


# A branch-and-cut-and-price algorithm for shared mobility considering customer satisfaction

Min Xu 

Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

## ARTICLE INFO

### Keywords:

Branch-and-cut-and-price  
Column generation  
Dial-a-ride  
Shared mobility  
Customer satisfaction

## ABSTRACT

This study determines the exact optimal fleet size, ride-matching patterns, and vehicle routes for shared mobility services (SMS) that maximize the profit of service operators considering ride-pooling and customer satisfaction. We make the first attempt to consider a nonlinear multivariate customer satisfaction function with respect to the features of the riders and the system under a ‘two riders-single vehicle’ ride-pooling scenario in a special case of dial-a-ride problem (DARP). A set packing model and a tailored branch-and-cut-and-price (BCP) approach are proposed to find the exact optimal solution of the problem. Unlike existing exact solution methods for DARP, we exploit the characteristic of the ride-pooling scenario and decompose the ride matching and vehicle routing in an effective two-phase method to solve the pricing problem of the BCP approach. Particularly, in Phase 1, feasible matching patterns subject to practical constraints are identified. In Phase 2, a heuristic and an exact label-correcting method with a bounded bi-directional search are sequentially employed to solve a new variant of elementary shortest path problem with time window (ESPTW) in a network constructed upon rides and feasible ride matching patterns identified in Phase 1. The labeling methods are further accelerated by a strengthened dominance test, the aggregate extension to other depots, and the decremental search space. Valid inequalities are also incorporated to further improve the upper bound. The proposed solution method is evaluated in randomly generated instances and the instances created from the real mobility data of Didi. Managerial insights are generated through impact analysis.

## 1. Introduction

The rise of shared mobility is reshaping the future of urban mobility (Shaheen and Cohen, 2019). It comes in many forms, including ride-sourcing, ride-sharing, and car-sharing. Distinctive as they are from one another today, these shared mobility business models are expected to be consolidated into two major types of door-to-door mobility service in the foreseeable future due to the advent of autonomous vehicle technology, i.e., shared mobility service without ride-pooling option (SMSw/oP) and the shared mobility service with ride-pooling option (SMSw/P) (Stocker and Shaheen, 2017). The high upfront purchase price of autonomous vehicles would make them more likely to be accessible to the broader public as part of a shared-fleet service model, instead of being privately owned. The service providers will thus be faced with fleet management decisions.

This study considers a profit-driven operator of a reservation-based multi-depot SMSw/P targeting at ordinary travelers, e.g., morning commuters, who are on average more time sensitive and have a narrow time window. For simplicity, we focus on the most-common ride-pooling

scenario, i.e., ‘two riders-single vehicle’, in which a maximum of two riders can be paired to be simultaneously served by a single vehicle and a rider may share part of his/her journey with multiple different strangers sequentially. In addition, the future SMS are expected to be customer-oriented and have to emphasize the service quality and ride experience to achieve wide acceptance. We impose the threshold in ride matching for a nonlinear customer satisfaction jointly determined by many features of the riders and ride-pooling system. Moreover, the operator allows the rejection of customers (subject to penalty) given the limited vehicle resources. Our objective is to determine the optimal fleet size, ride-matching patterns, and vehicle routes for SMSw/P by maximizing the profit of a service operator while taking the ride-pooling and customer satisfaction into consideration using an exact solution method. The considered problem is referred to as SMSP for short.

At its core, the SMSP of our interest is a variant of dial-a-ride problem (DARP) where a fleet of vehicles is dispatched to serve customer requests from pickup locations to drop-off locations within specific time windows (Cordeau, 2006; Cordeau and Laporte, 2007). Traditional application of DARP is non-profit public transit services for the elderly and disabled

E-mail address: [min.m.xu@polyu.edu.hk](mailto:min.m.xu@polyu.edu.hk).

<https://doi.org/10.1016/j.cor.2025.106998>

Received 13 June 2024; Received in revised form 7 December 2024; Accepted 27 January 2025

Available online 31 January 2025

0305-0548/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

with the objective to minimize the cost while satisfying all the requests. Another major application is the patient and intra-hospital transportations in ambulatory health care, which often involves complex constraints considering time urgency, equipment/staff compatibilities, etc. Recently, there has been a resurgence of interest in DARP due to the rise of shared mobility services for the general public (Ho et al., 2018). In what follows, we will mainly review the literature on DARP.

### 1.1. Literature review

Over the past decades, various solution approaches have been developed for DARP and these methods can be mainly classified into two categories: the exact methods and the heuristics/metaheuristics. Majority of research for DARP and its variants focused on developing fast heuristics/metaheuristics (Ho et al., 2018). For example, Cordeau and Laporte (2003) were among the first ones to develop a tabu search heuristic that incorporated several diversification strategies for the static multi-vehicle DARP. Ropke and Pisinger (2006) proposed an adaptive large neighborhood search (ALNS) heuristic for the pickup and delivery problem with time windows, which have inspired many studies of large neighborhood search heuristic for DARP and its variants. For example, Masson et al. (2014) allowed user transfer to different vehicles at specific locations during the trips in DARP and developed a solution method based on ALNS. Masmoudi et al. (2018) investigated a DARP with electric vehicles and battery swapping stations and proposed three efficient evolutionary variable neighborhood search algorithms. Zhao et al. (2022) investigated an interesting profitable DARP under time-dependent travel time with a single-vehicle. They proposed a hybrid *meta*-heuristic algorithm integrating ALNS and local search. For more information about different types of heuristics for DARP and its variants with various constraints, readers may refer to the survey paper by Ho et al. (2018).

Comparatively speaking, only a limited number of studies have focused on developing complex exact methods for DARP (Cordeau and Laporte, 2007; Ho et al., 2018). These exact methods are used to address a static DARP with all demand information known a priori. Though difficult to scale to large instances, exact methods can guarantee the highest solution quality and measure the optimality gap. The first exact branch-and-cut (B&C) method for multi-vehicle DARP was pioneered by Cordeau (2006) based on a three-index mixed integer programming (MIP) model. Ropke et al. (2007) proposed another B&C method for a two-index MIP model. The B&C algorithm has also been developed to address different DARP variants such as Parragh (2011) and Braekers and Kovacs (2016). Other than the B&C algorithm, the development of branch-and-price (B&P) as well as branch-and-cut-and-price (BCP) approaches that combine the advantages of B&P and B&C algorithms for DARP and its variants can be found in studies such as Parragh et al. (2015) and Luo et al. (2019). Those exact algorithms have also been proposed for many other problems, such as network design and operating room scheduling, and electric vehicle routing problems (Bargetto et al., 2023; Diao et al., 2024; Lam et al., 2022).

### 1.2. Research objective

In this study, in light of customers' concerns for sharing a ride with strangers in the context of SMS, we consider a special setting of DARP that allows the riders of at most two requests sharing their rides at a time, i.e., 'two riders-single vehicle' operation mode, with more emphasis on customer satisfaction. All the previous studies of exact methods for DARP and its variants considered the service quality by linear constraints such as imposing time windows and limiting the maximum ride time of each customer/client. Lavieri and Bhat (2019) have found that individuals' approval of strangers sharing the same vehicle is one of essential elements to the adoption of shared rides. Few studies, however, considered other influential factors for travel experience and incorporated nonlinear service quality metric such as the customer satisfaction to characterize the compatibility between pooled riders, which are highly relevant for shared

mobility services. To bridge the gap, we introduce a nonlinear multivariate customer satisfaction function with respect to the benefits and impedances of ride-pooling and the attributes of the concerned rider to ensure customers' ride experience when sharing a ride with a stranger. The proposed customer satisfaction function could be a more realistic and comprehensive characterization of customers' satisfaction to the shared mobility services. It serves as a nonlinear service quality metric (SQM), which is more general than linear SQMs such as the most frequently used 'maximum ride duration' in the studies for DARP. Moreover, many previous studies of exact methods for DARP and its variants often considered a single depot and/or required all customers to be served. We consider multiple depots and allow the rejection of customers in pursuit of profit maximization, which are practically relevant for future urban mobility services. Our objective is to develop an exact method to determine the fleet size, ride-matching patterns, and vehicle routes for SMS that maximize the profit of service operators.

To achieve the above objective, a set packing model will be built and a tailored BCP approach is subsequently developed to solve the model exactly. The pricing problem embedded in the BCP approach to determine the ride matching and vehicle routing strategy is a variant of NP-hard elementary shortest path problem with time windows, capacity, and pickup and delivery (ESPPTWCPD). Rather than employing a conventional approach for solving the ESPPTWCPD, we exploit the characteristic of 'two riders-single vehicle' ride-pooling scenario and propose an effective two-phase method that decomposes ride matching and vehicle routing to reduce the complexity of the pricing problem within BCP framework. We also propose tailored strategies such as the label extension rule, the dominance test, and the adaptive  $M$  value based on problem features to enhance the performance of BCP. Three speedup techniques, including a strengthened dominance test, the aggregate extension to other depots, and the decremental search space, are used to accelerate the algorithm. Valid inequalities are added to further strengthen the model and improve the upper bound. If the column generation method for solving the pricing problem produces a non-integer optimal solution, a branch-and-bound method is used to repeatedly solve the pricing problem until an integer solution is found. The proposed BCP approach can yield the optimal fleet sizing, ride matching, and vehicle routing decisions for SMSw/P. Numerical experiments on randomly generated instances and the instances created from the data from Didi are carried out to assess the efficacy of the proposed approach.

The remainder of this study is organized as follows. Section 2 elaborates on the assumptions, notations, and the description of the SMSP. A set packing model for the SMSP is formulated in Section 3. The BCP approach for solving the model is developed in Section 4. Through the numerical experiments of randomly generated instances and the instances created from the mobility data of Didi, the efficiency of the proposed solution methods and the impacts of ride-pooling and nonlinear SQM on system performance are evaluated in Section 5. Conclusions and future research directions are presented in Section 6.

## 2. Assumptions, notations, and problem description

This study considers a SMS provider who offers the reservation-based door-to-door mobility services with the option of ride-pooling using a fleet of homogeneous shared vehicles (SVs) within a service area. There are several depots clustered in the service area for SV parking in the low-demand period. These depots are grouped into a set denoted by  $W$  and we assume that SVs will return to their respective home depots after the operation period. Each depot  $w \in W$  has a limited parking capacity denoted by  $N_w$  and the location of depot  $w \in W$  is represented by  $s_w$ . Over a typical operation day, the service provider will receive a bunch of spatially and temporally distributed ride requests from customers. These orders are grouped into a set denoted by  $I$ . Each ride  $i \in I$  is characterized by a quadruple  $U_i = \{s_i^o, s_i^d, t_i^o, t_i^d\}$ , where  $s_i^o \in S$  represents the pickup location,  $s_i^d \in S$  stands for the drop-off location,  $t_i^o$  denotes the earliest departure time,  $t_i^d$  indicates the latest arrival time. The fixed cost of SV

per vehicle-day is denoted by  $AC$  and the operating cost per unit driving distance is represented by  $UC$ . To fully present the SMSw/P, the following subsections will elaborate on customer satisfaction, ride-matching pattern characterization, and vehicle route in Subsections 2.1–2.3, respectively. Kindly note that Subsections 2.1 and 2.2 describe how riders are pooled together, while Subsection 2.3 further discusses how to connect these solo rides and shared rides to form complete vehicle routes in order to facilitate the model building in Section 3. The notations used throughout this study are provided in the appendix.

### 2.1. Customer satisfaction

With the increasing penetration of shared mobility services, more and more customers will care about the level of service. To capture customers' satisfaction to the SMSw/P, each rider is assumed to be associated with a value of time (VOT) denoted by  $q_{i1}$ . We also quantify the travelers' acceptance and adoption of shared rides by associating each rider  $i \in I$  with willingness-to-pool (WTP) denoted by  $q_{i2}$  (Lavieri and Bhat, 2019). For profit maximization, we assume that the service operator allows the rejection of some orders as long as it can boost the overall profit. The penalty incurred by rejecting rider  $i$  is denoted by  $P_i$ . In the same vein, riders for whom no feasible ride-matching patterns are found will be assigned to travel individually. A service charge denoted by  $G_i$  will be placed on rider  $i \in I$  without ride-pooling, while a discount rate denoted by  $\nu$  applies for ride-pooling to compensate for the impedance of sharing rides with strangers. Therefore, the rider  $i \in I$  will enjoy a discounted service charge  $\hat{G}_i = \nu \cdot G_i$  if he/she shares the ride with another rider.

We assume that the satisfaction of customers in SMSw/P is jointly determined by the attributes of the concerned rider (e.g., the VOT and WTP) and the benefits and impedances of ride-pooling, including a discounted service charge, the duration of ride-pooling, and the additional ride time due to ride-pooling. In line with the customers' satisfaction function against service quality in the field of management and marketing, the satisfaction function of a rider  $i$  is assumed to be a nonlinear multivariate function with respect to the attributes of the concerned rider  $i$  and the benefits and impedances of ride-pooling with another rider  $j$ , i.e.,  $F_{ij}(\nu, \mathbf{q}_i, st_{ij}, et_{ij})$ , where  $\mathbf{q}_i$  is the vector of attributes of concerned rider  $i$ , e.g., the VOT  $q_{i1}$  and WTP  $q_{i2}$ ,  $st_{ij}$  is the duration of ride-pooling, and  $et_{ij}$  is the additional ride time compared to the time of traveling alone.

### 2.2. Ride-matching pattern characterization

As for the ride-pooling option in the SMSw/P, we first consider the

simplest ride-pooling scenario, i.e., 'two riders-single vehicle,' and assume that the vehicle will not initiate any new pickups before a shared ride is completed. In other words, a maximum of two riders are paired to be simultaneously served by a single vehicle. If a vehicle is in the process of serving two shared riders, it cannot pick up any other riders until both riders are dropped off. Given the itineraries and schedules of any two riders, denoted by Rider  $i$  and Rider  $j$ , a vehicle may pick up  $i$  and  $j$ , and drop  $i$  and  $j$  in sequence as shown in Fig. 1 (a), or do so in any of the other ways shown in Fig. 1 (b)–(d). Each way corresponds to a ride-matching pattern of rider  $i$  and  $j$ . We can see that there are at most four possible ride-matching patterns for the two riders. A ride-matching pattern denoted by  $(i-j-j-i)$  shown in Fig. 1 (a), is deemed feasible if the detour leads to a reduced travel distance compared with serving two riders individually while respecting the time window and satisfaction threshold of each rider, i.e.,

$$l_{s_i^o, s_j^o} + l_{s_j^d, s_i^d} < l_{s_i^o, s_i^d} \tag{1}$$

$$\max\{t_i^o, t_j^o - \tau_{s_i^o, s_j^o}\} \leq \min\{t_i^d - \tau_{s_j^d, s_i^d} - \tau_{s_i^o, s_j^o}, t_j^d - \tau_{s_j^d, s_j^o} - \tau_{s_i^o, s_j^o} - \tau_{s_j^d, s_i^d}\} \tag{2}$$

$$F_{ij}(\nu, \mathbf{q}_i, \tau_{s_i^o, s_j^o}, \tau_{s_j^d, s_i^d} + \tau_{s_i^o, s_j^o} + \tau_{s_j^d, s_i^d} - \tau_{s_i^o, s_i^d}) \geq \underline{E}_i \tag{3}$$

$$F_{ji}(\nu, \mathbf{q}_j, \tau_{s_j^o, s_i^d}, 0) \geq \underline{E}_j \tag{4}$$

where the distance and travel time between two locations, e.g., from  $s_i^o$  to  $s_i^d$ , are represented by  $l_{s_i^o, s_i^d}$  and  $\tau_{s_i^o, s_i^d}$  respectively;  $\underline{E}_i$  and  $\underline{E}_j$  are the minimal customer satisfaction of riders  $i$  and  $j$  that are assured by the service provider. Kindly note that the left-hand and right-hand sides of constraint (1) represent the travel distance of the matched ride for the two riders and the sum of travel distance of the two independent rides, respectively. Constraint (2) indicates that the matched ride should respect the time window of each rider. Constraints (3) and (4) enforce that the satisfactions of riders  $i$  and  $j$  are not less than the thresholds  $\underline{E}_i$  and  $\underline{E}_j$ , respectively. It can be seen that the fourth and the fifth variables in the satisfaction functions of constraints (3) and (4) represent the duration of ride-pooling and the additional ride time with respect to the original ride time of riders  $i$  and  $j$ , respectively. In addition to cost saving, time window and customer satisfaction constraint, other feasibility conditions or matching criteria related to level-of-service such as the threshold for waiting time and customers' particular preferences in matching can also be considered.

