Journal of Management Science and Engineering 5 (2020) 264–286

Science Press

Contents lists available at ScienceDirect

Journal of Management Science and Engineering

journal homepage: www.keaipublishing.com/en/journals/journal-of-management-science-and-engineering/

Who is more likely to get a ride and where is easier to be picked up in ride-sharing mode?

Yue Yang, Qiong Tian*, Yuqing Wang

School of Economics and Management, Beihang University, Beijing, 100191, China

A R T I C L E I N F O

Article history:
Available online 23 September 2020

Keywords:
Ride-matching
Heuristic algorithm
Passenger ratio
Chengdu data

A B S T R A C T

The ubiquity of information and communication technology (ICT) and application of global positioning system (GPS) enabled cell phones provide new opportunities to implement ride-sharing in many ride-hailing platforms, where matching proposals with multiple riders are established on very short notice. In this paper, the travelers joining in the ridesharing are assumed to be homogeneous in terms of having their own vehicles. When they have announced their travel requests, the ride-sharing platform will check whether they can be picked up by any other travelers. If failed, they will drive by themselves and become a driver who would like to pick up other passengers in the system. To solve this problem, the ride-matching problem is formulated as a set-partitioning problem and a so-called ordered greedy (OG) method is presented to get the approximately optimum under the large-scale circumstance. The results of simulation examples prove that the proposed method can achieve a reasonable matching result through Cplex within a few seconds but at most 3.8% worse than the exact optimum. Furthermore, several interesting results are also found via simulating generated data and the real-world data of Chengdu in China. In simulation experiments, with a higher level of demand density, the easiest place to find a ride is not in the center but a ring close by it, which is determined by traffic flows, OD distance and vehicles' utilization. As a contrast, the optimal strategy for participants to be a rider is going to other specific regions rather than staying in the city center in real-world experiments.

1. Introduction

With the growing concern for traffic congestion, fuel shortage and environmental pollution, people alternatively using transportation modes in a flexible way. Ridesharing which is for the purpose of taking the great use of the existing passenger-movement capacity on the vehicles has been widely adopted by citizens. While sharing a vehicle, individual travelers split travel costs such as gas, toll, and parking fees with people who have similar itineraries and time schedules and thus saving part of their travel costs. Aside from participants, ridesharing is also beneficial to the society by mitigating traffic congestion, conserving fuel, and leading to a low carbon economy through cutting down motor vehicle exhaust. (Ferguson, 1997; Kelley, 2007; Morency, 2007; Chan et al., 2012).

* Corresponding author.
E-mail address: tianqiong@buaa.edu.cn (Q. Tian).

https://doi.org/10.1016/j.jmse.2020.09.003

Over the past few years, traditional taxi industries have experienced and witnessed radical changes. One representative is these widely used online ride-hailing platforms, such as Uber, Lyft and Didi Chuxing, enable idle taxi drivers to be automatically matched with passengers without spending amount of time searching on the street. Specifically, these platforms would collect all the idle drivers and the new coming requests during each short time slot (say one or 2 s), and then make decisions based on a combinatorial optimization algorithm. Under this circumstance, the taxis' ride-matching optimization problems of E-hailing platforms naturally fall into the category of bipartite matching problems, which have been widely discussed in some researches (Agatz et al., 2012; Zhan et al., 2016; Xu et al., 2018). Some well-known solution approaches for maximum bipartite matching has been employed to solve these problems in both academic researches and industrial application. Compared to taxi-matching problem, ride-sharing matching problems will be more complex and intractable due to the nature of service mode. To begin with, there will be more than one rider picked up in a ridesharing trip, to reduce the operating cost and boost total revenue, the platform needs to schedule the route and determine the order of serving riders. In addition, the participants in ridesharing do not work as agency employees of the platform but have their own vehicle and would like to share the ride with others with limited detour. In this case, the platforms will determine their roles (rider or driver) and provide them with schedules to satisfy their demands. The above facts show that the matching process in ridesharing systems is similar to traditional carpooling and dial-a-ride problem (DARP). Although carpooling is a regular, advanced, and cost-effective means of transportation (Ferguson, 1997; Morency, 2007), it does not accommodate unexpected changes of schedule. By contrast, DARP provides shared seats to responsd to the advanced requests between any origins and destinations within specific depots (Berbeglia et al., 2010), whereas in a ridesharing system each driver may have a unique and flexible origin–destination (OD) pair.

From the perspective of travelers, overly long waiting time is essentially caused by the imbalance and mismatching among travelers, in other words, travelers' personal preferences, including travel distance and origin locations, will exert a critical effect on the performance of ride-sharing system. Upon tackling the inefficiency in this stage, Santi et al. (2014) suggested a mathematical framework for the understanding of the tradeoff between collective benefits of sharing and individual passenger discomfort is lacking. Stiglic et al. (2016) demonstrated the time flexibility of each participant in ridesharing will result in different efficiencies of the system. Long et al. (2018) also proved that the serval factors, including unit cost of driving, travelers' VOTs, travel time uncertainty, have significant impacts on the performance of the proposed ride-sharing system with travel time uncertainty. Nonetheless, compared to these aforementioned factors, the spatial distribution of participants' origins also has intense impacts on the outcomes of ride-sharing system, while there are few studies focus on this topic.

To ameliorate the matching efficiency in ridesharing, in this paper we concentrate our attention on figure out the optimal strategies for each traveler to find a ride as soon as possible. Firstly, we consider a ride-sharing system setting in which all the trips are known in advance and suppose each trip owns a vehicle. With information of location and time requirements provided by each participant, the ride-sharing system automatically matchs potential drivers and passengers over time via ordered greedy (OG) algorithm. Secondly, according to the spatial distribution of "passenger ratio" in aforementioned system, we discover several factors affecting the matching efficiency. Finally, combining with real-world identification in Chengdu, we figure out the optimal choices of origins for each participant based on extensive simulations. The main contributions of this paper can be summarized as follows:

- A new heuristic method is introduced to solve the ride-sharing matching problem, and results prove that our method achieves remarkable performance (about 3.8% worse than the verified exact optimum which is solved by Cplex within a few seconds).
- Based on the outcomes of ridesharing system, several factors affecting matching efficiency have been discovered to offer recommendations to ridesharing participants to find a ride but also help the platforms to improve the ride-sharing service.
- Comprehensive numerical experiments are also conducted based on simulation data and the real-world taxi data to identify who is more likely to be a passenger and where is easier to find a ride in ridesharing.

The rest of paper is organized as follows. In Section 2, we give a review of the related works refer to ride-sharing models and ride-matching methods. Then we propose a reduction mechanism and set-partitioning model in Section 3. In Section 4, the solution algorithm is developed. And in order to identify the performance of proposed algorithm and further explore that who and in where is easier to be a passenger in ride-sharing, extensive experiments are conducted in Section 5. Finally, a conclusive discussion is presented in Section 6.

2. Literature review

2.1. Shared taxi, carpooling and dial-a-ride problem

Several types of ridesharing systems have been proposed in the research literature and many of them are being used in Taxi-sharing, Carpooling and Dial-a-Ride problem (DARP).

Taxis are a private mode of demand-responsive transportation alternative. Along with higher cost, the door-to-door transportation provided by taxi is not affordable for everyone. While nowadays, shared taxi services make it possible for passengers to cut down the cost of the journey by sharing their trips and potentially result in a reduction in the total miles traveled in the network. In many researches, lots of different shared-use mobility services have been designed. Flexible itinerary transit systems (Quadrifoglio et al., 2008; Li and Quadrifoglio, 2010; Qiu et al., 2014), and High-Coverage Point-to-Point Transit (HCPPT) (Cortés and Jayakrishnan, 2002) are a few examples.

Carpooling is another concept of using shared cars by persons with similar travel needs who decide to carry out common car journeys between certain origins and destinations. The carpooling problem can be divided into two main types in practice: the traditional carpooling problem and the flexible (casual) carpooling problem. Traditional carpooling requires a long-term commitment among two or more people to travel together on recurring trips for a particular purpose, often for traveling to work (Baldacci et al., 2004; Wolfler et al., 2004). And the other type is characterized by no prearrangements or fixed schedules for matching drivers and passengers. Example of flexible carpooling is showed in the Washington D.C. area, where the participants are free of charge (LeBlanc, 1999; Spielberg and Shapiro, 2000), and casual carpooling is in San Francisco Bay Area and Houston, with a fixed-price for each rideshare itinerary (Burris and Winn, 2006; Kelley, 2007).

Dial-a-ride services are offered in the context of demand responsive transportation. Users are transported from a specific origin (e.g., from home) to a specific destination (e.g., to a medical facility). Typically, a user has two related transport requests on a single day: an outbound request from home to the desired destination and an inbound request, i.e., the return trip home. DARP is a generalization of vehicle routing problem with picking-up and delivery under a time window (VRPPDTW), where people are transported instead of goods. Usually, DARP optimizes the pick-up and delivery of passengers in special settings including door-to-door transportation and is commonly used in para-transit systems or shuttle-like services. In the basic form of DARP, vehicles and riders are assumed to be homogeneous. All vehicles start from and return to the same depot (Savelsbergh et al., 1995; Cordeau et al., 2007; Mahmoudi et al., 2016).

In recent years, traditional Dial-a-ride problems have experienced and witnessed radical changes. One representative is these widely used online ride-hailing platforms, which enable idle taxi drivers to be automatically matched with passengers without spending amount of time searching on the street. To facilitate the implementation of ride-hailing application, Alonso-Mora et al. (2017) proposed a real-time multitask framework to tackle the large-scale real-world taxi assignment. Given a collection of trips, Vazifeh et al. (2018) also provide a network-based solution to determine the minimum number of vehicles needed to satisfy all the trips in the city.

2.2. Ride-matching problem

With effective use of new communication modes including mobile technology and global positioning system (GPS), the ride-matching problem was concerned by scholars in recent years (Furuhata et al., 2013; Agatz et al., 2012). Ride-matching problems share some of the characteristics of the more advanced DARPs and VRPPDTW, such as multiple depots, heterogeneous vehicles and passengers. Drivers in ridesharing systems are traveling to perform activities and have distinct origin and destination locations (multi-depot), different vehicle capacities (heterogeneity), and rather narrow travel time windows. Generally, these factors may lead the matching in ridesharing systems into a spatiotemporally sparse problem. In its simplest form, the ride-matching problem matches a single rider to each driver. This can be modeled as a maximum-weight bipartite matching problem that minimizes the total rideshare cost (Agatz et al., 2011).

There are new trends in modeling the ride-matching problem recently.

(1) Variations of ridesharing through introducing meeting points: Stiglic et al. (2015) introduced a ridesharing system with meeting points instead of pickups or drop-offs at a series of points and validated the efficiency in terms of the number of shared trips and system wide travel distance savings. This idea was first mentioned by Kaan and Olinick (2013) while considering vanpooling. All the commuters gathered at one park-and-ride location and ride to another, but the destination was fixed by then, which made the problem simpler. The application of meeting points can also be found in school bus transportation, Schittekat et al. (2013) gave a comprehensive consideration of the station and itinerary choices by means of a new hybrid parameter-free metaheuristic. More relevant to our study, Varone and Aissat (2015) narrowed down to individual riders. Setting limitations to the meeting points on each way and keeping the detour constraint in mind. Aïvodji et al. (2016) considered the cost of ridesharing user privacy while setting meeting points, and developed a privacy preserving procedure to deploy meeting points without sacrificing the ridesharing usage.

(2) Scaling to large system through partitions and developing three types of partitions, geography based, user based, and model based: Ma et al. (2015) and Pelzer et al. (2015) proposed to partition the study area into small homogeneous subregions based on predefined grids and road network topology, by which real-time sharing requests could be well arranged under time, capacity and cost limitations. Apart from geography-based matching and scheduling, personal preference, the number of passengers for instance, was also considered by Lyu et al. (2017) as a basic condition while grouping. Moreover, Masoud et al. (2017) located users and potential available drivers in a narrow search space and model in each subset to reduce problem size. As for model-based development, Bent and Van Hentenryck (2006) creatively divided the matching process into two stages, minimized the fleet size and total distance in turn. As a result, larger instances could be solved faster through a population-based metaheuristic (Cherkesly et al., 2015). Also, for neighborhood search (VNS), Parragh et al. (2010) added a competitive variable into heuristic method. Hosni et al.

(2014) applied the Lagrange decomposition approach to reduce the mixed integer ride-matching problem into smaller problems and solved them separately.

