Vehicle routing for customized on-demand bus services

Lu Zhen [1], Xueting He [1], Shuaian Wang [2*], Jingwen Wu [1], Kai Liu [3]

[1] *School of Management, Shanghai University, Shanghai, China*

[2] *Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Hong Kong*

[3] *School of Transportation and Logistics, Dalian University of Technology, Dalian, China*

* Corresponding author: hans.wang@polyu.edu.hk

Abstract: This study investigates a variant of the vehicle routing problem (VRP) for customized on-demand bus service platforms. In this problem, the platform plans customized bus routes upon receiving a batch of orders released by passengers and informs the passengers of the planned pick-up and drop-off locations. The related decision process takes into account some passenger-side time window-related requirements, walking limits, the availability and capacities of various types of buses. A mixed-integer linear programming model of this new VRP variant with floating targets (passengers) is formulated. To solve the model efficiently, a solution method is developed that combines the branch-and-bound and column generation algorithms and includes embedded acceleration techniques such as the multi-labeling algorithm. Experiments based on real data from Dalian, China are conducted to validate the effectiveness of the proposed model and efficiency of the algorithm; the small-scale experimental results demonstrate our algorithm can obtain optimal results in majority of instances. Additionally, sensitivity analysis is conducted and model extensions are investigated to provide customized bus service platform operators with potentially useful managerial insights; for example, a platform need not establish as many candidate stops as possible, a wide range of walking distance may not bring early arrival at destinations for customers, more mini-buses should be deployed than large buses in our real-world case. Moreover, the rolling horizon based context and zoning strategies are also investigated by extending our proposed methodology.

Keywords: Bus transportation; vehicle routing; customized bus service; pick-up and drop-off locations; column generation; multi-labeling algorithm.

1. Introduction

According to the Ministry of Transport of the People's Republic of China, the car ownership was up to 297 million in 2021 (Guan et al. 2022). In addition, the rapid urbanization has led to an increase in the average number of daily passenger trips; the number of daily trips per capita in private cars in Beijing reached 3.33 on weekdays (BTDRC 2020). The large number of cars and daily trips have caused serious traffic congestion and accelerated energy consumption (Xia et al. 2022). In recent years, a new demand-driven user-oriented customized bus service has emerged as a novel component of municipal public transportation services in China since its launch in Qingdao in August 2013 (Wang et al. 2019). Such services provide customized and efficient transit services to groups of commuters with similar travel demands with respect to their origins or destinations. In addition to user convenience, these services also reduce the use of private cars, which provides benefits such as reducing carbon emissions and alleviating traffic congestion in metropolitan areas. Different from the well-known ride sharing mode that improves the urban transportation efficiency by increasing the private cars' occupancy rates, the mode of customized bus service reduces the usage of the private cars, and is benefit for reducing the transportation's carbon emissions (Agatz et al. 2012).

As the name suggests, a customized bus service system designs the routes of buses to serve a batch of passenger groups according to the information submitted about those passengers' origins and destinations. Each scheduled bus route is customized for a batch of passengers and executed only once. The schedules are designed automatically by an intelligent module as part of a customized bus service platform, rather than manually. Specifically, the platform should contain a decision model to make timely decisions about routes according to the input information provided by passengers in real time, and then deliver the output information to passengers to enable them to board the scheduled buses on time. The design of such a decision model is an important issue in the operation of a flexible, timely, and demand-responsive customized bus service.

This study aims to develop a mathematical model and an efficient algorithm for designing customized bus routes according to information submitted by passengers on the locations of their origins and destinations, as well as time-related information such as the earliest times passengers can depart from their origins and the latest times they can arrive at their destinations. Unlike dial-a-ride services, a customized bus service does not provide door-to-door service; instead, passengers must walk from their origins to pick-up points and to their destinations from drop-off points. In reality, these pick-up and drop-off points are usually routine bus stops, as these are convenient for the implementation of a customized bus service in a realistic transportation network. The mathematical model also decides the bus routes and locations of the pick-up and drop-off points. For each pick-up point, the bus's arrival time should be carefully decided so that the passengers assigned to the point can walk from their origins and arrive at the point before the bus's arrival time. In the model, this decision is related to the passengers' submitted times of departure from their origins. The decisions behind this problem are complex and intertwining; in addition, the input data for the model are more comprehensive than those used in related models in the literature. The model introduced in this paper is embedded in a customized bus service platform and is solved for each batch of passengers in an iterative manner. Specifically, the platform collects demand information from a batch of passengers and delivers this to the model as input data. Subsequently, the model produces the following outputs: the bus route, the pick-up and drop-off points for groups of passengers with similar origins or destinations, and the bus's arrival times at these points, which should accommodate not only the passengers' ability to walk to the points from their origins but also their arrival at the destinations no later than the required time.

Using mathematical programming methodology, a mixed integer linear programming (MILP) model is designed to solve in a timely manner the above-described complex decision problem in the setting of a demand-responsive customized bus service platform. As mentioned at the beginning of this paper, the model must be solved separately for each batch of passengers and the solution algorithm must be computationally efficient to ensure its applicability in real-world settings. Therefore, this study proposes a novel solution method in which the branch-and-bound (BB) and column generation (CG) algorithms are combined in a nested manner. In addition, a multi-label setting algorithm is designed and embedded in the CG algorithm to further accelerate the solution. The results of numerical experiments show that the BB-CG nested solution method can solve the model much faster than commercial solver (CPLEX), while producing a solution with a tiny optimality gap. Through a series of sensitivity analyses, this study also yields managerial insights. Finally, we demonstrate our proposed methodology in a case study based on real data from Dalian, China.

The remainder of this paper is organized as follows. Relevant studies on customized bus services are reviewed in Section 2. The background of the problem is described in Section 3. An MILP model to address the problem is formulated in Section 4. The proposed BB-CG nested algorithm is elaborated in Section 5. The results of the numerical experiments are reported in Section 6. A real case study is presented in Section 7. The conclusions are outlined in Section 8.

2. Related works

Customized bus services, an emerging mode of public transportation, have attracted attention from both academics and practitioners, leading to the recent publication of many papers on this topic. The literature in this field largely describes the use of mathematical programming methodology, with the exception of some game-theory-based studies (Li et al. 2021, Liu et al. 2021). In this section, representative studies are reviewed in terms of the following three aspects: analysis of the routing of customized bus services, special features of proposed methods for solving the problem, and special features considered in related models.

Routing optimization, customer assignment and time scheduling are the common themes among operations research-related studies on customized service systems including the bus systems, and are also the focus of the current study (Matin-Moghaddam and Sefair, 2021). To solve a customized bus service network design problem from a spatial–temporal network perspective, Tong et al. (2017) use a multi-commodity network flow model as the basis to establish an integrated bus routing and timetable planning optimization model. Tong and colleagues implement a novel hybrid algorithm based on Lagrangian decomposition and a space–time prism and perform experiments based on real-world large-scale cases to validate their methodology. Gambella et al. (2018) propose a mixed-integer second order cone programing model as well as an exact algorithm based on the branch-and-price for a vehicle routing problem (VRP) with nonstationary (floating) pick-up locations of the targets. Fielbaum et al. (2021) investigate an on-demand ridesharing optimization problem by considering the pick-up and drop-off walking locations; some heuristics are proposed to solve the problem. The above two problems are similar with this study to some extent; however, more constraints and more realistic factors are considered in this study than the above two studies. Chen et al. (2021) also use a spatial–temporal network as the basis for an integrated optimization method that can make decisions about bus stop deployment, route design, and timetable design optimization. They further propose the passenger inconvenience index to measure service quality, which they optimize in the model accompanied with another objective on the used bus stops. Lyu et al. (2019) formulate mathematical models that optimize bus stops, routes, timetables, and the probability that passengers will choose customized bus services; specifically, they design a heuristic framework containing a grid-density-based clustering method, a bus stop deployment algorithm, and a routing and timetabling algorithm based on dynamic programming. Guo et al. (2021) design a tabu search- and variable neighborhood search-based algorithm to solve an integrated model involving bus routing, path choice, and passenger assignment, and conduct experiments to investigate the effects of the time window and traffic congestion and the benefits of path flexibility. Guan et al. (2022) implement a genetic algorithm for solving a customized bus routing problem considering passengers with multi-trip requests and pickup/delivery-time-window constraints, which are verified in the application of the 2022 Beijing Winter Olympic Games. Xue et al. (2022) design an improved adaptive large neighborhood search algorithm for solving the heterogeneous customized bus service planning problem containing multiple pickup and delivery candidate locations. Shu and Li (2022) design an ant colony optimization (ACO)-based algorithm for the joint planning of bus lines and stations in a demand-responsive customized bus system and conduct simulations to validate its effectiveness. Although our study also focuses on bus routing, we concentrate on a different problem than other studies in the literature: we consider passengers' detailed travel process (travel duration, start/end time) before boarding or disembarking. In addition, all bus routing decisions in our problem are in response to customers' demands in real-time rather than prediction information. To suit this new context, we explicitly formulate and solve an MILP model.