Following the above analysis, we continue to derive the feasibility conditions for the other three ride-matching patterns in Fig. 1 (b)–(d) as

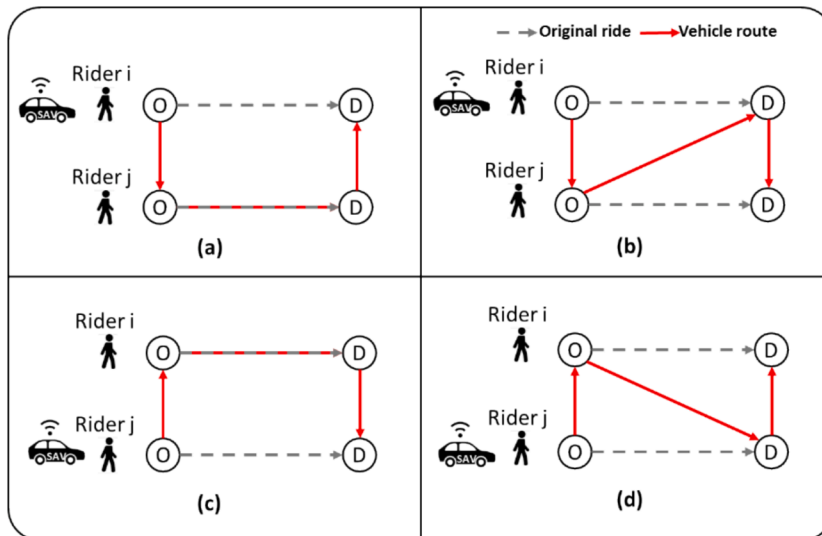


Fig. 1. Illustration for ride-matching patterns of two riders.

follows:

Ride-matching pattern  $(i-j-i-j)$  in Fig. 1 (b):

$$L_{s_j^o s_i^o} + L_{s_i^d s_j^d} + L_{s_i^d s_i^o} < L_{s_i^d s_i^o} + L_{s_j^o s_j^d} \quad (5)$$

$$\max\{t_i^o, t_j^o - \tau_{s_j^o s_i^o}\} \leq \min\{t_j^d - \tau_{s_j^o s_i^o} - \tau_{s_i^o s_j^d} - \tau_{s_i^d s_j^d}, t_i^d - \tau_{s_j^o s_i^o} - \tau_{s_j^o s_i^d}\} \quad (6)$$

$$F_{ij}(v, \mathbf{q}_i, \mathbf{p}_j, \tau_{s_j^o s_i^o}, \tau_{s_i^o s_j^d} + \tau_{s_i^d s_j^d} - \tau_{s_j^o s_i^d}) \geq \underline{E} \quad (7)$$

$$F_{ji}(v, \mathbf{q}_j, \mathbf{p}_i, \tau_{s_j^o s_i^o}, \tau_{s_i^o s_j^d} + \tau_{s_i^d s_j^d} - \tau_{s_j^o s_i^d}) \geq \underline{E} \quad (8)$$

Ride-matching pattern  $(j-i-i-j)$  in Fig. 1 (c):

$$L_{s_j^o s_i^o} + L_{s_i^o s_j^d} + L_{s_i^d s_j^d} < L_{s_i^d s_i^o} + L_{s_j^o s_j^d} \quad (9)$$

$$\max\{t_j^o, t_i^o - \tau_{s_j^o s_i^o}\} \leq \min\{t_j^d - \tau_{s_j^o s_i^o} - \tau_{s_i^o s_j^d} - \tau_{s_i^d s_j^d}, t_i^d - \tau_{s_j^o s_i^o} - \tau_{s_j^o s_i^d}\} \quad (10)$$

$$F_{ij}(v, \mathbf{q}_i, \mathbf{p}_j, \tau_{s_j^o s_i^o}, 0) \geq \underline{E} \quad (11)$$

$$F_{ji}(v, \mathbf{q}_j, \mathbf{p}_i, \tau_{s_j^o s_i^o}, \tau_{s_i^o s_j^d} + \tau_{s_i^d s_j^d} - \tau_{s_j^o s_i^d}) \geq \underline{E} \quad (12)$$

Ride-matching pattern  $(j-i-j-i)$  in Fig. 1 (d):

$$L_{s_j^o s_i^o} + L_{s_i^o s_j^d} + L_{s_i^d s_j^d} < L_{s_i^d s_i^o} + L_{s_j^o s_j^d} \quad (13)$$

$$\max\{t_j^o, t_i^o - \tau_{s_j^o s_i^o}\} \leq \min\{t_j^d - \tau_{s_j^o s_i^o} - \tau_{s_i^o s_j^d} - \tau_{s_i^d s_j^d}, t_i^d - \tau_{s_j^o s_i^o} - \tau_{s_j^o s_i^d}\} \quad (14)$$

$$F_{ij}(v, \mathbf{q}_i, \mathbf{p}_j, \tau_{s_j^o s_i^o}, \tau_{s_i^o s_j^d} + \tau_{s_i^d s_j^d} - \tau_{s_j^o s_i^d}) \geq \underline{E} \quad (15)$$

$$F_{ji}(v, \mathbf{q}_j, \mathbf{p}_i, \tau_{s_j^o s_i^o}, \tau_{s_i^o s_j^d} + \tau_{s_i^d s_j^d} - \tau_{s_j^o s_i^d}) \geq \underline{E} \quad (16)$$

In real-world shared mobility services, there are also cases when three or more riders are served sequentially while still respecting the vehicle capacity constraint. In these cases, customer satisfaction function of rider  $i$  will become  $F_{i^*}(v, \mathbf{q}_i, st_{i^*}, et_{i^*})$ , where  $st_{i^*}$  is the duration of ride-pooling with all the riders that are pooled with rider  $i$  during rider  $i$ 's trip, and  $et_{i^*}$  is the additional ride time of rider  $i$  compared to the time of traveling alone. The ride-matching pattern is feasible if the detour leads to a reduced travel distance compared with serving each rider individually while respecting the time window and satisfaction threshold of each rider.

### 2.3. Vehicle route

For ease of presentation, we refer to each ride-matching pattern as a shared ride. Vehicle route delineates the sequence of solo rides and the shared rides assigned to an SV as well as the relocation of an idle SV. An SV may serve several customers during the daily operation period, and vehicle relocation may be implemented between any two adjacent solo rides or shared rides to ensure that they are connected seamlessly. For ease of elaboration, we refer to the series of solo or shared rides and relocations underwent by an SV as the route of that vehicle. A vehicle route is feasible if the feasibility conditions of all covered rides, including solo rides and shared rides, are satisfied. An SV route  $r$ , which consists of a depot  $w$  and a series of solo rides and feasible shared rides sorted in an ascending order in terms of their departure times, i.e.,  $i_1, i_2, \dots, i_n$ , and several relocations linking these orders, can be represented by

$$r = w \rightarrow s_{i_1}^o \rightarrow s_{i_1}^d \Rightarrow s_{i_2}^o \rightarrow s_{i_2}^d \Rightarrow s_{i_3}^o \rightarrow s_{i_3}^d \Rightarrow \dots \Rightarrow s_{i_n}^o \rightarrow s_{i_n}^d \rightarrow w \quad (17)$$

where the single and double lined arrows denote the route segment serving solo/shared rides and the vehicle relocation, respectively. Suppose we have a total of 6 rides in a shared mobility system. These rides are represented by R1-R6 in an ascending order of their departure times.

Fig. 2 illustrates an SV route originating from Depot 1 and returning back to the same location after going through 2 solo rides (i.e., R1 and R6) and 1 shared ride (i.e., R2 paired with R3) and 2 relocations (i.e., Location 2  $\rightarrow$  3, 6  $\rightarrow$  4). The feasible route is "Depot 1  $\rightarrow$  Location 1  $\rightarrow$  Location 2  $\rightarrow$  Location 3  $\rightarrow$  Location 6  $\rightarrow$  Location 4  $\rightarrow$  Location 5  $\rightarrow$  Depot 1".

The diverse ride-matching patterns and the nonlinearity of customers' satisfaction function motivate us to formulate the SMSP based on SV route. As the vehicle dispatching strategy has already been reflected in SV route, the objective of the SMSP is to maximize the daily profit of the SMS providers by finding the optimal set of feasible routes in which riders are paired satisfactorily, SVs are assigned and relocated appropriately, and the selected solo/shared rides are served successfully. There is no doubt that the proposed SMSP is NP-hard because the simplest form of vehicle routing problem (VRP) with vehicle capacity of 1 as the special case of SMSP is a well-known NP-hard problem.

### 3. Set packing model

Let  $R$  denote the set of all the feasible routes; then the proposed SMSP is formulated by the following set packing model:

[SMSP]

$$\max_{x_r} \sum_{r \in R} \left( R_r + \sum_{i \in I} \delta_r^i P_i \right) x_r \quad (18)$$

subject to

$$\sum_{r \in R} \delta_r^i x_r \leq 1, \quad \forall i \in I \quad (19)$$

$$\sum_{r \in R} \theta_r^w x_r \leq N_w, \quad \forall w \in W \quad (20)$$

$$x_r \in \{0, 1\}, \quad \forall r \in R \quad (21)$$

where  $R_r$  is the amortized net profit of vehicle route  $r$  calculated by  $R_r = G_r - UC \bullet L_r - AC$ , in which  $G_r$  and  $L_r$  denote the total service charge of all covered rides served by route  $r$  and the total traveling distance of route  $r$ , respectively;  $x_r$  is the binary decision variable that equals 1 if the optimal route of a SV in the fleet is  $r$  and 0 otherwise;  $\delta_r^i$  is the coefficient that equals 1 if ride  $i$  is served by a SV through route  $r$ , and 0 otherwise; and  $\theta_r^w$  is the coefficient that equals 1 if the vehicle route  $r$  starts and ends at depot  $w$ , and 0 otherwise.

Note that the amortized daily profit of service operator is  $\sum_{r \in R} (R_r + \sum_{i \in I} \delta_r^i P_i) x_r - \sum_{i \in I} P_i$ . The objective of SMSP is to maximize the daily profit of service operator which is equivalent to Eq. (18). We consider the penalty incurred by rejecting riders in the objective function to incorporate the long-term negative impact of customer denial on service profitability. This will lead to a general model because by setting the penalty to 0, the model will reduce to the case where the penalty is not considered. Constraint (19) ensures that each ride is served at most once. Constraint (20) limits that the number of vehicles for each depot does not exceed the depot parking capacity.

Note that the above set packing model is formulated upon a given set of feasible routes. Subsection 2.2 has outlined the feasibility conditions of the shared rides along these routes. By the above approach, the optimization of the fleet size, ride-matching patterns, and vehicle routes considering ride-pooling and customer satisfaction is equivalent to seeking, among all the feasible routes, the most 'profitable' one for each SV in the fleet and accordingly the total number of them, such that every ride is 'covered' either individually or as a shared ride by the selected routes at most once. However, since the model [SMSP] has a great number of columns, which are difficult to be formulated explicitly, it cannot be solved by MIP solver. Therefore, in the next section, we will design a BCP approach, a leading exact algorithm for solving many classes of VRP (Barnhart et al., 1998; Costa et al., 2019).

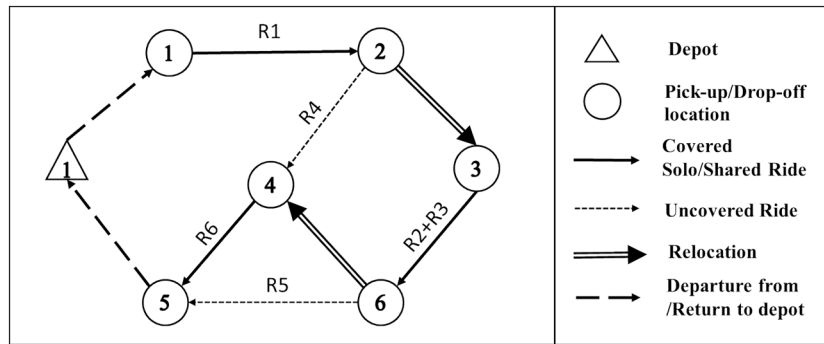


Fig. 2. An example of SV route.

#### 4. Branch-and-cut-and-price approach

Branch-and-cut-and-price (BCP) approach is essentially a branch-and-bound (B&B) framework in which the linear relaxation problem at each B&B node is iteratively solved by column generation (i.e., pricing problem) and strengthened by cutting planes. To apply the BCP approach for SMSP, we will first create initial columns for the linear relaxation of model [SMSP], referred to as master problem (MP) and formulate the restricted master problem (RMP) with a subset of routes in  $\bar{R} \subset R$  at the root node. The optimal dual values of RMP will be used to formulate the pricing problem to generate feasible columns. We propose a customized two-phase method to address the pricing problem and a primal-dual stabilization strategy to mitigate the tailing-off effect suffered by column generation, detailed in Subsections 4.1 and 4.2, respectively. If some columns are found by the pricing problem, we add these columns into RMP and solve it again. The iterative process stops when no column with positive reduced cost can be found, which implies that we have obtained the optimal solution to the MP. The MP will be further iteratively strengthened by valid inequalities identified from the fractional solution to the incumbent MP in Subsection 4.3 until there is no valid inequality. If the optimal solution to the reinforced MP is still fractional and is larger than the incumbent global lower bound of the problem, a branching strategy proposed in Subsection 4.4 will be used to branch this node into two child nodes; and otherwise, the corresponding branch will be pruned and the global lower bound may be updated. Again, a new node in the B&B will be explored in the same way described above until all nodes in the B&B tree have been examined.

