Proceedings of the 2017 Joint Rail Conference
JRC2017
April 4-7, 2017, Philadelphia, Pennsylvania, USA

Paper Number: JRC2017-2251

PLANNING THE RIDESHARING ROUTE FOR THE FIRST-MILE SERVICE LINKING TO RAILWAY PASSENGER TRANSPORTATION

Zheyong Bian
Department of Civil and Environmental Engineering
Rutgers, The State University of New Jersey
CoRE 606, 96 Frelinghuysen Road
Piscataway, NJ 08854-8018

Xiang Liu, Ph.D.
Department of Civil and Environmental Engineering
Rutgers, The State University of New Jersey
CoRE 606, 96 Frelinghuysen Road
Piscataway, NJ 08854-8018

ABSTRACT

First-mile and last-mile travel is the bottleneck of using the public transportation service. This paper considers the passenger matching and vehicle routing problem in the first-mile ridesharing service connecting train schedules. Then a mixed integer model is proposed to formulate the problem. Since the problem is NP hard, we develop a simulated annealing (SA) algorithm with four neighborhood structures to solve this problem. Experiments are designed to test the proposed model and algorithm. The experimental results verify the effectiveness of the algorithm and demonstrate that the proposed algorithm can obtain satisfactory solutions within a reasonably short time.

1. INTRODUCTION

Public transit is essential for supporting societies undergoing population growth, urban development, and climate change. A key challenge for using public transit is the first-and-last-mile accessibility from/to transit hubs (e.g. train stations). The $61 billion public transportation industry, consisting of 7,200 organizations in the U.S., faces an enduring challenge - how transit users travel between their locations and transit hubs. Literally, this is described as the “first mile and last mile” bottleneck. Numerous studies have found that travelers’ choice of public transportation is significantly affected by the first-mile-last-mile accessibility from/to transit stations (e.g. railway station, bus stop). Individual transit users typically drive their personal vehicles or take taxis between their homes and transit hubs neither of which fully utilizes available seats, thereby sustaining a high traveling cost and increasing road congestion and parking demand. Using shared mobility service is an effective solution to these problems.

In recent years, the number of shared mobility APPs in the market has been increasing exponentially, which provides travelers with a variety of alternative transportation services. Table 1 summarizes the existing smart phone mobility APPs in the market.

Table 1 Summary of existing mobility APPs

Characteristics	Existing APPs							
	Flywheel, Curb	BlablaCar, CarpooleAR	Scoop, Ford, Wazerider	Transit	Uber, Lyft, Via, Easy-taxi	Zipcar, Car2go	Spinlister	Chariot
On demand	√	√	√		√	√	√	
Ridesharing		√	√	√	√			√
Vehicles	Taxi	Private cars	Private cars	Public vehicles	Private cars	Business smart electric cars	Smart bikes	Business vehicles
Application	Door-to-door short distance	Long distance: inter-city	Door-to-workplace: commute	Short distance fixed routes	Door-to-door short distance	Short distance carsharing	Short distance bikesharing	Short distance fixed routes

The table classifies the existing smart phone mobility APPs into eight categories.

1) Flywheel [1], Curb [2] and etc. smartphone APPs provide passengers with the taxi hailing service. Passengers can send a

request for taxi using and pay for the service automatically with the phone after the passengers arrive at the destination. 2) Transit [3] APPs are passengers' real-time urban travel companion using public transportation means including bus, trains, ferries, etc. They can help passengers navigate public transit system and make an optimal travel plan. 3) Chariot [4] is similar as Transit, but unlike Transit, Chariot only has one type of vehicles in one city. In addition, vehicles are owned by the company and thus can change the route based on the needs. 4) Zipcar [5], Car2go [6], Volug [7] etc. are car rental APPs that provide members with carsharing service – accessing automobiles for short-term use (usually hourly). 5) Spinlister [8] etc. APPs are similar as Zipcar and Car2go, but is designed for bikesharing. 6) Scoop[9], Wazerider [10], etc. APPs are used to provide commuters with the ridesharing service to commute between their homes and workplaces every day. Drivers, who are also commuters, can use these APPs to pick up other commuters and arrive at the same workplaces. 7) BlablaCara [11] and CarpooleAR [12] are another type of ridesharing mobility APPs. These APPs is used for city-to-city ridesharing service. The driver can offer a ride to other members in his or her car if he/she will have a long-distance travel. 8) The most widely used mobility APPs is the category of Uber [13], Lyft [14], Via [15] and Easy-taxi [16]. They can provide demand responsive door-to-door ridesharing service.

The existing mobility services have achieved significant economic and social benefits. However, none of them are specifically developed for passengers' first-and-last-mile transportation, and all these APPs except Transit are owned by private companies whose primary goal is maximizing their revenue. Therefore, almost all these APPs have certain limitations when being used for first-and-last-mile transportation.

APPs, such as BlablaCar, CarpooleAR, Scoop, Ford and Wazerider, which are designed for specific purposes (e.g. long-distance travel and commuting), are unable to provide the service of first-and-last-mile transportation. Taxi APPs (Flywheel and Curb) can help people to achieve the first-and-last-mile travel by requesting a ride with the smartphone, but will not reduce the traveling cost, roadway congestion and emission without a ridesharing function. Fixed-route APPs (Transit and Chariot) have a high-level ridesharing service, but lack accessibility (passengers have to reach the pre-determined stop in time to get in the vehicle). It is inconvenient for travelers' first-and-last-mile travel. Vehicle rental APPs, such as Zipcar, Car2go and Spinlister, can be used for first-and-last-mile travel, and can reduce traveling cost and emissions. However, travelers have to find vehicles nearby when they start to travel and have to find a nearby place to park the vehicles after they finish the journey. These inconveniences significantly reduce its accessibility. Uber, Lyft, Via, Easy-taxi etc. are the only category APPs that can provide the service of first-and-last-mile on-demand ridesharing transportation. However, their APPs are not linked to public transit schedules, and the system cannot provide drivers and passengers with information of transit schedules. Thus, if passengers choose to use these services, it is possible that they may miss the train or wait the train for a long time after they arrive at the train station (first-mile). In addition, they may have to wait for a long time to be picked up at the train station before the last-mile travel. Many previous studies have been proposed to satisfy the practical demand of mobility. For example, researchers studied technology, impact, policy and future prospect of some formations of shared mobility, such as carsharing [17-19], bikesharing [20-29], and ridesourcing [30-34]. Ridesharing is another emerging mode of shared mobility and has been widely studied in academic fields. Several ridesharing patterns have been studied in literature.

