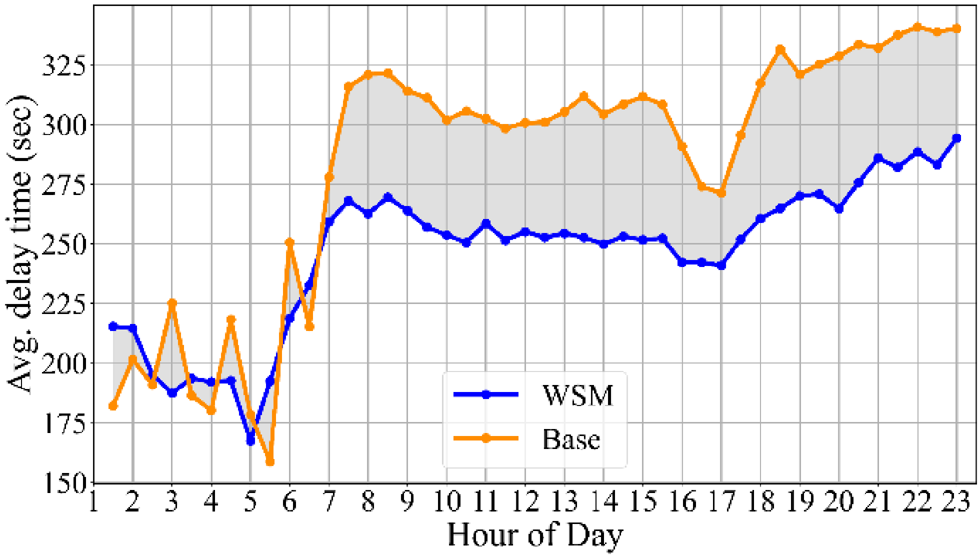

Figure 6 Hourly distribution of (d) delay time

Figure 7 shows heat maps weighted by additionally serviced requests by WSM. This figure depicts when and where the WSM serves the unmet demands of the original model. To see where the effect of walking appears markedly, we selected three time periods, i.e., 08:00–10:00, 14:00–16:00, and 22:00–24:00, when differences in service rates are substantial, as described above. As shown in **Figure 7 (a)**, during 08:00–10:00, additionally serviced requests are highly concentrated in some points in mid–Manhattan, near Penn Station, in particular. During 14:00–16:00 in **Figure 7 (b)**, additionally serviced requests show more dispersed distribution around mid–Manhattan and the Upper East Side. Finally, as in **Figure 7 (c)**, between 22:00 and 24:00, requests in mid–Manhattan, especially near Columbus Circle, are served more with WSM.