Customized bus systems face some very complex decision problems, and accordingly solution methods are often designed with multiple objectives, stages, phases (Pasha et al. 2022). Asghari et al. (2022a) design an ACO-based algorithm for solving a the multi-objective customized bus routing problem so as to investigate the social and environmental implications of incorporating carpooling in a customized bus service platform. Chen et al. (2021) investigate a customized bus route design problem and build a bi-objective (operating cost and passenger profit), multi-trip, multi-pickup and delivery optimization model with time windows; they further propose a two-stage meta-heuristic method to derive a pareto solution to the problem. Ma et al. (2021) propose a three-stage, hybrid non-dominated sorting genetic algorithm-II (NSGA-II) that both uses bi-objective programming and considers uncertainties to address customized bus route optimization under uncertain conditions. They also build a model using a robust optimization methodology, with the objectives of minimizing passengers' travel times and carbon emissions from buses. From a hierarchical decision-making perspective, Huang et al. (2020) propose a two-phase model for optimizing a demand-responsive customized bus network in an interactive manner between a customized bus operator and passengers. In this model, insertion and BB-based algorithms are designed to address the dynamic phase and static phase, respectively, and passengers' travel behavior is measured using a discrete choice model. Huang et al. (2020) also adopt a two-phase framework to realize real-time route decisions on a customized bus service platform. Using data from the Sioux Falls network, a computational study is performed to validate that real-time decisions on route optimization can be made in less than 0.4 seconds. Yu and Shen (2020) also propose a two-phase approach for solving a scheduling problem in an integrated car-and-ride sharing system. The BB-CG solution method proposed in our study also adopts two methodologies, namely, BB and CG; however, these are set in a nested rather than a two-phase framework. Thus, our BB-CG nested algorithm may also contribute to the research on algorithmic approaches to customized bus systems.

The relevant literature also considers special features such as uncertainty and zoning strategies (Kim et al. 2019; Wang and Wu 2021). Dou et al. (2021) considering uncertain factors and propose an MILP model for an integrated design problem under the condition of uncertain travel demand; their proposed methodology can help service operators to make decisions about customized bus routes, timetabling, and bus deployment. In that study, a branch-and-price based algorithm and a CG-based heuristic are implemented to solve small-scale and large-scale instances, respectively. To handle uncertain demand environments, Lee et al. (2021a) propose and develop a zone-based flexible bus service that considers the uncertainties of passenger volume and origin–destination requirements; they further propose a two-stage stochastic programing model and gradient-based solution method to minimize the costs of regular service operations and expected ad hoc services. The same research team further considers the elastic feature of demands in this flexible service (Lee et al. 2021b). Lee et al. (2021b) also consider in detail the different types of stochasticity with respect to demand, detour time, and demand elasticity with respect to flexible bus service price and quality, and design a gradient-based greedy search algorithm to design flexible bus routing plans while optimizing the associated reliability measures. Gong et al. (2021) consider passenger transfer in a customized bus system to further reduce the operational cost of the system; they design a particle swarm optimization algorithm to solve the problem with large-scale instances. The methodologies proposed in these three studies by Lee et al. and Gong et al. are validated by experiments using real data from Chengdu, China. As mentioned earlier, in this work we use real data from Dalian, China to perform our case study and validate our model and algorithm.

Our study is notably different from the published work summarized in this section. By comparing with the related literature, the contributions of this study may include the following two aspects. First, this paper investigates a pickup and delivery VRP with floating customers and time-related constraints, and defines this VRP variant in the context of a customized bus service system. Some factors are newly considered in this study; for example, this study considers the selection of the pick-up and drop-off locations for each order, the passengers' walking time from their origins to the pick-up locations (or from the drop-off locations to their destinations), and so on. Second, this paper implements a novel BB-CG nested algorithm with an embedded acceleration technique (i.e., a multi-labelling algorithm) to solve this new VRP variant rapidly. The proposed model and algorithm thus contribute to the literature on customized bus services.

3. Problem Background

Suppose that a platform provides a customized bus service to customers. Customers can submit orders for transportation on the platform. Each order contains the following information: the quantity of passengers who wish to take the ordered journey, the journey's origin and destination, the earliest time when the passengers could depart from the origin, and the latest time when the passengers should arrive at the destination. Using this information, the platform schedules customized bus routes and informs each customer about the locations of their pick-up and drop-off stops and the times when they should arrive at their pick-up stops.

The customized bus service uses routine bus stops as the pick-up and drop-off stops; accordingly, customers may need to walk from their origins to their assigned pick-up stops, and from their assigned drop-off stops to their destinations. For each customer order, a maximum walking distance is specified to improve passenger satisfaction; that is, the pick-up and drop-off stops assigned to the order's passengers should ensure that the total walking distance does not exceed the maximum limit.

Figure 1 shows an example in which two scheduled bus routes serve six orders. Twenty routine bus stops are candidates for selection as the pick-up or drop-off stops for these orders.

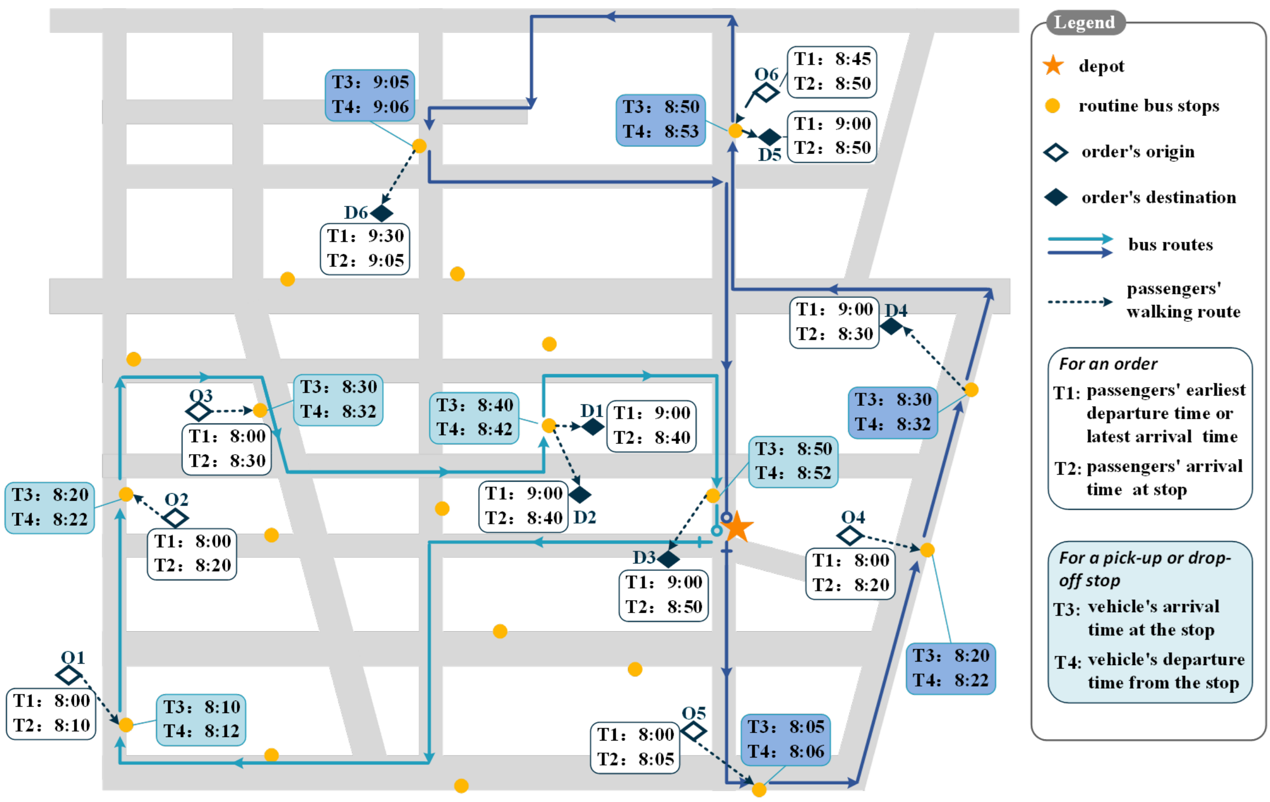


Figure 1: An example of two customized bus routes serving six customer orders

The objective of the platform is to decide the routes of some buses to minimize the number and total travel time (or distance) of the used buses. In this decision-making process, appropriate pick-up and drop-off stops should be assigned to each customer order to ensure that the total walking distance does not exceed the set limits. In addition, the departure time from the origin and arrival time at the destination for each order should be within the corresponding time window. Before describing the development of an MILP model in Section 4, the assumptions are clarified as follows.

(1) The bus travel time between any pair of two locations is known as deterministic parameters. The traffic congestion as well as uncertain travel time is not considered.

(2) Passenger walking time between any pair of two locations is known as deterministic parameters.

(3) The pick-ups and drop-offs of customer orders are among the set of routing bus stops in an area rather than any locations in the area.

4. Model formulation

The known parameters and decision variables are defined formally according to the problem description in the previous section. A mathematical model is then formulated to select and sequence the candidate bus stops as pick-up and drop-off stops so that the customers' required departure and arrival times are satisfied. Some nonlinear parts of the model are also linearized to formulate an MILP model.

4.1 Notations

The parameters and decision variables used in the MILP model are defined in this subsection. For the ease of understanding, we use the Roman letters and the Greek letters to denote the parameters and the variables, respectively.

Indices and sets:

K	set of vehicles, indexed by k.
S	set of vehicle types, indexed by s.
P	set of pick-ups of customer orders, indexed by r.
D	set of drop-offs of customer orders, $D = \{\lvert P\rvert + 1, \lvert P\rvert + 2, \dots, 2\lvert P\rvert\}$, indexed by r.
R	set of pick-ups and drop-offs of customer orders, $R = P \cup D$, indexed by r.
N	set of candidate stops for being a pick-up or a drop-off, indexed by i, j.
$\widetilde{N}$	set of stops and the start depot "0" and the end depot "$n + 1$"; $\widetilde{N} = N \cup \{0\} \cup \{n + 1\}$.

Parameters:

q_r	quantity of passengers in order $r \in P$, and $q_{r+\lvert P\rvert} = -q_r$.
Q_s	capacity of vehicle of type s, $s \in S$.
N_s	number of vehicles with type s, $s \in S$.
s_k	type of vehicle k, $k \in K$.
e_r	earliest time when order r's passengers could depart from their origins, $r \in P$.
l_r	latest time when order r's passengers should arrive at their destinations, $r \in D$.
$t_{i,j}^{V}$	vehicle's travel time between node i and node j.
$t_{i,r}^{WO}$	walking time for passengers in order r from their origin to stop i, $r \in P$.
$t_{i,r}^{WD}$	walking time for passengers in order r from stop i to their destination, $r \in P$.
T_r^{W}	maximum walking time for passengers in customer order r, $r \in P$.
c_s^{F}	fixed cost for vehicle of type s.
c_s^{V}	unit variable cost of vehicle of type s with respect to its traveling time.