3. Model formulation

In this section, we consider a ride-sharing platform serves for a particular metropolitan area, where a sequence of travel requests generated over time from potential participants. All the participants sending their trip request to the ride-sharing platform imply their willingness to share the ride with others. Every trip request s has an earliest departure time, $t_{o_s}^{ED}$, from her/his origin o_s and a latest arrival time, $t_{d_s}^{LA}$, at her/his destination d_s. The participants can decide to either be drivers, who serve as many as possible passengers considering her/his time and the vehicle capacity constraints, or be a passenger if any driver can pick up and drop off in her/his time window $[t_{o_s}^{ED}, t_{d_s}^{LA}]$. Let $S = P \cup D$ denotes the set of participants, which is divided into two sub-sets, namely P, represents the set of passengers who are looking for rides and D, stands for the set of drivers who are willing to provide rides along the journey. Each vehicle is homogeneous in terms of the same capacity CP and constant travel speed v. It is assumed that all the travel requests are known in advance before the execution of the ride-matching process. The platform, for some purpose (for instance minimizing the total vehicle-km) decides how to sort the travel requests into itineraries. Let's considering a typical participant s who acts as a driver. After a series of matching processes, driver s totally serves n passengers. Let $R(s)$ denotes the itinerary of driver s, which includes $2(n+1)$ nodes:

$$R(s) = \{o_s, \ldots, o_{s_j}, \ldots, d_{s_j}, \ldots, d_s\}$$

The itinerary $R(s)$ is an ordered sequence which consists of origins and destinations not only of the driver s but also of other passengers sharing the ride with s. To focus on the ride matching process and without any loss of generality, the network structure is ignored in the model and the vehicle travel along straight line between successive nodes. Due to the simplicity, the rest of the paper will use r_i to represent the ith node in the itinerary R.

For any itinerary $R = \{r_1, \ldots, r_i, \ldots r_{2n+2}\}$, the number of occupied seats is denoted as a vector $M = \{m_1, \ldots, m_i, \ldots m_{2n+2}\}$, where m_i is the number of occupied seats of itinerary R after visiting the i th node along the itinerary. In addition, because each travel request has its own earliest departure time and latest arrival time, for each node of an itinerary there are four types of time constraint. Let EA, ED, LA and respectively denote the sets of the earliest arrival time, the earliest departure time, the latest arrival time and the latest departure time for the itinerary.

$$EA = \{ea_1, \ldots, ea_i, \ldots, ea_{2n+2}\}$$

$$ED = \{ed_1, \ldots, ed_i, \ldots, ed_{2n+2}\}$$

$$LA = \{la_1, \ldots, la_i, \ldots, la_{2n+2}\}$$

$$LD = \{ld_1, \ldots, ld_i, \ldots, ld_{2n+2}\}$$

Obviously, the earliest departure time ed_1 of the first node is $t_{o_{s_1}}^{ED}$ and the latest arrival time la_{2n+2} of the final node is $t_{d_{s_1}}^{LA}$. It is essential to note that there are no arriving time existing at o_{s_1} and no departure time existing at d_{s_1}, and for simplicity, we can assume that $ea_1 = 0, ld_{2n+2} = \infty$. Thus, EA_r, ED_r LA_r, LD_r can be derived from:

$$ea_i = ed_{i-1} + t_{r_{i-1},r_i} \tag{1}$$

$$ed_i = ea_i + w_{r_i} \tag{2}$$

$$ld_i = la_{i+1} - t_{r_i,r_{i+1}} \tag{3}$$

$$la_i = ld_i - a_{r_i} \tag{4}$$

where t_{r_{i-1},r_i} is the travel time from r_{i-1} to r_i, and w_{r_i}, a_{r_i} indicate the probable waiting time and probable advance time at the node r_i, respectively.

As intuitively depicted in Fig. 1, it is compulsory for the driver to wait at r_i if ea_i is earlier than $t_{r_i}^{ED}$, that is, the time gap w_{r_i} between $t_{r_i}^{ED}$ and ea_i ($t_{r_i}^{ED} \geq ea_i$) is the waiting duration when driver arrives at r_i. Meanwhile, compared to latest departure time ld_i, the driver has to reach r_i no later than $t_{r_i}^{LA}$. In other words, the gap a_{r_i} between $t_{r_i}^{LA}$ and ld_i ($t_{r_i}^{LA} \leq ld_i$) is length of time to calculate the latest arrival time. Specifically, we should note that waiting time w_{r_i} will only exist at origin node, while advance time a_{r_i} will happen only at destination node. As a consequence, w_{r_i}, a_{r_i} can be represented as:

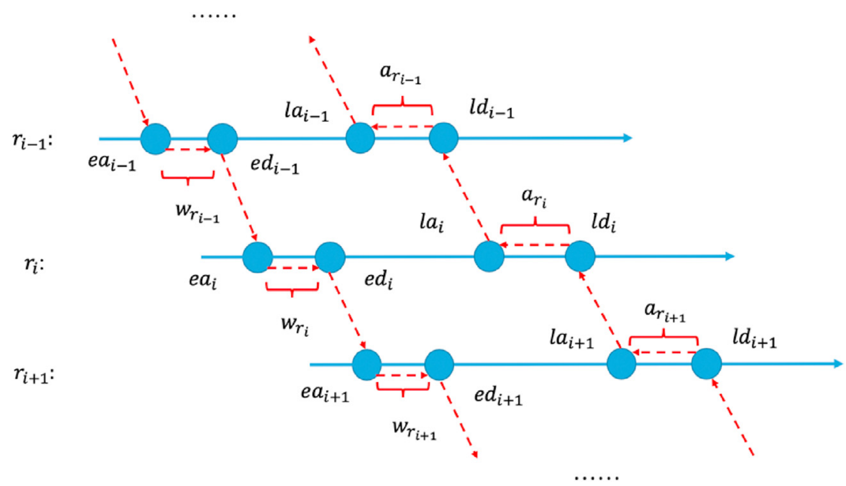

Fig. 1. The procedure of calculating EA_r, ED_r LA_r, LD_r.

$$w_{r_i} = \begin{cases} 0 & \text{if } r_i \text{ is destiontion} \\ \max\left(0, t_{r_i}^{ED} - ea_i\right) & \text{if } r_i \text{ is origin} \end{cases} \tag{5}$$

$$a_{r_i} = \begin{cases} 0 & \text{if } r_i \text{ is origin} \\ \max\left(0, ld_i - t_{r_i}^{LA}\right) & \text{if } r_i \text{ is destination} \end{cases} \tag{6}$$

Specifically, the equation (1) represents the earliest arrival time ea_i of r_i is generated from ed_{i-1} of previous node r_{i-1}, and the equation (2) indicates the earliest departure time ed_i of r_i is derived from ea_i and w_{r_i}. Moreover, the equation (3) shows the latest departure time ld_i of r_i is calculated from la_{i+1} of latter node r_{i+1}, finally the equation (4) indicates the latest arrival time la_i of la_i is deduced from ld_i and a_{r_i}. The detailed procedure of acquiring four sets is depicted in Fig. 1:

After constructing the capacity and time constraints, we realize that the location r_i and r_{i+1}, the earliest departure time ed_i and the latest arrival time la_{i+1} can be used to define a serving region in the network, which is the so-called time-space range (Masoud et al., 2017). In this paper, we put all the spatial nodes into two-dimension coordinate, that is, each node in itinerary R will own horizontal and vertical coordinates. Consequently, the time-space range will be formed as an ellipse, whose focal points o are two adjacent nodes r_i and r_{i+1}. In addition, the length of the major axes is the Euclidean distance between r_i and r_{i+1}, and the transverse diameter is an upper bound of the travel (detour) distance between r_i and r_{i+1} (see Fig. 2). According to R, LA ED, we can easily obtain the time-space range set $\mathbb{E}$ of the itinerary R, which states the detour (serving) range of driver s_1 without conflicting with the time constraints. See as follows:

$$\mathbb{E} = \{e_1, \ldots, e_i, \ldots, e_{2n+1}\}, \forall i \in \{1, \ldots, 2n+1\}$$

where e_i, $\forall i \in \{1, \ldots, 2n+1\}$ is the detour range when the driver starts from r_i. Obviously, if a new node can be covered by any of subrange e_i in $\mathbb{E}$, hence this node can be visited by the itinerary R without violating its time constraints.

Note that the range of e_i is also determined by the new coming node, assume that there is a new coming node (x, y, t) with a coordinate (x, y) and a time buffer t, which represents a probable waiting time or a probable advance time at this node. In particular, since the new node can be origin or destination, for taking time characters of the current node into consideration, here we introduce a different notation t, which is equivalent to a probable waiting time if the node is origin, otherwise a probable advance time. Therefore, the subrange e_i covering the new coming node can be defined as the following form:

$$e_i = \left\{(x, y, t) \middle| \left(\sqrt{(x - x_{r_i})^2 + (y - y_{r_i})^2} + \sqrt{(x - x_{r_{i+1}})^2 + (y - y_{r_{i+1}})^2}\right) \leq (la_{i+1} - ed_i - t) \times v\right\} \tag{7}$$

where (x_{r_i}, y_{r_i}), $(x_{r_{i+1}}, y_{r_{i+1}})$ are the two-dimension coordinates of the node r_i and r_{i+1}, respectively. In particular, the time buffer t defined here will be further discussed in the next section.

To mathematically model the ride-matching problem, we use three sets of decision variables, as defined in (7)-(9). And the mathematical notations used in the paper are summarized in Table 1.

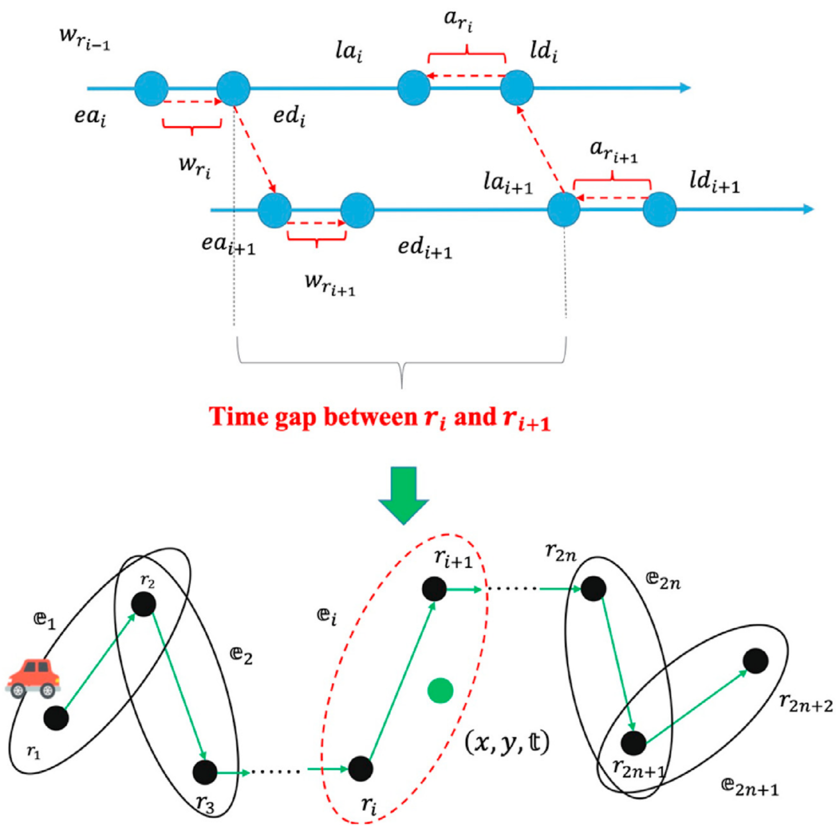

Fig. 2. The example of acquiring the time-space range set $\mathbb{E}$.

Table 1
Notations.