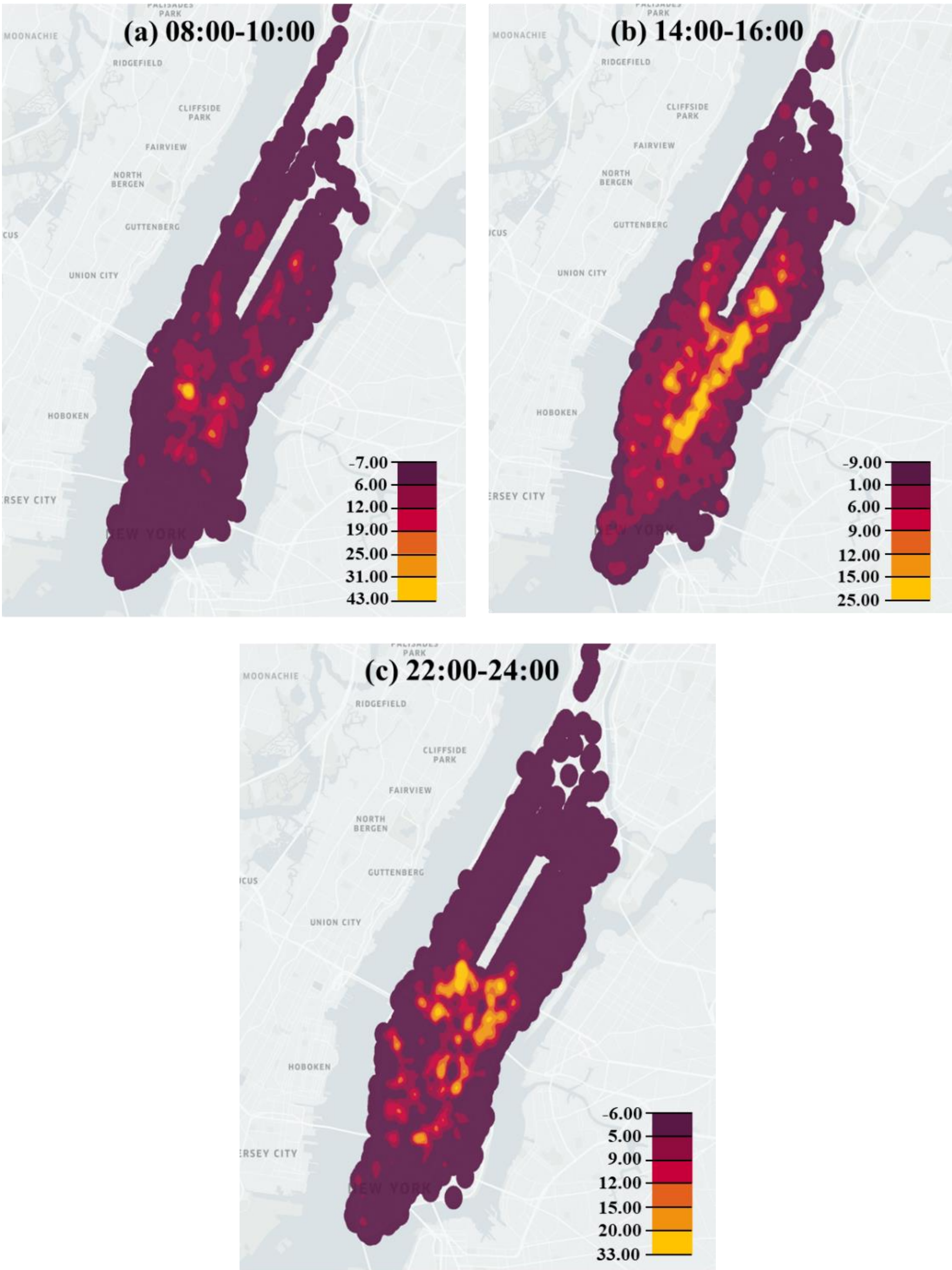

Figure 7 Heat maps weighted by additionally serviced requests by WSM during (a) 08:00−10:00; (b) 14:00−16:00; (c) 22:00−24:00

4.2.2. Analysis of WSM

In this part, we look further at walking−related indicators in WSM to reveal how it works in detail. First, as described in **Table 4**, the number of serviced requests with walking accounts for 86.63% of total serviced requests. The average walking distance to pick−up points and from drop−off points are 223 m and 256 m, respectively. Considering the maximum walking radius is a total of 10 minutes, riders usually walk half of the threshold, which might be affected by decreasing acceptance probability up to 30% near 10 minutes. Thus, the average acceptance probability is 49.81%, determined by total walking distance. Since the acceptance probability decreases to less than 50% if required to walk more than 5 minutes, if we want to make riders walk longer, which enables the search for better walking points, pricing strategies should be provided to entice them.