Decision variables:

α_k	binary, equals one if vehicle k accepts at least one order, otherwise zero.
$\beta_{i,j,k}$	binary, equals one if vehicle k visits stop j immediately after stop i, otherwise zero.
$\zeta_{i,r,k}$	binary, equals one if vehicle k uses stop i as a pick-up or drop-off for order r, and zero otherwise.
$\lambda_{i,k}$	vehicle k's arrival time at stop i.
$\mu_{i,k}$	vehicle k's departure time at stop i.

ξ_r^o walking time from order r's origin location to pick-up stop, $r \in P$.

ξ_r^d walking time from drop-off stop to order r's destination location, $r \in P$.

$Q_{k,i}$ integer, number of passengers on vehicle k after visiting stop i.

4.2 Mathematical model

Based on the above defined parameters, sets and decision variables, a mathematical programming model is formulated to minimize the total cost of operating the customized bus routes for serving a set of customers.

Minimize $\sum_{k\in K} c_{s_k}^F \alpha_k + \sum_{k\in K} c_{s_k}^V \sum_{i\in \widetilde{N}} \sum_{j\in \widetilde{N}} t_{i,j}^V \beta_{i,j,k}$ (1)

s.t.

$\sum_{k\in K} \sum_{i\in N} \zeta_{i,r,k} = 1$ $\quad \forall r \in R$ (2)

$\sum_{i\in N} \zeta_{i,r,k} = \sum_{i\in N} \zeta_{i,r+|P|,k}$ $\quad \forall k \in K, r \in P$ (3)

$\sum_{j\in N} \beta_{0,j,k} = \alpha_k$ $\quad \forall k \in K$ (4)

$\sum_{i\in N} \beta_{i,n+1,k} = \alpha_k$ $\quad \forall k \in K$ (5)

$\sum_{j\in N\cup\{0\}} \beta_{j,i,k} = \sum_{j\in N\cup\{n+1\}} \beta_{i,j,k}$ $\quad \forall i \in N, k \in K$ (6)

$\sum_{r\in R} \zeta_{i,r,k} \geq \sum_{j\in N\cup\{n+1\}} \beta_{i,j,k}$ $\quad \forall i \in N, k \in K$ (7)

$\sum_{r\in R} \zeta_{i,r,k} \leq M \sum_{j\in N\cup\{n+1\}} \beta_{i,j,k}$ $\quad \forall i \in N, k \in K$ (8)

$\beta_{i,j,k} \leq \alpha_k$ $\quad \forall i,j \in \widetilde{N}, k \in K$ (9)

$\beta_{i,j,k} + \beta_{j,i,k} \leq 1$ $\quad \forall i,j \in \widetilde{N}, k \in K$ (10)

$\mu_{i,k} + t_{i,j}^V \leq \lambda_{j,k} + M(1 - \beta_{i,j,k})$ $\quad \forall i,j \in \widetilde{N}, k \in K$ (11)

$\sum_{k\in K} \sum_{i\in N} \zeta_{i,r,k}\, t_{i,r}^{WO} = \xi_r^o$ $\quad \forall r \in P$ (12)

$\sum_{k\in K} \sum_{i\in N} \zeta_{i,r+|P|,k}\, t_{i,r}^{WD} = \xi_r^d$ $\quad \forall r \in P$ (13)

$\xi_r^o + \xi_r^d \leq T_r^W$ $\quad \forall r \in P$ (14)

$\sum_{j\in N} \lambda_{j,k} \zeta_{j,r+|P|,k} \geq \sum_{i\in N} \mu_{i,k} \zeta_{i,r,k}$ $\quad \forall r \in P, k \in K$ (15)

$\sum_{i\in N} \mu_{i,k} \zeta_{i,r,k} \geq e_r + \xi_r^o$ $\quad \forall r \in P, k \in K$ (16)

$\sum_{i\in N} \lambda_{i,k} \zeta_{i,r,k} \leq l_r - \xi_{r-|P|}^d$ $\quad \forall r \in D, k \in K$ (17)

$Q_{k,j} \geq Q_{k,i} + \sum_{r\in P\cup D} q_r \zeta_{j,r,k} - M(1 - \beta_{i,j,k})$ $\quad \forall i \in N \cup \{0\}, j \in N \cup \{n+1\}, k \in K$ (18)

$Q_{k,i} \leq Q_{s_k}$ $\quad \forall i \in \widetilde{N}, k \in K$ (19)

$\alpha_k \in \{0,1\}$ $\quad \forall k \in K$ (20)

$\beta_{i,j,k} \in \{0,1\}$ $\quad \forall i,j \in \widetilde{N}, k \in K$ (21)

$\zeta_{i,r,k} \in \{0,1\}$ $\quad \forall i \in \widetilde{N},\ r \in R, k \in K$ (22)

$\lambda_{i,k} \geq 0$ $\quad \forall i \in \widetilde{N}, k \in K$ (23)

$\xi_r^o, \xi_r^d \geq 0 \qquad \forall r \in P \qquad (24)$

$Q_{k,i} \geq 0 \qquad \forall i \in \widetilde{N}, k \in K. \qquad (25)$

Objective (1) minimizes the total cost of using vehicles and their travel costs. Constraints (2) guarantee that each pick-up or drop-off is related to one vehicle's visit at a stop. Constraints (3) ensure that the pick-up and the drop-off of a customer should be visited by the same vehicle. Constraints (4) and (5) guarantee that each vehicle must depart from the depot if it is used and return to the depot. Constraints (6) are flow conservation constraints. Constraints (7) and (8) link two variables $\beta_{i,j,k}$ and $\zeta_{i,r,k}$. Constraints (9) and (10) link two variables related to the usage of a vehicle and visiting sequence of the vehicle. Constraints (11) are travel time conservation constraints for vehicles. Constrains (12) and (13) calculate the required walking time from an origin location to a pick-up stop and from a drop-off stop to a destination location, respectively. Constrains (14) guarantee the walking time for the passengers in order r is no greater than their limits T_r^W. Constraints (15) ensure the precedence relationship between a vehicle's arrival time at an order's pick-up stop and the arrival time at the order's drop-off stop. Constrains (16) and (17) ensure that the pick-up time for the passengers in order r is no earlier than their earliest arrival at the pick-up stop, and the drop-off time is no later than their latest departure from the drop-off stop. Constraints (18) and (19) guarantee that the number of passengers on a vehicle between any two consecutive stops should be no greater than the capacity of the vehicle. Constraints (20) and (21) define decision variables.

4.3 Linearization for the model

The above model contains some nonlinear parts, which could be linearized before using some solvers to solve it (Asghari et al. 2022b). More specifically, Constraints (15)–(17) are nonlinear because of containing product of two variables such as $\lambda_{i,k}\zeta_{i,r,k}$. Thus, we define a nonnegative variable $\theta_{r,k}$ to denote the time when the order r's pick-up or drop-off is visited by vehicle k. $\theta_{r,k}$ equals $\lambda_{i,k}$ if $\zeta_{i,r,k}$ equals one. Two more constraints are defined as follows:

$\theta_{r,k} + M(1 - \zeta_{i,r,k}) \geq \lambda_{i,k} \qquad \forall i \in N, r \in R, k \in K \qquad (26)$

$\theta_{r,k} - M(1 - \zeta_{i,r,k}) \leq \mu_{i,k} \qquad \forall i \in N, r \in R, k \in K. \qquad (27)$

Then Constraints (15)–(17) are replaced by the following constraints:

$\theta_{r+|P|,k} \geq \theta_{r,k} \qquad \forall r \in P, k \in K \qquad (28)$

$\theta_{r,k} \geq e_r + \xi_r^o \qquad \forall r \in P, k \in K \qquad (29)$

$\theta_{r,k} \leq l_r - \xi_{r-|P|}^d \qquad \forall r \in D, k \in K \qquad (30)$

$\theta_{r,k} \geq 0 \qquad \forall r \in R, k \in K. \qquad (31)$

Eventually, the linearized model is: (1)–(14); (18)–(31).

5. BB-CG nested algorithm

To solve the proposed MILP model in large-scale instances, we design a novel solution method based on the BB and CG algorithms. These algorithms are nested in the proposed solution method. The framework of the solution method is elaborated in Section 5.1, and details of the embedded BB and CG algorithms are presented in Sections 5.2 and 5.3, respectively.

5.1 Algorithmic framework

This section provides an overview of the proposed BB-CG nested algorithm. A nested framework is adopted in the algorithm design mainly because the problem studied in this paper can be decomposed into two interrelated subproblems: an assignment problem (AP), wherein candidate stops (i.e., routine bus stops) are assigned to customer orders as the pick-up and drop-off stops, and a pick-up and delivery problem with time windows (PDPTW). The AP and PDPTW correspond to the group of Constraints (2) and (3) and the group of Constraints (4)–(19), respectively.

In the algorithmic framework, the decision-making processes of the AP and PDPTW can be regarded as outer and inner loops, respectively, such that the decisions made in the outer loop are the input parameters in the inner loop. Specifically, we define a binary parameter $Z_{i,r}$ as a bridge to connect the AP and the PDPTW; here, $Z_{i,r}$ equals 1 if order r uses stop i as its pick-up or drop-off and 0 otherwise. The AP decides the pick-up and drop-off stops for each order, after which the value of $\sum_{k \in K} \zeta_{i,r,k}$ is determined. The value of $Z_{i,r}$ is set as the value of $\sum_{k \in K} \zeta_{i,r,k}$, and $Z_{i,r}$ acts as a parameter in the PDPTW. Given the value of $Z_{i,r}$, the optimal bus route can be derived by solving the PDPTW.