1) Scheduled carpooling [35-49]: scheduled service for travelers that share ride in a private vehicle with other appointed travelers. Typically, drivers and other participants have similar origins and destinations or the drivers has convenient pickup and drop-off routes for other participants. This service is usually provided for those who have regular travel plans, such as daily commuters.

2) Flexible carpooling [50-56]: all participants will be matched at a predetermined spot and time, which are publicly known in advance, after they come to this spot. This type of ridesharing usually happens in long-distance travel cases.

3) Dynamic ridesharing [57-68]: providing an automated process of ride-matching (routing, scheduling, and pricing) between drivers and passengers on very short notice [69]. Passengers send on-demand requests. The system will respond to requests in a very short time and then generate passenger matching and vehicle routing plan.

However, no researchers have specifically designed the service for first-mile and last-mile transportation. More precisely, none of these researches connect the ridesharing service to train schedule. Based on the analysis of limitations and disadvantages of existing APPs and researches in providing the first-and-last-mile transportation service, it is obvious that to reduce the first-and-last-mile accessibility from/to transit hubs (e.g. train stations) is still a key challenging problem for using the public transportation service. To narrow the first-and-mast-mile gap for public transit, some transportation companies such as Transit start to specifically develop an economic, convenient and efficient transportation service – first-and-last-mile ridesharing service. This service will transform the way travelers use public transit via demand-driven ridesharing. This paper will provide the forthcoming service system with methodology to eliminate the gap between the existing smartphone mobility APPs & travelers' demand for more convenient, economic, and efficient first-mile transportation service.

This paper contributes to solving the passenger matching and vehicle routing problem in the first-mile ridesharing transportation (we name it as first-mile ridesharing problem for short, FMRP). The last-mile ridesharing problem is not considered in this paper because it is relatively easy to solve. Last-mile ridesharing problem is a classical vehicle routing problem (VRP), which can be solved by many existing approaches in literature [70-73].

FMRP also has similarities to the classical VRP, but it has unique characteristics, which makes it difficult to handle.

1) In FMRP, available vehicles are randomly distributed at any locations. Vehicles finish their task when all they arrive at the train station dropping off all passengers. In traditional VRP, all vehicles are dispatched from one depot. After all tasks are finished, all vehicles should return to the depot.

2) The first-mile ridesharing transportation service is linked to train schedule in the train station, and passengers will specify which train they will take. The train schedule will impose a "hard deadline" before which passengers must arrive at the train station.

3) FMRP should take passenger matching problem into account, which is not considered in the classical VRP.

This paper is structured as follows: Section 2 builds a mathematical model – mixed integer programming – to formulate the problem. Section 3 introduces the algorithm to solve the model. Experiments are designed in Section 4 to verify the effectiveness of the model and algorithm. Finally, conclusions are drawn and future work is proposed in Section 5.

2. FORMULATION

2.1 Problem Statement

Suppose m passengers in different regions near the train station are requesting the first-mile ridesharing service. They all have the same destination (e.g. the train station). The service company has n available cars distributed randomly in the nearby regions to provide the first-mile ridesharing service. Each passenger will specify a train they will take. The train schedule imposes each passenger' deadline before which they must arrive at the train station. The problem is to determine a simultaneous optimal passenger matching plan and vehicle routing plan with the objective of minimizing travel cost. After the plan is determined, related information will be sent to drivers' and passengers' smartphones. Drivers can execute the transportation task based on the plan, and passengers can view the information of the upcoming travel.

2.2 Assumptions

The proposed mathematical model has the following assumptions.

(1) Each passenger is served by exactly one vehicle. The passenger does not change the vehicle on the way.

(2) All vehicles are the same type, i.e. all vehicles have the same capacity.

The following part introduces what we have considered or not considered.

(1) Only one transit hub is considered in this research. In other words, we only optimize the passenger matching and vehicle routing plan for only one transit hub.

(2) We do not consider passengers' waiting time as an objective function. As long as passengers can catch the train, the plan is feasible.

(3) We only consider the optimization problem within a specific time period, and do not consider the impact of future passengers' pickup request on the optimization plan.

The model has the following inputs:

a. Number of passengers;
b. Number of vehicles;
c. Location of each passenger;
d. Location of each vehicle;
e. Travel cost between each passenger pair;
f. Travel cost from each passenger to the train station;
g. Vehicle capacity;
h. The train departure time for each passenger.

2.3 Notations

Suppose that n cars are available to pick up passengers. $V=\{1,2,\ldots,n\}$ represents the set of vehicles and the vehicles are indexed by k. The initial locations of all vehicles are donated by $LV=\{1,2,\ldots,n\}$. m passengers are waiting to be picked up. $P=\{n+1,n+2,\ldots,n+m\}$ represents the set of passengers and each passenger is indexed by i. The train station is denoted by $H=\{0\}$.

Let

$$L=LV\bigcup P$$

$$M=P\bigcup H$$

$$W=LV\bigcup P\bigcup H$$

$$y_i^k=\begin{cases}1 & \text{vehicle } k \text{ picks up passenger } i\\ 0 & \text{Otherwise}\end{cases}$$

$$k\in V, i\in P$$

$$x_{ij}^k = \begin{cases} 1 \text{ vehicle } k \text{ travels from node } i \text{ to node } j \\ 0 \text{ Otherwise} \end{cases}$$

$k \in V, i \in L, j \in M$

c_{ij}: the transportation cost from node i to node j, $i \in L, j \in M$

t_{ij}: the travel time from node i to node j, $i \in L, j \in M$;

Q: the seat capacity of a vehicle;

T_w^k: the time moment when the vehicle k arrives at node w, $w \in M \bigcup \{k\}$.