TABLE 4 Results of WSM regarding walking

	WSM
Serviced requests with walking	76,289 (86.63% of total serviced requests)
Avg. walking distance (time) to pick−up points	223 m (2.67 min)
Avg. walking distance (time) from drop−off points	256 m (3.07 min)
Avg. acceptance probability	49.81%
Avg. OD distance	3,287 m
Avg. updated OD distance	2,725 m
Avg. waiting time of vehicle	3.72 s
Avg. waiting time or rider (except walking time)	10.42 s

The average shortest travel distance between actual origin and destination is 3,287 m, but it is shortened by 562 m with replaced pick-up and drop-off points. It means that to reduce the travel detour, WSM optimizes the walking points toward shortening the route. In the process of finding better combinations with other riders' pick-up and drop-off points, it actually contributes to providing faster routes leading to the shorter travel time of vehicles while serving more riders. Furthermore, we also consider the situation in which vehicles arrive earlier than riders walking to the pick-up points. In this case, the vehicles need to wait for additional waiting time, which might lower the utility rate. Although we do not set the threshold waiting time for vehicles, the average time a given vehicle must wait is only 3.72 seconds, mainly due to in-vehicle riders' detour constraints. The average waiting time of riders, except for walking time, is 10.42 seconds, which is approximately 5% of the total waiting time. This indicates that riders of WSM usually exchange waiting time for walking time.

Actual origins of serviced requests	Relocated pick-up points of serviced requests

Figure 8 Heat maps of actual origins and relocated pick-up points of serviced requests around high-demand areas in peak hours; (a) Columbus Circle (08:00-10:00); (b) Penn Station (19:00-21:00)

Figure 8 shows some cases of how optimized walking points rebalance requests in high-demand areas during peak hours within the walkable range of 461 m. **Figure 8 (a) and (b)** describe actual origins and relocated pick-up points of serviced requests by walking near Columbus Circle from 08:00 to 10:00, respectively. As shown in the figures, the origins of requests concentrated around Columbus Circle disperse, and some requests are replaced to lower street. **Figure 8 (c) and (d)** depict the area near Penn Station from 19:00 to 21:00. Similarly, requests previously around Penn Station are relocated to nearby intersections. Considering these areas are one of the high-demand areas in Manhattan, WSM can function as dispersing or relocating requests, thereby mitigating the imbalance between demand and supply. These effects of walking are noticeable in high-demand areas during peak hours.

Chapter 5. Conclusion

In this study, a walking points search algorithm was developed to enable riders to participate actively in the current dispatching framework of ride-pooling services. With the aim of mitigating increasing travel delays with higher capacity common in ride-pooling, the proposed algorithm searches cost-minimizing neighborhood nodes that riders can walk to and walk from. With these optimized walking points, riders can get crucial information to be matched with vehicles, and operators can highly enhance the utility rates of the fleet. Furthermore, we take into account riders' probability of acceptance of suggested walking points. By reflecting riders' sensitivity to walking, which also is affected by hourly demand, we penalize our model more realistically.

The key contribution of this study is proving the enhancement achieved by walking and showing the effects both spatially and temporally with a simulation for a day. The proposed real-time walking points search algorithm shows significant improvement in all indicators used and the ability to deal with long-lasting over-demand, especially in high-demand areas. In our simulation results, the service rate increases by 8.28%, with decreases in delay time and in-vehicle time of 18.2% and 8.6%, respectively. The average waiting time shows a slight increase of 12 seconds; however, 95% of the waiting time is replaced by walking rather than standing still. While serving more requests, vehicles' average travel time and empty travel time are reduced by 1.5% and 12.8%, respectively. By saving previously wasted time, vehicles can spend more time rebalancing to serve more unassigned requests.

In addition, the results show that the performance of WSM is remarkable when high demand is maintained for hours. While over-demand hinders the utility rate of vehicles, dropping the service rate of the base model by as much as 50%, WSM is more resistant to it, serving a maximum of 12.62% more requests in an hour than the base model. Furthermore, WSM achieves this significant improvement with only 5 minutes walks on average. It is also observed that WSM disperses or relocates concentrated requests in high-demand areas to be matched. In this way, WSM serves additional requests that might not be served in the base model not including walking.

There is some future work to be done. First, since we could not obtain the riders' revealed preference data for walking time, we had to use the disutility parameter of bus riders. By estimating the disutility of the walking of riders in ride-sharing services, we can better describe their behavior in the framework. In addition, as shown in the results, the acceptance probability is lowered as the walking time increases. Therefore, price discounts should be optimized to encourage riders to walk more often to exploit the benefit of walking fully. Joint optimization of price and walking points would benefit both operators and riders by increasing the total profit and giving riders more options.