##### 4.1. Pricing problem

Let  $\pi_i, \forall i \in I$  and  $\rho_w, \forall w \in W$  denote the dual variables corresponding to Constraint (19) and (20), respectively. The pricing problem for the MP of model [SMSP], named by [SMSP-PP], is presented as follows:

[SMSP-PP]

$$p^* = \max_{r \in \bar{R}} R_r + \sum_{i \in I} \delta_i^s P_i - \sum_{i \in I} \delta_i^d \pi_i - \sum_{w \in W} \theta_w^w \rho_w \quad (22)$$

The objective of the pricing problem is to find the route with the maximum profit in the network with an additional revenue  $P_i - \pi_i$  from each covered ride and an additional revenue  $-\rho_w$  for originating from depot  $w \in W$  among the rest routes. The existence of time window may induce cycles. This is particularly true for problem with wide time window, although it is less likely to happen for level-of-service-emphasized transportation systems where customers often have very narrow time windows. According to the definition of vehicle route in Subsection 2.3, the pricing problem by nature is a variant of ESPPTWCPD, which is NP-hard in the strong sense (Dror, 1994). Instead of using a general approach for solving the ESPPTWCPD (as often done in studies for DARPs), we propose a customized two-phase method that decomposes the ride matching and vehicle routing. Specifically, in Phase

1, feasibility matching patterns between any two or more rides are identified. After the feasibility check for each ride matching pattern, we obtain the attributes of feasible ride-matching patterns for processing in the next phase. For example, the ride-matching pattern in Fig. 1 (a) can be equivalently viewed as a ride associated with the origin  $s_i^o$ , destination  $s_i^d$ , a revenue  $\widehat{G}_i + \widehat{G}_j$ , a distance  $l_{s_i^o s_j^o} + l_{s_j^o s_i^d} + l_{s_i^d s_j^d}$ , and a time window  $[\max\{t_i^o, t_j^o - \tau_{s_i^o s_j^o}\}, \min\{t_j^d - \tau_{s_j^o s_i^d} - \tau_{s_i^o s_j^o}, t_i^d - \tau_{s_j^o s_i^d} - \tau_{s_i^o s_j^o} - \tau_{s_i^d s_j^d}\}]$ . In Phase 2, a labeling method is iteratively employed to solve a new variant of elementary shortest path problem with time window (ESPPTW) in a network constructed upon feasible ride matching patterns identified in Phase 1. The decomposition of ride matching and vehicle routing is motivated by several considerations:

- (i) The operation mode and complex feasibility check of ride matching patterns, e.g., Eqs. (1)-(4), entails an additional attribute of the label to track the rider that has been pooled with other riders, probably making the available labeling method for conventional ESPPTWCPD more computationally intensive if solved directly;
- (ii) The pre-generation of ride-matching pattern in Phase 1 can further relax the vehicle capacity constraint and reduce the dimension of resources to be considered in the labeling algorithm for ESPPTW;
- (iii) Since at most two riders are simultaneously pooled together, and the time windows are averagely narrow, the pre-generation of ride matching patterns is plausible. The restrictions imposed on these patterns further eliminate the number of feasible patterns and accordingly the size of network for solving the ESPPTW in Phase 2;
- (iv) The BCP method requires the pricing problem to be solved many times. It is desirable that few modifications are needed in the subsequent and repetitive labeling method in Phase 2 once all feasible matching patterns are identified in Phase 1.

##### 4.1.1. Ride-matching pattern pre-generation (Phase 1)

We will first discuss the ride-matching pattern with two riders. In theory, if we have  $|I|$  rides, then there will be  $2 \cdot |I| \cdot (|I| - 1)$  ride pairs at most to be generated. Nevertheless, in practice, the number of feasible ride pairs can be far smaller than this value. Note that the feasibility of a ride-matching pattern depends on both the spatial and temporal consistency between two rides. This means that a ride can only be paired with another ride that has a common period of travel at least. Given the narrow time window of travelers who generally prefer ad-hoc instant services in the context of urban mobility, the chance of pairing may not be that high. To generate these feasible ride pairs more efficiently, we will first sort all rides in ascending order in terms of the earliest departure time and check them in sequence. For example, suppose we aim to generate all ride pairs that include the first ride and the other rides.

We will check the second ride till the last ride in the sequence. For each ride, before we perform the feasibility check in Eqs. (1)-(16), we will examine whether this ride potentially has a same travel period with the first ride. If not, the generation process can be stopped because all the subsequent rides will definitely have no common travel time with the first ride.

After obtaining the set of feasible ride-matching patterns with two riders, referred to as RM-2, we can further generate the ride-matching pattern with three or more riders by using efficient insertion heuristics (Campbell and Savelsbergh, 2004; Gendreau et al., 1998) in an iterative way. Our proposed insertion heuristic will mainly check the feasibility of the ride-matching arrangement. Specifically, we will start by generating the ride-matching patterns with three riders, named RM-3, and group all those feasible RM-3 in a set. Given the set of RM-3, we will further implement the insertion heuristic to generate the ride-matching patterns with four riders, and so on. For example, let us consider the generation of RM-3 by the insertion of a ride into a RM-2. Intuitively, we will check all the possible insertion positions for both pick-up and drop-off operations of the rider, iterating from the first position to the last one in the possible insertion position sequence. For a specific pick-up operation insertion position, we will first examine whether inserting the drop-off operation of this rider would exceed the vehicle capacity. If it is violated, the checking process for the following drop-off operation insertion positions can be terminated, as all the subsequent drop-off operation insertion positions will definitely lead to violations. If no violations occur, we will proceed to assess the feasibility of travel distance and time windows and check the customers' satisfaction with the proposed insertion.

Let  $\beta$  denote the ride-matching pattern. For each generated feasible ride-matching pattern  $\beta$ , we will obtain its attributes, including the set of covered rides  $I_\beta$ , the first boarding ride  $\bar{l}_\beta$ , the last alighting ride  $\underline{l}_\beta$ , travel duration  $\tau_\beta$ , net profit  $G_\beta := \sum_{i \in I_\beta} \widehat{G}_i - UC \cdot l_\beta$ , where  $l_\beta$  denotes the travel distance, the earliest departure time  $\bar{t}_\beta := \max_{i \in I_\beta} \{t_i^d - \bar{\tau}_i\}$ , where  $\bar{\tau}_i$  is the travel duration of the ride-matching pattern till the pick-up of ride  $i$ , the latest arrival time  $\underline{t}_\beta := \min_{i \in I_\beta} \{t_i^d - \underline{\tau}_i\}$ , where  $\underline{\tau}_i$  is the travel duration of the ride-matching pattern till the drop-off of ride  $i$ .

#### 4.1.2. Labeling methods for solving the ESPPTW variant (Phase 2)

Even if we have reduced the ESPPTWCPD to an ESPPTW variant by ride-matching pattern pre-generation, the ESPPTW itself is still NP-hard (Dror, 1994). Following the hierarchical procedure proposed by Desaulnier et al. (2008), we will call two pricing algorithms, i.e., a heuristic labeling method and an exact labeling method in sequence, to solve the ESPPTW problem. For ease of exposition, in the following subsections, we will elaborate them in the reverse order with respect to their execution. Usually, at each column generation iteration, we identify a number of columns with positive reduced cost (for maximization problem) and then add them into the current RMP. A pricing algorithm is called only if the previous pricing algorithm cannot find a pre-specified target number of columns.

**4.1.2.1. Exact labeling method.** We propose a tailored label-correcting algorithm with a bounded bi-directional search to solve the ESPPTW variant. Label-correcting algorithm is a widely used method for solving the elementary shortest path problem (Feillet et al., 2004; Irnich and Desaulniers, 2005). The bounded bi-directional search was proposed by Righini and Salani (2006), in which forward and backward partial paths are first generated respectively from the depot and then join together to form complete paths. Different from conventional approaches for ESPPTW problem where each node in the network represents only one ride, we derive a customized labeling method for a generic variant of ESPPTW where each node may represent two or more rides as a shared ride. Details are elaborated as follows.

##### (1) Network construction

To apply the label-correcting algorithm, we will first create a copy of the depots grouped in set  $W$  and construct a pseudo-network denoted by

$G = (N, A)$ , where  $N := I \cup \Theta \cup W \cup W'$  in which  $\Theta$  denotes the set of ride-matching patterns generated in Subsection 4.1.1, and  $A$  is the set of time-feasible links connecting these nodes in  $N$ . Any node  $n \in N$  is associated with a node cost  $c_n$ , a node service duration  $\tau_n$ , the index of the ride  $\bar{l}_n$ , referred to as 'start-ride', of which the service at node  $n$  starts at the pick-up location, the index of the ride  $\underline{l}_n$ , referred to as 'end-ride', of which the service at node  $n$  ends at the drop-off location, the set of rides included in the node  $\Delta_n$ , and a time window  $[\underline{t}_n, \bar{t}_n]$  within which the service of node  $n$  must start.

Particularly, any individual ride  $i \in I$  represented by a node in the network is associated with the node cost  $-(G_i - UC \cdot l_{s_i^d} + P_i - \pi_i)$ , the service duration  $\tau_{s_i^d}$ , the start-ride  $i$ , the end-ride  $i$ , the set of served rides  $\{i\}$ , and the time window  $[t_i^d, t_i^d - \tau_{s_i^d}]$ . Any feasible ride matching pattern  $\beta$  is represented by a node in the network associated with the node cost  $-(G_\beta + P_i + P_j - \pi_i - \pi_j)$ , the service duration  $\tau_\beta$ , the start-ride  $\bar{l}_\beta$ , the end-ride  $\underline{l}_\beta$ , the set of served rides  $I_\beta$ , and the time window  $\{\bar{t}_\beta, \underline{t}_\beta\}$ . Kindly note that because we address the pricing problem by solving a variant of ESPPTW in the pseudo-network, we need to define the cost of solo-ride node as  $-(G_i - UC \cdot l_{s_i^d} + P_i - \pi_i)$ , which is calculated by the negative profit  $-(G_i - UC \cdot l_{s_i^d})$  minus the additional revenue  $P_i - \pi_i$  as required by Eq. (22). Likewise, the cost of ride-matching pattern node in the pseudo-network will be  $-(G_\beta + P_i + P_j - \pi_i - \pi_j)$ , which is calculated by the negative net profit of the ride-matching pattern  $-G_\beta$  minus the additional revenue  $-(P_i + P_j - \pi_i - \pi_j)$ . Any link  $(n, m) \in A$  denoting the relocation operation from the drop-off location of ride/ride pair  $n$  to the pick-up location of another ride/ride pair  $m$  is associated with link cost  $c_{nm} = UC \cdot l_{s_m^d, s_n^o}$  and travel time  $\tau_{nm} = \tau_{s_m^d, s_n^o}$ .

Each path originating from a depot  $w \in W$  and returning to physically the same depot  $w' \in W'$ , i.e., a copy of depot  $w$  in the constructed network, is a vehicle route, and the ride matching and vehicle routing strategy has been implicitly implied in the nodes and links of the optimal paths. The objective to find the routes with the largest profit is thus equivalent to solving the ESPPTW in the constructed network  $G$ .

##### (2) Label extension

The label-correcting algorithm with a bounded bi-directional search works by associating both forward labels grouped in the set  $L_n^{forward}$  and backward labels grouped in the set  $L_n^{backward}$  with each node  $n \in N$  of the network. Each forward and backward label represent the partial path originating from a depot and ending at the correspondent copy of the same depot, respectively.

We code any forward label  $k$  at the node  $n$  as  $l_k^{forward} := [S_k, \widehat{c}_k, \widehat{\tau}_k, \gamma_k, \kappa_k]$ , where  $S_k$  is the set of served rides, often referred to as customer resources in the literature;  $\widehat{c}_k$  is the cost of the corresponding partial path originating from a depot;  $\widehat{\tau}_k$  is the earliest service ending time at node  $n$ ;  $\gamma_k$  is the node index of label  $k$ , i.e.,  $\gamma_k := n$ ; and  $\kappa_k$  is the index of the end-ride, i.e.,  $\kappa_k := \underline{l}_n$ . Likewise, any backward label  $g$  at the node  $n$  will be coded as  $l_g^{backward} := [S_g, \widehat{c}_g, \widehat{\tau}_g, \gamma_g, \kappa_g]$ , where  $S_g$  is again the set of covered rides;  $\widehat{c}_g$  is the cost of the corresponding partial path ending at a depot;  $\widehat{\tau}_g$  is the minimum time that must be consumed since the start of the service at node  $n$  up to the arrival at the depot  $w$  not later than at time  $T = \max_{i \in I} \{t_i^d + \tau_{s_i^d, s_w}\}$ ;  $\gamma_g$  is the node index of label  $g$ , i.e.,  $\gamma_g := n$ ; and  $\kappa_g$  is the index of the start-ride, i.e.,  $\kappa_g := \bar{l}_n$ .

For conventional ESPPTW, labels are explored according to the nodes they are associated with (Feillet et al., 2004; Costa et al., 2019). However, since the labels of different nodes in our network may represent the partial paths ending at the same physical location, i.e., the labels of node  $n$  and node  $m$  with  $\underline{l}_n = \underline{l}_m$ , we will explore the labels by the end-rides (for forward label) or the start-rides (for backward label). This means once a ride is selected, the labels, probably at multiple nodes in the network, should be extended simultaneously, and both the forward and backward

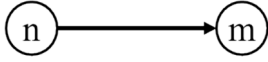


Fig. 3. Two nodes with a link.

labels will be extended. Specifically, the forward extension starts from an initial forward label at a depot with the set of served rides initialized as  $\emptyset$  and the earliest service ending time initialized as a sufficiently large negative number to ensure that the vehicle can depart earlier than the first ride. We use the example in Fig. 3 to illustrate the label extension rule.

Suppose we have a forward label  $k$  at node  $n$  represented by  $l_k^{forward} := [S_k, \hat{c}_k, \hat{\tau}_k, n, \hat{t}_n]$ ; then a forward label  $u$  denoted by  $l_u^{forward} := [S_k \cup \Delta_m, \hat{c}_k + c_m + c_{nm}, \max\{\hat{\tau}_k + \tau_{nm} + \tau_m, \bar{t}_m + \tau_m\}, m, \hat{t}_m]$  at node  $m$  will be generated if there exists a link  $(n, m) \in A$ . The new label is feasible if and only if we have  $S_k \cap \Delta_m = \emptyset$  and  $\max\{\hat{\tau}_k + \tau_{nm}, \bar{t}_m\} \leq \bar{t}_m$ , otherwise it is fathomed. As for the backward extension, we will start from an initial backward label at the copy of the correspondent depot with the set of served rides initialized as  $\emptyset$  and the other items initialized at 0. Again, for the example in Fig. 3, if we have a backward label  $g$  at node  $m$  represented by  $l_g^{backward} := [S_g, \hat{c}_g, \hat{\tau}_g, m, \bar{t}_m]$ ; then a backward label  $v$  denoted by  $l_v^{backward} := [S_g \cup \Delta_n, \hat{c}_g + c_n + c_{nm}, \max\{\hat{\tau}_g + \tau_{nm} + \tau_n, T - \bar{t}_n\}, n, \bar{t}_n]$  at node  $n$  will be generated if there exists a link  $(n, m) \in A$ . The new label is feasible if and only if we have  $S_g \cap \Delta_n = \emptyset$  and  $\max\{\hat{\tau}_g + \tau_{nm} + \tau_n, T - \bar{t}_n\} \leq T - \bar{t}_n$ , otherwise it is fathomed.