The aforementioned BB and CG algorithms are used to solve the AP and PDPTW, respectively. When using the BB to solve the AP, we propose two rules to reduce the number of nodes to be explored during branching and thus reduce the overall solution time. When using the CG to solve the PDPTW, we implement a multi-labelling algorithm to solve the CG's pricing problem (PP) and thus accelerate the solving process. In the next two subsections, we elaborate the BB and the CG-based algorithms designed to solve the AP and PDPTW, respectively. Last, it should be noted that the proposed BB-CG nested algorithm is a heuristic and thus differs from the branch-and-price algorithm, which is exact. According to our preliminary tests, The traditional branch-and-price algorithm may not be very efficient in solving our problem, especially in some large-scale instances. Thus we design this BB-CG nested algorithm to solving the problem with large-scale instances during a reasonable time.

5.2 BB algorithm for solving the AP

A pair comprising a pick-up stop and drop-off stop for an order is defined as a PD-pair. Let $\mathcal{A}_r$ be the set of all possible PD-pairs for order r; each element in $\mathcal{A}_r$ is denoted by a_r, which represents one PD-pair for order r; $a_r \in \mathcal{A}_r$. For each order $r \in P$, we choose one a_r from $\mathcal{A}_r$ and construct an assignment plan $(a_1, a_2, \ldots, a_{|P|})$, which is a solution to the AP. This plan represents an assignment of PD-pairs for all of the orders. Given an assignment plan $(a_1, a_2, \ldots, a_{|P|})$, we solve the PDPTW using the CG algorithm. The number of all possible assignment plans is $\prod_{i=1}^{|P|} |\mathcal{A}_i|$, which implies that we need to solve the PDPTW for $\prod_{i=1}^{|P|} |\mathcal{A}_i|$ times using the CG algorithm.

Before introducing the algorithm of solving the AP, a formal definition of the AP is given as follows for the purpose of expressing the relationship between the nested two subproblems (i.e., the AP and the PDPTW). Suppose ϑ_{a_r} is the binary decision variable in the AP; it equals one if PD-pair $a_r \in \mathcal{A}_r$ for order r is selected, and zero otherwise. The AP is: Minimize $f\left(\left\{\vartheta_{a_r}\right\}_{r \in R, a_r \in \mathcal{A}_r}\right)$, subject to $\sum_{a_r \in \mathcal{A}_r} \vartheta_{a_r} = 1$ for $r \in R$. Here $f\left(\left\{\vartheta_{a_r}\right\}_{r \in R, a_r \in \mathcal{A}_r}\right)$ is the objective value for the assignment plan $(a_1, a_2, \ldots, a_{|P|})$, i.e., $\left\{\vartheta_{a_r}\right\}_{r \in R, a_r \in \mathcal{A}_r}$; the objective value is actually the PDPTW's objective. The constraints "$\sum_{a_r \in \mathcal{A}_r} \vartheta_{a_r} = 1$, for $r \in R$" ensure each order is assigned with one plan.

To improve the efficiency of solving the AP, we design a tailored BB algorithm, which is elaborated as follows.

In the BB process, the elements in $\bigcup_{r \in P} \mathcal{A}_r$ (i.e., PD-pairs) are regarded as nodes. First, two rules are proposed to reduce the number of nodes by removing infeasible PD-pairs.

Rule 1: Eliminate PD-pairs that violate Constraint (14) to ensure that the walking time of passengers in order r does not exceed the maximum walking time T_r^W.

Rule 2: Eliminate PD-pairs that violate the case wherein the sum of the earliest time to reach the pick-up stop i plus the time needed to travel to the drop-off stop j should be no later than the latest departure time from drop-off stop j (i.e., $e_r + t_{i,r}^{WO} + t_{i,j}^{V} \leq l_r - t_{j,r}^{WD}$).

After removing nodes according to the above rules, the set $\mathcal{A}_r$ is updated for all orders and the branching procedure is executed. For each node, we define an order waiting list Λ, which includes all of the orders that are not served when the node is explored. The branching rules and node selection strategy of the BB algorithm are introduced as follows.

Branching rule: We select the order r with the fewest PD-pairs from the current order waiting list Λ of the parent node. We branch the parent node into $|\mathcal{A}_r|$ child nodes, each of which is a PD-pair a_r in the updated set $\mathcal{A}_r$. We then assign the order waiting list Λ of the parent node individually to each child node, and remove the order r from the list Λ of each child node (i.e., $a_r \in \mathcal{A}_r$).

Node selection strategy: If the current BB node is not pruned, the depth-first-search rule is used. The order to which the next exploring node is related has the fewest PD pairs in the current order waiting list Λ of the parent node.

The procedure of the BB for solving the AP is summarized as follows.

Step 1: Initialize an assignment plan $(a_1, a_2, \dots, a_{|P|})$ and use it to obtain the values of $Z_{i,r}, \forall i \in N, r \in P$.Invoke the CG algorithm to solve the PDPTW with all $|P|$ orders; in the PDPTW model, $\{Z_{i,r}\}_{\forall i \in N, r \in P}$ are the parameters. The solved objective value is set as the upper bound (UB), which represents the incumbent best objective value. Define $Z_{i,r}^*$ as the incumbent best feasible solution for the AP, $Z_{i,r}^* \leftarrow Z_{i,r}$. Initialize the root node (the initial parent node) and the node list as null.

Step 2: Execute branching from the current parent node. Let the obtained child nodes join the node list, then select and remove the last node from this list as the next parent node. For the next parent node, execute the following procedures. (i) If $|\Lambda| \geq |P|/2$, do not invoke the CG algorithm to solve the PDPTW for the parent node, and set the node's objective values as zero directly, $Obj \leftarrow 0$. (ii) Otherwise, determine the assignment plan $(a_1, a_2, \dots)$ and the values of $Z_{i,r}$. Invoke the CG algorithm to solve the PDPTW with the orders contained in the set $P \backslash \Lambda$. Obj equals the solved objective value.

Step 3: (i) If $Obj < UB$ and $\Lambda \neq \emptyset$, go to **Step 2**. (ii) If $Obj < UB$ and $\Lambda = \emptyset$, update $UB \leftarrow Obj(Z_{i,r})$ and $Z_{i,r}^* \leftarrow Z_{i,r}$, and go to **Step 4**. (iii) If $Obj \geq UB$, prune this node and go to **Step 4**.

Step 4: If the node list is empty, terminate the whole algorithm; otherwise, select and remove the last node from the node list as the parent node. Invoke the CG algorithm to solve the PDPTW with the orders in the set $P \backslash \Lambda$. Obj equals the solved objective value. Go to **Step 3**.

A judgement condition "$|\Lambda| \geq |P|/2$" is included in Step 2 because the Obj of a node is likely to be less than UB when the number of orders is small. We invoke the CG algorithm to solve the PDPTW model only when the level of the node being explored is larger than $|P|/2$, i.e., half of the total orders, at which point pruning can be executed. The above judgement condition can accelerate the solution of the AP.

5.3 CG algorithm for solving the PDPTW

For each node during the branching process of solving the AP, the CG algorithm may be invoked to solve a PDPTW, which is constructed on the basis of $Z_{i,r}$. As aforementioned, the values of $Z_{i,r}$ are obtained according to one solution of the AP. The PDPTW model is difficult to solve, especially in the context of some large-scale instances. As the CG algorithm is widely used to solve large-scale MILP models, this study adopts this approach to solve the PDPTW.

The usual practices used to implement the CG algorithm, the set partitioning model, the PP, the tailored algorithm used to solve the PP, and the strategies for obtaining integer solutions, are elaborated in subsections 5.3.1–5.3.4, respectively. The whole CG procedure is summarized in subsection 5.3.5.

5.3.1 A set-partitioning model

In this section, we reformulate the problem as a master problem (MP) model using Dantzig–Wolfe decomposition (Dantzig and Wolfe, 1960). We define $\mathcal{P}_s$ as the set of all feasible routes for a vehicle of type s, $s \in S$. A binary variable x_{p_s} is defined for each route $p_s \in \mathcal{P}_s$, $k \in K$. When route p_s is chosen for vehicle type s, x_{p_s} equals one, and zero otherwise. A binary parameter y_{r,p_s} is used to denote whether customer order r is served by route p_s. The cost of a vehicle associated with route p_s is denoted by C_{p_s}. Using these definitions, the set-partitioning model is formulated as follows.

[**MP**] *Minimize* $\sum_{k\in K}\sum_{p_k\in\mathcal{P}_k} C_{p_k}x_{p_k}$ (32)

s.t.

$\sum_{k\in K}\sum_{p_s\in\mathcal{P}_s} y_{r,p_s}x_{p_s} = 1$ $\quad \forall r \in P\backslash\Lambda$ (33)

$\sum_{p_s\in\mathcal{P}_s} x_{p_s} \le N_s$ $\quad \forall s \in S$ (34)

$x_{p_s} \in \{0,1\}$ $\quad \forall s \in S, p_s \in \mathcal{P}_s.$ (35)

Objective (32) minimizes the total costs of the routes selected in the solution. Constraints (33) guarantee that each customer order is served once by a vehicle route. Constraints (34) state that vehicles of type s are assigned with at most N_s route. Constraints (35) define the domain of the decision variables.

The above formulation cannot be solved directly due to the huge number of variables corresponding to the set of all the feasible routes. Hence, in practice a CG procedure solves the linear programming (LP) relaxation of a restricted master problem (LR-RMP), which is based on subsets $\mathcal{P}'_s \subseteq \mathcal{P}_s$ for all vehicle types $s \in S$. The LR-RMP model is defined as follows.

[**LR-RMP**] *Minimize* $\sum_{s\in S}\sum_{p_s\in\mathcal{P}'_s} C_{p_s}x_{p_s}$ (36)

s.t.