DT_i: the departure time of the train that the passenger i will catch

2.4 Model formulation

The FMRP can be formulated as follows.

Objective function:

Minimize:

$$\sum_{i \in L} \sum_{j \in M} \sum_{k \in V} x_{ij}^k c_{ij} \tag{1}$$

Subject to

$$\sum_{k \in V} \sum_{i \in L} x_{ij}^k = 1 \quad \text{for all } j \in P \tag{2}$$

$$\sum_{k \in V} \sum_{j \in M} x_{ij}^k = 1 \quad \text{for all } i \in P \tag{3}$$

$$\sum_{i \in P} y_i^k \le Q \quad \text{for all } k \in V \tag{4}$$

$$\sum_{i \in L} x_{ij}^k = y_j^k \quad \text{for all } k \in V, j \in P \tag{5}$$

$$\sum_{j \in M} x_{ij}^k = y_k^i \quad \text{for all } k \in V, i \in P \tag{6}$$

$$T_k^k = 0, \quad \text{for all } k \in V \tag{7}$$

$$T_j^k = \sum_{i \in \{k\} \bigcup P} x_{ij}^k \left(T_i^k + t_{ij} \right) \text{ for all } j \in M, k \in V \tag{8}$$

$$T_0^k \le (1 - y_i^k) M + DT_i \quad \text{for all } k \in V, i \in P \tag{9}$$

In the above formulation, the objective function Formula (1) is to minimize the total transportation cost. Formulas (2)-(9) are the constraints that the decision variables must satisfy. Formula (2) and (3) ensure that all passengers will be picked up by one vehicle and only be served once. Formula (4) represents that the capacity of each vehicle should not be exceeded. Formula (5) signifies that if passenger j is picked up by vehicle k, the vehicle k must come from one site. Formula (6) means that if passenger i is picked up by car k, the car k must travel to the next site, either the destination (train station) or the location of the next passenger. Formula (7) states that the departure times of vehicles are set to be zero. Formula (8) guarantees that if vehicle k travels from site i to site j, the arrival time at site j should be the arrival time at site i plus the travel time from site i to j. Formula (9) indicates that all the passengers should be transported to the train station before the departure times of their trains.

3. APPROACH

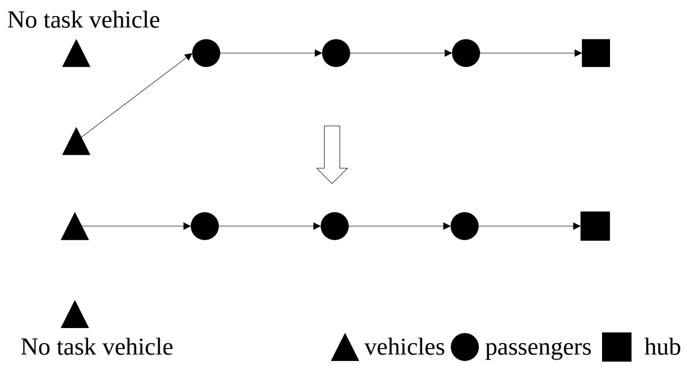
The problem defined is NP-hard [70]. Thus, it cannot be solved within a reasonable time using exact algorithms when the scale of problem is large. Regarding our problem defined, the locations of vehicles are updating very quickly, and passengers will be impatient if the waiting time is too long. Thus the reasonable computing time is very short for our problem. For example, if we spend one hour getting a satisfactory solution (a reasonable vehicle assignment plan and a short routing length), the result cannot be used because locations of cars has changed significantly within one hour, and passengers have already been impatient when they need to wait for one hour to be assigned to a vehicle. Furthermore, within one hour, there may be newly emerging passengers placing new order for pickups. Thus, for our problem, the reasonable time must be very short. For example, the computational time should be within one minute.

Since the problem defined has a high requirement for the computing time, we need to develop an efficient heuristic algorithm. Simulated annealing (SA) is a promising algorithm for routing-related problems, whose effectiveness is demonstrated by several researchers [70, 72, 73]. We attempt to employ SA to solve the FMRP. However, the accuracy of simulated annealing is significantly influenced by neighborhood structure. In order to improve the efficiency and accuracy of simulated annealing algorithm, we test various neighborhood structures to find effective neighborhood structures. Finally, we select four neighborhood structures and combined them into the simulated annealing with a given probability for each neighborhood structure. We find the mixed neighborhood structures that can have the highest computational efficiency and accuracy.

3.1 Neighborhood structures

We firstly introduce the four neighborhood structures. Note that all the four neighborhood structures below have the possibility of generating infeasible solutions (constraints of train departure time and vehicles' capacity are not satisfied, formula (4) and (9)). Thus, when infeasible solutions are generated, we re-generate a solution until the constraints are satisfied.