Bibliography

Agrawal, A. W., M. Schlossberg, and K. Irvin. How Far, by Which Route and Why? A Spatial Analysis of Pedestrian Preference. Journal of Urban Design, Vol. 13, No. 1, 2008, pp. 81–98. https://doi.org/10.1080/13574800701804074.

Aïvodji, U. M., S. Gambs, M. J. Huguet, and M. O. Killijian. Meeting Points in Ridesharing: A Privacy-Preserving Approach. Transportation Research Part C: Emerging Technologies, Vol. 72, 2016, pp. 239–253. https://doi.org/10.1016/j.trc.2016.09.017.

Al-Abbasi, A. O., A. Ghosh, and V. Aggarwal. DeepPool: Distributed Model-Free Algorithm for Ride-Sharing Using Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems, Vol. 20, No. 12, 2019, pp. 4714–4727. https://doi.org/10.1109/TITS.2019.2931830.

Alonso-Mora, J., S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus. On-Demand High-Capacity Ride-Sharing via Dynamic Trip-Vehicle Assignment. Proceedings of the National Academy of Sciences of the United States of America, Vol. 114, No. 3, 2017, pp. 462–467. https://doi.org/10.1073/pnas.1611675114.

Bertsimas, D., P. Jaillet, and S. Martin. Online Vehicle Routing: The Edge of Optimization in Large-Scale Applications. Operations Research, Vol. 67, No. 1, 2019, pp. 143–162. https://doi.org/10.1287/opre.2018.1763.

Browning, R. C., E. A. Baker, J. A. Herron, and R. Kram. Effects of Obesity and Sex on the Energetic Cost and Preferred Speed of Walking. Journal of Applied Physiology, Vol. 100, No. 2, 2006, pp. 390–398. https://doi.org/10.1152/japplphysiol.00767.2005.

Castillo, J. C., D. Knoepfle, and G. Weyl. Surge Pricing Solves the Wild Goose Chase. EC 2017 - Proceedings of the 2017 ACM Conference on Economics and Computation, No. July, 2017, pp.

241–242. https://doi.org/10.1145/3033274.3085098.
Chia, J., J. Lee, and M. D. Kamruzzaman. Walking to Public Transit: Exploring Variations by Socioeconomic Status. International Journal of Sustainable Transportation, Vol. 10, No. 9, 2016, pp. 805–814. https://doi.org/10.1080/15568318.2016.1156792.
Daniels, R., and C. Mulley. Explaining Walking Distance to Public Transport: The Dominance of Public Transport Supply. Journal of Transport and Land Use, Vol. 6, No. 2, 2013, pp. 5–20. https://doi.org/10.5198/jtlu.v6i2.308.
Duan, Y., N. Wang, and J. Wu. Optimizing Order Dispatch for Ride-Sharing Systems. Proceedings - International Conference on Computer Communications and Networks, ICCCN, Vol. 2019-July, 2019. https://doi.org/10.1109/ICCCN.2019.8847177.
Fielbaum, A., X. Bai, and J. Alonso-Mora. On-Demand Ridesharing with Optimized Pick-up and Drop-off Walking Locations. Transportation Research Part C: Emerging Technologies, Vol. 126, No. March, 2021, p. 103061. https://doi.org/10.1016/j.trc.2021.103061.
Heineke, K., B. Kloss, T. Möller, and C. Wiemuth. McKinsey Center for Future Mobility. McKinsey & Comapany, No. August, 2021.
Hosni, H., J. Naoum-Sawaya, and H. Artail. The Shared-Taxi Problem: Formulation and Solution Methods. Transportation Research Part B: Methodological, Vol. 70, 2014, pp. 303–318. https://doi.org/10.1016/j.trb.2014.09.011.
Ian Ker, S. G. Myths and Realities in Walkable Catchments: The Case of Walking and Transit. Journal of Allergy and Clinical Immunology, Vol. 130, No. 2, 1998, p. 556.
Jung, J., R. Jayakrishnan, and J. Y. Park. Dynamic Shared-Taxi Dispatch Algorithm with Hybrid-Simulated Annealing. Computer-Aided Civil and Infrastructure Engineering, Vol. 31, No. 4, 2016, pp. 275–291. https://doi.org/10.1111/mice.12157.
Lee, D. H., H. Wang, R. L. Cheu, and S. H. Teo. Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions. Transportation Research Record, No. 1882, 2004, pp. 193–200. https://doi.org/10.3141/1882-23.