$\sum_{s\in S}\sum_{p_s\in\mathcal{P}'_s} y_{r,p_s}x_{p_s} = 1$ $\quad \forall r \in P\backslash\Lambda$ (37)

$\sum_{p_s\in\mathcal{P}'_s} x_{p_s} \le N_s$ $\quad \forall s \in S$ (38)

$x_{p_s} \ge 0$ $\quad \forall s \in S, p_s \in \mathcal{P}'_s.$ (39)

Let π_r and φ_s be the dual variables associated with constraints (37) and (38), respectively. Following the scheme of a CG procedure, the optimal dual variables π_r and φ_s are used to construct the PP for generating new routes at each iteration of the algorithm. The CG procedure terminates whenever no negative reduced cost columns or routes are generated, and the resulting LR-RMP solution corresponds to the optimal solution of the LP-relaxation of the set-partitioning based formulation. Otherwise, the columns identified by solving problem PP are added to problem LR-RMP, and a new iteration is executed. To ensure the existence of an initial feasible LR-RMP solution, an initial solution generation method is shown in the Appendix A.

5.3.2 Pricing problem (PP)

The PP aims to generate columns (i.e., routes) with negative reduced costs. These columns are then added to the LR-MRP. Due to its special feature, the PP can be decomposed into $|S|$ independent pricing subproblems, each of which corresponds to one vehicle of type $s \in S$. The pricing subproblem for a vehicle of type s is denoted by PP_s and is defined as follows.

Decision variables in PP_s:

α	binary, equals one if vehicle of type s accepts at least one customer order, zero otherwise.
$\beta_{i,j}$	binary, equals one if vehicle of type s visits stop j immediately after stop i, zero otherwise.
λ_i	arrival time at stop i for a vehicle of type s.
μ_i	departure time at stop i for a vehicle of type s.
ξ_r^o	walking time from order r's origin location to a pick-up stop.
ξ_r^d	walking time from a drop-off stop to the order r's destination location.
Q_i	integer, number of passengers on vehicle of type s after visiting stop i.

Mathematic model:

[$\mathbf{PP}_k$] *Minimize* $\sigma_k = C_{p_k} - \sum_{r \in P \backslash \Lambda} \pi_r \sum_{i \in N} Z_{i,r} - \varphi_s$ (40)

s.t.

$\sum_{j \in N} \beta_{0,j} = \alpha$ (41)

$\sum_{i \in N} \beta_{i,n+1} = \alpha$ (42)

$\sum_{j \in N \cup \{0\}} \beta_{j,i} = \sum_{j \in N \cup \{n+1\}} \beta_{i,j}$ $\quad \forall i \in N$ (43)

$\sum_{r \in R \backslash (\Lambda \cup \{r+|P|\}_{\forall r \in \Lambda})} Z_{i,r} \geq \sum_{j \in N \cup \{n+1\}} \beta_{i,j}$ $\quad \forall i \in N$ (44)

$\sum_{r \in R \backslash (\Lambda \cup \{r+|P|\}_{\forall r \in \Lambda})} Z_{i,r} \leq M \sum_{j \in N \cup \{n+1\}} \beta_{i,j}$ $\quad \forall i \in N$ (45)

$\beta_{i,j} \leq \alpha$ $\quad \forall i \in \widetilde{N}, \forall j \in \widetilde{N}$ (46)

$\beta_{i,j} + \beta_{j,i} \leq 1$ $\quad \forall i \in \widetilde{N}, \forall j \in \widetilde{N},$ (47)

$\mu_0 + t_{0,i}^V \geq \lambda_i - M(1 - \beta_{0,i})$ $\quad \forall i \in N$ (48)

$\mu_i + t_{i,j}^V \leq \lambda_j + M(1 - \beta_{i,j})$ $\quad \forall i \in \widetilde{N}, j \in \widetilde{N}$ (49)

$\lambda_{n+1} \leq \mu_i + t_{i,n+1}^V + M(1 - \beta_{i,n+1})$ $\quad \forall i \in N$ (50)

$$\sum_{i\in N} Z_{i,r}\, t_{i,r}^{WO} = \xi_r^o \qquad \forall r \in P\backslash\Lambda \qquad (51)$$

$$\sum_{i\in N} Z_{i,r+|P|}\, t_{i,r}^{WD} = \xi_r^d \qquad \forall r \in P\backslash\Lambda \qquad (52)$$

$$\xi_r^o + \xi_r^d \le T_r^W \qquad \forall r \in P\backslash\Lambda \qquad (53)$$

$$\theta_r + M(1 - Z_{i,r}) \ge \lambda_i \qquad \forall i \in N, r \in R\backslash(\Lambda \cup \{r + |P|\}_{\forall r\in\Lambda}) \qquad (54)$$

$$\theta_r - M(1 - Z_{i,r}) \le \mu_i \qquad \forall i \in N, r \in R\backslash(\Lambda \cup \{r + |P|\}_{\forall r\in\Lambda}) \qquad (55)$$

$$\theta_{r+|P|} \ge \theta_r \qquad \forall r \in P\backslash\Lambda \qquad (56)$$

$$\theta_r \ge e_r + \xi_r^o \qquad \forall r \in P\backslash\Lambda \qquad (57)$$

$$\theta_r \le l_r - \xi_{r-|P|}^d \qquad \forall r \in D\backslash\{r + |P|\}_{\forall r\in\Lambda} \qquad (58)$$

$$Q_j \ge Q_i + \sum_{r\in P\cup D} q_r Z_{j,r} - M(1 - \beta_{i,j}) \qquad \forall i \in N \cup \{0\}, j \in N \cup \{n + 1\} \qquad (59)$$

$$Q_i \le Q_k \qquad \forall i \in \widetilde{N} \qquad (60)$$

$$C_{p_k} = c_k^F \alpha + c_k^V \sum_{i\in\widetilde{N}} \sum_{j\in\widetilde{N}} t_{i,j}^V \beta_{i,j} \qquad (61)$$

$$\alpha \in \{0,1\} \qquad (62)$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall i \in \widetilde{N}, j \in \widetilde{N} \qquad (63)$$

$$\lambda_i \ge 0 \qquad \forall i \in \widetilde{N} \qquad (64)$$

$$\xi_r^o, \xi_r^d \ge 0 \qquad \forall r \in P\backslash\Lambda \qquad (65)$$

$$Q_i \ge 0 \qquad \forall i \in \widetilde{N} \qquad (66)$$

$$\theta_r \ge 0 \qquad \forall r \in R\backslash(\Lambda \cup \{r + |P|\}_{\forall r\in\Lambda}). \qquad (67)$$

Objective (40) minimizes the reduced cost. Constraint groups (41)–(53), (54)–(58), (59)–(60) correspond to Constraint groups (4)–(14), (26)–(30), (18)–(19), respectively. Constraints (61) is the calculation of the cost of a column, which is the sum of vehicle's fixed cost and its travel cost. Constraints (62)–(67) define the decision variables.

5.3.3 Multi-labelling algorithm for solving PP

The PP_s can be defined using a graph $G = (R^+, A)$, with the node set $R^+ = P\backslash\Lambda \cup \{r + |P|\}_{\forall r\in P\backslash\Lambda} \cup \{0, 2|P| + 1\}$ and arc set $A = \{(r, r')|r, r' \in R^+: r \ne r', r \ne 2|P| + 1, r' \ne 0\}$. A multi-labelling algorithm is designed to solve the PP_s. The multi-labelling algorithm first generates partial routes and then extends them to complete routes. In this algorithm, time and bus capacity are the crucial resources, such that the algorithm only extends labels whose used time and capacity resources satisfy the time window and bus capacity limits, respectively.

Specifically, this algorithm uses a label to represent a partial route that departs from the depot and ends at an order's pick-up or drop-off stop, indexed by r. The label is denoted by $(g_r, t_r, \mathbb{S}_r, \mathbb{P}_r, \mathbb{D}_r, \delta_r)$; here, g_r is the number of passengers that the vehicle carries after visiting node r; t_r is the arrival time at node r; $\mathbb{S}_r$

is the set of nodes included in the current partial route; $\mathbb{P}_r$ is the subset of $\mathbb{S}_r$ containing all of the pick-up nodes that have been visited; $\mathbb{D}_r$ is the subset of $\mathbb{P}_r$ containing all of the pick-up nodes that have been visited but whose associated drop-off nodes have not been visited; and δ_r is the reduced cost of the route. The calculation of δ_r is based on the dual variables (π_r and φ_s) obtained from the LR-RMP and is elaborated later. A label $(g_r, t_r, \mathbb{S}_r, \mathbb{P}_r, \mathbb{D}_r, \delta_r)$ with $r \neq 0$ and $\mathbb{S}_r \neq \emptyset$ represents a route in graph G.