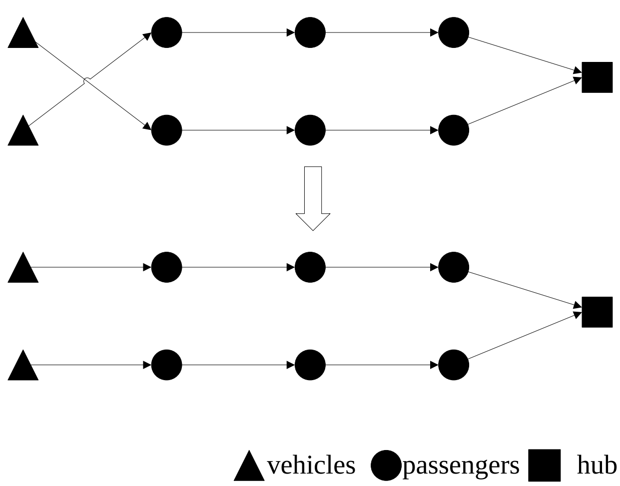
The first neighborhood structure is to hand over all pickup tasks from one vehicle to another vehicle. We randomly select one vehicle (V_1) without assigned passengers, and select another vehicle (V_2) with assigned passengers. Then let the vehicle V_1 implements the transportation task instead of vehicle V_2, as shown in Figure 1(a)



Hand over all pickup tasks to another vehicle

Figure 1(a) The first neighborhood structure

The second neighborhood structure is to exchange all tasks between two vehicles. We randomly select two vehicles with assigned passengers, and then exchange assigned passengers to finish transportation tasks. This neighborhood structure is shown in Figure 1 b.

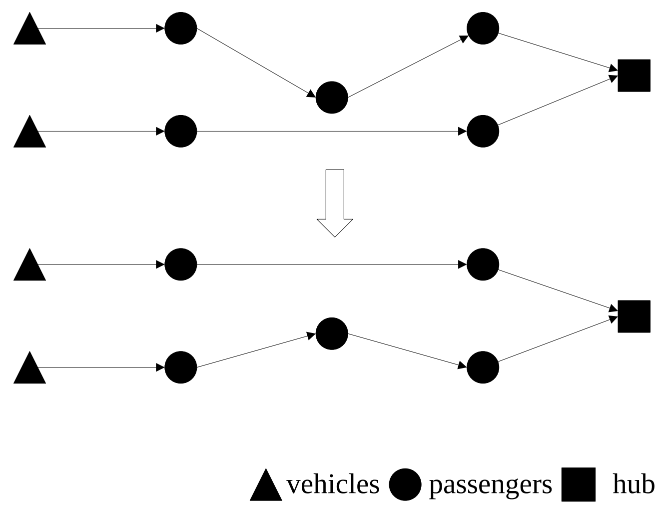


Exchange all pickup tasks between two vehicles

Figure 1(b) The second neighborhood structure

The third neighborhood structure is to hand over one pickup task from one vehicle to another vehicle. We randomly select one vehicle and one of the passengers assigned to this vehicle, and select another vehicle which has available seats. Then the second vehicle replaces the first vehicle to transport this passenger to the train station, as shown in Figure 1(c).

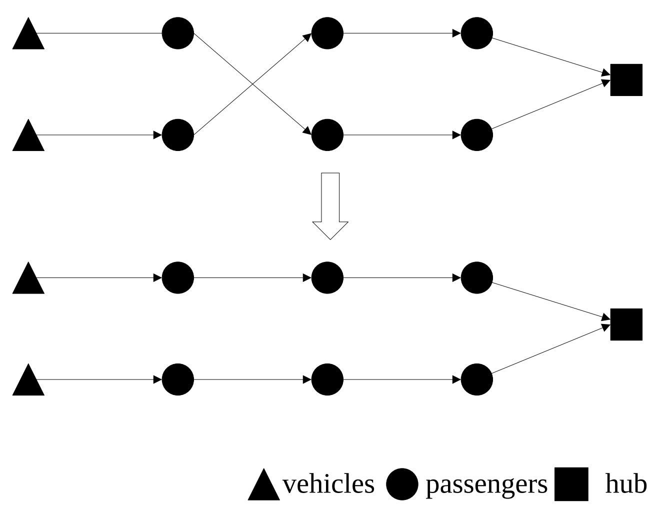
.



Hand over one passenger to another vehicle

Figure 1(c) The third neighborhood structure

Finally, the fourth neighborhood structure is to randomly break two links and build two new links based on the original routes. This neighborhood is the famous 2-opt neighborhood structure which is widely used in route-related problems.



Break two links and re-build two new links (2-opt)

Figure 1(d) The fourth neighborhood structure

3.2 Simulated annealing

We use simulated annealing algorithm to conduct iterations with the four mixed neighborhood structures. The parameters and variables of simulated annealing are defined as follows.

t_{start} : initial temperature

t_{end} : final temperature

$t_{current}$: current temperature

X_0 : initial feasible solution

$X_{current}$: current solution during iterations

$E_{current}$: objective function value of the current solution $X_{current}$

X_{best} : best solution obtained during the execution of the algorithm

E_{best} : objective function value of the solution X_{best}

Len: epoch length (number of inner-loop iterations)

α : coefficient of temperature decrease

it: accumulative number of current iterations

$Z(X)$: the objective function value of a solution *X*

The simulated annealing begins at the initial temperature (t_{start}), and ends at the final temperature (t_{end}). The temperature is decreased at an exponential rate via multiplication by a cooling coefficient (α) which is slightly lower than 1. At each temperature, SA performs *Len* (epoch length) iterations, and accepts a newly generated solution with a probability of ρ at each iteration.

Algorithm: simulated annealing

Using insertion heuristic algorithm to construct a feasible solution X_0.

Initialize $t_{current} = t_{start}, X_{current} = X_0$, $X_{best} = X_0, E_{current} = Z(X_0), E_{best} = Z(X_0)$ and $it = 0$.

Do while $t_{current} \geq t_{end}$

Do while $it \leq Len$

Use the mixed four neighborhood structures to generate a neighbor solution (X') of $X_{current}$

If $Z(X') \leq Z(X_{current}) \,||\, exp(\frac{Z(X_{current}) - Z(X')}{t_{current}}) < rand$

Accept X' as the current solution $X_{current}$, i.e. $X_{current} = X'$

$E_{current} = Z(X')$

If $Z(X') \leq E_{best}$

$X_{best} = X', E_{best} = Z(X')$

End if

End if

it=*it*+1

End do

$t_{current} = \alpha \cdot t_{current}$

End do

Output X_{best} and E_{best}

4. NUMERICAL EXAMPLE

4.1 Data setting

In order to test the proposed algorithm, we use the computer to randomly generate numerical examples. We totally generate seven numerical examples with increasing scales. The data generated are show in Table 2, where *N* (0, 5) means that the coordinates, including abscissa and ordinate, are drawn from normal distribution with mean of 0 and standard deviation of 5. Traveling time between two nodes is $3d_{ij} + N(0, 0.1)$, d_{ij} is the distance between two nodes, and $N(0, 0.1)$ is number drawn from normal distribution, with mean of 0 and standard deviation of 0.1. Note that the traveling time is not direct proportional to the distance considering different congestion conditions of different roads in practice.