### (3) Dominance test

To avoid creating a huge number of labels, during the extension of labels, a dominance test is done to eliminate labels that cannot lead to an optimal solution. Unlike the dominance test for conventional ESPPTW that is performed rightly after a new label at a node is generated (Irnich and Desaulniers, 2005), we will conduct the dominance test for generated but not yet extended labels associated with the same end-ride index  $\kappa$ , after all labels generated from all nodes with the same end-ride (for forward labels) and start-ride (for backward labels) have been extended. Suppose we have two labels  $l_k := [S_k, \hat{c}_k, \hat{\tau}_k, \gamma_k, \kappa_k]$  and  $l_u := [S_u, \hat{c}_u, \hat{\tau}_u, \gamma_u, \kappa_u]$  such that  $\kappa_k = \kappa_u$ ; Then the former dominates the latter if the following conditions are satisfied:

$$S_k \subseteq S_u \quad (23)$$

$$\hat{c}_k \leq \hat{c}_u \quad (24)$$

$$\hat{\tau}_k \leq \hat{\tau}_u \quad (25)$$

and at least one of the inequalities is strict.

### (4) Label joining

To reduce the unnecessary labels and avoid the duplication of the paths, we consider the time as the critical resource and extend only forward labels and backward labels whose consumed time resources are less than half of the maximal time resource, i.e.,  $T/2$  (Righini and Salani, 2006). A forward label  $k$  at the node  $n$ , i.e.,  $l_k^{forward} := [S_k, \hat{c}_k, \hat{\tau}_k, n, \hat{t}_n]$  and a backward label  $g$  at node  $m$ , i.e.,  $l_g^{backward} := [S_g, \hat{c}_g, \hat{\tau}_g, m, \bar{t}_m]$  can join together to form a complete feasible path if we have  $S_k \cap S_g = \emptyset$  and  $\hat{\tau}_k + \hat{\tau}_g + \tau_{nm} \leq T$ . The cost of the resulting path is  $\hat{c}_k + \hat{c}_g + c_{nm}$ . The minimum cost among all complete paths is the optimal solution to the pricing problem.

Let  $\bar{L}_n^{forward}$  and  $\bar{L}_n^{backward}$  denote the set of un-extended forward and backward labels of node  $n$ , and  $E$  denote the set of rides to be explored. The end-rides/start-rides of nodes with newly generated forward/backward labels after an iteration are grouped in the set  $\bar{P}^f$  and  $\bar{P}^b$ , respectively. The above procedure of exact label correcting method for ESPPTW in the network  $G = (N, A)$  is summarized in Algorithm 1.

```

1 Initialize  $\bar{L}_w^{forward} \leftarrow \{(\emptyset, 0, -\infty, w, -)\}$ ;  $\bar{L}_w^{backward} \leftarrow \{(\emptyset, 0, 0, w, -)\}$ ;  $\bar{L}_n^{forward} \leftarrow \emptyset$  and
 $\bar{L}_n^{backward} \leftarrow \emptyset, n \in \mathcal{I} \cup \Omega$ ;  $\leftarrow \max_{i \in \mathcal{I}} \{t'_i + \tau_{s'_i}\}$ ;  $T$ 
2  $\mathcal{Z} \leftarrow \{w, w'\}$ ;
3 While  $\mathcal{Z} \neq \emptyset$  Do
4   Select  $i \in \mathcal{Z}$ ;  $\mathcal{Z} \leftarrow \mathcal{Z} \setminus i$ ;  $\bar{P}^f \leftarrow \emptyset$ ;  $\bar{P}^b \leftarrow \emptyset$ ;
5   For all  $n \in \mathcal{N}$  s.t.  $\hat{t}_n = i$  Do //forward extension
6     For all  $l_k^{forward} = [S_k, \hat{c}_k, \hat{\tau}_k, n, \hat{t}_n] \in \bar{L}_n^{forward}$  Do
7       If  $\hat{\tau}_k < \frac{T}{2}$ , then
8         For all  $m$  s.t.  $(n, m) \in \mathcal{A}$  Do
9           If  $S \cap \Delta_m = \emptyset$  and  $\max\{\hat{\tau}_k + \tau_{nm}, \bar{t}_m\} \leq \bar{t}_m$ , then
10             $\bar{L}_m^{forward} \leftarrow [S \cup \Delta_m, \hat{c}_k + c_m + c_{nm}, \max\{\hat{\tau}_k + \tau_{nm} + \tau_m, \bar{t}_m + \tau_m\}, m, \hat{t}_m]$ ;
11            If  $\hat{t}_k \notin \bar{P}^f$ , then
12               $\bar{P}^f \leftarrow \bar{P}^f \cup \hat{t}_k$ ;
13            EndIf
14          EndIf
15        EndFor
16      EndIf
17    EndFor
18     $\bar{L}_n^{forward} \leftarrow \bar{L}_n^{forward} \cup \bar{L}_m^{forward}$ ;  $\bar{L}_n^{backward} \leftarrow \emptyset$ ;
19  EndFor
20  For all  $m \in \mathcal{N}$  s.t.  $\bar{t}_m = i$  Do //backward extension
21    For all  $l_g^{backward} = [S_g, \hat{c}_g, \hat{\tau}_g, m, \bar{t}_m] \in \bar{L}_m^{backward}$  Do
22      If  $\hat{\tau}_g < \frac{T}{2}$ , then
23        For all  $n$  s.t.  $(n, m) \in \mathcal{A}$  Do
24          If  $S \cap \Delta_n = \emptyset$  and  $\max\{\hat{\tau}_g + \tau_{nm} + \tau_n, T - \bar{t}_n\} \leq T - \bar{t}_n$ , then
25             $\bar{L}_n^{backward} \leftarrow [S \cup \Delta_n, \hat{c}_g + c_n + c_{nm}, \max\{\hat{\tau}_g + \tau_{nm} + \tau_n, T - \bar{t}_n\}, n, \bar{t}_n]$ ;
26            If  $\bar{t}_g \notin \bar{P}^b$ , then
27               $\bar{P}^b \leftarrow \bar{P}^b \cup \bar{t}_g$ ;
28            EndIf
29          EndIf
30        EndFor
31      EndIf
32    EndFor
33     $\bar{L}_m^{backward} \leftarrow \bar{L}_m^{backward} \cup \bar{L}_n^{backward}$ ;  $\bar{L}_m^{forward} \leftarrow \emptyset$ ;
34  EndFor
35  For all  $i \in \bar{P}^f$  Do //forward dominance check
36     $\bar{L}_n^{forward} \leftarrow \text{DOMINANCE}(\bar{L}_n^{forward}, \bigcup_{m \in \mathcal{N}, s.t. \hat{t}_m = i} (\bar{L}_n^{forward} \cup \bar{L}_m^{forward}))$ ;
37  EndFor
38  For all  $i \in \bar{P}^b$  Do //backward dominance check
39     $\bar{L}_m^{backward} \leftarrow \text{DOMINANCE}(\bar{L}_m^{backward}, \bigcup_{n \in \mathcal{N}, s.t. \bar{t}_n = i} (\bar{L}_m^{backward} \cup \bar{L}_n^{backward}))$ ;
40  EndFor
41 EndWhile
42 For all  $i \in \mathcal{I}$  Do //label joining
43   For all  $l_k^{forward} \in \bigcup_{n \in \mathcal{N}, s.t. \hat{t}_n = i} (\bar{L}_n^{forward} \cup \bar{L}_m^{forward})$  Do
44     For all  $j \in \mathcal{I} \setminus i$  Do
45       For all  $l_g^{backward} \in \bigcup_{m \in \mathcal{N}, s.t. \bar{t}_m = j} (\bar{L}_m^{backward} \cup \bar{L}_n^{backward})$  Do
46         If  $S_k \cap S_g = \emptyset, \hat{\tau}_k + \hat{\tau}_g + \tau_{nm} \leq T$ , and  $\hat{c}_k + \hat{c}_g + c_{nm} < 0$ , then
47            $\mathcal{R}_i \leftarrow l_k^{forward} + l_g^{backward}$ ;
48         EndIf
49       EndFor
50     EndFor
51   EndFor
52 EndFor

```

4.1.2.2. *Accelerating strategies.* Three accelerating strategies will be employed to speed up the labeling method, which are described as follows:

**(1) Strengthened dominance test**

According to Feillet et al. (2004), if the traveling times satisfy the triangle inequality, the dominance conditions can be further relaxed and thus made more efficient in eliminating un-optimal labels by including in the set of served rides  $S_k$  the unreachable rides that cannot be served in any feasible extension of a given label due to the limitation of time resource.

**(2) Aggregate extension to other depots.**

Instead of solving the pricing problem for each depot individually, we solve the pricing problems for all depots at once using a single bounded directional labeling algorithm because two paths associated with different depots and covering the same rides and ride pairs in the same order only differ by their initial and final links connecting the depot with the first and the last respectively. This means that the labels for different depots only differ in cost by a constant value and thus the label extension, feasibility and dominance checks within the labeling algorithm have nothing to do with the depot. Suppose we have obtained a feasible complete path from depot  $w$  to its counterpart depot  $w'$  with the cost denoted by  $C_w$ . Then for any other depot  $\bar{w} \in W \setminus w$ , we will also have a feasible path covering the same rides and ride pairs in the same order with the cost  $C_{\bar{w}} = C_w + UC \cdot \left( l_{s_w^{so}}^{i_{first}} + l_{i_{last}^{s_w}}^{d} - l_{s_w^{so}}^{i_{first}} - l_{i_{last}^{s_w}}^{d} \right)$ , where  $i_{first}$  and  $i_{last}$  denote the first and last rides covered by the obtained path for depot  $w$ .

**(3) Decremental search space**

The third technique is called decremental search space (Boland et al., 2006). It is an iterative procedure that works on an iteratively enlarged set of served rides  $S_k$  in the configuration of labels. Basically, it starts by solving the pricing problem without considering any elementarity requirements, that is, without the component  $S_k$  in label. If the computed most profitable route is nonelementary, a subset  $\hat{S}_k \subseteq S_k$  that can only include the rides that are served more than once in the obtained path of the previous iteration is considered to forbid this path and the pricing problem is solved again. This iterative process is repeated until an elementary shortest path is found. In a column generation context, it can, however, be stopped before optimality when either positive reduced cost elementary paths are found or the length of the optimal path computed at an iteration is nonpositive (for maximization problem). Our implementation of the decremental search space technique is similar to the one described in Desautniers et al. (2008). Instead of starting decremental search with an empty set of  $S_k$  at each column generation iteration, we start it using the  $\hat{S}_k$  of the preceding iteration.

4.1.2.3. *Heuristic labeling method.* Although aforementioned speedup techniques have been used, the exact labeling method is still time consuming. In theory, there is no need to solve the pricing problem exactly except the last iteration of column generation and heuristics can help identify suitable columns in most iterations. This can greatly reduce the number of calls to the exact labeling algorithm, which often leads to a substantial reduction of the total computation time. Therefore, for the implementation of column generation in previous literature, heuristic algorithms are often used first to solve the subproblem (Costa et al., 2019; Desautniers et al., 2008). If the heuristics succeed to identify negative reduced cost columns, these columns are added to the RMP and another iteration is started. Otherwise, we invoke the exact algorithm for solving the subproblem to optimality, ensuring the exactness of the overall method. In this study, we will implement a heuristic version of the proposed exact labeling algorithm that relies on an aggressive dominance rule. It stipulates that a label dominates a label if conditions (24) and (25) hold. Hence, condition (23) on the customer resources is not tested. This strongly increases the chances of dominance, thus yielding much more dominated labels to be discarded, and accordingly

much fewer generated labels overall. Note that the label extension and feasibility check remain unchanged.

To sum up, the main novelty of the proposed method for the pricing problem in Subsection 4.1 lies in the way to construct a vehicle route that reduces the computational complexity of the original pricing problem. Traditional labeling method builds routes by connecting individual rides. Our method innovatively decomposes the ride matching and vehicle routing. It first builds many ride-matching patterns and then connects these ride-matching patterns to construct a route. The idea that works on ride-matching patterns and the labeling method for ride-matching patterns rather than a labeling method that works on each individual ride one by one is new. This can efficiently handle complex constraints related to ride pooling such as time windows, capacity, and pickup and delivery, and especially the nonlinear customer satisfaction considered in this study. Particularly, once we have generated the ride-matching patterns, the pricing problem will reduce from ESPPTWCPD to ESPPTW. Kindly note that ESPPTWCPD is more complex than ESPPTW due to the additional constraints, leading to a larger state space and more intricate feasibility checks. The efficiency gain from problem complexity reduction from ESPPTWCPD to ESPPTW will be significantly magnified when applied in the BCP approach, where the pricing problem needs to be solved repeatedly. Detailed discussions can be found in Subsection 4.1.

In addition to the above innovative idea, we also consider problem features to enhance the computational performance of the proposed two-phase method for the pricing problem. For example, In Phase 1 concerning the generation of ride-matching patterns with two riders, we will check the riders in ascending order of their departure times and see if any two rides have same travel period, and we will terminate the checking of subsequent riders if the current rider is infeasible to be pooled. Besides, when applying the labeling method in Phase 2, we need to pay attention to label extension and dominance test, since we now work on a network consisting of solo rides and shared rides. For example, for conventional ESPPTW, labels are explored according to the nodes they are associated with. However, since the labels of different nodes in our network may represent the partial paths ending at the same physical location, we will explore the labels by the end-rides (for forward label) or the start-rides (for backward label). We also propose adaptive  $M$  values to address the tailing-off effect in column generation (see Subsections 4.2 and 4.4).