The extension starts from the initial label $(g_0, t_0, \mathbb{S}_0, \mathbb{P}_0, \mathbb{D}_0, \delta_0)$, where $g_0 = 0$, $t_0 = 0$, $\mathbb{S}_0 = \emptyset$, $\mathbb{P}_0 = \emptyset$, $\mathbb{D}_0 = \emptyset$, and $\delta_0 = (c_s^F - \varphi_s)/2$. During the extension process, a label that ends at node r becomes a new label that ends at another node $r' \in R^+$, i.e., the arc $(r, r') \in A$. This extension may generate multiple new labels. To reduce the number of newly generated labels (i.e., routes), we eliminate in advance some infeasible arcs in graph $G = (R^+, A)$ that violate Constraint (56) and the time windows. Specifically, the arcs (r, r'), i.e., $r = 0, r' \in D$, $r \in D, r' \in P$, or $r \in P, r' = 2|P| + 1$, violate Constraints (56); the arcs (r, r'), i.e., $r \in P, r' \in D$ and $e_r + \sum_{i \in N} Z_{i,r}\, t_{i,r}^{WO} + \sum_{i,j \in N} Z_{i,r} Z_{j,r'}\, t_{i,j}^{V} > l_{r'} - \sum_{j \in N} Z_{jr'+|P|}\, t_{i,r'}^{WD}$, violate the time windows. After eliminating these infeasible arcs, the arc set A is updated. Then, a partial route can be extended by linking its ending node r to another node r' via an arc $(r, r') \in A$, and a new label $(g_{r'}, t_{r'}, \mathbb{S}_{r'}, \mathbb{P}_{r'}, \mathbb{D}_{r'}, \delta_{r'})$ is generated. The detailed formulas for updating the parameters in the new label are listed as follows.

$$g_{r'} = g_r + q_{r'} \tag{68}$$

$$t_{r'} = \max\{e_{r'} + \textstyle\sum_{i \in N} Z_{j,r'}\, t_{j,r'}^{WO}, t_r + \sum_{i,j \in N} Z_{i,r} Z_{j,r'}\, t_{i,j}^{V}\} \tag{69}$$

$$\mathbb{S}_{r'} = \mathbb{S}_r \cup r', if\ r' \notin \mathbb{S}_r \tag{70}$$

$$\mathbb{P}_{r'} = \begin{cases} \mathbb{P}_r \cup r', & \text{if } r' \in P \backslash \Lambda \text{ and } r' \notin \mathbb{P}_r \\ \mathbb{P}_r & , \text{if } r' \notin P \backslash \Lambda \end{cases} \tag{71}$$

$$\mathbb{D}_{r'} = \begin{cases} \mathbb{D}_r \cup r' & , \text{if } r' \in P \backslash \Lambda \text{ and } r' \notin \mathbb{D}_r \\ \mathbb{D}_r \backslash \{r' - |P|\}, & \text{if } r' - |P| \in P \backslash \Lambda \text{ and } r' - |P| \in \mathbb{D}_r \\ \mathbb{D}_r & , \text{otherwise} \end{cases} \tag{72}$$

$$\delta_{r'} = \begin{cases} \delta_r + \sum_{i,j \in N} Z_{i,r} Z_{j,r'}\, t_{i,j}^{V} - \pi_r/2 & , \text{if } r' \in R^+ \backslash \{0, 2|P| + 1\} \\ \delta_r + \sum_{i,j \in N} Z_{i,r} Z_{j,r'}\, t_{i,j}^{V} + (c_k^F - \varphi_k)/2, & \text{if } r' = 2|P| + 1 \end{cases}. \tag{73}$$

The extension stops when no new labels can be generated. In each iteration of the extension, some inequalities are used to judge the feasibility of a node to which a label's ending node is extended. These inequalities are addressed in the following proposition.

Proposition 1: Given a label $(g_r, t_r, \mathbb{S}_r, \mathbb{P}_r, \mathbb{D}_r, \delta_r)$ and an arc (r, r') with $r' \notin \mathbb{S}_r$, we can derive $t_{r'} = max\{e_{r'} + \sum_{i \in N} Z_{j,r'}\, t_{j,r'}^{WO}, t_r + \sum_{i,j \in N} Z_{i,r} Z_{j,r'}\, t_{i,j}^{V}\}$. Extending the label's ending node r to node r' is feasible if and only if the following constraints are satisfied.

$$g_r + q_{r'} \leq Q_s \tag{74}$$

$$t_{r'} \leq l_{r'} - \sum_{j \in N} Z_{j,r'+|P|}\, t_{i,r'}^{WD} \qquad \forall r' \in \mathbb{D}_r \tag{75}$$

$$t_{r'} + \sum_{j,h \in N} Z_{j,r'} Z_{h,r''}\, t_{j,h}^{V} \leq l_{r''} - \sum_{h \in N} Z_{h,r''+|P|}\, t_{h,r''}^{WD} \qquad \forall r'' \in \mathbb{D}_{r'}. \tag{76}$$

Proof: See Appendix B. ■

The above proposition can help to judge the feasibility of numerous possible extensions and thus accelerate the solution of the PP. The following elimination of dominated routes is another tactic used to accelerate the solution of the PP.

Proposition 2: Given two labels, $L^1 = (g_r^1, t_r^1, \mathbb{S}_r^1, \mathbb{P}_r^1, \mathbb{D}_r^1, \delta_r^1)$ and $L^2 = (g_r^2, t_r^2, \mathbb{S}_r^2, \mathbb{P}_r^2, \mathbb{D}_r^2, \delta_r^2)$, both end at the same node r. If (i) $g_r^1 \leq g_r^2$, (ii) $t_r^1 \leq t_r^2$, (iii) $\mathbb{P}_r^1 = \mathbb{P}_r^2$, (iv) $\mathbb{D}_r^1 \supseteq \mathbb{D}_r^2$, and (v) $\delta_r^1 \leq \delta_r^2$, label L^1 dominates label L^2.

Proof: See Appendix C. ■

Any complete route that is extended from a dominated label cannot exist in an optimal solution. According to Proposition 2, we can identify the dominated labels and eliminate them from the set of labels, which can accelerate the solution of the PP$_s$.

The steps of the multi-labelling algorithm for solving the PP$_s$ are summarized as follows.

Step 0: Let Γ be the set of all labels that have not been extended and Φ_r be the set of all labels that end at node r. Define an initial label $(g_0, t_0, \mathbb{S}_0, \mathbb{P}_0, \mathbb{D}_0, \delta_0)$ and add it to Γ.

Step 1: Select and remove a label L_{min} with the smallest reduced cost from Γ, and set r_{min} as the ending node of label L_{min}.

Step 2: For each extended arc (r_{min}, r_{ext}) in the updated arc set A, if node r_{ext} does not satisfy Constraints (76)–(78) as proposed by Proposition 1, return to the beginning of **Step 2** and find the next arc $(r_{min}, r_{ext}) \in A$; otherwise, extend L_{min} to r_{ext} and generate a new label L_{ext}. Next, execute the following procedures. (i) If L_{ext} is dominated by a label in $\Phi_{r_{ext}}$, return to the beginning of **Step 2** and find the next arc $(r_{min}, r_{ext}) \in A$. (ii) Otherwise, add L_{ext} to Γ and $\Phi_{r_{ext}}$, and remove the labels dominated by label L_{ext} from Γ and $\Phi_{r_{ext}}$.

Step 3: if $\Gamma = \emptyset$, terminate the whole algorithm, and add all labels (i.e., routes) with negative reduced costs in $\Phi_{2|P|+1}$ to RMP. Otherwise, go to **Step 1**.

The above algorithm is used to generate complete routes with negative reduced costs. Note that in the remainder of this paper, *route* refers to the complete route. This algorithm can be solved much faster than CPLEX. In addition, CPLEX often cannot be used solve large-scale instances, whereas the above-described algorithm can solve them quickly.

5.3.4 Strategies for obtaining a feasible integer solution

The CG procedure described above only solves the linear relaxation of the set-partitioning model and does not guarantee that feasible integer solutions are obtained. Therefore, we propose some strategies for route selection so as to obtain near-optimal integer solutions. The routes are chosen from the subset of feasible routes maintained in the RMP. This subsection proposes three strategies to select a route from the column pool.

Strategy 1: Select the route corresponding to the largest fractional value of the decision variables. If there are two routes with the same fractional value, select the one with a lower route cost.

Strategy 2: Select the route corresponding to the lowest route cost. If there are two routes with the same route cost, select the one with the larger fractional value of the decision variables.

Strategy 3: Select the route corresponding to the lowest reduced cost. If there are two routes with the same reduced cost, select the one with a lower route cost.

5.3.5 Framework of CG-based heuristic

This subsection describes the proposed CG-based heuristic used to obtain near-optimal integer solutions. The heuristic framework contains an outer procedure and an inner procedure. The outer procedure is the selection heuristic used to obtain an integer solution. The inner procedure is the CG procedure proposed in Section 5.3.1. Before elaborating the heuristic framework, we define two resources limited in the RMP and initially set their values: $Vehicle_s = N_s, \forall s \in S$ (vehicle resource), and $Order_r = 1, \forall r \in R$ (order resources). These two resources correspond to the right-hand sides of Constraints (37) and (38), respectively, and are set as the input parameters for the right-hand sides of the constraints in the algorithm. The detailed framework of our algorithm is as follows:

Step 0: Initialize the set Ω as empty for the final solution routes. Pass the initial resources (i.e., $Vehicle_s = N_s, Order_r = 1$) to the right-hand sides of the constraints in the RMP.

Step 1: Invoke the CG procedure (**Steps 1.1–1.4**). An LP solution is obtained when the procedure ends. If the LP solution is an integer solution, the algorithm is terminated.

Step 1.1: Generate the initial set of feasible routes with the algorithm described in **Appendix A**. Input the initial routes to the RMP formulated in Section 5.3.1.

Step 1.2: Solve the RMP using an LP solver (e.g., CPLEX) and obtain the dual vectors (π_r,φ_s).

Step 1.3: Pass the dual vectors to the PP defined in Section 5.3.2. Delete the node for which the order resource (i.e.,$Order_r$) equals 0 from the node set R^+ in graph G. Use the multi-labelling algorithm described in Section 5.3.3 to find routes with negative reduced costs for each vehicle of type s with $Vehicle_s > 0$.

Step 1.4: Add the routes with negative reduced costs to the RMP. If all routes have a nonnegative reduced cost, stop the CG procedure; otherwise, go to **Step 1.2**.

Step 2: Select one route $\mathcal{P}_s$ from the column pool based on the strategies proposed in Section 5.3.4, and pass it to the set Ω. Update the two resources based on the selected route. For example, if the selected route $\mathcal{P}_s$ occupies order r, then set $Vehicle_s = Vehicle_s - 1, Order_r = 0$. After the update, pass the current two resources to the right-hand sides of the constraints in the RMP.