Table 2 Data setting

Data	Numerical examples						
	1	2	3	4	5	6	7
Number of vehicles	10	20	20	40	30	60	100
Number of passengers	10	10	20	20	30	30	100
Locations of cars and	*N* (0, 5)	*N* (0, 5)	*N* (0, 5)	*N* (0, 5)	*N* (0, 5)	*N* (0, 5)	*N* (0, 5)

passengers							
Location of transit hub	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Traveling time between two nodes	$3d_{ij} + N(0, 0.1)$						

4.2 Running condition

The algorithm is tested with software implemented in Matlab on a 3.10GHz Windows 8 PC with 8GB RAM.

4.3 Running results

Each numerical example is performed for 10 times, and the running results are listed in Table 3. Row "minimum" means the minimum objective function value obtained within the ten times' running. "Average" is the average objective function value of the ten solutions obtained. "standard deviation" is the standard deviation for the ten objective function values. Number of optimal solutions obtained means number of minimum objective function values obtained among the ten solutions. Running time is measured in seconds.

Table 3 Running results

Experiment results	Numerical example						
Passenger number × vehicle number	10×10	10×20	20×20	20×40	30×30	30×60	100×100
Minimum	47.9	41.9	83.8	83.9	118.4	115.9	298.7
Average	47.9	41.9	84.1	85.7	119.5	116.6	304.7
Standard deviation	0	0	0.41	1.59	1.81	0.66	3.93
Number of optimal solutions obtained	10	10	4	2	4	3	1
Running time (seconds)	3.4	6.4	13.4	23.8	33.4	41.7	57.6

From Table 3, we find that the algorithm can obtain ten same solutions for the small scale problems, 10×10, and 10×20 problems. Furthermore, with the scale of problem increasing, the standard deviation still remains small. The standard deviations are all smaller than 4 for the seven numerical examples. Moreover, we can see that the algorithm can complete the computation within a reasonable time. Solutions of small scale problems (10×10, and 10×20) can be obtained within 10 seconds, and solutions of larger scale problem (100×100 problem, which is very large in practice) can be obtained in less than 1 minute.

We also generate the figures of the best solutions, which is shown in Figures 2-8. The red points represent available cars, the stars represent passengers, and the circle in the middle represents the transit hub. We can see that routing plans are reasonable and all solutions obtained are satisfactory. The cars and routes selected are reasonable and length of all routes are short. Figure 9 shows the iteration process, which indicates that the quality of the solution is improved significantly from the initial feasible solution.

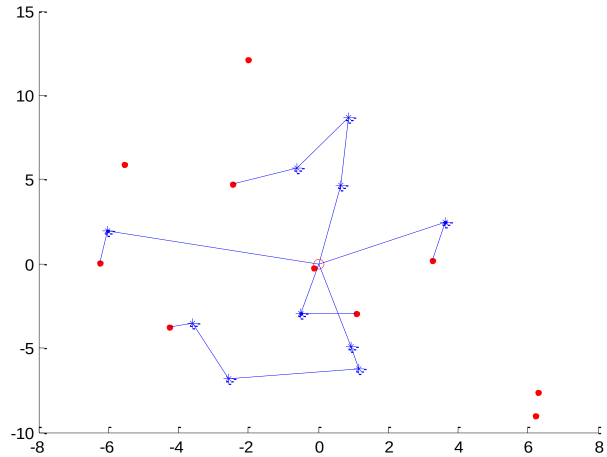


Figure 2 Best solution of passenger number 10 × car number 10 (the blue stars represent passengers, red points represent available vehicles and red circle is the train station)