The Description of Notations	
D	The set of ridesharing drivers.
P	The set of passengers.
$\mathbb{R}$	The set of candidate routes.
R	A feasible itinerary and $R \in \mathbb{R}$.
M	The occupied seats vector of the itinerary R.
ED	The earliest departure time vector of the itinerary R.
EA	The earliest arrival time vector of the itinerary R.
LD	The latest departure time vector of the itinerary R.
LA	The latest arrival time vector of the itinerary R.
$\mathbb{E}$	The time-space range set of the itinerary R.
CP	The capacity of a vehicle, which is constant in this paper.
$t^d_{r_i}$	The departure time of node r_i in the itinerary R.
$t^a_{r_i}$	The arrival time of node r_i in the itinerary R.
t_{o_p}	The probable waiting time at origin o_p of passenger p.
t_{d_p}	The compulsory advanced arriving gap at origin d_p of passenger p.
c_R	The cost saving of the itinerary R.
P_R	All the passengers joining the itinerary R.
δ_R	The total time buffer of the itinerary R.
μ^s_R	A binary decision variable.
$u^s_{r_i, r_{i+1}}$	A binary decision variable.
$z^{s,p}_R$	A binary decision variable.
λ_R	A binary decision variable.

$$\mu_R^s = \begin{cases} 1, \text{Driver } s \text{ takes the itinerary } R \\ 0, \text{ Otherwise} \end{cases} \tag{8}$$

$$u_{r_i,r_{i+1}}^s = \begin{cases} 1, \text{Driver } s \text{ travels on the arc from } r_i \text{ to } r_{i+1} \text{ in } R \\ 0, \text{ Otherwise} \end{cases} \tag{9}$$

$$z_R^{s,p} = \begin{cases} 1, \text{Passenger } p \text{ take a ride in } R \text{ driven by } s \\ 0, \text{ Otherwise} \end{cases} \tag{10}$$

Therefore, the ridesharing matching problem can be modeled as follows:

$$MaxDistance_{saving} = \sum_{R\in\mathbb{R}} \left(c_R \times \sum_{s\in D} \mu_R^s \right) \tag{11}$$

Subject to:

$$\sum_{r_i=o_s} u_{r_i,r_{i+1}}^s - \sum_{r_j=o_s} u_{r_{j-1},r_j}^s = \mu_R^s, \ \forall s\in D, \forall r_i, r_j \in R \tag{12}$$

$$\sum_{r_j=d_s} u_{r_{j-1},r_j}^s - \sum_{r_i=d_s} u_{r_i,r_{i+1}}^s = \mu_R^s, \ \forall s\in D, \forall r_i, r_j \in R \tag{13}$$

$$\sum_{r_{i+1}} u_{r_i,r_{i+1}}^s - \sum_{r_{i+1}} u_{r_{i+1},r_{i+2}}^s = 0, \ \forall s \in D, \forall r_{i+1} \in R \setminus \{o_s, d_s\} \tag{14}$$

$$\sum_{r_i=o_p} u_{r_i,r_{i+1}}^s - \sum_{r_j=o_p} u_{r_{j-1},r_j}^s = z_R^{s,p}, \ \forall s\in D, \forall p\in P, \forall r_i, r_j\in R, \forall R\in\mathbb{R} \tag{15}$$

$$\sum_{r_j=d_p} u_{r_{j-1},r_j}^s - \sum_{r_i=d_p} u_{r_i,r_{i+1}}^s = z_R^{s,p}, \ \forall s\in D, \forall p\in P, \forall r_i, r_j\in R, \forall R\in\mathbb{R} \tag{16}$$

$$ed_i \le t_{r_i}^d \le ld_i, \forall r_i \in R \tag{17}$$

$$ea_i \le t_{r_i}^a \le la_i, \forall r_i \in R \tag{18}$$

$$m_i \le CP, \forall r_i \in R \tag{19}$$

$$z_R^{s,p} \le \mu_R^s, \forall s\in D, \forall p\in P, \forall R\in\mathbb{R} \tag{20}$$

$$\mu_R^s \le \sum_{r_i} u_{r_i,r_{i+1}}^s, \ \forall s\in D, \forall r_i\in R, \forall R\in\mathbb{R} \tag{21}$$

Equation (11) presents the objective function of the problem. A ride-matching problem may have various objectives, ranging from maximizing profits to minimizing the total system-wide vehicle-miles. This objective can vary among the nature of the agency who is managing the system (public or private). If the system is private and operates for profit, the platform may pursue to maximize the number of participants because revenues are linked to the number of successful ride-sharing arrangements. However, public systems may have a societal objective like minimal system-wide vehicle-miles, which is important from a societal point of view since it helps to reduce pollution and congestion. Hence, in this paper we suppose that the ride-sharing system attempts to minimize a system-wide vehicle-miles driven by all potential participants traveling to their destinations, specifically, this objective also coincides with maximizing total travel distance savings.

The sets of constraints that define the ridesharing system are presented in (12) -(21). Constraint sets (12)–(14) route drivers in the network. Constraint set (12) directs drivers in set D out of their origin o_s, and (13) ensures that they end their trips at their destination d_s. Moreover, constraint set (14) is for flow conservation, which enforces that a driver entering a passenger's origin or destination, will exit this node and continue his trip. Constraint sets (15)–(16) route riders in the network and are analogous to (12)–(13), except for a small variation. While the optimization problem generates itineraries for all drivers, matched or not, this is not the case for riders. Only riders who are successfully matched will receive itineraries. This difference is reflected in the formulation by replacing 1 on the ride hand side of constraint sets (12) and (13) by $z_R^{s,p}$ in constraint sets (15) and (16). Constraints (17)–(18) guarantee the departure time $t_{r_i}^d$ should fall into time windows $[ed_i, ld_i]$ and arrival time $t_{r_i}^a$ also need to satisfy time windows $[ea_i, la_i]$, and Constraint (19) ensure occupied seats does not exceed the vehicle capacity. Constraint sets (20) and (21) register drivers who contribute to each passenger's itinerary. Finally, all decision variables of the problem defined in (8)-(10) are binary variables.

In particular, c_R in this paper indicates the travel distance saving of itinerary R, which is defined as:

$$c_R = \sum_{p \in P_R} dis\left(R_p^{solo}\right) + dis\left(R_s^{solo}\right) - dis(R) \tag{22}$$

where $dis(\bullet)$ indicates the function calculating the travel distance (Euclidean distance) of the given itinerary, and R_s^{solo} represents the itinerary when the driver s drives alone.

3.1. Set partitioning formulation

Therefore, based on the feasible itineraries set $\mathbb{R}$ satisfying constraints (12–19), the ride-sharing assignment problem in equation (11) can be converted to a set-partitioning problem as follows:

$$MaxDistance_{saving} = \sum_{R \in \mathbb{R}} c_R \times \lambda_R$$

Subject to:

$$\sum_{R \in \mathbb{R}_p} \lambda_R = 1 \ \forall p \in P \tag{23}$$

$$\sum_{R \in \mathbb{R}_s} \lambda_R = 1 \ \forall s \in D \tag{24}$$

$$\lambda_R \in \{0, 1\}, \forall R \in \mathbb{R} \tag{25}$$

where λ_R is the binary decision variable equals to 1 if the itinerary R is included in the final schedule $\mathbb{R}^*$ and 0 if not. As shown in equation (23)-(24), each participant can join only one itinerary whether he acts as a driver or a passenger, and $\mathbb{R}_p$, $\mathbb{R}_s$ respectively represents the subset of itineraries sets containing the passenger p and driver s.

3.2. Reduction mechanism

In the previous sections, we have constructed a mathematical model to solve the optimal assignment of ridesharing problem. To acquire the feasible itineraries set $\mathbb{R}$, we further introduce a reduction mechanism to limit the number of matching relationships among the participants and eliminate drivers who cannot be part of a rider's itinerary due to lack of spatiotemporal compatibility between their trips. It should be noted, this procedure does not limit the search space of the optimization problem, but only narrows it by cutting down practically infeasible ranges, and therefore it does not affect the optimality of the solution.

Suppose there is a feasible itinerary $R = \{r_1, \ldots, r_i, \ldots, r_{2n+2}\}$, $\forall i \in \{1, \ldots, 2n+2\}$ and a new arriving request p with an origin o_p, a destination d_p and time window $[t_{o_p}^{ED}, t_{d_p}^{LA}]$, we need to identify whether p can be a passenger to join the itinerary R. To do so, we propose a reduction mechanism with four steps as follows:

Step1 Identify whether the origin o_p of the participant s can join the itinerary r if it can be covered by any of subrange e_i, $\forall i \in \{1, \ldots, 2n+1\}$ in $\mathbb{E}$.

Firstly, given the subrange e_i, the origin o_p can be converted to $(x_{o_p}, y_{o_p}, t_{o_p})$, where t_{o_p} is the probable waiting time at o_p:

$$t_{o_p} = \max\left(0, t_{o_p}^{ED} - ed_i - t_{r_i, o_p}\right) \tag{26}$$

Secondly, we will check the origin o_p whether can be cover by e_i:

$$\left(\sqrt{\left(x_{o_p} - x_{r_i}\right)^2 + \left(y_{o_p} - y_{r_i}\right)^2} + \sqrt{\left(x_{o_p} - x_{r_{i+1}}\right)^2 + \left(y_{o_p} - y_{r_{i+1}}\right)^2}\right) \leq \left(la_{i+1} - ed_i - t_{o_p}\right) \times v \tag{27}$$

Finally, if the origin o_p can be visited in t_i without violating time constraints, we need to verify the capacity constraints as below:

$$m_i + 1 \leq CP \tag{28}$$

Step2 If o_p satisfies the constraints mentioned in equations (27) and (28), we can obtain a new itinerary R^{+o_p}, which indicates o_s has been inserted into the itinerary R, and then the corresponding sets M^{+o_p}, EA^{+o_p}, ED^{+o_p}, LA^{+o_p}, LD^{+o_p}, $\mathbb{E}^{+o_p}$ can be easily derived from equations (1)–(7).

Step3 Suppose o_p is the kth node of the itinerary R^{+o_p}, we need to identify whether the destination d_p of the request p can be visited by the itinerary R^{+o_p} if it can be covered by any of subrange t_j, $\forall j \in \{k, \ldots, 2n+2\}$ in $\mathbb{E}^{+o_p}$.

Initially, we also need to represent d_p as a form $(x_{d_p}, y_{d_p}, t_{d_p})$ according to the subrange t_j, where t_{d_p} defined as:

$$t_{d_p} = \max\left(0, la_{j+1} - t_{d_p, r_{j+1}} - t_{d_p}^{LA}\right) \tag{29}$$

Last, the required covering condition should be verified:

$$\begin{aligned}&\left(\sqrt{\left(x_{d_p} - x_{r_j}\right)^2 + \left(y_{d_p} - y_{r_j}\right)^2} + \sqrt{\left(x_{d_p} - x_{r_{j+1}}\right)^2 + \left(y_{d_p} - y_{r_{j+1}}\right)^2}\right)\\ &\leq \left(la_{j+1} - ed_j - t_{d_p}\right) \times v\end{aligned} \tag{30}$$

Step4 If d_p satisfies the constraint mentioned in equation (30), we can obtain a new itinerary $R^{+o_p+d_p}$, which indicates d_p has been inserted into the itinerary R^{+o_p}, and then the corresponding sets $M^{+o_p+d_p}$, $EA^{+o_p+d_p}$, $ED^{+o_p+d_p}$, $LA^{+o_p+d_p}$, $LD^{+o_p+d_p}$, $\mathbb{E}^{+o_p+d_p}$ can be easily derived from equations (1)–(7).

Consequently, if a request p satisfies all the verification in the four steps mentioned above, namely he can join the itinerary r to establish a new feasible itinerary scheme $R^{+o_p+d_p}$ (summary of the reduction mechanism can be seen in Fig. 3). In this paper, the objective of ridesharing matching problem is to obtain the optimum schedule $\mathbb{R}^*$ among the feasible itinerary set $\mathbb{R}$.

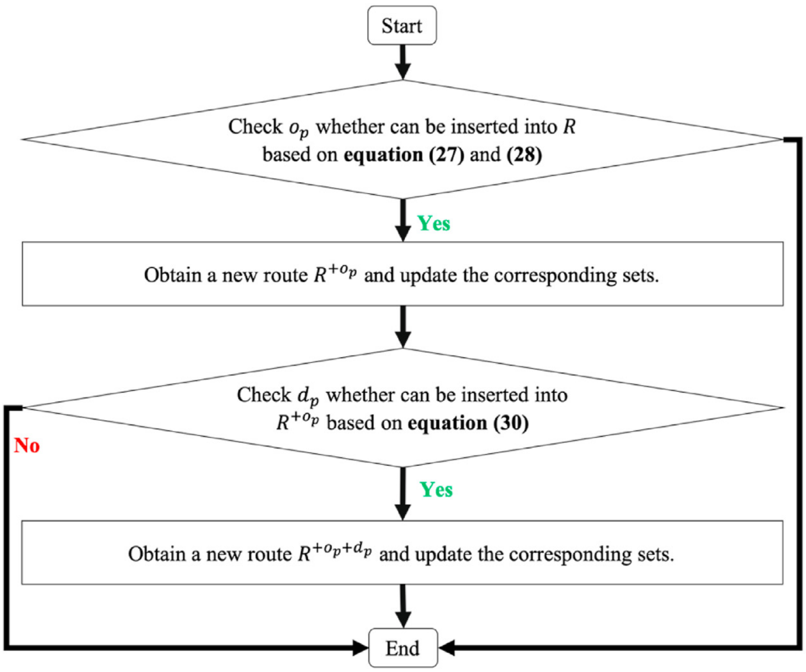

Fig. 3. The procedure of reduction mechanism.