Step 3: Repeat **Step 1** and **Step 2** until $Order_r = 0, \forall r \in R$. At the end of the algorithm, an integer solution for the problem can be derived from the set Ω.

6 Computational experiments

We conduct computational experiments to validate the efficiency of the proposed BB-CG nested algorithm. The experiments mainly contain three parts. In the first part, we tune the configuration of the BB-CG nested algorithm; i.e., we seek the best combination of components embedded in the algorithm. The components are related to two types of options used in the algorithm: the option of strategies to solve the CG PP (the CPLEX or multi-labeling algorithm), and the option of three strategies for obtaining feasible solutions to the CG algorithm. In the second part, we validate the quality of the solution yielded by our BB-CG nested algorithm under the best algorithmic configuration (i.e., the best combination of embedded elements). In the third part, we use the validated algorithm to solve several series of problem instances under different parameter settings to derive managerial implications. In this study, we use four instance groups (ISGs) to conduct the three-part computational experiments. The scales of the ISGs are shown in Table 1. The walking speeds of customers range from 4 to 5 km/h (Montufar et al. 2007), and the number of customers in each order ranges between 5 and 10. The experiments consider three types of bus: mini-bus, medium bus, and large bus. The vehicle capacity, fixed cost, and unit variable cost of each type of bus are tabulated in Table 2 (Dou et al. 2021). The BB-CG nested algorithm is coded in C# and run on a PC equipped with two Xeon E5-2643 V4 CPUs (12 cores) with a 3.4 GHz and 256 GB of RAM.

Table 1: Scales of instance groups in experiments

Group ID	Bus quantity ($\|K\|$)	Order quantity ($\|P\|$)	Stop quantity ($\|N\|$)	Num. of var.	Num. of cons.
ISG 1	3	6	20	1×10^3	5×10^3
ISG 2	4	8	25	3×10^3	1×10^4
ISG 3	6	15	40	1×10^4	3×10^4
ISG 4	8	20	50	2×10^4	8×10^4

Notes: Num. of var. and Num. of cons. denote approximate number of variables and constraints contained in the model.

Table 2: Parameters on different types of buses

Vehicle type	Capacity	Fixed cost	Unit variable cost
Mini-bus	15	10	1.5
Medium bus	40	20	2.0
Large bus	60	30	2.5

6.1. Investigation of embedded components in the algorithm

In the proposed method, the components needed to obtain a feasible solution and solve the PP are essential. As we propose three strategies for obtaining a feasible solution, this subsection first investigates which of these strategies is best. In addition, we propose a multi-labeling algorithm for solving the PP, which can also be solved directly by the CPLEX and thus compare the ability of both algorithms to solve the pricing problem.

Table 3. Comparison of three strategies for obtaining feasible integer solution

Instances	Strategy 1			Strategy 2			Strategy 3		
	Obj.	Gap	Time(s)	Obj.	Gap	Time(s)	Obj.	Gap	Time(s)
ISG1	427.80	0.00%	2.01	427.80	0.00%	1.68	427.80	0.00%	2.31
	457.48	1.20%	35.19	504.55	11.61%	36.94	452.06	0.00%	36.83
	449.26	0.00%	9.36	481.41	7.16%	10.12	449.28	0.00%	10.23
	591.00	4.61%	27.64	591.00	4.61%	28.58	564.94	0.00%	23.55
	591.64	5.53%	1.93	591.64	5.53%	2.02	560.62	0.00%	1.91
ISG2	676.70	2.23%	8.34	669.56	1.15%	8.48	661.96	0.00%	8.07
	672.45	0.00%	2.40	692.59	2.99%	2.35	672.45	0.00%	2.42
	663.32	0.14%	3.35	665.09	0.41%	3.52	662.36	0.00%	3.41
	539.40	1.32%	28.14	633.60	19.02%	24.55	532.35	0.00%	25.31
	1027.47	10.52%	2.78	929.66	0.00%	2.49	935.73	0.65%	2.70
Average		2.56%			5.25%			0.07%	

Notes: "Obj." is the objective value of the solution obtained by using one of the three strategies in the BB-CG nested algorithm. "Gap" is the gap between the minimum of the solutions obtained by using three different strategies and the solution obtained by using the corresponding strategy. "Time(s)" is the solution time (in seconds).

We also conduct extensive numerical experiments to compare the performance of the proposed algorithm when using three strategies to obtain feasible solutions. The small-scale instance groups ISG1 and ISG2 are used in this experiment. The results in Table 3 show that the choice of strategy has almost no influence on the algorithm solution time but has a considerable influence on the solution quality, which is reflected by the relative gap values (i.e., three columns of Gap values). The results in Table 3 also show that Strategy 3 outperforms Strategies 1 and 2 with respect to solution quality, as Strategy 3 achieves the lowest relative gap value (0.07% on average). Therefore, Strategy 3 is chosen to obtain feasible integer solutions in subsequent experiments. The three strategies in the comparison adopt different criteria in column selection for in the CG's last stage of constructing feasible solution. Strategy 3 adopts the column's reduced cost as the primal criterion; while Strategy 1 and Strategy 2 adopt the solved value of variable x_{p_s} in LR-RMP and the column's cost as the primal criteria in the CG's last stage, respectively. The best performance achieved by Strategy 3 may imply that the reduced cost of columns is the most suitable criterion for constructing feasible solution in the problem studied in this paper.

Table 4. Comparison of multi-labeling algorithm and the CPLEX in solving the pricing problem

Instances	Solving PP by CPLEX		Solving PP by multi-labeling Algorithm		Time ratio
	Obj.	Time(s)	Obj.	Time(s)	
ISG1	530.79	945.61	530.79	0.36	0.04%

	541.62	80.11	541.62	0.77	0.97%
	758.94	1355.35	758.94	0.26	0.02%
	755.04	755.53	755.04	0.26	0.03%
	413.34	1395.62	413.33	3.08	0.22%
ISG2	412.46	1295.32	412.46	2.06	0.16%
	507.29	138.69	507.29	0.22	0.16%
	723.46	30.91	723.46	0.27	0.87%
	741.37	252.03	741.37	0.50	0.20%
	645.88	50.00	645.88	0.37	0.75%
Average					0.34%

Notes: "Time ratio" is the computational time of solving the pricing problem (PP) by multi-labeling algorithm divided by the computational time by CPLEX.

Next, we can use the CPLEX to solve the PP directly or use the multi-labeling algorithm, which is proposed in Section 5.3.3. We conduct experiments to compare the above two ways with respect to the objective value and the solution time. The results are listed in Table 4. From the columns labeled Obj., we can see that both approaches yield the same objective values for each instance. However, the multi-labeling algorithm yields a much shorter computation time than the CPLEX. The average value in the column labeled Time ratio is 0.34%, which validates the proposal and use of the multi-labeling algorithm to solve the PP embedded in our BB-CG nested algorithm.

6.2. Investigating the solution quality of the BB-CG nested algorithm

After determining the configuration of the BB-CG nested algorithm, we investigate the quality of the solutions provided by the algorithm. The quality of these solutions is investigated through comparison with the optimal results obtained by the CPLEX and by a rival heuristic, i.e., the large neighborhood search algorithm (LNS). Details of the LNS algorithm are elaborated in Appendix D.

Table 5: Performance evaluation in small-scale problem instances

Instances	**CPLEX**		**BB-CG**		**LNS**		**Gap**	
	Obj_C	t_C	Obj_B	t_B	Obj_L	t_L	gap_1	gap_2
	537.60	446.55	537.60	0.85	597.46	0.64	0.00%	11.13%
	807.95	522.29	820.85	0.86	898.83	0.48	1.60%	11.25%
ISG1	408.20	116.29	408.21	1.50	458.84	0.57	0.00%	12.40%
	749.47	477.09	749.47	0.77	876.93	0.38	0.00%	17.01%
	537.60	889.34	537.60	1.20	598.18	0.93	0.00%	11.27%
Average							0.32%	12.61%
	662.00	3600.00	662.00	54.96	749.18	1.95	0.00%	13.17%
	662.72	3600.00	662.72	10.73	759.54	0.69	0.00%	14.61%
ISG2	523.90	3600.00	523.90	14.69	676.21	0.58	0.00%	29.07%
	1123.29	3600.00	970.73	1.99	1108.57	1.31	-13.58%	-1.52%
	674.52	3600.00	618.03	5.37	734.03	1.11	-8.37%	9.63%
Average							-4.39%	12.99%

Notes:(1) obj_C, obj_B and obj_L denote the objective value obtained by CPLEX, the proposed BB-CG nested algorithm and LNS algorithm (elaborated in Appendix D), respectively. (2) t_C, t_B, and t_L denote the solution time of the CPLEX, the BB-CG nested algorithm and LNS algorithm, respectively. (3) $gap_1 = (obj_B - obj_C)/obj_C$, $gap_2 =$

$(obj_L - obj_C)/obj_C$. (4) In ISG 2, the CPLEX cannot solve the model within one hours of computing time, and we report the upper bound values computed by CPLEX within the imposed time limit (i.e., 3600 seconds).

Table 5 illustrates the comparisons between the CPLEX, the proposed BB-CG nested algorithm, and the LNS algorithm. Although the CPLEX can find the optimal solution to the ISG1 within a reasonable time, the BB-CG nested algorithm and LNS algorithm can solve the problem within 2 seconds. The average optimality gaps of the solutions yielded by the BB-CG nested algorithm and the LNS algorithm are about 0.32% and 12.61%, respectively. For ISG2, the CPLEX cannot solve the model within 1 hour of computing time. Therefore, we record the upper bound values returned by the CPLEX when the computation time reaches 1 hour, and all of these values are greater than or equal to the objective values obtained by our algorithm within 1 minute. However, most of the solutions yielded by the CPLEX are smaller than the objective values yielded by the LNS algorithm within a very short duration (i.e., 10 seconds). The above results validate the advantages of our proposed BB-CG nested algorithm with respect to the optimality gap and solution time. In addition, our BB-CG nested algorithm outperforms the well-known LNS algorithm, as demonstrated by a significant gap between the objective values obtained by the two algorithms.