Li, C., D. Parker, and Q. Hao. Optimal Online Dispatch for High-Capacity Shared Autonomous Mobility-on-Demand Systems. 2021, pp. 779–785. https://doi.org/10.1109/icra48506.2021.9561281.

Liu, Y., and S. Samaranayake. Proactive Rebalancing and Speed-Up Techniques for On-Demand High Capacity Ridesourcing Services. IEEE Transactions on Intelligent Transportation Systems, Vol. 23, No. 2, 2022, pp. 819–826. https://doi.org/10.1109/TITS.2020.3016128.

Ma, S., Y. Zheng, and O. Wolfson. Real-Time City-Scale Taxi Ridesharing. IEEE Transactions on Knowledge and Data Engineering, Vol. 27, No. 7, 2015, pp. 1782–1795. https://doi.org/10.1109/TKDE.2014.2334313.

Pelzer, D., J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt. A Partition-Based Match Making Algorithm for Dynamic Ridesharing. IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 5, 2015, pp. 2587–2598. https://doi.org/10.1109/TITS.2015.2413453.

Qian, X., W. Zhang, S. V. Ukkusuri, and C. Yang. Optimal Assignment and Incentive Design in the Taxi Group Ride Problem. Transportation Research Part B: Methodological, Vol. 103, 2017, pp. 208–226. https://doi.org/10.1016/j.trb.2017.03.001.

Qiu, H., R. Li, and J. Zhao. Dynamic Pricing in Shared Mobility on Demand Service. 2018, pp. 1–9.

Santi, P., G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti. Quantifying the Benefits of Vehicle Pooling with Shareability Networks. Proceedings of the National Academy of Sciences of the United States of America, Vol. 111, No. 37, 2014, pp. 13290–13294. https://doi.org/10.1073/pnas.1403657111.

Seow, K. T., N. H. Dang, and D. H. Lee. A Collaborative Multiagent Taxi-Dispatch System. IEEE Transactions on Automation Science and Engineering, Vol. 7, No. 3, 2010, pp. 607–616. https://doi.org/10.1109/TASE.2009.2028577.

Shah, S., M. Lowalekar, and P. Varakantham. Neural Approximate Dynamic Programming for On-Demand Ride-Pooling. AAAI 2020 - 34th AAAI Conference on Artificial Intelligence, 2020, pp. 507–515. https://doi.org/10.1609/aaai.v34i01.5388.

Simonetto, A., J. Monteil, and C. Gambella. Real-Time City-Scale Ridesharing via Linear Assignment Problems. Transportation Research Part C: Emerging Technologies, Vol. 101, No. February, 2019, pp. 208–232. https://doi.org/10.1016/j.trc.2019.01.019.

Stiglic, M., N. Agatz, M. Savelsbergh, and M. Gradisar. The Benefits of Meeting Points in Ride-Sharing Systems. Transportation Research Part B: Methodological, Vol. 82, 2015, pp. 36–53. https://doi.org/10.1016/j.trb.2015.07.025.

Wang, Z., Z. Qin, X. Tang, J. Ye, and H. Zhu. Deep Reinforcement Learning with Knowledge Transfer for Online Rides Order Dispatching. Proceedings - IEEE International Conference on Data Mining, ICDM, Vol. 2018-Novem, 2018, pp. 617–626. https://doi.org/10.1109/ICDM.2018.00077.