4. Algorithm

Since the ILP model mentioned in Section 3.1 is NP-hard, even with a small number of requests, the feasible itineraries set $\mathbb{R}$ may be large and the decision variable λ_R results in an exponentially large search space. Hence in this section we propose a heuristic approach to arrive at the approximately optimal ride-matching assignment, which aims to reach the best compromise between solution quality and computational efficiency for large-sized real-world instances.

The philosophy of the heuristic approach is to reduce the value of the exponential growth of decision variables λ_R. Ferrari et al. (2003) has proposed a straightforward application of well-known greedy heuristic to solve the set partitioning problem. In addition, Huang et al. (2013) applied a kinetic tree algorithm capable of scheduling dynamic requests and adjusting itineraries on-the-fly. Furthermore, Qian et al. (2017) introduced a graph conversion method to acquire the optimal solution. However, the exact algorithm may work efficiently for small or medium sized instances, nevertheless, it will fail with growing size. Xing et al. (2018) also proposed to apply the greedy strategy in solving the one-to-one ride-matching problem combining stability of matching.

Therefore, in order to efficiently solve the large-scale ridesharing cases in this paper, we develop an improved heuristic algorithm so-called ordered greedy (OG) algorithm, and the process of the OG Algorithm can be summarized as follows:

Step1 Initialize: Given the participants set $S = \{s_1, s_2, \ldots, s_N\}$, we can construct the candidate assignment $\mathbb{R}^* = \{R^{solo}_{s_1}, R^{solo}_{s_2}, \ldots, R^{solo}_{s_N}\}$, which comprised of the itinerary driven alone by each participant in set S. Moreover, the driver set D and passenger set P are initially set to $\varnothing$.

Step2 Order: Given the candidate assignment $\mathbb{R}^*$, then we need to sort the itinerary R from largest to smallest in set $\mathbb{R}^*$ based on δ_R, which can be denoted as:

$$\delta_R = (la_{2n+2} - ed_1) - \sum_{i\in\{2,\ldots,2n+2\}} t_{r_{i-1},r_i} \tag{31}$$

where $(la_{2n+2} - ed_1)$ is the totally extra time of the driver and $\sum_{i\in\{2,\ldots,2n+2\}} t_{r_{i-1},r_i}$ represents the sum of travel time consuming. Hence, we can observe the remaining time (to serve other participants if feasible) of the itinerary R via introducing δ_R.

Step3 Select: Given the ordered candidate assignment $\mathbb{R}^*$, we select the itinerary R_i from $\mathbb{R}^*$ in order, and further obtain a feasible subset $\mathbb{R}_{R_i,+1}$ deriving from R_i inserted only one participant. In particular, all the itineraries in $\mathbb{R}_{R_i,+1}$ will have a positive distance saving (the itineraries with a negative distance saving will be also seen as infeasible and remove from $\mathbb{R}_{R_i,+1}$). Hence:

- If $\mathbb{R}_{R_i,+1}$ is nonempty, we will go directly to Step4. It's also should be noted that if R_i is selected for the first time, the driver of R_i will be removed from the set S.
- If $\mathbb{R}_{R_i,+1}$ is empty, which means the itinerary R_i can't share a ride with any other participants in remaining set S, we will select the next itinerary in $\mathbb{R}^*$ and repeat Step3.

Step4 Update: Based on local greedy strategy, we can easily obtain the itinerary R^* in $\mathbb{R}_{R_i,+1}$ with maximum travel distance saving as:

$$R^* = argmax_R\left(\{c_R, \forall R \in \mathbb{R}_{R_i,+1}\}\right) \tag{32}$$

Compared to R_i, the new joined passenger in $\mathbb{R}^*$ will be added into the passenger set P and removed from the set S. Moreover, the itinerary driven alone by this passenger will also be removed from the set $\mathbb{R}^*$. Consequently, R_i will be updated to R^*.

Step5 Iterate: Repeat the Step2, Step3, Step4 until set S is empty.

Step6 Terminate: Final the assignment set $\mathbb{R}^*$ will be an approximately optimal solution (more detailed procedure can be referred in Table 2).

In order to explain the OG algorithm mentioned above from a more intuitive view, then we construct the following example to demonstrate the procedure of the Algorithm1 in detail. As shown in the Fig. 4A, we suppose that participants set $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, the steps for applying OG Algorithm to assign the announcements are as below:

Firstly, suppose δ_R of each itinerary in $\mathbb{R}^*$ equal to 60,50,40,30,20,10 min, respectively, and we can construct the candidate itinerary set $\mathbb{R}^*$ from the biggest to the smallest as follows:

Table 2
Pseudo-code of OG algorithm.

Algorithm OG Algorithm
Initialize $\mathbb{R}^* = \{R^{solo}_{s_1}, R^{solo}_{s_2}, \ldots, R^{solo}_{s_N}\}$, *where* $S = \{s_1, s_2, \ldots, s_N\}$.
$D = \varnothing, P = \varnothing$.
Input: $S, D, P, \mathbb{R}^*$
Output: $\mathbb{R}^*$
$i \leftarrow 1$
While $S \neq \varnothing$ **do**
Sort the itinerary set $\mathbb{R}^*$ based on the time gap δ_R, $\forall R \in \mathbb{R}^*$.
Select the i th itinerary R_i in the set $\mathbb{R}^*$
If driver in R_i is selected for the first time **do**
$d \leftarrow$ the driver of R_i
$D \leftarrow$ *append d into D*, $S \leftarrow$ *remove d from S*
End
Enumerate feasible schedules set $\mathbb{R}_{R_i,+1}$
If $\mathbb{R}_{R_i,+1} \neq \varnothing$ **do**
Calculate the travel distance saving set $\{c_R, \forall R \in \mathbb{R}_{R_i,+1}\}$ for each $R \in \mathbb{R}_{R_i,+1}$
$R^* \leftarrow$ *Obtained by* ***Equation*** (20)
$p \leftarrow$ the new joined passenger in R^* compared to R_i
$P \leftarrow$ *append p into P*, $S \leftarrow$ *remove p from S*
$R^* \leftarrow$ *remove* R^{solo}_p *from* $\mathbb{R}^*$
Replace R_i with R^* in the set $\mathbb{R}^*$
$i \leftarrow 1$
Else
$i \leftarrow i + 1$
End
End
Return $\mathbb{R}^*$

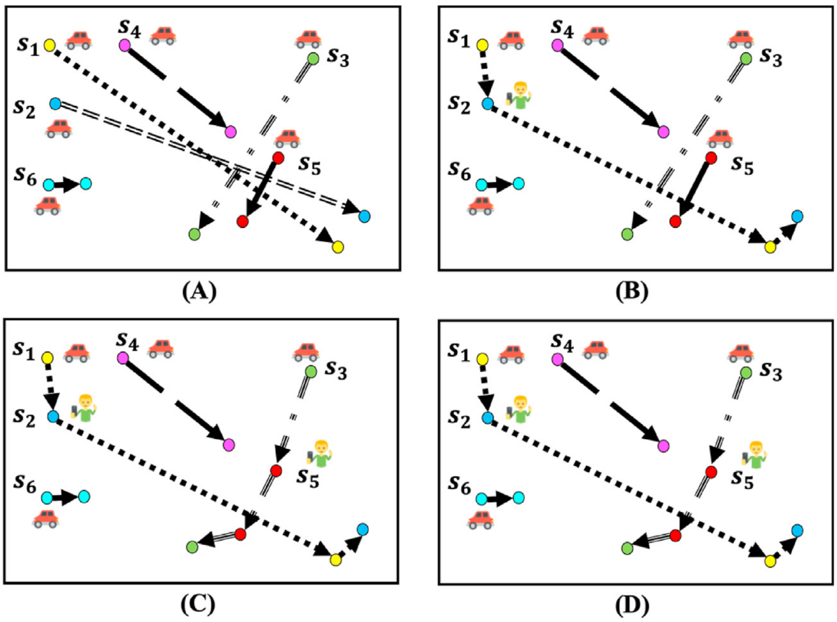

Fig. 4. The example of OG algorithm.

$$\mathbb{R}^* = \left\{R^{solo}_{s_1}, R^{solo}_{s_2}, R^{solo}_{s_3}, R^{solo}_{s_4}, R^{solo}_{s_5}, R^{solo}_{s_6}\right\}$$

Secondly, $R^{solo}_{s_1}$ will be selected and suppose $\mathbb{R}_{R^{solo}_{s_1},+1} = \{R_{s_1+s_2}, R_{s_1+s_4}, R_{s_1+s_5}\}$, and s_1 will be assigned as a driver, hence, $D = \{s_1\}$ and $S = \{s_2, s_3, s_4, s_5, s_6\}$.

Thirdly, suppose c_R of each R in $\mathbb{R}_{R^{solo}_{s_1},+1}$ are 2.5, 1.5 and 1 km, respectively, and according to the local greedy strategy ($R_{s_1+s_2}$ owns the maximum saving), thus s_2 will be a passenger and $P = \{s_2\}$, $S = \{s_3, s_4, s_5, s_6\}$. Aa a consequence, $\mathbb{R}^* = \{R_{s_1+s_2}, R^{solo}_{s_3}, R^{solo}_{s_4}, R^{solo}_{s_5}, R^{solo}_{s_6}\}$ (shown in Fig. 4B).

Fourthly, suppose δ_R of each itinerary in $\mathbb{R}^*$ equals 15,40,30,20,10 min, respectively. Then the set $\mathbb{R}^*$ will be sorted as:

$$\mathbb{R}^* = \left\{R_{s_3}^{solo}, R_{s_4}^{solo}, R_{s_5}^{solo}, R_{s_1+s_2}, R_{s_6}^{solo}\right\}$$

Accordingly, $R_{s_3}^{solo}$ will be selected and suppose n $\mathbb{R}_{R_{s_3}^{solo},+1} = \{R_{s_3+s_5}\}$, s_3 will be assigned as a driver and obviously s_5 will be a passenger, hence, $D = \{s_1, s_3\}$, $P = \{s_2, s_5\}$, $S = \{s_4, s_6\}$ and $\mathbb{R}^* = \{R_{s_1+s_2}, R_{s_3+s_5}, R_{s_4}^{solo}, R_{s_6}^{solo}\}$ (shown in Fig. 4C).

Fifthly, suppose δ_R of each itinerary in $\mathbb{R}^*$ equals 15,15,30,10 min, respectively. Then the set $\mathbb{R}^*$ will be sorted as:

$$\mathbb{R}^* = \left\{R_{s_4}^{solo}, R_{s_1+s_2}, R_{s_3+s_5}, R_{s_6}^{solo}\right\}$$

Obviously, $R_{s_4}^{solo}$ will be selected and $\mathbb{R}_{R_{s_4}^{solo},+1}$ is supposed to be empty. Therefore, both s_4 and s_6 will be assigned as drivers, and further $D = \{s_1, s_3, s_4, s_6\}$, $P = \{s_2, s_5\}$.

Consequently, $S = \varnothing$ and the $\mathbb{R}^* = \{R_{s_1+s_2}, R_{s_3+s_5}, R_{s_4}^{solo}, R_{s_6}^{solo}\}$ (shown in Fig. 4D) will be obtained as the approximately optimal assignment.

5. Numerical study

We now present the results of a set of computational experiments, designed to generate the performance of the proposed algorithm and analyses upon where is easier to get a ride. The algorithm was coded in Matlab2015b, and the experiments were conducted on a desktop with @1.60 GHz processor, and 4 GB RAM. Each result reported in the following sections are averaged over 10 runs for each problem instance. The origins and destinations of participants in this experiment will be randomly generated in a circle with a L mile radius. Besides, the rule of time window of each participant can be set as $[t_{o_s}^{ED}, t_{o_s}^{ED} + \alpha \times t_{o_s,d_s}]$, where $t_{o_s}^{ED}$ is generated randomly within $[0, tw]$, and $t_{d_s}^{LA}$ simply equals to the sum of $t_{o_s}^{ED}$ and $\alpha \times t_{o_s,d_s}$. Note that the parameter α is a travel time budget factor, which indicates the degree of tightness of travel time (Masoud et al., 2017).