## 4.2. Tailing-off effect

The column generation method will find an upper bound for SMSP. Nevertheless, this method was found slow convergence when approaching the optimal solution to the MP, often known as ‘tailing-off effect’. To mitigate this unfavorable effect, Ben Amor et al. (2006) proposed the use of dual-optimal inequalities in the context of cutting stock and bin packing problems. We extend this to the shared mobility service problem. Specifically, the column generation procedure will be pre-terminated based on the following proposition:

**Proposition 1.** *The column generation procedure can be terminated with the MP upper bounded by  $LpObj \times (1 + \epsilon_1)$  if the optimal solution to the pricing problem satisfies  $p^* \leq \frac{LpObj \times \epsilon_1}{M}$  where  $M \geq |I|$ , where  $LpObj$  denotes the optimal objective value of the RMP in the current iteration.*

**Proof.** *The MP can be augmented by a null constraint  $\sum_{r \in R} x_r \leq M$  as follows:*

$$\begin{aligned} & \text{[SMSP-R]} \\ & \max_{x_r} \sum_{r \in R} \left( R_r + \sum_{i \in I} \delta_i^r P_i \right) x_r \\ & \text{subject to} \end{aligned} \quad (26)$$



$$\sum_{r \in R} \delta_r^i x_r \leq 1, \quad \forall i \in I \quad (27)$$

$$\sum_{r \in R} \theta_r^w x_r \leq N_w, \quad \forall w \in W \quad (28)$$

$$\sum_{r \in R} x_r \leq M \quad (29)$$

$$x_r \geq 0, \quad \forall r \in R \quad (30)$$

The dual problem of model [SMSP-R] is formulated as follows:  
[SMSP-R *dual*]

$$\min_{(\pi_i, \rho_w, p) \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}} \sum_{i \in I} \pi_i + \sum_{w \in W} N_w \rho_w + M \times p \quad (31)$$

subject to

$$\sum_{i \in I} \delta_r^i \pi_i + \sum_{w \in W} \theta_r^w \rho_w + p \geq R_r + \sum_{i \in I} \delta_r^i P_i, \quad \forall r \in R \quad (32)$$

$$\pi_i \geq 0, \rho_w \geq 0, \lambda \geq 0, \quad \forall i \in I, w \in W \quad (33)$$

Let  $\pi_i^*, \forall i \in I$  and  $\rho_w^*, \forall w \in W$  be the optimal dual solutions corresponding to constraints (19) and (20) of the current RMP, respectively. Then  $(\pi_i^*, \rho_w^*, p^*)_{i \in I, w \in W}$  will be a feasible solution to the model [SMSP-R *dual*]. By substituting this feasible solution to the objective function (31), we have  $LpObj + M \times p^* \geq ObjR^*$ , where  $ObjR^*$  is the optimal objective value of the model [SMSP-R *dual*]. Let  $Obj^*$  be the optimal objective value of the model [SMSP-R]. Then it follows from the strong duality theorem that  $Obj^* = ObjR^* \leq LpObj + M \times p^* \leq LpObj \times (1 + \varepsilon_1)$ . Hence, we can terminate the column generation earlier while respecting the relative optimality  $\varepsilon_1$ . In the implementation of the pre-termination, an adaptive value of  $M$  can be adopted to improve the overall computational efficiency (see Subsection 4.4).□

### 4.3. Valid inequalities

In order to provide a better upper bound, we further strengthen the MP with valid inequalities. The combination of valid inequalities and column generation in B&B framework is referred to as BCP method. Over the past years, many studies have been conducted to propose and incorporate the many families of valid inequalities into B&P algorithm, i.e., the combination of column generation and B&B scheme, in the context of VRP (Costa et al., 2019). Among these cuts, the well-known subset row inequalities (SRIs) proposed by Jepsen et al. (2008) are promising to improve the computed upper bound in the B&P algorithm for the considered problem. We thus consider a special and the most popular case of SRIs, i.e., SRIs of size 3 (3-SRIs), which defines for subsets of three riders as follows:

$$\sum_{r \in R} \varphi_r^U x_r \leq 1, \quad \forall U \subseteq I, |U| = 3 \quad (34)$$

where  $U$  is the subset of riders with a cardinality of 3, and  $\varphi_r^U$  is a coefficient that equals 1 if route  $r$  serves at least two riders in set  $U$ , and 0 otherwise.

It can be seen that 3-SRIs suggest that in any feasible integer solution, at most one route that serves two or more riders in a subset of riders with the cardinality of 3, can be selected; otherwise there will be a ride served twice, thus violating constraint (19). To illustrate this, suppose we have two routes, i.e., P1 and P2, and both routes serve two riders in a set consisting of three riders, i.e., R1, R2, R3. Let R1 and R2 be the arbitrary two riders served by P1. No matter what rides are served by P2 (for example, R1 and R2, R1 and R3, or R2 and R3), P1 and P2 cannot be included in the optimal solution; otherwise, there will be rider(s) that are served twice, which violates the condition that each rider will be served at most once. Therefore, if the fractional solution to the MP

obtained by column generation violates some of the inequalities (34), we will add these violated 3-SRIs into the MP and restart the column generation process to further improve the upper bound.

Although it has been found that the SRIs can significantly improve the upper bound and thus result in a smaller branch tree, it increases the complexity of the labeling algorithm for solving the pricing problem. Specifically, let  $\Omega_C$  denote the set of all identified violated inequalities (34),  $C_b \in \Omega_C$  be any one identified violated inequality, and  $\eta_{C_b}$  be the corresponding dual variable, respectively. Then the pricing problem for the augmented MP of the model [SMSP], named by [SMSP-PP-SRIs], is presented as follows:

$$\text{[SMSP-PP-SRIs]} \\ p^* = \max_{r \in R \setminus \bar{R}} R_r + \sum_{i \in I} \delta_r^i P_i - \sum_{i \in I} \delta_r^i \pi_i - \sum_{w \in W} \theta_r^w \rho_w - \sum_{C_b \in \Omega_C} \varphi_r^{C_b} \eta_{C_b} \quad (35)$$

For each identified violated inequality  $C_b$ , an additional resource should be added to the definition of label to count the number of riders in  $C_b$  that have been served in the associated path. For example, the forward label  $k$  at the node  $n$  is now coded as  $l_k^{forward} := [S_k, \hat{c}_k, \hat{\tau}_k, \gamma_k, \kappa_k, D_k]$ , where  $D_k := \{D_k^b\}_{C_b \in \Omega_C}$  is the vector representing the number of riders (mod 2) in all violated inequities that have been served. As for the label extension rule, suppose we have a forward label  $k$  at node  $n$  with  $D_k$ ; then  $D_u$  of a forward label  $u$  at the subsequent node  $m$  in the network will be given by

$$D_u^b = (D_k^b + |\Delta_m \cap C_b|) \text{ mod } 2, \quad \forall C_b \in \Omega_C \quad (36)$$

According to the definition of the augmented label and the revised pricing problem, the cost of the forward label  $u$  at node  $m$  will be updated to  $\hat{c}_k + c_m + c_{nm} + \sum_{C_b \in \Omega_C: D_k^b + |\Delta_m \cap C_b| \geq 2} \eta_{C_b}$ . Condition (24) in the dominance test will be revised to be  $\hat{c}_k + \sum_{C_b \in \bar{\Omega}_C} \eta_{C_b} \leq \hat{c}_u$ , where  $\bar{\Omega}_C = \{C_b : \eta_{C_b} > 0 \wedge D_k^b > D_u^b\}$ . Similar treatments also apply to backward label extension and dominance test. When joining a forward label  $l_k^{forward} := [S_k, \hat{c}_k, \hat{\tau}_k, n, \hat{l}_n, D_k]$  and a backward label  $l_g^{backward} := [S_g, \hat{c}_g, \hat{\tau}_g, m, \hat{l}_m, D_g]$ , the cost of the resulting path is modified to be  $\hat{c}_k + \hat{c}_g + c_{nm} + \sum_{C_b \in \Omega_C: D_k^b + D_g^b - 1 \geq 2} \eta_{C_b}$ .

We use the simplest enumeration method to separate all 3-SRIs. Since the SRIs make the pricing problem even harder to solve and thus may negatively affect the overall efficiency of BP algorithm if the number of 3-SRIs are too large. Therefore, we limit the generation of 3-SRIs using the following rules: (i) Cuts are generated only when the number of total labels in the previous column generation is less than a threshold  $Q_{threshold}$  and the most violated cuts are violated by at least  $\bar{V}_{min}$ ; (ii) a maximum of  $C_{max}$  the most violated 3-SRIs are added; and (iii) a cut can be added only if it is violated by at least  $V_{min}$ . Last, to accelerate the re-optimization, we prematurely halt the column generation for 3-SRIs augmented linear programming relaxation if the current objective value is larger than the value of the best lower bound found so far. Kindly note that Proposition 1 still holds for the MP augmented by 3-SRIs.

### 4.4. Tailored hybrid branching scheme

Previous studies have proposed different branching strategies in the context of different problems (Costa et al., 2019). In this study, we propose to use a combination of three branching strategies, i.e., branching on the number of vehicles originating from a depot, branching on the flow through a ride, and branching on the flow of two rides, in the order of implementation. In particular, the strategy of branching on the flow of two rides was proposed by Ryan and Foster (1981). Specifically, if the solution to the MP is fractional, then we can identify a depot such that  $\sum_{r \in R} \theta_r^w x_r$  is fractional, or a ride  $i \in I$  such that  $0 < \sum_{r \in R} \delta_r^i x_r < 1$ , or a two rides  $i, j \in I$  such that  $0 < \sum_{r \in Q(i,j)} x_r < 1$ , where  $Q(i, j)$  denotes the set of routes covering the ride  $i$  (individually or in a ride-

matching pattern with other rides) and ride  $j$  (individually or in a ride-matching pattern with other rides) successively or serving ride  $i$  and ride  $j$  in a ride-matching pattern. Therefore, we develop the following hybrid branching scheme:

**Case 1:** The first branching rule concerns the values of  $\tilde{N}_w = \sum_{r \in R} \theta_r^w x_r$ . Specifically, if there exists a depot  $w \in W$  such that  $\tilde{N}_w$  is fractional, we will impose two branches: (i)  $\sum_{r \in R} \theta_r^w x_r \leq \lfloor \tilde{N}_w \rfloor$  and (ii)  $\lceil \tilde{N}_w \rceil \leq \sum_{r \in R} \theta_r^w x_r \leq N_w$ . For the former branch, the number of vehicles departing from this depot is no more than  $\lfloor \tilde{N}_w \rfloor$ , whereas for the latter branch, the number of vehicles departing from this depot is no less than  $\lceil \tilde{N}_w \rceil$ .

**Case 2:** The second branching rule concerns the values of  $\sum_{r \in R} \delta_r^i x_r$ . Specifically, we look for a ride  $i \in I$  such that  $0 < \sum_{r \in R} \delta_r^i x_r < 1$  and impose two branches on this ride: (i)  $\sum_{r \in R} \delta_r^i x_r = 1$  and (ii)  $\sum_{r \in R} \delta_r^i x_r = 0$ . For the former branch, ride  $i$  must be served, while for the latter branch, ride  $i$  is rejected.

**Case 3:** The third branching rule concerns the values of  $\sum_{r \in Q(i,j)} x_r$  for a pair of rides  $i, j$ . Specifically, we will search for a pair of rides  $i, j$  satisfying  $0 < \sum_{r \in Q(i,j)} x_r < 1$  and impose the following two branches: (i)  $\sum_{r \in Q(i,j)} x_r = 1$  and (ii)  $\sum_{r \in Q(i,j)} x_r = 0$ . For the former branch, ride  $j$  will be served immediately after ride  $i$  (i.e., ride  $j$  is picked up after the drop-off of ride  $i$ ), or ride  $i$  and ride  $j$  are served jointly in a ride-matching pattern by the same set of routes, while for the latter branch, ride  $i$  and ride  $j$  will not be served successively or jointly in a ride-matching pattern by the same set of routes.

During the course of the BCP algorithm, the MP at a node would be associated with upper  $\bar{N}_w$  and/or lower bound  $\underline{N}_w$  of capacity of each depot, a long list of rides in the sets of satisfied rides  $SI$  and rejected rides  $RI$ , as well as a long list of ride pairs in the sets of included pairs of rides  $IP$  and excluded pairs of rides  $EP$ . Accordingly, Constraints (19) and (20) will be updated to

$$\underline{N}_w \leq \sum_{r \in R} \theta_r^w x_r \leq \bar{N}_w, \quad \forall w \in W \quad (37)$$

$$\sum_{r \in R} \delta_r^i x_r = 1, \quad \forall i \in SI \quad (38)$$

$$\sum_{r \in R} \delta_r^i x_r = 0, \quad \forall i \in RI \quad (39)$$

$$\sum_{r \in R} \delta_r^i x_r \leq 1, \quad \forall i \in I \setminus (SI \cup RI) \quad (40)$$

where the upper  $\bar{N}_w$  and/or lower bound  $\underline{N}_w$  of a node will be the same with the upper and/or lower bound of its parent node, except for the case that this node is branched from the parent node based on Case 1 of the branching scheme (In this case, the upper bound of one node will be  $\lfloor \tilde{N}_w \rfloor$ , and the lower bound of another node will be  $\lceil \tilde{N}_w \rceil$ ). The updated pricing problem incorporating the above changes can be handled by network re-construction. Regarding the tailing-off effect, Proposition 1 holds with an updated value  $M = |I| - |RI| - |IP|$  at a node with sets  $RI$  and  $IP$ .

In summary, this study proposes an exact BCP approach, which is a sophisticated method used to solve integer programming problems, particularly those that are large and complex, such as vehicle routing and network design problems. It combines column generation and cutting planes within the B&B framework. The time complexity of the BCP algorithm is not straightforward to define in a simple closed form. In general, the number of nodes in the B&B tree can be exponential in the worst case. However, for many practical problems, the BCP approach can be very efficient, especially when the problem has a large number of variables but a relatively small number of constraints, as is the case with

the shared mobility problem in this study (see model [SMSPP]). We will conduct numerical experiments to evaluate its performance in the next section.

## 5. Numerical experiments

This section reports the results of computational experiments on randomly generated instances and instances based on the mobility data from Didi. First, we will elaborate the test instances used for our tests. We will then evaluate the performance of the proposed approach and examine the efficiency of the valid inequalities. Finally, impact analysis is conducted to explore how the ride-pooling option and nonlinear SQM affect the system performance of SMS. We code the algorithms in Matlab calling CPLEX on a personal computer with Intel (R) Core (TM) Duo 3.4 GHz CPU.

### 5.1. Test instances

Two sets of instances will be tested in the experiments. The first set is composed by randomly generated instances that mimic the travel pattern of commuters. To create these random instances, we first uniformly choose  $|S| = 1000$  location points from a 50 km by 50 km grid. The pick-up and drop-off locations of  $|D|$  ride requests, i.e.,  $s_i^o$  and  $s_i^d$ , are randomly chosen from the generated locations. Let  $dis(s_i^o, s_i^d)$  be the Euclidean distance between the pick-up location and the drop-off location of ride  $i$ . Given an average travel speed  $v = 30$  km/hr, the ride duration of trip  $i$  would be  $dis(s_i^o, s_i^d)/v$  hrs. Similarly, the locations of  $|W| = 4$  depots are uniformly distributed in the study area, i.e., (12.5, 12.5), (12.5, 37.5), (37.5, 12.5), and (37.5, 37.5) in the 50 km by 50 km grid. The capacity of each depot is 20. We consider three study periods with different demand patterns. One is the ordinary demand period of 12 h from 7 am to 7 pm with more riders required to depart in the first and last two hours. Specifically, if 7 am is taken as the time benchmark and the time duration is measured in minutes, the earliest departure time of each trip  $i$ , i.e.,  $t_i^o$ , is an integer randomly chosen with a 25 % probability of being from interval [0, 120], a 50 % probability of being from interval [120, 600], and a 25 % probability of being from interval (600, 720]. The second is the peak-hour demand period of 3 h from 7 am to 10 am for the morning peak-hours, or 4 pm to 7 pm for the afternoon peak-hours. The last is the transition period of 6 h from 10 am to 4 pm between the morning and afternoon peak-hours. The numerical experiments in the peak-hour and transition demand period can shed light on the computational performance of the proposed algorithm when applied in a rolling-time horizon under a dynamic and stochastic problem setting, which is one of our future research directions. Specially, for the morning peak-hour period, if 7 am is taken as the time benchmark, the departure time of each trip  $i$ , i.e.,  $t_i^o$ , is an integer randomly chosen with a 20 % probability of being from interval [0, 60], a 60 % probability of being from interval [60, 120], and a 20 % probability of being from interval (120, 180]. For the transition period, the departure time of each trip is randomly and uniformly generated from the interval [180, 540]. Let  $TW$  denote the slack time of all rides, i.e.,  $(t_i^d - t_i^o - dis(s_i^o, s_i^d)/v)$ ,  $\forall i$ , measured in minutes. The latest arrival time  $t_i^d$  can finally be inferred from the earliest departure time and the slack time.