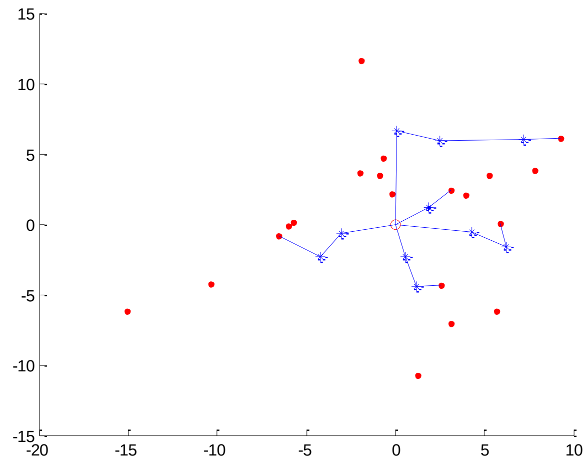


Figure 3 Best solution of passenger number 10 × car number 20

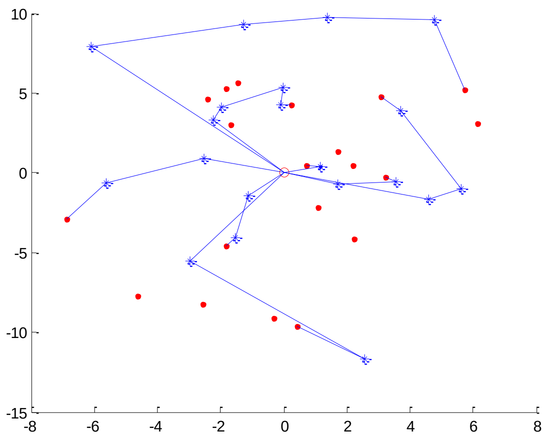


Figure 4 Best solution of passenger number 20 × car number 20

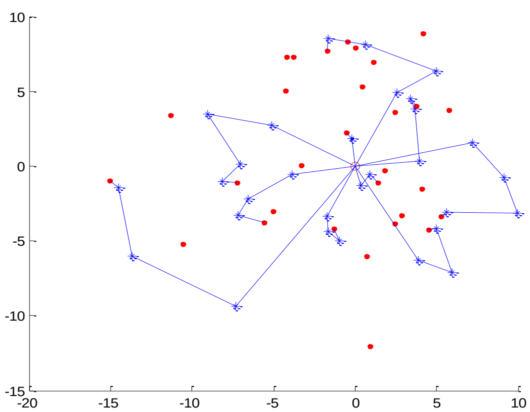


Figure 5 Best solution of passenger number 20 × car number 40

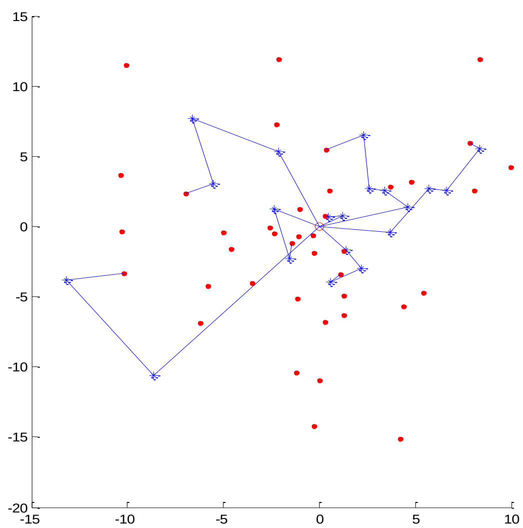


Figure 6 Best solution of passenger number 30 × car number 30

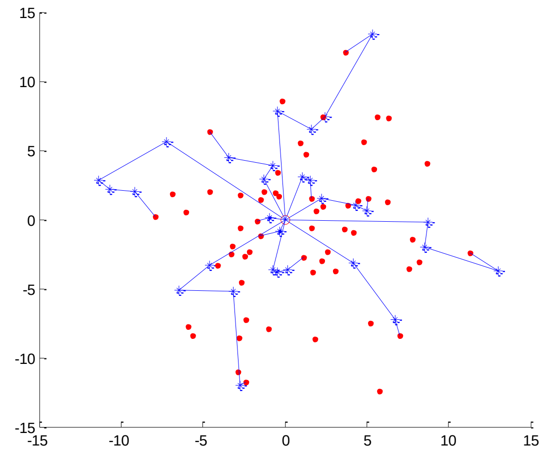


Figure 7 Best solution of passenger number 30 × car number 60

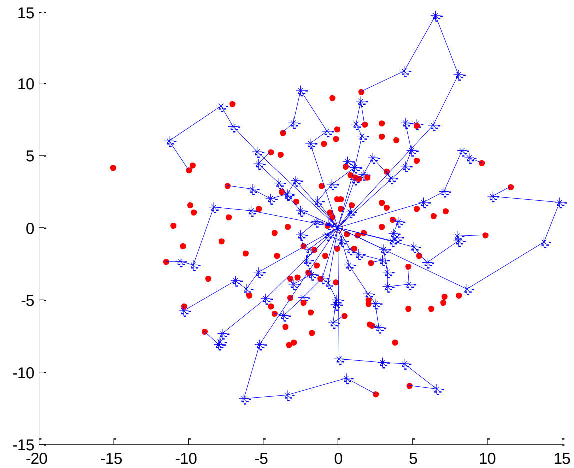


Figure 8 Best solution of passenger number 100 × car number 100

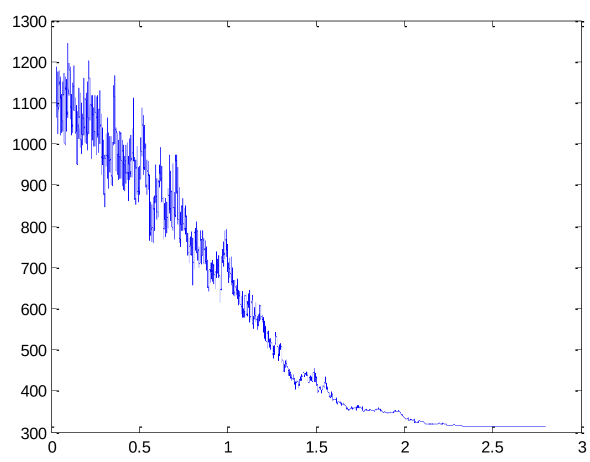


Figure 9 Iteration process

5. Conclusions

This paper studies the first-mile ridesharing problem for passengers to arrive at train station. A mathematical model is developed for the optimal passenger matching and vehicle routing to minimize total transportation cost. To achieve computational efficiency for a large-scale optimization problem, we develop an efficient metaheuristic algorithm called simulated annealing with four mixed neighborhood structures to solve this problem. Seven numerical examples with different scales are generated, and an experiment is designed to verify the effectiveness of the proposed algorithm. From the experimental results, we find that our method can obtain satisfactory solutions within very short time periods (seconds or minutes).

6. Future research

Our ongoing research includes the following:

(1) This paper only considers static occasions. In other words, we only consider the optimization problem of a specific time period, and do not consider the impact of emerging passengers on the optimization plan. In practice, the problem is dynamic, so it is necessary to develop dynamic model and the corresponding algorithm for the problem.

(2) Only a single transit hub is involved in the optimization problem. In real word, there may be more than one transit hub in a nearby region, and thus it is necessary to consider multiple transit hubs for the problem.

(3) Future work can consider another objective, which is to minimize passengers' waiting time, including the time they wait for the car coming and the time they spend in the car.