In the next three sections, we will use the proposed algorithm to find the optimal solutions for the randomly generated problems and real-world problems through simulations. In section 5.1, we will compare the performance of the proposed algorithm with Cplex in terms of computation time and solution quality. And an in-depth analysis upon which factors affected ridesharing participants to become riders is conducted in section 5.2. Finally, the numerical experiments are carried out by using multiple real-world datasets from Chengdu in China to identify the practical results in comparison to Section 5.2. In addition, we introduce three indicators in this paper as below:

(1) The request density ρ

$$\rho = |S| \Big/ \left(\pi \cdot L^2 \cdot tw\right) \tag{33}$$

(2) The passenger ratio δ

$$\delta = {|P|}/{|S|} \tag{34}$$

(3) The distance ratio ε

$$\varepsilon = \sum_{R \in \mathbb{R}^*} dis(R) \Big/ \sum_{s \in S} dis\left(R_s^{solo}\right) \tag{35}$$

In equation (33), $\pi \cdot L^2$ represents the area size of the given region and tw is the length of the time range, thus ρ also indicates the arriving rate of the requests per minute per square kilometer. Moreover, $|S|$ and $|P|$ in equation (34) are the

Table 3
Parameter setting.

Parameter	Values
L	5 km
tw	30 min
α	2
CPP	4

number of participants and passengers, respectively. Furthermore, $\sum_{r\in R^*} dis(r)$ is the total distance after ridesharing while $\sum_{s\in S} dis(r_s^{solo})$ is the total distance without ridesharing in equation (35). It should be noted, that the objective mentioned in ILP model can be equivalent to maximizing the distance ratio ε.

For more detailed parameters, see the following Table 3:

5.1. Algorithm comparison

For the first experiment, in order to gain a comprehensive comparison of the algorithm in terms of computation time and solution quality, we generate and solve five random instances with various $\rho = 0.01, 0.02, 0.05, 0.1, 0.2$, ten runs for each. As shown in Table 4, The passenger ratio δ, total distance and computation time cost are compared between proposed method and IBM Cplex.

It can be observed that the Cplex is better than our method in minimizing the total distance. However, it will take much more seconds to find the optimal solution $\mathbb{R}^*$ compared to proposed algorithm. Specifically, even with small increase of ρ (such as $\rho = 0.02, 0.1$), the computation time may differ significantly. The reason for the huge differences is mainly because of the increase of the feasible itinerary set $\mathbb{R}$. For instance, $\rho = 0.02$ has only 917 feasible itineraries due to the loosely matched relations among participants, which can be easily solved by enumerating within 3 s. As ρ increases to 0.1, it becomes harder to enumerate 34233 feasible itineraries for a large amount of computation time. In addition, when ρ equals to 0.2, the computing cost is too high to arrive at the optimal solutions by Cplex.

On the contrary, it can be illustrated that the proposed OG algorithm not only performs efficiently for all the cases, but also obtains reasonably good solution quality. It generates optimal or approximately optimal solutions for cases with verified optimal solution and solves $\rho = 0.2$ case in less than 20s. In particular, the solutions of heuristic algorithm are optimal solution in $\rho = 0.01$ case and are from 2.0% ($\rho = 0.02$) to 3.8% ($\rho = 0.1$) worse than the optimal solutions in other cases. Consequently, the heuristic algorithm is proved to be efficient and effective for solving large-scale ride-matching instances based on our numerical experiments.

Since the efficiency and effectiveness of the heuristic algorithm has been shown, we further generate and solve 11 additional instances with $\rho = 0.1, \ldots, 1.5$. The results are shown in Table 5, and Fig. 5. A measure the passenger ratio δ and the distance ratio ε based on different level of ρ. As intuition suggests, for the given region and time scale, the passenger ratio δ increases with the participant density ρ in the system. This implies that the higher density will result in much more riders in ridesharing system, which also means more income for a platform. On the other hand, the distance ratio ε presents a downward trend with the increase of ρ, which suggests that system-wide vehicle-miles will significantly reduce when the participant density ρ is high enough. Another interesting observation is that there is a downward trend in both the increase rate of δ and the decrease rate of ε, moreover, it's depicted in Fig. 5. B that the computation cost will also become expensive when ρ is too high. As a result, the figure and table lead to the conclusion that ride-sharing platform should collect the appropriate requests (e.g. ρ can be set to 0.8 in Fig. 5) before executing assignments to achieve at a better condition.

5.2. Mining the factors affecting passenger ratio

In the previous section, we have presented the overall results of the ridesharing system and introduced the indicator so-called "passenger ratio" δ. Obviously in the system, δ may vary among places in the given region and higher δ can reflect more

Table 4
Comparison of algorithm's performance.

$\lvert S\rvert$	ρ	$\lvert\mathbb{R}\rvert$	Cplex			OG algorithm		
			δ	Total Distance (km)	Time Cost(s)	δ	Total Distance (km)	Time Cost(s)
30	0.01	83	0.26	141.75	3.52	0.26	141.75	0.96
60	0.02	917	0.4833	220.85	248.24	0.55	225.38	2.93
150	0.05	2758	0.4800	504.59	2465.06	0.57	517.59	9.09
300	0.1	34233	0.5733	901.73	53456.35	0.59	937.14	10.17
600	0.2	–	–	–	–	0.64	1689.28	19.13

Table 5
The result of 11 additional instances with $\rho = 0.1, \ldots, 1.5$.

$\lvert S \rvert$	ρ	δ	$\lvert P \rvert$	$\lvert D \rvert$	Total distance (km)	ε	Time cost(s)
300	0.1	0.596	178.8	121.2	937.14	0.593818	10.16928
600	0.2	0.642667	385.6	214.4	1689.28	0.536523	19.13027
900	0.3	0.661778	595.6	304.4	2384.21	0.508735	35.4628
1200	0.4	0.678667	814.4	385.6	3043.66	0.487774	73.9648
1500	0.5	0.686333	1029.5	470.5	3704.49	0.474363	113.8377
1800	0.6	0.6965	1253.7	546.3	4344.61	0.463184	156.251
2100	0.7	0.70481	1480.1	619.9	4971.99	0.453732	257.574
2400	0.8	0.710458	1705.1	694.9	5576.22	0.444812	377.086
2700	0.9	0.714185	1928.3	771.7	6174.31	0.437665	563.427
3000	1.0	0.717467	2152.4	847.6	6809.35	0.434076	738.268
4500	1.5	0.731467	3291.6	1208.4	9675.83	0.412705	1239.95

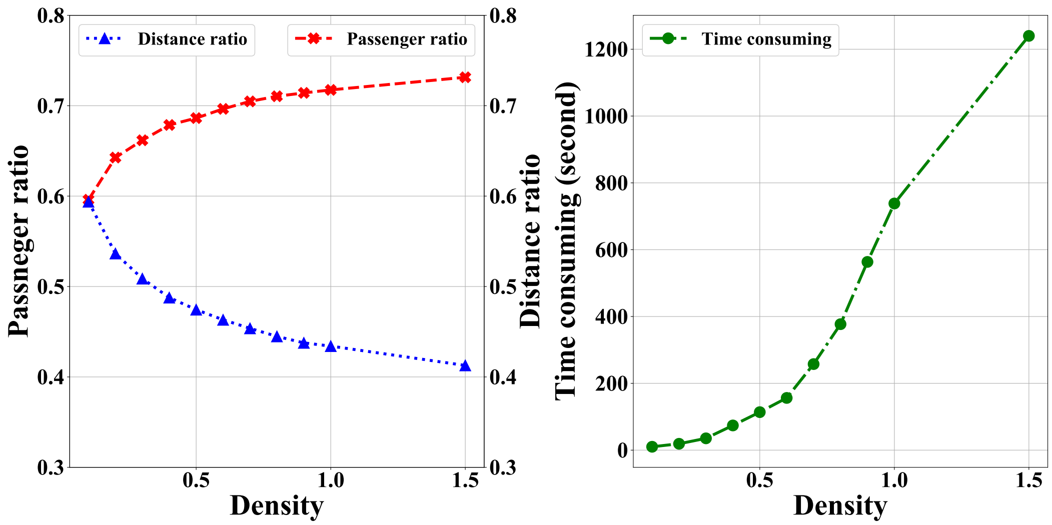

Fig. 5. The curves of δ, ε and time consuming under different ρ.

probabilities for participants to find a ride. Therefore, in order to investigate the changes of δ in ridesharing system, we will observe its performance in terms of spatial distribution based on grid-level and ring-level in this section. In the case of this scenario, the region will be divided into 100 grids, which are 1 km long and 1 km wide. Furthermore, we divide the grids into five rings based on the distance away from the center, and notice that a ring can be approximately considered as the sub-region with same distance to the center.

Fig. 6 depicts the heatmaps of δ in each grid, in this experiment the origin o_s will be used to represent a participant s for simplistic, that is, δ will be equivalent to the ratio between the number of passengers' origins and the number of participants' origins in each grid. From the figure we can observe that the distribution of δ shows an irregular trend because of randomness in the small participant size (such as 300 and 600 participants). However, the increase in participant size (such as 3000 participants) results in uniform-distributed δs among the grids of inner four rings.

Furthermore, Fig. 7 displays δ upon each ring, for a given small participant size (such as 300 and 600 participants), the passenger ratio δ will decrease with the ring number (equivalent to the distance to the center of the region), which suggests that the participants in places closed to the center are easier to find a ride. While we set more participants (such as 1500 or 3000 participants), along with the increase in ring number, δ reaches a peak value at the third rings and then reduces. This result suggests that the participants who start from the third ring will be more likely to be a passenger.

Despite the intuitive difference in Figs. 6 and 7, there is little knowledge of which factors will affect the spatial distribution of δ. However, understanding the significance of each factor and its quantitative influence will not only offer recommendations to ridesharing participants to find a ride but also help the platforms to improve the ride-sharing service. To achieve this goal, we utilize the several factors as below:

Definition 1. Given a specific sub-region, τ_i is the flow per participant of grid i:

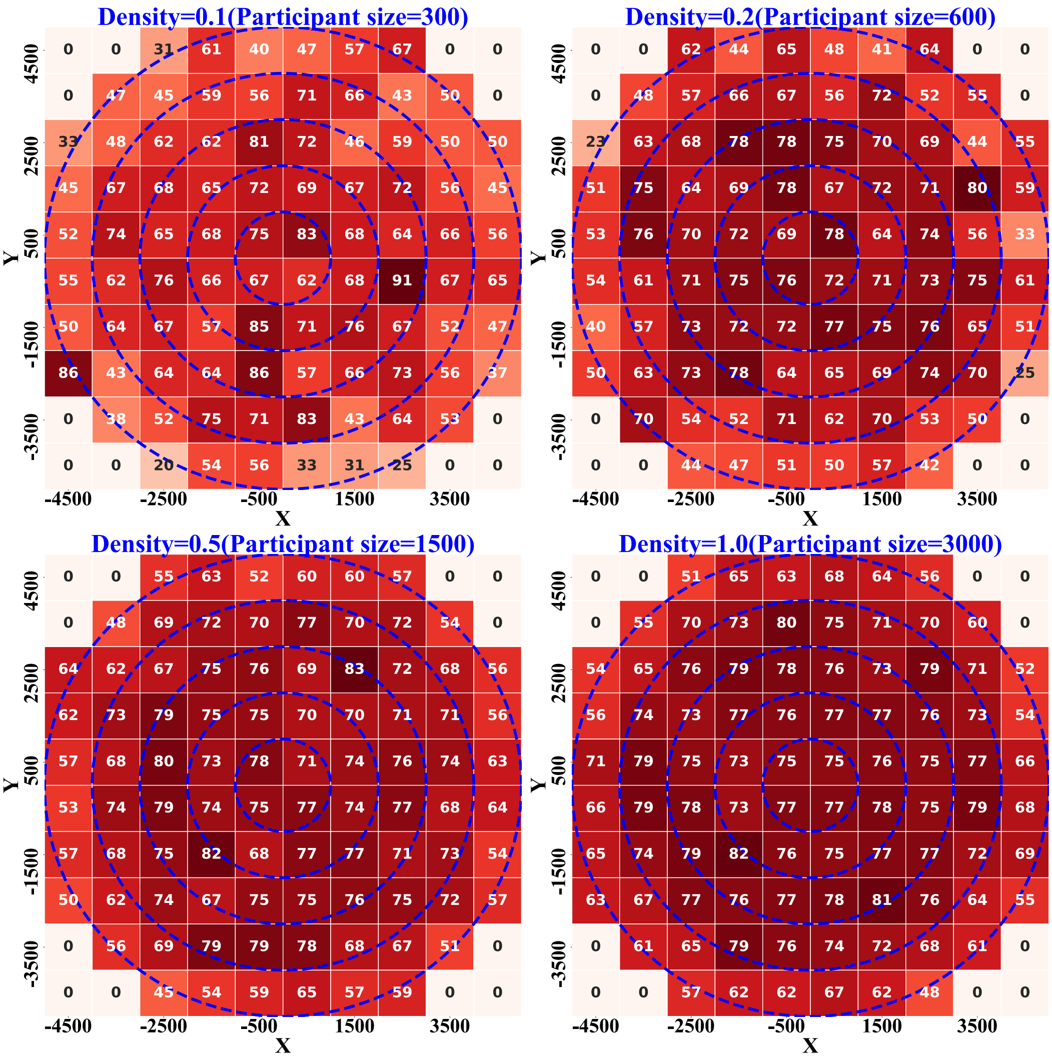

Fig. 6. δ percentage varies among different grids.

$$\tau_i = \frac{D_i}{|S_i|} \tag{36}$$

where D_i denotes the flows (the number of drivers) passing grid i, and S_i is the set of participants originating from grid i.