The second set of instances is created by randomly sampling  $|D|$  ride requests from the historical trip records retrieved from Didi Gaiya Open data (<https://outreach.didichuxing.com/research/opendata/en/>) in Haikou, China in June 2017. We randomly choose a typical working day in June and the mobility dataset on that day contains about 60,000 trip records. Each trip record contains the information of pick-up and drop-off locations, the departure and arrival times. The area of Haikou is around 70 km<sup>2</sup> and its network structure extracted from OpenStreetMap is illustrated in Fig. 4. Dijkstra's shortest path algorithm is used to calculate the travel distance between any two points in the network. Data cleansing and pre-processing are conducted prior to the analysis.

Specifically, we remove the abnormal trip records if the trip duration is less than 1 min or larger than 2 h, or the travel distance is less than 0.2 km or larger than 50 km. Analogous to the aforementioned randomly generated instances, we consider three periods including an ordinary demand period (7 am-7 pm), a peak-hour demand period (7 am-10 am), and a transition period (10 am-4 pm). Again the ride duration is calculated by assuming an average travel speed  $v = 30$  km/h. A total of  $|W| = 4$  depots, each with a capacity of 20, are assumed to be located at

customer satisfaction function. The VOT is randomly and uniformly chosen from the set of normalized values  $\{0.1, 0.2, 0.3, \dots, 1\}$ , while the WTP is chosen as a uniformly random integer from the set  $\{1, 2, 3, 4, 5\}$ . Only ride-matching patterns with two riders are considered. Without loss of generality, the satisfaction function of a rider  $i$  is assumed to be multivariate concave function given by

$$F_{ij}(v, q_{i1}, q_{i2}, st_{ij}, et_{ij}) = 1 - \frac{(v/v_{\max})^2}{5} - \frac{(q_{i1}/q_{i1,\max})^2}{5} - \frac{(1 - q_{i2}/q_{i2,\max})^2}{5} - \frac{(st_{ij}/st_{\max})^2}{5} - \frac{(et_{ij}/et_{\max})^2}{5} \tag{41}$$

4 major intersections in the network as highlighted in Fig. 4.

For both sets of instances, we assume for simplicity that the SMS is charged by travel distance and the unit service charge is  $UG = \$1/\text{km}$ . Hence the service charge of ride  $i$  is calculated by  $G_i = UG \cdot \text{dis}(s_i^o, s_i^d)$ . A rider will enjoy 10 % discount upon the original service charge for a shared ride, i.e.,  $v = 0.9$  and  $\hat{G}_i = 0.9 \cdot G_i$ . The penalty of rejecting the ride  $i$ , i.e.,  $P_i$ , is assumed to be 30 % of the service charge of that trip. The operating cost per unit driving distance of SV is set to be  $UC = 0.2\$/\text{km}$ . The fixed cost of SV is set to be  $AC = 25$  per vehicle-day. We consider the VOT and WTP as the attributes of the concerned rider in the

where  $v_{\max}$ ,  $q_{i1,\max}$ , and  $q_{i2,\max}$  are the maximal values of discount rate, VOT, and WTP, and  $st_{\max}$  and  $et_{\max}$  are the maximal acceptable ride-pooling duration and additional ride time of rider  $i$ . We can see that the maximal customer satisfaction value is 1 and is achieved at an idealized scenario of  $v = 0$ ,  $q_{i1} = 0$ ,  $q_{i2} = 5$ ,  $st = 0$ , and  $et = 0$ ; and the customer satisfaction will always be non-negative if both the ride-pooling duration and additional ride time do not exceed the maximal acceptable values, i.e.,  $F_i \geq 0$  if  $st_{ij} \leq st_{\max}$  and  $et_{ij} \leq et_{\max}$ . Both the  $st_{\max}$  and  $et_{\max}$  are set to be 30 min. Unless stated otherwise, the threshold of customer satisfaction value  $\underline{F}$  for ride-matching pattern generation is



Fig. 4. Road network of Haikou obtained from OpenStreetMap.

assumed to be 0.5, and the slack time of all rides is assumed to be 10 min.

## 5.2. Assessment of solution methods

We now evaluate the overall performance of the proposed BCP method and the effectiveness of the valid inequalities, i.e., 3-SRIs, in obtaining the optimal integer solution in the above two sets of instances. Kindly note that without 3-SRIs, the proposed method becomes a B&P approach. Since the number of rides and the slack time may influence the computational efficiency of the solution methods, test instances associated with various combinations of the number of rides  $|I| \in \{10, 20, 30, 40, 50\}$  (for ordinary demand period)/ $|I| \in \{30, 60, 90, 120\}$  (for peak-hour and transition demand periods) and the slack time  $TW_{\max} \in \{5, 10, 15\}$  are used to test the performance of the proposed method with and without valid inequalities. The two variants of method are applied independently for the same set of instances. For a particular combination or scenario of  $|I|$  and  $TW_{\max}$ , five instances are randomly generated and the average results are reported for the ordinary, peak-hour, and transition demand periods, respectively. The relative optimality gap is controlled by  $\varepsilon_1$  and  $\varepsilon_2$ . By setting  $\varepsilon_1 = \varepsilon_2 = 0.0005$ , the overall relative optimality gap is about 0.001. The thresholds for selecting the violated 3-SRIs are set as follows:  $Q_{\text{threshold}} = 50,000$ ,  $\bar{V}_{\min} = 0.1$ ,  $C_{\max} = 30$ , and  $V_{\min} = 0.01$ . A limit of 2 h is imposed for solving each of these instances.

Tables 1-3 show the results of the proposed approach with and without 3-SRIs for these test instances corresponding to the ordinary, peak-hour, and transition demand periods, respectively. Each row corresponds to the average results obtained for the five instances for a particular scenario named as  $\langle r/d/o/p/t \rangle \langle |I| \rangle \langle TW_{\max} \rangle$ , where 'r' and 'd' are used for randomly generated instances and the instances created from the mobility data of Didi, respectively, whereas 'o', 'p', and 't' are used for the instances in ordinary, peak-hour, and transition demand periods, respectively, in the first column of the tables. We report several output parameters in the tables, including the number of solved linear relaxation instances within the time limit (#LPSolved), the number of solved instances within the time limit (#Solved), the number of instances solved at the root node (#SolvedR), the total CPU time to obtain the optimal integer solution on average (T\_CPU Time), the number of priced out columns (#Column), the number of generated cuts (#Cut) [for 'with 3-SRIs' only], and the number of nodes traversed (#Node) in the B&B search tree of instances that are solved to optimality.

We can see from Table 1 that most instances in the ordinary demand period can be solved to optimality within 2 h if the number of rides is no larger than 40. The CPU time for obtaining the optimal integer solution would increase rapidly as the number of rides increases. Sometimes finding a valid bound within the time limit is a difficult task. In addition to the ride number, the computational efficiency is also negatively affected by the slack time, especially for instances with a relatively larger number of rides, i.e.,  $|I|=40$  or 50. For example, from the right below part of Table 1, we notice that it takes 62.81 s on average to solve an instance with  $|I| = 40$  and  $TW_{\max} = 5$  min, while the average computation time dramatically increases to 1994.77 s for an instance with the same number of rides but a larger slack time, i.e.,  $TW_{\max}=15$  min. Moreover, the number of solved instances at the root node tends to decrease with an increased number of rides. Similar findings can be observed in the instances for the peak-hour and transition demand periods as well.

Among the three demand scenarios, the instances in the ordinary demand period are the most computationally intensive ones, probably because of the long time horizon. On the contrary, the instances in the peak-hour demand period are the easiest to solve. In fact, all the randomly generated instances in Table 2 are solved at the root node except one instance 'r-p-120-5'. The transition demand period lies in the middle. The results in Table 3 show that the proposed approach can solve some instances in the transition demand period with up to 120

rides. A further refined examination of the total CPU time in Tables 1-3 suggests that the computational efficiency of the proposed approach is significantly influenced by the length of time horizon. For example, the BCP approach takes 8.51 s and 26.91 s to solve an instance with  $|I| = 60$  and  $TW_{\max} = 5$  min in the peak-hour and transition demand period respectively, whereas the average CPU time increases greatly to 3721.45 s even for a smaller-sized instance with  $|I| = 50$  and  $TW_{\max} = 5$  min in the ordinary demand period. Note that the time horizons of peak-hour, transition, and ordinary demand period are 3 h, 6 h, and 12 h, respectively. The results favorably demonstrate the efficacy and potential of the proposed approach when applied in a rolling-time horizon for an online and dynamic setting.

Compared with the randomly generated instances, the instances of Didi appear easy to solve when the number of rides is small. However, this may not be true for large-sized instances. Take the ordinary demand period for example, Table 1 shows that the BCP method averagely takes 24.86 s to solve randomly generated instances with 10 rides and slack time of 5 min, whereas only 7.94 s are needed to address the same-sized instances of Didi. However, when the number of rides rises to 50, we find an obviously small number of instances of Didi solved to optimality within the time limit and an averagely longer CPU time for these limited number of solved instances. This phenomenon becomes more apparent in the instances of peak-hour demand period as shown in Table 2. It can be seen that the CPU time for randomly generated instances and the instances of Didi is comparable when the number of rides is no more than 60, while for larger instances, especially the instances with 120 rides, nearly 3 times of CPU time is required for solving the Didi instances compared with randomly generated instances of the same size. This may be attributed to the spatially clustered demand pattern of morning commuters in the mobility data from Didi, e.g., from residential areas to CBD during the morning peak hour, which creates more ride-pooling opportunities and thus more nodes in the network and accordingly more time to solve the pricing problem.

As for the valid inequalities, since we set some restrictions for the 3-SRIs generation, the 3-SRIs are generated only in a few instances. Those instances can potentially benefit from the 3-SRIs in the implementation of column generation. According to Tables 1-3, these instances include 'r-o-30-5', 'r-o-30-10', 'd-o-20-15', 'd-o-30-5', 'r-p-120-5', 'd-p-60-15', 'd-p-90-10', 'd-p-90-15', 'r-t-60-10', 'r-t-60-15', 'r-t-90-10', 'r-t-120-5', 'd-t-30-15', and 'd-t-60-15'. By comparing the results of these instances solved by the proposed approach with and without 3-SRIs, we can find that the 3-SRIs increase the likelihood of an instance being solved at the root node. For example, there are two additional instances of 'r-t-60-15' and one additional instance of 'r-o-30-10', 'd-o-20-15', 'd-p-60-15', 'd-p-90-15', 'r-t-120-5', and 'd-t-60-15' solved to optimality at the root node when we apply the 3-SRIs. The total number of columns and the computation time may also decrease with the help of 3-SRIs. A remarkable example is the instance 'd-p-60-15' where the CPU time has reduced significantly from 326.87 s to 15.64 s after applying the valid inequalities. Other than diminishing the nodes explored in the B&B tree and increasing the likelihood of instances being solved at the root node with less computation time, the valid inequalities can also help to solve some instances to optimality which cannot be solved otherwise. For example, more instances in 'r-t-90-10' and 'r-t-120-5' are solved thanks to 3-SRIs.

## 5.3. Impact analysis

In this subsection, we will examine the impact of ride-pooling option by comparing the results of SMSw/P and SMSw/oP. Given the high probability of ride-matching between peer riders in peak-hour demand period, instances created from Didi mobility dataset in that period will be used in the analysis. In particular, we compare the optimal fleet size, the profit, the number of satisfied rides, and the usage rate of SV of the two systems. The differences of these parameters are reported both in absolute value and in percentage in Table 4. It can be seen from the table

**Table 1**  
Comparison of the results for instances in ordinary demand period.

Instance	Without 3-SRIs						With 3-SRIs						
	#LPSolved	#Solved	#SolvedR	T_CPU Time (s)	#Column	#Node	#LPSolved	#Solved	#SolvedR	T_CPU Time (s)	#Column	#Cut	#Node
r-o-10-5	5	5	5	25.71	504	1.0	5	5	5	24.86	504	0	1.0
r-o-10-10	5	5	5	24.52	509	1.0	5	5	5	25.13	509	0	1.0
r-o-10-15	5	5	5	24.90	537	1.0	5	5	5	25.29	537	0	1.0
r-o-20-5	5	5	5	30.91	5,957	1.0	5	5	5	29.95	5,957	0	1.0
r-o-20-10	5	5	5	30.16	6,663	1.0	5	5	5	30.13	6,663	0	1.0
r-o-20-15	5	5	5	30.89	8,132	1.0	5	5	5	30.39	8,132	0	1.0
r-o-30-5	5	5	4	99.78	37,645	2.2	5	5	4	94.63	37,238	5.0	1.4
r-o-30-10	5	5	3	80.01	44,143	1.6	5	5	4	64.13	43,332	1.8	1.2
r-o-30-15	5	5	4	93.35	50,886	1.4	5	5	4	80.25	50,886	0	1.4
r-o-40-5	5	5	5	648.60	139,754	1.0	5	5	5	666.02	139,754	0	1.0
r-o-40-10	5	5	3	1382.89	157,888	1.8	5	5	3	1361.47	157,888	0	1.8
r-o-40-15	5	5	5	1691.85	187,641	1.0	5	5	5	1670.77	187,641	0	1.0
r-o-50-5	4	4	4	2772.61	229,654	1.0	4	4	4	2771.35	229,654	0	1.0
r-o-50-10	4	2	2	5073.44	286,334	1.0	4	2	2	5130.64	286,334	0	1.0
r-o-50-15	2	0	0	—	—	—	2	0	0	—	—	—	—
d-o-10-5	5	5	5	8.32	442	1.0	5	5	5	7.94	442	0	1.0
d-o-10-10	5	5	5	7.96	525	1.0	5	5	5	8.01	525	0	1.0
d-o-10-15	5	5	5	8.10	557	1.0	5	5	5	7.98	557	0	1.0
d-o-20-5	5	5	5	11.76	7,320	1.0	5	5	5	11.81	7,320	0	1.0
d-o-20-10	5	5	5	11.95	8,790	1.0	5	5	5	11.94	8,790	0	1.0
d-o-20-15	5	5	3	47.63	10,599	4.6	5	5	4	37.13	10,093	5.2	4.2
d-o-30-5	5	5	4	25.31	39,392	1.4	5	5	4	25.87	39,392	2.0	1.4
d-o-30-10	5	5	4	50.89	49,409	1.4	5	5	4	45.03	49,409	0	1.4
d-o-30-15	5	5	5	62.97	57,656	1.0	5	5	5	62.81	57,656	0	1.0
d-o-40-5	5	5	4	1030.81	128,478	2.2	5	5	4	943.15	128,478	0	2.2
d-o-40-10	5	5	4	2084.63	158,971	1.4	5	5	4	1994.77	158,971	0	1.4
d-o-40-15	5	5	5	2963.15	215,338	1.0	5	5	5	2971.89	215,338	0	1.0
d-o-50-5	4	1	1	3726.90	291,064	1.0	4	1	1	3721.45	291,064	0	1.0
d-o-50-10	2	1	1	6831.15	354,626	1.0	2	1	1	6842.59	354,626	0	1.0
d-o-50-15	0	0	0	—	—	—	0	0	0	—	—	—	—

Remarks: #LPSolved: the number of solved linear relaxation instances within the time limit. #Solved: the number of solved instances within the time limit. #SolvedR: the number of instances solved at the root node. T\_CPU Time: the total CPU time to obtain the optimal integer solution on average. #Column: the number of priced out columns. #Node: the number of nodes traversed in the B&B search tree. #Cut: the number of generated cuts.