Wong, K. I., and M. G. H. Bell. The Optimal Dispatching of Taxis under Congestion: A Rolling Horizon Approach. Journal of Advanced Transportation, Vol. 40, No. 2, 2006, pp. 203–220. https://doi.org/10.1002/atr.5670400207.

Xu, Z., Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye. Large-Scale Order Dispatch in on-Demand Ride-Hailing Platforms: A Learning and Planning Approach. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 905–913. https://doi.org/10.1145/3219819.3219824.

Yan, C., H. Zhu, N. Korolko, and D. Woodard. Dynamic Pricing and Matching in Ride-Hailing Platforms. Naval Research Logistics, Vol. 67, No. 8, 2020, pp. 705–724. https://doi.org/10.1002/nav.21872.

Yu, X., and S. Shen. An Integrated Decomposition and Approximate Dynamic Programming Approach for On-Demand Ride Pooling. IEEE Transactions on Intelligent Transportation Systems, Vol.

21, No. 9, 2020, pp. 3811–3820. https://doi.org/10.1109/TITS.2019.2934423.

Zhang, H., Y. Ding, W. Zhang, S. Feng, Y. Zhu, Y. Yu, Z. Li, C. Liu, Z. Zhou, and H. Jin. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, 2019, pp. 3620–3624. https://doi.org/10.1145/3308558.3314139.

Zhao, F., L. F. Chow, M. T. Li, I. Ubaka, and A. Gan. Forecasting Transit Walk Accessibility: Regression Model Alternative to Buffer Method. Transportation Research Record, No. 1835, 2003, pp. 34–41. https://doi.org/10.3141/1835-05.

Abstract

Ride-pooling 서비스는 기존의 ride-sharing 서비스의 시스템 효율성을 크게 증대시켰다. 하지만 여러 명의 승객들이 하나의 차량에 동승하여 운행하는 특성으로 인해, 동승하는 승객들이 많을수록 승객당 통행 지체 시간이 길어진다는 단점이 존재한다. 이러한 문제를 해결하기 위해, 본 연구는 승객들의 기존 승하차 지점으로부터 걸어서 도달 가능한 대안 승하차 지점을 최적화하는 알고리즘을 제안한다. 기존의 승객-차량 배정 프레임워크를 기반으로, 비용을 최소화하는 대안 승하차지점 탐색 알고리즘을 구현하였다. 또한, 도보 이동 거리에 대한 승차 수락률 모델을 통해 승객들의 도보 이동에 대한 민감도를 반영하였다. 뉴욕시티의 옐로우캡 택시 데이터를 이용한 하루 동안의 시뮬레이션을 통해 제안한 모델을 검증하고 기존 모델과의 비교를 수행하였다. 시뮬레이션 결과, 서비스율은 69.56%에서 77.84%로 증가하였으며, 통행 지체 시간은 평균적으로 18.2% 감소하였고, 차내 시간은 8.6% 감소하였다. 서비스율의 증가와 함께, 차량들의 평균 총 운행 시간은 1.5% 감소하였고 이는 차량 재배치 시간의 증가로 이어짐을 확인하였다. 또한, 시뮬레이션 결과를 시공간 상에서 분석함으로서, 피크 시간대에 수요가 밀집되는 지역에서 도보 이동의 효과가 극대화됨을 보였다. 본 연구는 도보 이동을 통한 대안 승하차지점 이용이 ride-pooling 서비스의 운영 효율성을 향상시키며, 제한된 수의 공급 대수로 수요-공급의 불균형을 해소시킴을 증명하였다. 제안된 모델은 택시 뿐 아니라 수요응답형 셔틀 등 다양한 종류의 다인승 차량 서비스에서 승객과 차량의 승하차 지점을 최적화하는 데에 활용될 수 있다.

Keyword : Ride-sharing, Ride-pooling, On-demand mobility, Walking to ride, Meeting points

Student Number : 2021-26822