Definition 2. Given a specific sub-region, ϕ_i is the average OD distance of passengers in grid i:

$$\phi_i = \frac{\sum_{s \in P_i} R_s^{solo}}{|P_i|} \tag{37}$$

where P_i is the set of passengers originating from grid i, and $\sum_{s \in P_i} R_s^{solo}$ indicates the sum of solo distance of all passengers in the given sub-region i.

Definition 3. Given a specific sub-region, ω_i is the rate of vehicles' utilization in grid i:

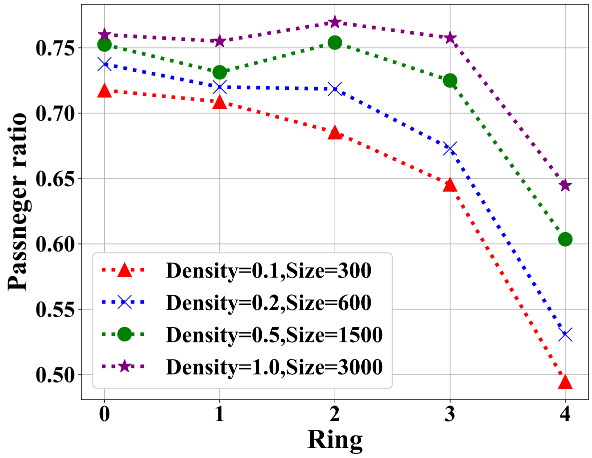

Fig. 7. δ varies among different rings.

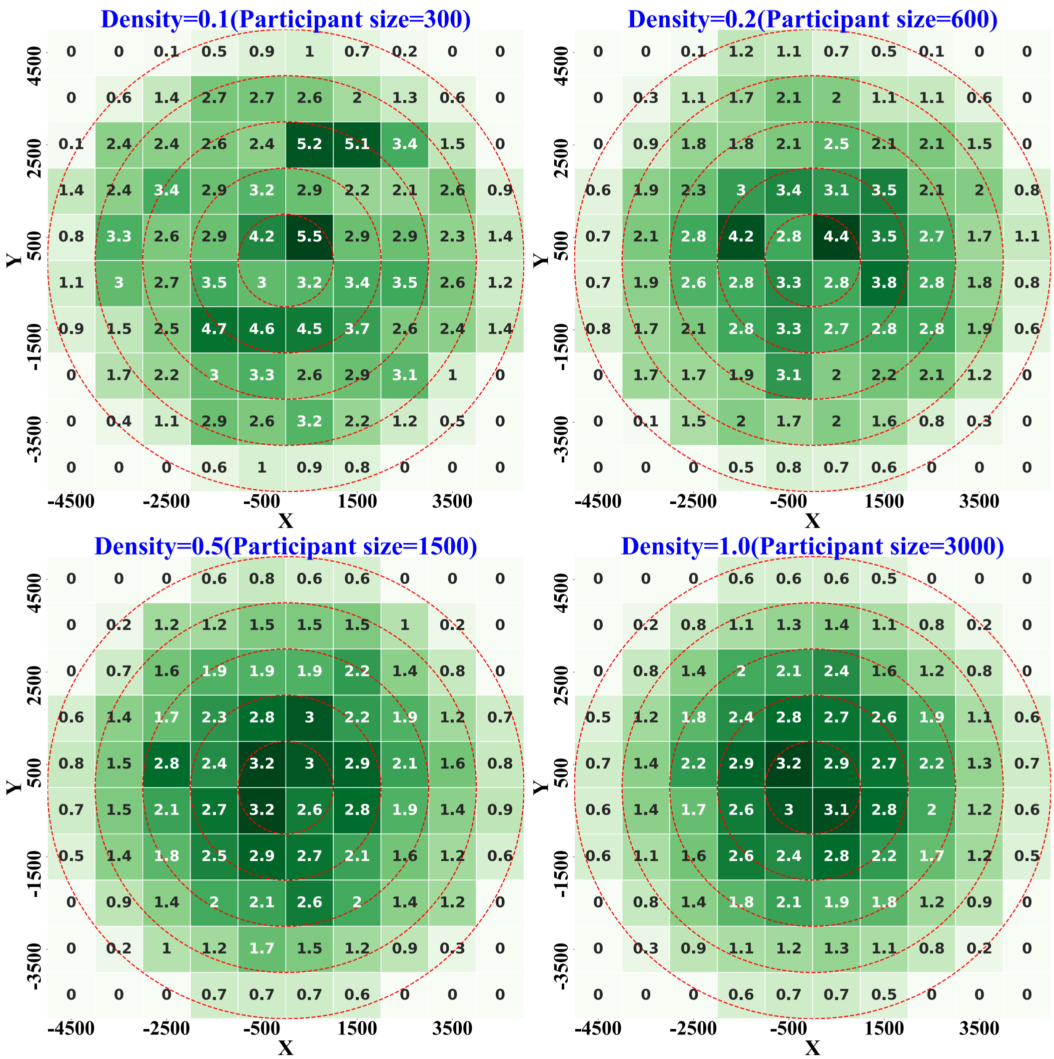

Fig. 8. τ varies among different grids.

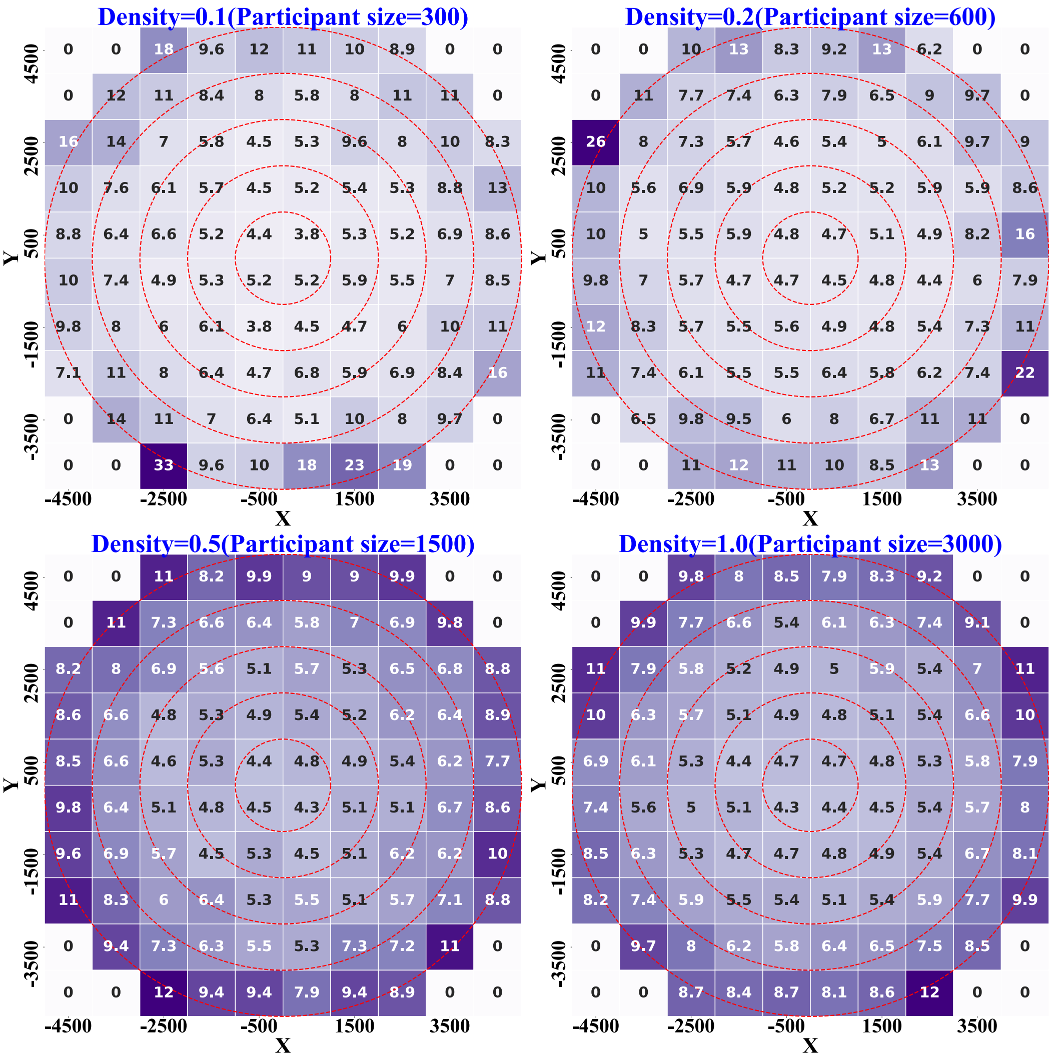

Fig. 9. ϕ varies among different grids.

$$\omega_i = \frac{\sum_{d \in D_i} m_i^d}{|D_i| \cdot CP} \tag{38}$$

where m_i^d represents the occupied seats when driver d visits grid i, thus $\sum_{d \in D_i} m_i^d$ is the sum of total occupied seats of all the drivers passing grid i.

The distributions of τ, ϕ, ω are depicted in Fig. 8, Fig. 9, Fig. 10, respectively. As an intuition suggests, both τ and ω decrease with the distance to the center, while ϕ drops down as the increasing of the distance to the center. Here we further investigate the correlations between factors and δ in Fig. 11. In the first place, there exists a rough linear rise for δ with τ in four cases, and another interesting observation is that δ will keep stable when τ reaches 2 as requests are sufficient (such as 3000 trips). This is due to the fact that τ is no longer the major determinants if given enough trips. In the second place, δ significantly decreases with ϕ in four cases, which means that the participant with a short distance is easier to find a ride in the system. Finally, δ also demonstrates an increasing trend with ω, and as a same trend, δ will fluctuate within a small range if ω exceeds 0.5 when we collect enough requests (such as1500 or 3000 trips).