**Table 2**  
Comparison of the results for instances in peak-hour demand period.

Instance	Without 3-SRIs						With 3-SRIs						
	#LPSolved	#Solved	#SolvedR	T_CPU Tim (s)	#Column	#Node	#LPSolved	#Solved	#SolvedR	T_CPU Time (s)	#Column	#Cut	#Node
r-p-30-5	5	5	5	2.79	294	1.0	5	5	5	2.57	294	0	1.0
r-p-30-10	5	5	5	2.63	489	1.0	5	5	5	2.62	489	0	1.0
r-p-30-15	5	5	5	2.73	680	1.0	5	5	5	2.69	680	0	1.0
r-p-60-5	5	5	5	5.62	1,466	1.0	5	5	5	5.63	1,466	0	1.0
r-p-60-10	5	5	5	6.04	2,323	1.0	5	5	5	6.00	2,323	0	1.0
r-p-60-15	5	5	5	6.68	3,848	1.0	5	5	5	6.72	3,848	0	1.0
r-p-90-5	5	5	5	11.51	3,328	1.0	5	5	5	11.24	3,328	0	1.0
r-p-90-10	5	5	5	13.31	6,320	1.0	5	5	5	13.32	6,320	0	1.0
r-p-90-15	5	5	5	17.69	11,187	1.0	5	5	5	17.98	11,187	0	1.0
r-p-120-5	5	5	4	67.78	7,363	21.8	5	5	4	56.02	6,735	7.6	11.8
r-p-120-10	5	5	5	118.31	15,045	1.0	5	5	5	117.93	15,045	0	1.0
r-p-120-15	5	5	5	359.40	29,132	1.0	5	5	5	360.13	29,132	0	1.0
d-p-30-5	5	5	5	3.60	210	1.0	5	5	5	3.44	210	0	1.0
d-p-30-10	5	5	5	3.48	301	1.0	5	5	5	3.51	301	0	1.0
d-p-30-15	5	5	5	3.60	467	1.0	5	5	5	3.62	467	0	1.0
d-p-60-5	5	5	5	8.47	2,019	1.0	5	5	5	8.51	2,019	0	1.0
d-p-60-10	5	5	5	7.41	3,259	1.0	5	5	5	7.45	3,259	0	1.0
d-p-60-15	5	5	4	326.87	6,325	17.4	5	5	5	15.64	6,251	1.4	1.0
d-p-90-5	5	5	5	13.79	6,915	1.0	5	5	5	13.60	6,915	0	1.0
d-p-90-10	5	5	4	119.11	11,518	10.6	5	5	4	102.43	11,132	7.0	7.0
d-p-90-15	5	5	4	180.51	19,955	4.2	5	5	5	131.67	15,473	1.0	1.0
d-p-120-5	5	5	5	127.29	14,169	1.0	5	5	5	126.23	14,169	0	1.0
d-p-120-10	5	5	5	319.10	27,795	1.0	5	5	5	323.03	27,795	0	1.0
d-p-120-15	5	5	3	994.25	58,922	12.6	5	5	3	984.19	58,922	0	12.6

Remarks: #LPSolved: the number of solved linear relaxation instances within the time limit. #Solved: the number of solved instances within the time limit. #SolvedR: the number of instances solved at the root node. T\_CPU Time: the total CPU time to obtain the optimal integer solution on average. #Column: the number of priced out columns. #Node: the number of nodes traversed in the B&B search tree. #Cut: the number of generated cuts.

**Table 3**  
Comparison of the results for instances in transition demand period.

Instance	Without 3-SRIs						With 3-SRIs						
	#LPSolved	#Solved	#SolvedR	T_CPU Tim (s)	#Column	#Node	#LPSolved	#Solved	#SolvedR	T_CPU Time (s)	#Column	#Cut	#Node
r-t-30-5	5	5	5	12.31	2,116	1.0	5	5	5	12.17	2,116	0	1.0
r-t-30-10	5	5	4	13.98	2,636	2.2	5	5	4	12.45	2,636	0	2.2
r-t-30-15	5	5	5	12.46	3,353	1.0	5	5	5	12.37	3,353	0	1.0
r-t-60-5	5	5	5	24.56	18,213	1.0	5	5	5	24.22	18,213	0	1.0
r-t-60-10	5	5	3	844.09	25,766	23.4	5	5	3	886.87	25,766	1.4	23.4
r-t-60-15	5	5	2	634.55	35,484	19.4	5	5	4	307.69	32,942	15.0	5.8
r-t-90-5	5	5	4	1060.07	64,621	5.4	5	5	4	1004.92	64,621	0	5.4
r-t-90-10	5	3	2	2262.19	92,039	17.7	5	4	2	3181.02	95,789	3.0	3.0
r-t-90-15	5	2	2	2853.00	100,496	1.0	5	2	2	2628.99	100,496	0	1.0
r-t-120-5	3	2	1	3218.65	104,249	9.0	3	3	2	3536.62	142,401	1.0	6.3
r-t-120-10	1	0	0	–	–	–	1	0	0	–	–	–	–
r-t-120-15	0	0	0	–	–	–	0	0	0	–	–	–	–
d-t-30-5	5	5	5	19.52	2,114	1.0	5	5	5	18.46	2,114	0	1.0
d-t-30-10	5	5	5	19.06	2,810	1.0	5	5	5	18.53	2,810	0	1.0
d-t-30-15	5	5	4	168.96	3,597	9.4	5	5	4	92.97	3,014	7.2	2.6
d-t-60-5	5	5	5	27.60	19,915	1.0	5	5	5	26.91	19,915	0	1.0
d-t-60-10	5	5	3	148.44	27,538	13.8	5	5	3	130.01	27,538	0	13.8
d-t-60-15	5	4	3	161.84	37,756	2.5	5	4	4	150.01	37,087	0.5	1.0
d-t-90-5	5	5	4	384.39	70,407	1.4	5	5	4	370.99	70,407	0	1.4
d-t-90-10	5	4	4	1059.51	99,200	1.0	5	4	4	1080.86	99,200	0	1.0
d-t-90-15	5	1	1	2624.39	140,259	1.0	5	1	1	2522.69	140,259	0	1.0
d-t-120-5	2	1	1	4126.54	176,773	1.0	2	1	1	4183.35	176,773	0	1.0
d-t-120-10	1	0	0	–	–	–	1	0	0	–	–	–	–
d-t-120-15	0	0	0	–	–	–	0	0	0	–	–	–	–

Remarks: #LPSolved: the number of solved linear relaxation instances within the time limit. #Solved: the number of solved instances within the time limit. #SolvedR: the number of instances solved at the root node. T\_CPU Time: the total CPU time to obtain the optimal integer solution on average. #Column: the number of priced out columns. #Node: the number of nodes traversed in the B&B search tree. #Cut: the number of generated cuts.

**Table 4**  
Result comparison of SMSw/P and SMSw/oP.

Instance	SMSw/P				SMSw/oP				SMSw/P V.S. SMSw/oP							
	FS	Profit	#SR	#SR/FS	FS	Profit	#SR	#SR/FS	Diff_FS		Diff_Profit		Diff_#SR		Diff_#SR/FS	
d-p-30-5	8.6	-20.90	20.4	2.37	8.6	-21.7	20.4	2.37	0.0	0 %	0.8	4 %	0.0	0 %	0.00	0 %
d-p-30-10	8.8	3.56	22.0	2.50	8.8	2.1	21.6	2.45	0.0	0 %	1.5	72 %	0.4	2 %	0.05	2 %
d-p-30-15	9.2	21.20	25.2	2.74	9.6	14.5	24.2	2.52	-0.4	-4%	6.7	46 %	1.0	4 %	0.22	9 %
d-p-60-5	18.8	8.51	44.4	2.36	18.8	1.7	42.4	2.26	0.0	0 %	6.8	407 %	2.0	5 %	0.11	5 %
d-p-60-10	19.4	61.24	49.0	2.53	19.6	48.1	47.2	2.41	-0.2	-1%	13.2	27 %	1.8	4 %	0.12	5 %
d-p-60-15	18.8	108.26	52.6	2.80	19.6	91.8	50.2	2.56	-0.8	-4%	16.4	18 %	2.4	5 %	0.24	9 %
d-p-90-5	27.6	88.57	74.0	2.68	27.4	75.0	71.6	2.61	0.2	1 %	13.5	18 %	2.4	3 %	0.07	3 %
d-p-90-10	26.8	160.28	79.6	2.97	28.0	134.5	76.4	2.73	-1.2	-4%	25.8	19 %	3.2	4 %	0.24	9 %
d-p-90-15	25.0	226.03	81.0	3.24	27.8	196.2	81.2	2.92	-2.8	-10 %	29.9	15 %	-0.2	0 %	0.32	11 %
d-p-120-5	35.4	187.26	96.2	2.72	36.0	169.1	94.2	2.62	-0.6	-2%	18.2	11 %	2.0	2 %	0.10	4 %
d-p-120-10	34.2	289.32	106.0	3.10	35.8	244.0	99.6	2.78	-1.6	-4%	45.3	19 %	6.4	6 %	0.32	11 %
d-p-120-15	32.8	381.52	111.0	3.38	35.8	330.0	105.4	2.94	-3.0	-8%	51.6	16 %	5.6	5 %	0.44	15 %

Remarks: FS: the optimal fleet size. #SR: the number of satisfied rides. #SR/FS: the usage rate of SV calculated by #SR/FS. Diff\_FS: the difference of the optimal fleet size obtained under a nonlinear SQM and linear SQM in absolute value and in percentage. Diff\_Profit: the difference of the profit obtained under a nonlinear SQM and linear SQM in absolute value and in percentage. Diff\_#SR: the difference of the number of satisfied rides obtained under a nonlinear SQM and linear SQM in absolute value and in percentage. Diff\_#SR/FS: the difference of the usage rate of SV obtained under a nonlinear SQM and linear SQM in absolute value and in percentage.

that the incorporation of ride-pooling in SMS helps to reduce the fleet size in most instances. In some instances, e.g., ‘d-p-90-15’, the reduction of fleet size can be up to 10% on average. Fortunately, the cut in fleet size in SMSw/P does not lead to the decrease of profit and satisfied rides. Instead, the ride-pooling option gives a great boost to the profit with an average increment ratio reaching more than 50% of all instances. This can be explained by an increased revenue due to an increased number of satisfied rides and a decreased cost because of the decrease of fleet size, penalty for denying customers, and operating cost for pooled rides in the SMSw/P. Moreover, the usage rate of SV in the SMSw/P is generally higher than that in SMSw/oP.

This study makes a contribution by incorporating the nonlinear customer satisfaction. Hence, it is important to evaluate the value of this nonlinear SQM against the most commonly used linear SQM, i.e., the maximum ride duration constraint, in DARP (Ho et al., 2018). The linear SQM can be mathematically represented by  $et_{ij} + \tau_{sp} \leq rd_{max}$ , where  $rd_{max}$  denotes the maximal acceptable ride duration. For ease of illustration, we compare the optimal fleet size, the profit, the number of satisfied rides, and the number of satisfied pooled rides under the nonlinear customer satisfaction constraint with the threshold being 0 and the results under the maximum ride duration constraint with the threshold  $et_{max}$  being 30 min. Again, we report in Table 5 the differences of these parameters both in absolute value and in percentage. According to the results, if the nonlinear customer satisfaction is considered, more rides will be satisfied by the fleet, resulting in a higher utilization of the

SV fleet. This is achieved by serving more riders in pair as the number of satisfied pooled rides increases significantly in most instances. For example, the increase of satisfied pooled rides can be up to 2 (29%) on average in instances ‘d-p-60-15’. This result suggests that the nonlinear customer satisfaction constraint allows more riders to be pooled together than the maximum ride duration constraint. This is because the proposed nonlinear SQM factors in many more customers’ concerns that may also influence the adoption of shared rides other than the additional ride time, such as the VOT, the WTP, the discounted service charge, etc. Therefore, the adversity of one aspect like the additional ride time can be offset by other factors. Take a ride-matching pattern with  $v = 0.8$ ,  $q_{i1} = 0.8$ ,  $q_{i2} = 1$ ,  $st_{ij} = 30$  min for example. Based on Eq. (41), we can find that the customer satisfaction would always be positive as long as the additional ride time  $et_{ij}$  is smaller than 40 min. Moreover, it is worthwhile to mention that the feasible ride-matching patterns identified by the linear SQM may not be satisfactory to the pooled riders, since enforcing  $et_{ij} \leq 30$  min may not guarantee a positive customer satisfaction. Last, as expected, the increased number of satisfied pooled rides contribute to an obvious rise in profit, with an average increment ratio reaching more than 8% of all instances.