REFERENCES

[1] https://play.google.com/store/apps/details?id=com.upstartmobile.Cabulous (Flywheel)
[2] https://play.google.com/store/apps/details?id=com.ridecharge.android.taximagic (Curb)
[3] https://play.google.com/store/apps/details?id=tapgap.transit (Transit)
[4] https://play.google.com/store/apps/details?id=com.chariot.riderapp (Chariot)
[5] https://play.google.com/store/apps/details?id=com.zc.android (Zipcar)
[6] https://play.google.com/store/apps/details?id=com.car2go (Car2go)
[7] https://www.vulog.com/solution/?gclid=Cj0KEQjw4rbABRD_gfPA2-uQqroBEiQA58MNdF2GeFuh8QmRsFyrY8K_IKApr13tYmOjT_HG4pDsa1IaAi1a8P8HAQ (Volug)
[8] https://play.google.com/store/apps/details?id=spinlister.prod.app (Spinlister)
[9] https://play.google.com/store/apps/details?id=com.takescoop.android.scoopandroid (Scoop)
[10] https://play.google.com/store/apps/details?id=com.ridewith (Wazerider)
[11] https://twitter.com/BlaBlaCar?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor (BlablaCara)
[12] https://play.google.com/store/apps/details?id=com.sts.carpoolear (CarpooleAR)
[13] https://play.google.com/store/apps/details?id=com.ubercab (Uber)
[14] https://play.google.com/store/apps/details?id=me.lyft.android (Lyft)
[15] https://play.google.com/store/apps/details?id=via.rider (Via)
[16] https://play.google.com/store/apps/details?id=br.com.easytaxi (Easy-taxi)
[17] Shaheen, Susan, and Adam Cohen. "Growth in worldwide carsharing: An international comparison." *Transportation Research Record: Journal of the Transportation Research Board* 1992 (2007): 81-89.
[18] Katzev, Richard. "Car sharing: A new approach to urban transportation problems." *Analyses of Social Issues and Public Policy* 3.1 (2003): 65-86.
[19] Martin, Elliot, Susan Shaheen, and Jeffrey Lidicker. "Impact of carsharing on household vehicle holdings: Results from North American shared-use vehicle survey." *Transportation Research Record: Journal of the Transportation Research Board* 2143 (2010): 150-158.
[20] Shaheen, Susan, Stacey Guzman, and Hua Zhang. "Bikesharing in Europe, the Americas, and Asia: past, present, and future." *Transportation Research Record: Journal of the Transportation Research Board* 2143 (2010): 159-167.
[21] Vogel, Patrick, Torsten Greiser, and Dirk Christian Mattfeld. "Understanding bike-sharing systems using data mining: Exploring activity patterns." *Procedia-Social and Behavioral Sciences* 20 (2011): 514-523.
[22] DeMaio, Paul. "Bike-sharing: History, impacts, models of provision, and future." *Journal of Public Transportation* 12.4 (2009): 3.
[23] Shaheen, Susan, et al. "China's Hangzhou public bicycle: understanding early adoption and behavioral response to bikesharing." *Transportation Research Record: Journal of the Transportation Research Board* 2247 (2011): 33-41.
[24] García-Palomares, Juan Carlos, Javier Gutiérrez, and Marta Latorre. "Optimizing the location of stations in bike-sharing programs: a GIS approach." *Applied Geography* 35.1 (2012): 235-246.
[25] Schuijbroek, Jasper, Robert Hampshire, and Willem-Jan van Hoeve. "Inventory rebalancing and vehicle routing in bike sharing systems." (2013).
[26] Shaheen, Susan A. "Public Bikesharing in North America: Early Operator and User Understanding, MTI Report 11-19." (2012).
[27] Midgley, Peter. "The role of smart bike-sharing systems in urban mobility." *Journeys* 2 (2009): 23-31.
[28] Dell'Amico, Mauro, et al. "The bike sharing rebalancing problem: Mathematical formulations and benchmark