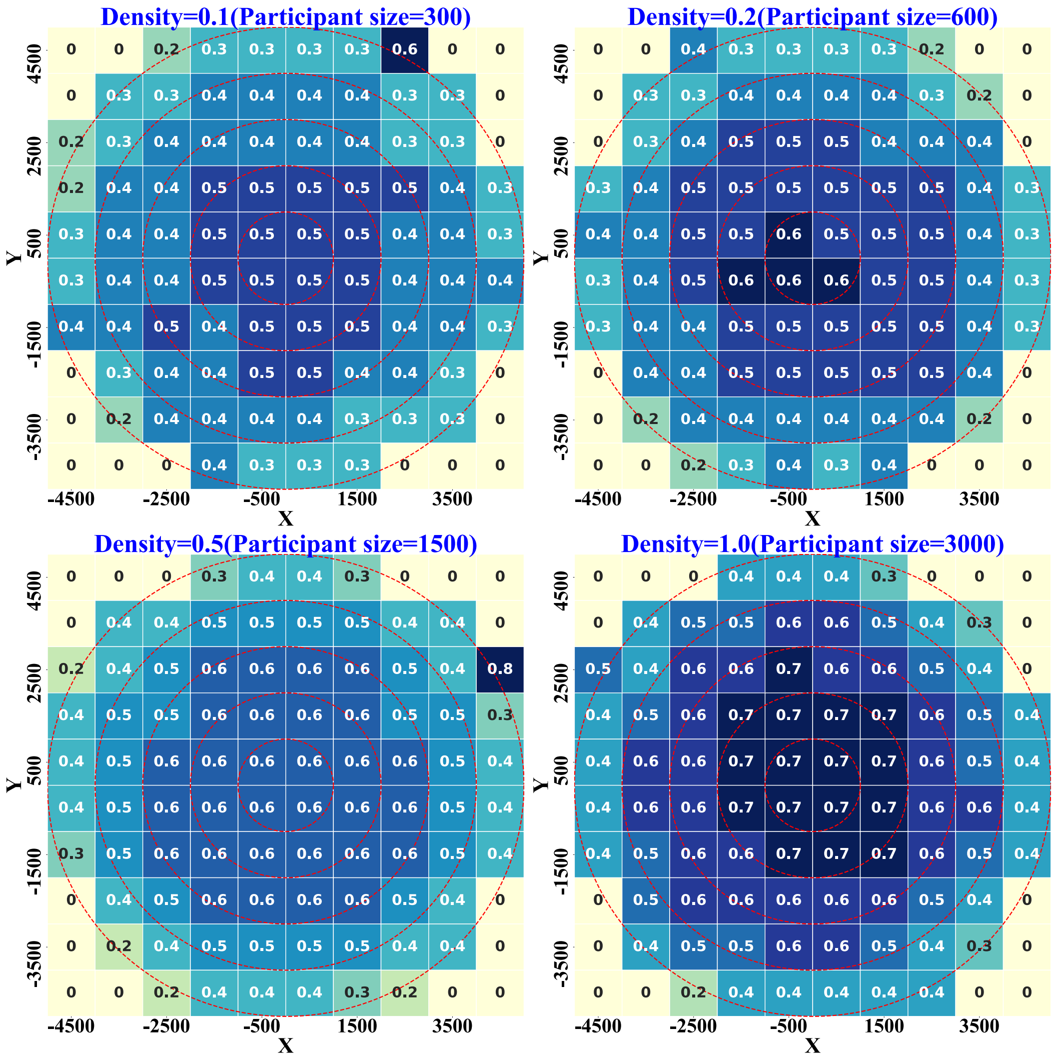

Fig. 10. ω varies among different grids.

5.3. Real-world experiments

Compared to the random distributions of origins and destinations in simulation experiments, the commercial and business districts are usually denser within certain geographical regions and are distinct from residential areas in reality. As depicted in Fig. 12, we pick a spatial range with 5 km wide and 5 long in Chengdu, and specifically, the center of our region is set as Tianfu Square. Thus, in this section we use taxi datasets in Chengdu provided by DiDi chuxing[1] to comprehensively understand the performance of our algorithm and the distribution of the passenger ratio.

In this experiment, all the data are pre-processed to remove erroneous records, such as coordinates located outside the city boundary or trips whose length is shorter than 1 min. After cleaning, we conduct the experiments within specific region centered at Tianfu Square with the same size compared to the previous experiments. In particular, we extract trip dataset during off-peak hours (00:00–00:30), morning peak hours (09:00–09:30), and evening peak hours (17:00–17:30) on November 2nd, 2016, which are regarded as representative time frames covering typical traffic states. And all the other parameters are identical with settings in Table 3.

Table 6 presents the result for different extracted intervals, similar to our previous simulation the passenger ratio δ demonstrates an increasing trend and the distance ratio ε presents a down trend with the increase of the density ρ. Another interesting point is that there is always a trade-off between the solution quality and the computational efficiency for this ride-

[1] https://outreach.didichuxing.com/research/opendata/.

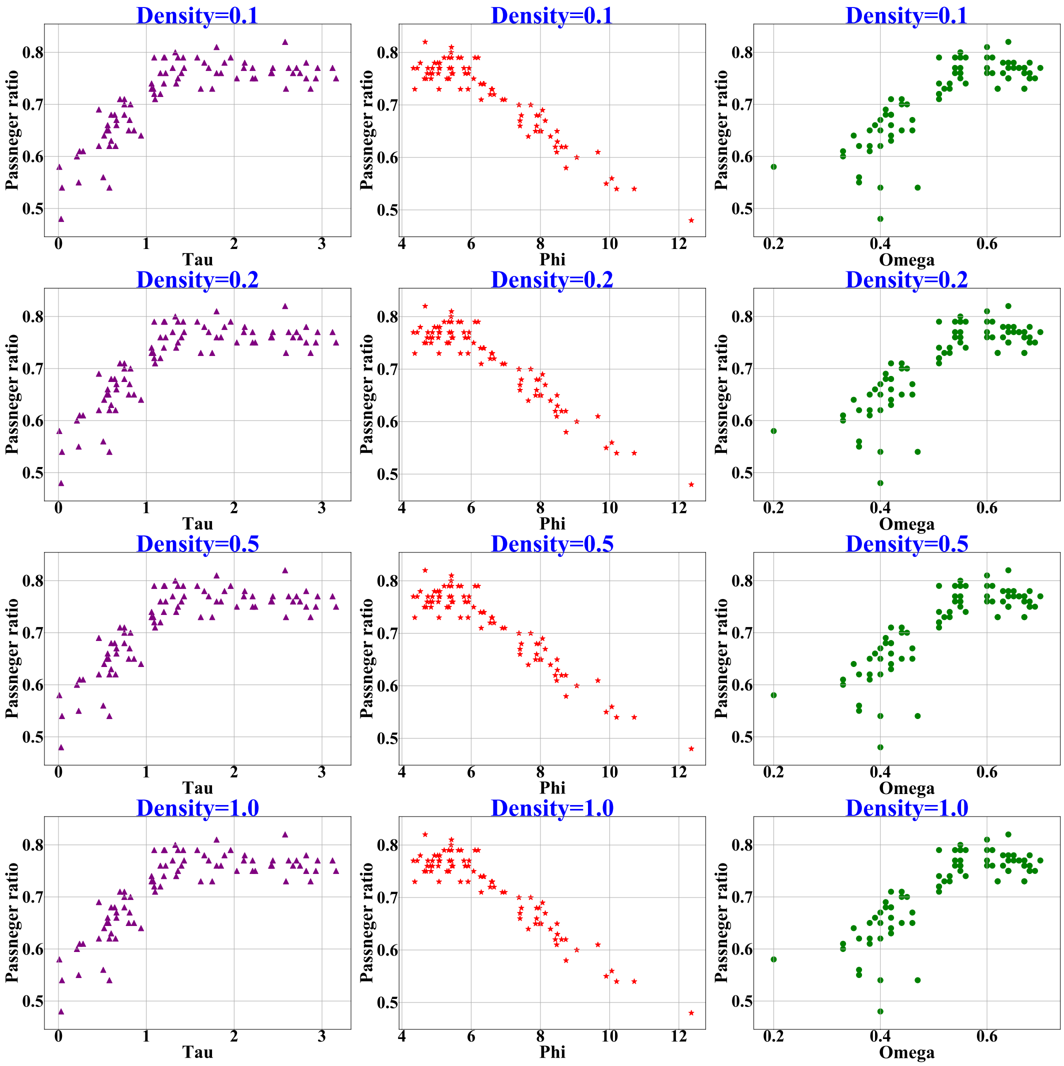

Fig. 11. The correlations between factors and δ.

matching problem, that is, higher income (higher δ) and lower vehicle-mile cost (lower ε) means more expensive computational cost for a platform. Apparently, compared to other scenarios, the result of morning peak (09:00–09:30) is optimal but it takes more than 17 min, which may beyond the participants' endurance. As a consequence, platforms in real-world should make a matching decision under an appropriate participant density.

To further investigate the passenger ratio under different locations in Chengdu, firstly, we plot the spatial distribution of origins and passenger ratio in Fig. 13. Contrast to our simulations, the origins in Chengdu display a high central tendency which is mainly concentrated in inner two rings. The main factors lead to the spatial concentration in ridesharing are the residential distribution and commuting pattern in real-world. Also shown in Fig. 14, we combine the passenger ratios among girds locating at the same ring, we can observe that the city center is not the places with the highest δ during 09:00–09:30, which suggests that the people close to the Tianfu Square (the central ring) need to go to other specific regions (the second ring) rather than staying in the center.

6. Conclusion

Nowadays ride-sharing has the potential to provide huge societal and environmental benefits, which is also an effective way to balance the demand and supply during the peak periods without increasing the number of vehicles. Hence in this paper, we consider a ride-sharing system setting in which all the trips are known in advance and suppose each trip is

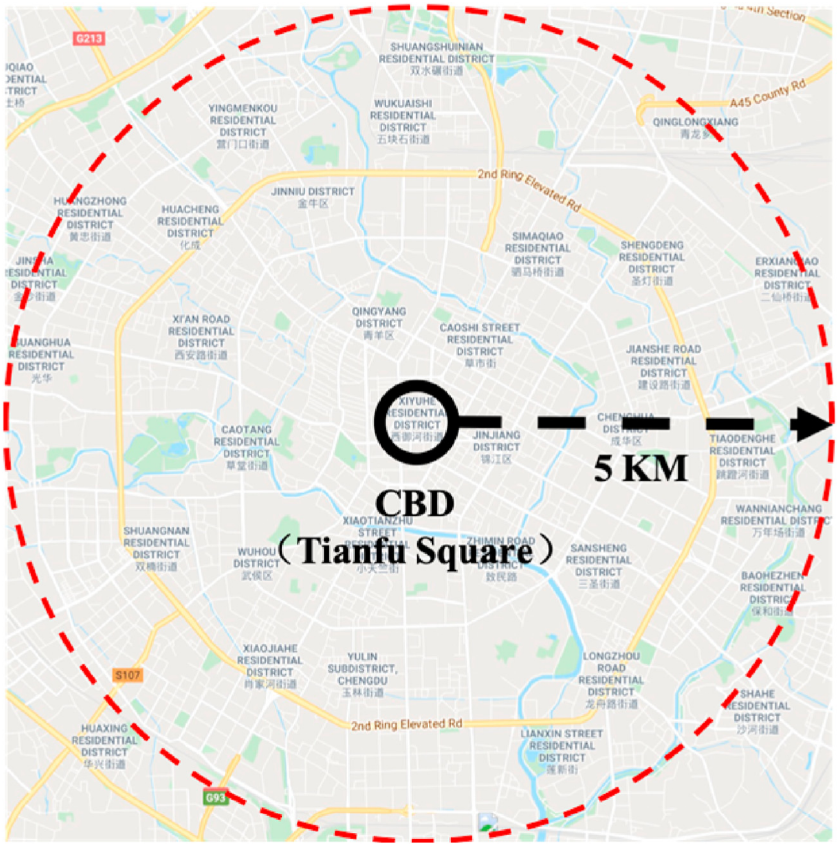

Fig. 12. The sketch map of selected region in Chengdu.

Table 6
The result generated by Chengdu data.

Interval	$\|S\|$	ρ	δ	Total distance (km)	ε	Time cost(s)
00:00–00:30	1126	0.3753	0.6687	1682.96	0.4896	102
09:00–09:30	3755	1.2517	0.7217	4700.51	0.4194	1073
17:00–17:30	3532	1.1773	0.7197	4434.58	0.4392	954

supposed to be made for self-interest. Therefore, the ride-matching problem in this study is formulated as a set partition problem and we further propose a heuristic method to solve the ride-matching problem, and the results prove that our method achieves remarkable performance (about 3.8% worse than the verified exact optimum and solve the problem within a few seconds).

We also conduct extensive numerical experiments to answer the question “Who is more likely and where is easier to be picked up in ride-sharing mode?” based on simulation data and Chengdu data. All these experiments demonstrate the platform should collect a proper level of density of trips before executing the matching process. The best place to find a ride changes with the demand density in the simulation experiments, which is significantly affected by traffic flow and vehicles’ utilization. In addition, the optimal strategy for participants to be a rider is going to other specific regions rather than staying in the city center from real-world experiments.

In the future, the proposed method could be applied to the online-hailing platforms, such as DiDi and Uber, and we believe that our findings could be beneficial for these platforms or governments to develop strategies in ridesharing and further significantly improve the level of ridesharing service.