The above results have shown the overwhelming advantages of SMSw/P over the services without ride-pooling and the potential merits of a nonlinear SQM as well as the significance of this study in considering the ride-pooling and customer satisfaction in the decision-making problems of SMS. Although the results of the impact analysis will depend on the parameter values and the specific forms of SQM, the nonlinear

**Table 5**  
Result comparison of nonlinear and linear SQMs.

Instance	Nonlinear SQM				Linear SQM				Nonlinear SQM V.S. Linear SQM							
	FS	Profit	#SR	#SP	FS	Profit	#SR	#SP	Diff_FS		Diff_Profit		Diff_#SR		Diff_#SP	
d-p-30-5	10.0	-7.25	19.4	0.6	10.2	-9.13	19.4	0.2	-0.2	-2%	1.9	26 %	0.0	0 %	0.4	67 %
d-p-30-10	9.6	6.70	19.6	1.2	9.6	3.19	19.2	0.8	0.0	0 %	1.5	22 %	0.4	2 %	0.4	33 %
d-p-30-15	11.0	33.87	24.0	1.8	11.0	32.45	23.6	1.4	0.0	0 %	6.7	20 %	0.4	2 %	0.4	22 %
d-p-60-5	23.2	52.18	45.8	2.0	23.6	43.79	45.0	0.4	-0.4	-2%	6.8	13 %	0.8	2 %	1.6	80 %
d-p-60-10	23.4	95.94	49.0	3.0	23.2	94.05	48.0	2.0	0.2	1 %	13.2	14 %	1.0	2 %	1.0	33 %
d-p-60-15	21.6	153.93	51.8	6.8	21.6	153.45	50.8	4.8	0.0	0 %	16.4	11 %	1.0	2 %	2.0	29 %
d-p-90-5	34.4	155.86	72.4	3.0	34.4	149.88	71.2	2.0	0.0	0 %	13.5	9 %	1.2	2 %	1.0	33 %
d-p-90-10	33.2	232.46	77.4	6.4	32.8	226.69	75.4	4.6	0.4	1 %	25.8	11 %	2.0	3 %	1.8	28 %
d-p-90-15	31.8	302.05	81.2	8.4	31.6	301.71	80.2	6.8	0.2	1 %	29.9	10 %	1.0	1 %	1.6	19 %
d-p-120-5	43.8	227.61	94.2	3.8	43.8	222.95	93.4	3.0	0.0	0 %	18.2	8 %	0.8	1 %	0.8	21 %
d-p-120-10	42.8	351.25	103.0	8.6	43.4	333.74	101.6	6.6	-0.6	-1%	45.3	13 %	1.4	1 %	2.0	23 %
d-p-120-15	41.8	455.55	107.6	11.2	41.2	451.81	107.2	12.0	0.6	1 %	51.6	11 %	0.4	0 %	-0.8	-7%

Remarks: FS: the optimal fleet size. #SR: the number of satisfied rides. #SP: the number of satisfied pooled rides. Diff\_FS: the difference of the optimal fleet size obtained under a nonlinear SQM and linear SQM in absolute value and in percentage. Diff\_Profit: the difference of the profit obtained under a nonlinear SQM and linear SQM in absolute value and in percentage. Diff\_#SR: the difference of the number of satisfied rides obtained under a nonlinear SQM and linear SQM in absolute value and in percentage. Diff\_#SP: the difference of the number of satisfied pooled rides obtained under a nonlinear SQM and linear SQM in absolute value and in percentage.

SQM is likely to be a more realistic characterization of customer satisfaction than the linear SQM.

### 6. Conclusions

This study optimized the fleet size, ride-matching patterns, and vehicle routes for SMS considering ride-pooling and customer satisfaction. A set packing model and customized BCP approach were developed to obtain an exact optimal solution to the problem. The pricing problem within the BCP approach is NP-hard in the strong sense and we proposed a customized two-phase method to effectively address it. In the first phase, we identified all the feasible matching patterns between any two rides. A labeling method was iteratively employed in the second phase to solve ESPPTW in a network constructed upon these feasible ride matching patterns. Both heuristic and exact version of the labeling algorithm were implemented and three speedup techniques were used to accelerate the algorithm. We further strengthened the model by adding the valid inequalities. A primal–dual stabilization strategy was adopted to mitigate the tailing-off effect in column generation. A compatible hybrid branching scheme was devised to guarantee the integrality of the optimal solution. We evaluated the solution method in numerical experiments. The impacts of ride-pooling option and nonlinear customer satisfaction constraint on the performance of SMS were also examined.

This study considered a special case of DARP that allows at most two requests sharing their rides in static and deterministic setting in which the information of rides is known a priori. Further research work can be undertaken in several aspects, among which the first and most important

future work is to develop efficient heuristic methods for solving large-scale and dynamic problems. In addition, motivated by the growing trend of vehicle electrification, it would be interesting to extend the proposed model and methods to SMS with EVs, by incorporating the limited driving ranges and charging requirements of these vehicles. Moreover, the future SMS may be subject to significant stochasticity and uncertainties in both the demand and the operating parameters (e.g., ride duration, electricity consumption, driving range of EVs, etc.). How to consider the uncertainties of those factors in the context of SMS is also an important avenue for future research.

### CRedit authorship contribution statement

**Min Xu:** Writing – original draft, Software, Methodology, Conceptualization.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgement

This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15210620).

### Appendix: Notations

$i$	Index for ride
$I$	Set of rides
$w$	Index for depot
$W$	Set of depots
$r$	Index for vehicle route
$R$	Set of feasible routes
$S$	Set of pick-up and drop-off locations
$N_w$	Capacity of depot $w \in W$
$s_w$	Location of depot $w \in W$
$s_i^o$	Pick-up location of ride $i \in I$
$s_i^d$	Drop-off location of ride $i \in I$
$t_i^o$	Earliest departure time of ride $i \in I$
$t_i^d$	Latest arrival time of ride $i \in I$
$AC$	Fixed cost of SV per vehicle-day
$UC$	Operating cost per unit driving distance
$P_i$	Penalty incurred by unserved rider $i \in I$
$G_i$	A service charge of ride $i \in I$
$v$	A discount rate for ride-pooling service
$\widehat{G}_i$	A discounted service charge of ride $i \in I$ for ride-pooling service
$F_{ij}(\bullet)$	Satisfaction function of rider $i \in I$ when sharing a ride with rider $j \in I \setminus \{i\}$
$F_{i^*}(\bullet)$	Satisfaction function of rider $i \in I$ when three or more riders are pooled with him/her sequentially
$q_i$	Vector of attributes of concerned rider $i$
$q_{i1}$	Value of time of concerned rider $i$
$q_{i2}$	willingness-to-pool of concerned rider $i$
$st_{ij}$	Duration of ride-pooling of rider $i$ and rider $j$
$et_{ij}$	Additional ride time due to ride-pooling of rider $i$ and rider $j$
$l_{s_i^o, s_i^d}$	Distance between two locations, e.g., from $s_i^o$ to $s_i^d$
$\tau_{s_i^o, s_i^d}$	Travel time between two locations, e.g., from $s_i^o$ to $s_i^d$
$E_i$	Minimal customer satisfaction of rider $i$ that is assured by the service provider
$R_r$	Amortized net profit of vehicle route $r$
$G_r$	Total service charge of all covered rides served by route $r$
$L_r$	Total traveling distance of route $r$
$x_r$	Binary decision variable that equals 1 if the optimal vehicle route of a SV in the fleet is $r$ and 0 otherwise
$\delta_i^r$	Coefficient that equals 1 if ride $i$ is served by a SV through route $r$ , and 0 otherwise
$\theta_r^w$	Coefficient that equals 1 if the route $r$ starts and ends at depot $w$ , and 0 otherwise
$\bar{R}$	A subset of routes
$\pi_i$	Dual variables corresponding to constraint
$\rho_w$	Dual variables corresponding to constraint
$\beta$	Index for ride-matching pattern

(continued on next page)



(continued)

$I_\beta$	Set of covered rides of ride-matching pattern $\beta$
$\bar{i}_\beta$	The first boarding ride of ride-matching pattern $\beta$
$\underline{i}_\beta$	The last alighting ride of ride-matching pattern $\beta$
$\tau_\beta$	Travel duration of ride-matching pattern $\beta$
$G_\beta$	Net profit of ride-matching pattern $\beta$
$l_\beta$	Travel distance of ride-matching pattern $\beta$
$\bar{t}_\beta$	Earliest departure time of ride-matching pattern $\beta$
$\bar{\tau}_i$	Travel duration of the ride-matching pattern till the pick-up of ride $i$
$\bar{t}_\beta$	Latest arrival time of ride-matching pattern $\beta$
$\bar{t}_i$	Travel duration of the ride-matching pattern till the drop-off of ride $i$
$G = (N, A)$	Constructed pseudo-network for pricing problem
$\Theta$	Set of ride-matching patterns
$c_n$	Cost of node $n \in N$ in the pseudo-network $G$
$\tau_n$	Service duration of node $n \in N$ in the pseudo-network $G$
$\bar{i}_n$	Index of start-ride, of which the service at node $n$ starts at the pick-up location
$\underline{i}_n$	Index of end-ride, of which the service at node $n$ ends at the drop-off location
$\Delta_n$	Set of rides included in the node $n \in N$ in the pseudo-network $G$
$[\bar{t}_n, \underline{t}_n]$	A time window within which the service of node $n \in N$ in the pseudo-network $G$ must start
$c_{nm}$	Cost of link $(n, m) \in A$ in the pseudo-network $G$
$\tau_{nm}$	Travel time of link $(n, m) \in A$ in the pseudo-network $G$
$L_n^{forward}$	Set of forward labels at node $n \in N$
$L_n^{backward}$	Set of backward labels at node $n \in N$
$l_k^{forward}$	A forward label $k$ at the node $n \in N$
$l_g^{backward}$	A backward label $g$ at the node $n$
$S_k$	Set of served rides of label $k$
$\hat{C}_k$	Cost of label $k$
$\hat{\tau}_k$	Earliest service ending time of label $k$
$\gamma_k$	Node index of label $k$
$\kappa_k$	Index of the end-ride of label $k$
$T$	Maximal time resource
$\bar{L}_n^{forward}$	Set of unextended forward labels of node $n \in N$
$\bar{L}_n^{backward}$	Set of unextended backward labels of node $n \in N$
$E$	Set of rides to be explored in the labeling algorithm
$\bar{P}^f$	End-rides of nodes with newly generated forward labels after an iteration in the labeling algorithm
$\bar{P}^b$	Start-rides of nodes with newly generated backward labels after an iteration in the labeling algorithm
$C_w$	Cost of a feasible complete path from depot $w$ to its counterpart depot $w'$
$\hat{S}_k$	A subset of set $S_k$ used in decremental search space method
$\epsilon_1$	Pre-specified tolerance for column generation
$lpObj$	Optimal objective value of the RMP
$M$	Pre-specified parameter to help fix the long tail effect of column generation
$p^*$	The largest reduced cost, i.e., the optimal objective value of the pricing problem
$\Omega_C$	Set of all identified violated inequalities
$C_b$	Any one identified violated inequality in set $\Omega_C$
$\eta_{C_b}$	Dual variable corresponding to the violated inequality $C_b$
$D_k := \{D_k^b\}_{C_b \in \Omega_C}$	Vector representing the number of riders (mod 2) in all violated inequities that have been served by label $k$
$Q_{threshold}$	Threshold for the number of total labels that triggers the cut generation
$\bar{V}_{min}$	Requested minimal violated value for most violated cut to trigger the cut generation
$C_{max}$	Maximal number of most violated 3-SRIs that can be added in the linear relaxation
$V_{min}$	Requested minimal violated value for a cut to be added in the linear relaxation
$\bar{N}_w$	Upper bound of parking capacity of depot $w \in W$
$\underline{N}_w$	Lower bound of parking capacity of depot $w \in W$
$SI$	Set of satisfied rentals
$RI$	Set of rejected rentals
$Q(i, j)$	Set of routes covering the ride $i$ and ride $j$ successively or serving ride $i$ and ride $j$ jointly in a ride-matching pattern
$IP$	Set of included pairs of rides
$EP$	Set of excluded pairs of rides
$\epsilon_2$	Pre-specified tolerance for branch and bound

Data availability

Data will be made available on request.

References

Bargetto, R., Garaix, T., Xie, X., 2023. A branch-and-price-and-cut algorithm for operating room scheduling under human resource constraints. *Computers & Operations Research* 152, 106136.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Research* 46 (3), 316–329.

Ben Amor, H., Desrosiers, J., Valério de Carvalho, J.M., 2006. Dual-optimal inequalities for stabilized column generation. *Operations Research* 54 (3), 454–463.

Boland, N., Dethridge, J., Dumitrescu, I., 2006. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters* 34 (1), 58–68.

Braekers, K., Kovacs, A.A., 2016. A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part b: Methodological* 94, 355–377.

Campbell, A.M., Savelsbergh, M., 2004. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* 38 (3), 369–378.

Cordeau, J.F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54 (3), 573–586.

Cordeau, J.F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part b: Methodological* 37 (6), 579–594.

Cordeau, J.F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. *Annals of Operations Research* 153 (1), 29–46.

- Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science* 53 (4), 946–985.
- Desaulniers, G., Lessard, F., Hadjar, A., 2008. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science* 42 (3), 387–404.
- Diao, X., Qiu, M., Xu, G., 2024. Electric vehicle-based express service network design with recharging management: A branch-and-price approach. *Computers & Operations Research* 162, 106469.
- Dror, M., 1994. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research* 42 (5), 977–978.
- Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal* 44 (3), 216–229.
- Gendreau, M., Hertz, A., Laporte, G., Stan, M., 1998. A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research* 46 (3), 330–335.
- Ho, S.C., Szeto, W.Y., Kuo, Y.H., Leung, J.M., Petering, M., Tou, T.W., 2018. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part b: Methodological* 111, 395–421.
- Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints. In: *Column Generation*. Springer, US, Boston, MA, pp. 33–65.
- Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D., 2008. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* 56 (2), 497–511.
- Lam, E., Desaulniers, G., Stuckey, P.J., 2022. Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. *Computers & Operations Research* 145, 105870.
- Lavieri, P.S., Bhat, C.R., 2019. Modeling individuals' willingness to share trips with strangers in an autonomous vehicle future. *Transportation Research Part a: Policy and Practice* 124, 242–261.
- Luo, Z., Liu, M., Lim, A., 2019. A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. *Transportation Science* 53 (1), 113–130.
- Masmoudi, M.A., Hosny, M., Demir, E., Genikomsakis, K.N., Cheikhrouhou, N., 2018. The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research Part e: Logistics and Transportation Review* 118, 392–420.
- Masson, R., Lehuédé, F., Péton, O., 2014. The dial-a-ride problem with transfers. *Computers & Operations Research* 41, 12–23.
- Parragh, S.N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part c: Emerging Technologies* 19 (5), 912–930.
- Parragh, S.N., Pinho de Sousa, J., Almada-Lobo, B., 2015. The dial-a-ride problem with split requests and profits. *Transportation Science* 49 (2), 311–334.
- Righini, G., Salani, M., 2006. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization* 3 (3), 255–273.
- Ropke, S., Cordeau, J.F., Laporte, G., 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks: An International Journal* 49 (4), 258–272.
- Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40 (4), 455–472.
- Ryan, D.M., Foster, B.A., 1981. An integer programming approach to scheduling. *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling* 269–280.
- Shaheen, S., Cohen, A., 2019. Shared ride services in North America: definitions, impacts, and the future of pooling. *Transport Reviews* 39 (4), 427–442.
- Stocker, A., Shaheen, S., 2017. Shared automated vehicles: Review of business models. accessed 3 May 2024 International Transport Forum Discussion Paper. <https://www.itf-oecd.org/sites/default/files/docs/shared-automated-vehicles-business-models.pdf>.
- Zhao, J., Poon, M., Zhang, Z., Gu, R., 2022. Adaptive large neighborhood search for the time-dependent profitable dial-a-ride problem. *Computers & Operations Research* 147, 105938.