instances." *Omega* 45 (2014): 7-19.
[29] Shaheen, Susan, Adam Cohen, and Elliot Martin. "Public bikesharing in North America: early operator understanding and emerging trends." *Transportation Research Record: Journal of the Transportation Research Board* 2387 (2013): 83-92.
[30] Rayle, Lisa, et al. *App-Based, On-Demand Ride Services: Comparing Taxi and Ridesourcing Trips and User Characteristics in San Francisco University of California Transportation Center (UCTC)*. UCTC-FR-2014-08, 2014.
[31] Rayle, Lisa, et al. "Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco." *Transport Policy* 45 (2016): 168-178.
[32] Zhen, Chen. *Impact of Ride-Sourcing Services on Travel Habits and Transportation Planning*. Diss. University of Pittsburgh, 2015.
[33] Zha, Liteng, Yafeng Yin, and Hai Yang. "Economic analysis of ride-sourcing markets." *Transportation Research Part C: Emerging Technologies* 71 (2016): 249-266.
[34] Ngo, Victor D. *Transportation network companies and the ridesourcing industry: a review of impacts and emerging regulatory frameworks for Uber*. Diss. University of British Columbia, 2015.
[35] Teal, R.F., 1987. Carpooling: who, how and why. Transportation Research Part A 21 (3), 203–214.
[36] Ferguson, Erik. "The rise and fall of the American carpool: 1970–1990." *Transportation* 24.4 (1997): 349-376.
[37] Morency, Catherine. "The ambivalence of ridesharing." *Transportation* 34.2 (2007): 239-253.
[38] Yang, Hai, and Hai-Jun Huang. "Carpooling and congestion pricing in a multilane highway with high-occupancy-vehicle lanes." *Transportation Research Part A: Policy and Practice* 33.2 (1999): 139-155.
[39] Giuliano, Genevieve, Douglas W. Levine, and Roger F. Teal. "Impact of high occupancy vehicle lanes on carpooling behavior." *Transportation* 17.2 (1990): 159-177.
[40] Galland, Stephane, et al. "Multi-agent simulation of individual mobility behavior in carpooling." *Transportation Research Part C: Emerging Technologies* 45 (2014): 83-98.
[41] Dewan, Kum Kum, and Israr Ahmad. "Carpooling: A Step To Reduce Congestion." *Engineering Letters* 14.1 (2007).
[42] Knapen, Luk, et al. "Exploiting graph-theoretic tools for matching in carpooling applications." *Journal of Ambient Intelligence and Humanized Computing* 5.3 (2014): 393-407.
[43] Abrahamse, Wokje, and Michael Keall. "Effectiveness of a web-based intervention to encourage carpooling to work: A case study of Wellington, New Zealand." *Transport policy* 21 (2012): 45-51.
[44] Vanoutrive, Thomas, et al. "What determines carpooling to workplaces in Belgium: location, organisation, or promotion?." *Journal of transport geography* 22 (2012): 77-86.
[45] Dobrosielski, John, et al. "Carpool. umd: community carpooling." *CHI'07 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2007.
[46] Chen, Yu-Tso, and Chen-Heng Hsu. "Improve the carpooling applications with using a social community based travel cost reduction mechanism." *International Journal of Social Science and Humanity* 3.2 (2013): 87.
[47] Zhang, Desheng, et al. "A carpooling recommendation system for taxicab services." *IEEE Transactions on Emerging Topics in Computing* 2.3 (2014): 254-266.
[48] Di Martino, Sergio, et al. "A Matching-Algorithm based on the Cloud and Positioning Systems to Improve Carpooling." *DMS*. 2011.
[49] Collotta, Mario, et al. "A novel trust based algorithm for carpooling transportation systems." *Energy Conference and Exhibition (ENERGYCON), 2012 IEEE International*. IEEE, 2012.
[50] Massaro, Dominic W., et al. "Carpoolnow-Just-in-Time Carpooling without Elaborate Preplanning." *WEBIST*. 2009.
[51] Xing, Xin, et al. "SMIZE: a spontaneous ride-sharing system for individual urban transit." *German Conference on Multiagent System Technologies*. Springer Berlin Heidelberg, 2009.
[52] Dorinson, Diana, et al. "Flexible carpooling: exploratory study." *Institute of Transportation Studies* (2009).
[53] LeBlanc, David E. *Slugging: The commuting alternative for Washington, DC*. Forel Pub., 1999.
[54] Spielberg, Frank, and Phillip Shapiro. "Mating habits of slugs: Dynamic carpool formation in the I-95/I-395 corridor of Northern Virginia." *Transportation Research Record: Journal of the Transportation Research Board* 1711 (2000): 31-38.
[55] Burris, Mark W., and Justin R. Winn. "Slugging in Houston—casual carpool passenger characteristics." *Journal of Public Transportation* 9.5 (2006): 2.
[56] Kelly, Kalon L. "Casual carpooling-enhanced." *Journal of Public Transportation* 10.4 (2007): 6.
[57] Di Febbraro, A., E. Gattorna, and N. Sacco. "Optimization of dynamic ridesharing systems." *Transportation Research Record: Journal of the Transportation Research Board* 2359 (2013): 44-50.
[58] Ma, Shuo, Yu Zheng, and Ouri Wolfson. "T-share: A large-scale dynamic taxi ridesharing service." *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013.
[59] Herbawi, Wesam, and Michael Weber. "The ridematching problem with time windows in dynamic ridesharing: a model and a genetic algorithm." *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012.
[60] Tian, Charles, et al. "Noah: a dynamic ridesharing

system." *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 2013.

[61] Agatz, Niels, et al. "Optimization for dynamic ride-sharing: A review." *European Journal of Operational Research* 223.2 (2012): 295-303.

[62] Agatz, Niels AH, et al. "Dynamic ride-sharing: A simulation study in metro Atlanta." *Transportation Research Part B: Methodological* 45.9 (2011): 1450-1464.

[63] Lee, Alan, and Martin Savelsbergh. "Dynamic ridesharing: Is there a role for dedicated drivers?." *Transportation Research Part B: Methodological* 81 (2015): 483-497.

[64] Kleiner, Alexander, Bernhard Nebel, and V. Ziparo. "A mechanism for dynamic ride sharing based on parallel auctions." (2011): 266-272.

[65] Herbawi, Wesam, and Michael Weber. "Evolutionary multiobjective route planning in dynamic multi-hop ridesharing." *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer Berlin Heidelberg, 2011.

[66] Stach, Christoph. "Saving time, money and the environment-vHike a dynamic ride-sharing service for mobile devices." *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. IEEE, 2011.

[67] Amey, Andrew, John Attanucci, and Rabi Mishalani. "Real-time ridesharing: opportunities and challenges in using mobile phone technology to improve rideshare services." *Transportation Research Record: Journal of the Transportation Research Board* 2217 (2011): 103-110.

[68] Agatz, Niels, et al. "Sustainable passenger transportation: Dynamic ride-sharing." (2010).

[69] Furuhata, Masabumi, et al. "Ridesharing: The state-of-the-art and future directions." *Transportation Research Part B: Methodological* 57 (2013): 28-46.

[70] Golden, Bruce L., Subramanian Raghavan, and Edward A. Wasil, eds. *The vehicle routing problem: latest advances and new challenges*. Vol. 43. Springer Science & Business Media, 2008.

[71] Gendreau, Michel, Alain Hertz, and Gilbert Laporte. "A tabu search heuristic for the vehicle routing problem." *Management science* 40.10 (1994): 1276-1290.

[72] Osman, Ibrahim Hassan. "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem." *Annals of operations research*41.4 (1993): 421-451.

[73] Prins, Christian. "A simple and effective evolutionary algorithm for the vehicle routing problem." *Computers & Operations Research* 31.12 (2004): 1985-2002.

[74] Geng X, Chen Z, Yang W, et al. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search[J]. Applied Soft Computing, 2011, 11(4): 3680-3689.