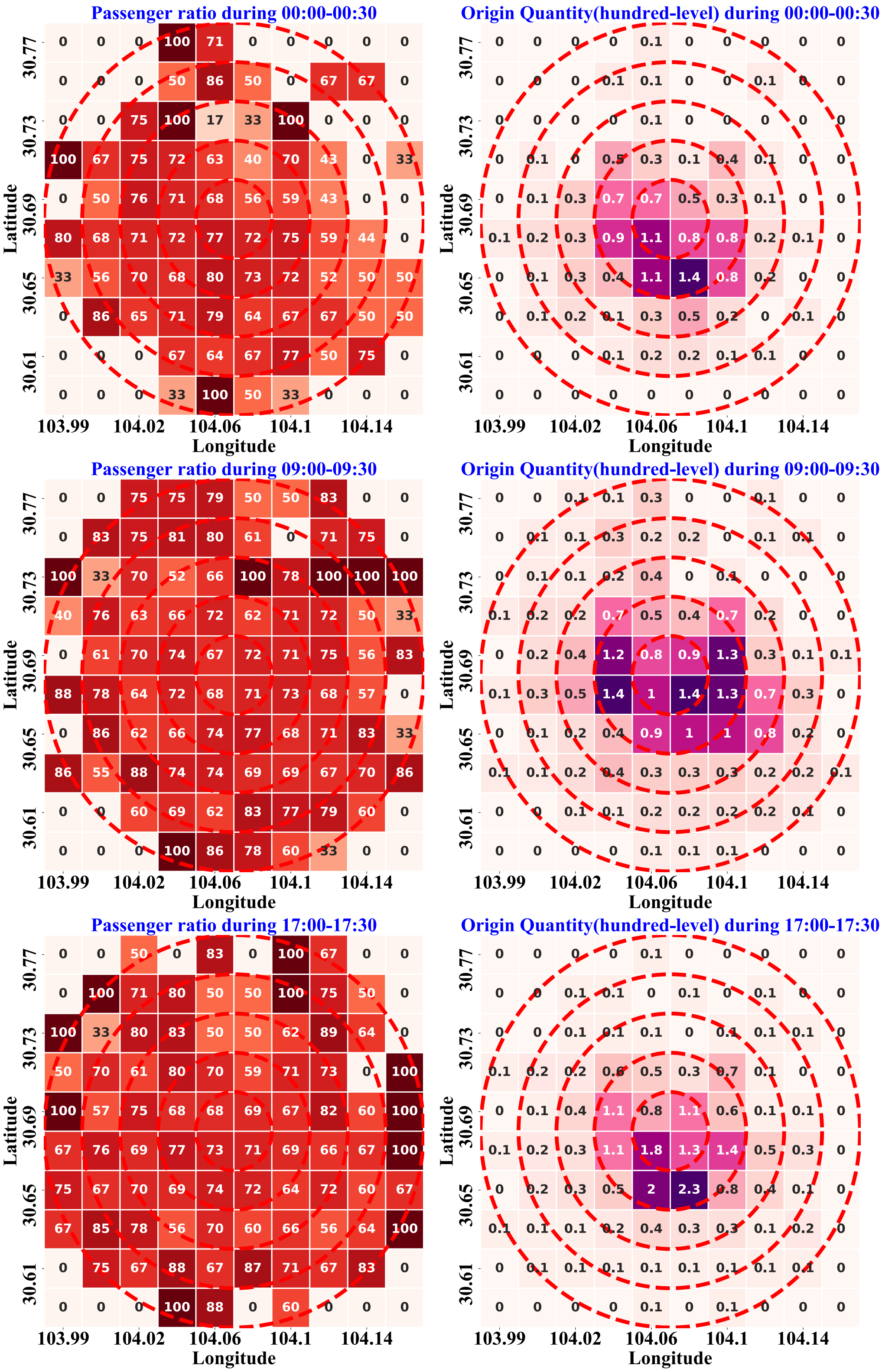

Fig. 13. The distribution of origins and δ in Chengdu.

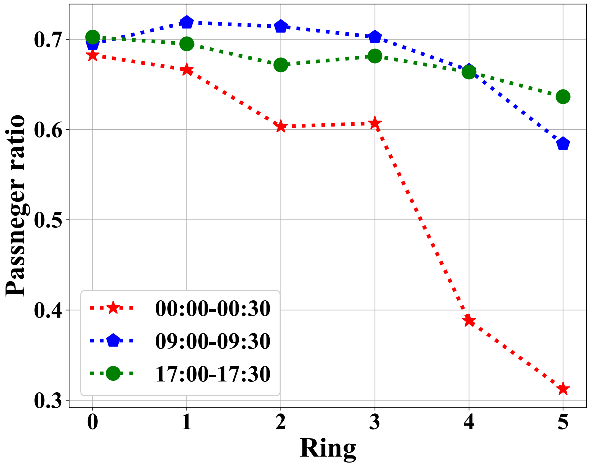

Fig. 14. δ varies among different rings in Chengdu.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgements

This research is jointly supported by the National Key Research and Development Program of China (2018YFB1600900) and National Natural Science Foundation of China (No. 72071013, 71890971/71890970, 71961137001). We also thank the participants at the 11th International Workshop on Computational Transportation Science for helpful comments.

References

Agatz, N. A., Erera, A. L., Savelsbergh, M. W., & Wang, X. (2011). Dynamic ride-sharing: a simulation study in metro atlanta. *Transport. Res. Part B, 45*(9), 1450–1464.
Agatz, N. A., Erera, A. L., Savelsbergh, M. W., & Wang, X. (2012). Optimization for dynamic ride-sharing: a review. *Eur. J. Oper. Res., 223*(2), 295–303.
Aïvodji, U. M., Gambs, S., Huguet, M. J., & Killijian, M. O. (2016). Meeting points in ridesharing: a privacy-preserving approach. *Transport. Res. Part C, 72*, 239–253.
Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc. Natl. Acad. Sci. Unit. States Am., 114*(3), 462–467.
Baldacci, R., Maniezzo, V., & Mingozzi, A. (2004). An exact method for the car pooling problem based on Lagrangean column generation. *Oper. Res., 52*(3), 422–439.
Bent, R., & Van Hentenryck, P. (2006). A two-stage hybrid algorithm for pick-up and delivery vehicle routing problems with time windows. *Comput. Oper. Res., 33*(4), 875–893.
Berbeglia, G., Cordeau, J. F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *Eur. J. Oper. Res., 202*(1), 8–15.
Burris, M. W., & Winn, J. R. (2006). Slugging in Houston—Casual carpool passenger characteristics. *J. Publ. Transp., 9*(5), 23–40.
Chan, N. D., & Shaheen, S. A. (2012). Ridesharing in North America: past, present, and future. *Transport Rev., 32*(1), 93–112.
Cherkesly, M., Desaulniers, G., & Laporte, G. (2015). A population-based metaheuristic for the pick-up and delivery problem with time windows and LIFO loading. *Comput. Oper. Res., 62*, 23–35.
Cordeau, J. F., & Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Ann. Opera. Res., 153*(1), 29–46.
Cortés, C., & Jayakrishnan, R. (2002). Design and operational concepts of high-coverage point-to-point transit system. *Transport. Res. Rec.: J. Trans. Res. Board., 1783*, 178–187.
Ferguson, E. (1997). The rise and fall of the American carpool: 1970–1990. *Transportation, 24*(4), 349–376.
Ferrari, E., Manzini, R., Pareschi, A., Persona, A., & Regattieri, A. (2003). The carpooling problem: heuristic algorithms based on savings functions. *J. Adv. Transport., 37*(3), 243–272.
Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M. E., Wang, X., & Koenig, S. (2013). Ridesharing: the state-of-the-art and future directions. *Transport. Res. Part B, 57*, 28–46.
Hosni, H., Naoum-Sawaya, J., & Artail, H. (2014). The shared-taxi problem: formulation and solution methods. *Transport. Res. Part B, 70*, 303–318.
Huang, Y., Jin, R., Bastani, F., & Wang, X. (2013). Large scale real-time ridesharing with service guarantee on road networks. *Proc. VLDB Endowm., 7*(14), 2017–2028.
Kaan, L., & Olinick, E. V. (2013). The vanpool assignment problem: optimization models and solution algorithms. *Comput. Ind. Eng., 66*(1), 24–40.
Kelley, K. (2007). Casual carpooling enhanced. *J. Publ. Transp., 10*(4), 119–130.
LeBlanc, D. E. (1999). *Slugging: the Commuting Alternative for.* Washington, DC: Forel Publishing.
Li, X., & Quadrifoglio, L. (2010). Feeder transit services: choosing between fixed and demand responsive policy. *Transport. Res. Part C, 18*(5), 770–780.
Long, J., Tan, W., Szeto, W. Y., & Li, Y. (2018). Ride-sharing with travel time uncertainty. *Transp. Res. Part B Methodol., 118*, 143–171.
Lyu, Y., Lee, V. C. S., Chow, C. Y., Ng, K. Y., Li, Y., & Zeng, J. (2017). R-sharing: rendezvous for personalized taxi sharing. *IEEE Access, 6*, 5023–5036.
Ma, S., Zheng, Y., & Wolfson, O. (2015). Real-time city-scale taxi ridesharing. *IEEE Trans. Knowl. Data Eng., 27*(7), 1782–1795.
Mahmoudi, M., & Zhou, X. (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: a dynamic programming approach based on state–space-time network representations. *Transport. Res. Part B, 89*, 19–42.

Masoud, N., & Jayakrishnan, R. (2017). A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem. *Transport. Res. Part B, 99*, 1–29.
Morency, C. (2007). The ambivalence of ridesharing. *Transportation, 34*(2), 239–253.
Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2010). Variable neighbourhood search for the dial-a-ride problem. *Comput. Oper. Res., 37*, 1129–1138.
Pelzer, D., Xiao, J., Zehe, D., Lees, M. H., Knoll, A. C., & Aydt, H. (2015). A partition-based match making algorithm for dynamic ridesharing. *IEEE Trans. Intell. Trans., 16*(5), 2587–2598.
Qian, X. W., Zhang, W. B., Ukkusuri, S. V., & Yang, C. (2017). Optimal assignment and incentive design in the taxi group ride problem. *Transport. Res. Part B, 103*, 208–226.
Qiu, F., Li, W., & Zhang, J. (2014). A dynamic station strategy to improve the performance of flex-itinerary transit services. *Transport. Res. Part C, 48*, 229–240.
Quadrifoglio, L., Dessouky, M. M., & Ordóñez, F. (2008). Mobility allowance shuttle transit (MAST) services: MIP formulation and strengthening with logic constraints. *Eur. J. Oper. Res., 185*(2), 481–494.
Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., & Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *Proc. Natl. Acad. Sci. Unit. States Am., 111*(37), 13290–13294.
Savelsbergh, M. W. P., & Sol, M. (1995). The general pickup and delivery problem. *Transport. Sci., 29*(1), 17–29.
Schittekat, P., Kinable, J., Sorensen, K., Sevaux, M., Spieksma, F., & Springael, J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. *Eur. J. Oper. Res., 229*(2), 518–528.
Spielberg, F., & Shapiro, P. (2000). Mating habits of slugs: dynamic carpool formation in the I-95/I-395 corridor of northern Virginia. *Transport. Res. Rec., 1711*, 31–38.
Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transport. Res. Part B, 82*, 36–53.
Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2016). Making dynamic ride-sharing work: the impact of driver and rider flexibility. *Transport. Res. E Logist. Transport. Rev., 91*, 190–207.
Varone, S., & Aissat, K. (2015). Routing with public transport and ride-sharing. In *Proceeding of the Sixth International Workshop on Freight Transportation and Logistics*.
Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., & Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Nature, 557*(7706), 534–538.
Wolfler, C., de Luigi, F., Haastrup, P., & Maniezzo, V. (2004). A distributed geographic information system for the daily carpooling problem. *Comput. Oper. Res., 31*(13), 2263–2278.
Xing, W., Agatz, N. A. H., & Erera, A. L. (2018). Stable matching for dynamic ride-sharing systems. *Transport. Sci.*, 1526–1554.
Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., & Ye, J. (2018). Large-scale order dispatch in on-demand ride-hailing platforms: a learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.
Zhan, X., Qian, X., & Ukkusuri, S. V. (2016). A graph-based approach to measuring the efficiency of an urban taxi service system. *IEEE Trans. Intell. Transport. Syst.*, 1–11.