To further evaluate the effectiveness of the proposed BB-CG nested algorithm, we conduct large-scale experiments on the basis of ISG3 and ISG4. As the CPLEX cannot solve problems involving large-scale instances, these experiments mainly compare the BB-CG nested algorithm and the LNS algorithm, and the results are listed in Table 6.

Table 6: Performance evaluation in large-scale problem instances

Instances	**BB-CG**		**LNS**		**Gap**
	Obj_B	t_B	Obj_L	t_L	gap
ISG3	1258.61	52.23	1408.86	3.97	11.94%
	978.60	155.78	1111.92	13.16	13.62%
	930.30	116.98	1024.65	16.10	10.14%
	1058.11	27.40	1137.47	8.21	7.50%
	1076.87	9.91	1275.19	0.25	18.42%
ISG4	1327.64	130.45	1766.36	26.71	33.05%
	1379.66	249.05	1516.93	26.90	9.95%
	1447.08	181.44	1710.09	7.11	18.18%
	1593.61	149.62	1781.72	4.63	11.80%
	1241.37	297.40	1326.77	24.79	6.88%
Average					14.15%

Notes: $gap = (Obj_L - Obj_B)/Obj_B$.

The comparison reveals that the proposed algorithm achieves significant cost savings relative to the LNS algorithm. When using the large-scale instances, as illustrated in Table 6, the average gap between the two algorithms is about 14.15%. Although the solution time of the LNS algorithm is shorter than that of the BB-

CG nested algorithm, these large-scale instances could be solved by the proposed BB-CG nested algorithm within 5 minutes, which is acceptable for a real-world application.

6.3 Deriving managerial insights from sensitivity analyses

We next conduct four series of sensitivity analyses to investigate the influence of important parameters on the final performance of the proposed algorithm and thus derive potentially useful managerial insights.

The first series of experiments involves sensitivity analysis of the ratio of the number of candidate stops (i.e., $|N|$) to the number of orders (i.e., $|R|$). The experiments are based on ISG2, ISG3, and ISG4. The results in Figure 2 show that as the ratio of $|N|$ to $|R|$ increases, the objective value decreases gradually to a certain value. This may occur because as the ratio of $|N|$ to $|R|$ increases, the flexibility of bus scheduling also increases; thus, the total cost decreases. However, this decreasing trend is not infinite. There exists a certain value of the ratio of $|N|$ to $|R|$ (e.g., 4 in Figure 2); when the ratio is larger than this value, the reduction in the total cost becomes insignificant. This result demonstrates the importance of determining a reasonable ratio of $|N|$ to $|R|$. It is not always a good choice to involve as many candidate stops in the system as possible.

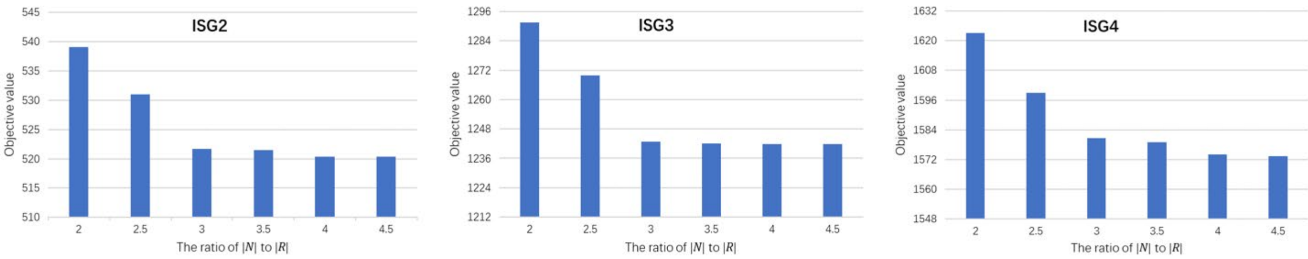


Figure 2: Sensitivity analysis of the influence of the ratio of number of candidate stops to number of orders

The second series of sensitivity analyses addresses the passengers' maximum walking distance in customer orders (i.e., T_r^W), which ranges between 1000 and 2250 meters in the experiments. The results are shown in Figure 3. In Figure 3, the y-axis represents the total arrival time of all orders, rather than the objective value of the original model; because the influence of the maximum walking distance on the objective value is intuitive, the latter analysis may be meaningless. Therefore, we analyze the influence of the maximum walking distance on the total arrival time of all orders, which actually reflects the utility for customers (passengers). The results in Figure 3 suggest a significant relationship between these two passenger-related factors. For example, a long walking distance may not reduce a passenger's time to arrival at a destination because the bus stops selected by the customized bus service platform may require passengers to walk for relatively long distances, thus delaying arrival at their destinations.

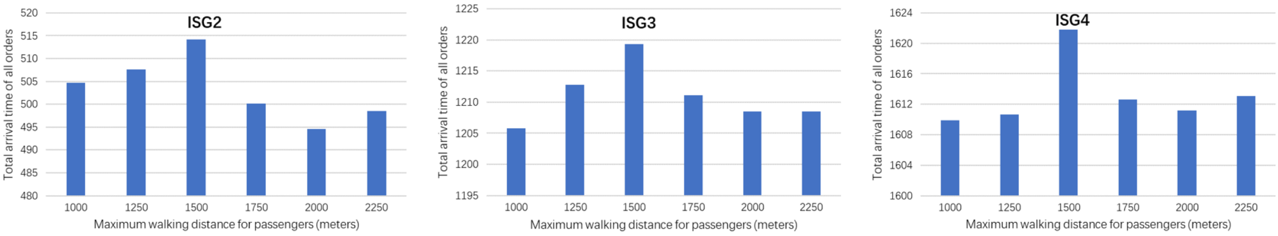


Figure 3: Sensitivity analysis of the influence of passengers' maximum walking distance

The third series of sensitivity analyses addresses the time window length in an order. Because the time window length (i.e., $l_r - e_r$) varies among customers, we use the unified parameter h to reflect the degree of the time window length, i.e., $h = (l_r - e_r)/T_r^{OD}$; here, T_r^{OD} is the normal door-to-door travel time for an order r. In Figure 4, the x-axis of each graph represents the coefficient h of the time window. Figure 4 shows that as the coefficient of the time window increases, the objective value decreases, and the magnitude of this decreasing trend also decreases. The results indicate the importance of setting an appropriate coefficient of the time window. The larger the coefficient of the time window, the lower the total cost and the lower the level of passenger satisfaction. The platform should choose an appropriate coefficient to achieve a balance between passenger satisfaction and travel cost.

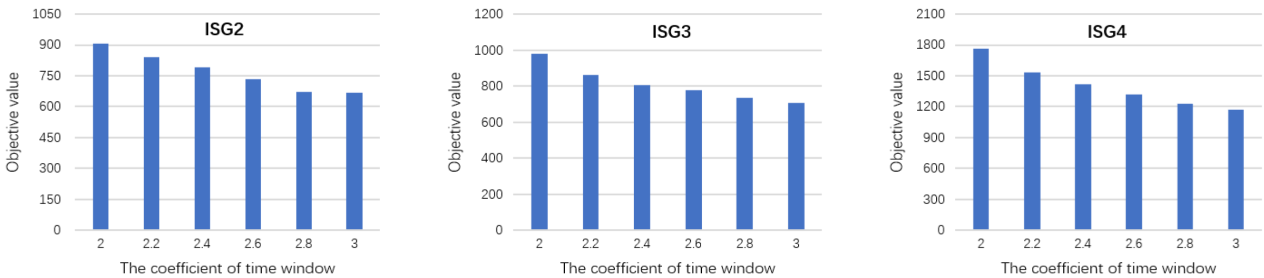


Figure 4: Sensitivity analysis of the influence of the time window length in an order

The fourth series of sensitivity analyses addresses the ratio of the three types of vehicles, namely, the mini-bus, medium bus, and large bus, which have respective capacities of 15, 40, and 60 passengers. The three types of vehicles also have different fixed and variable costs. The results are shown in Figure 5, in which the x-axis on each graph denotes the ratios of the three types of vehicles. Figure 5 demonstrates that the objective value generally decreases as the proportion of large buses decreases or that of mini-buses increases. Although the above trend does not strictly exit, the results may be instructive. For example, decision-makers for customized bus services may wish to invest more on mini-buses than on large buses in real-world settings.

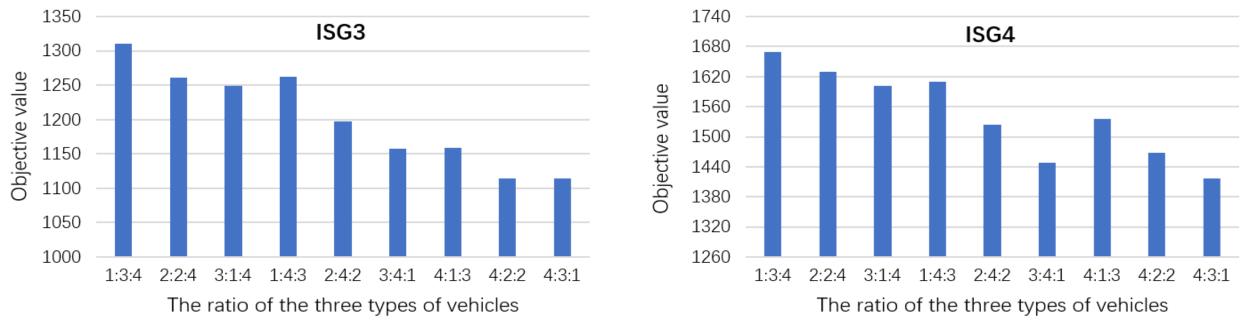
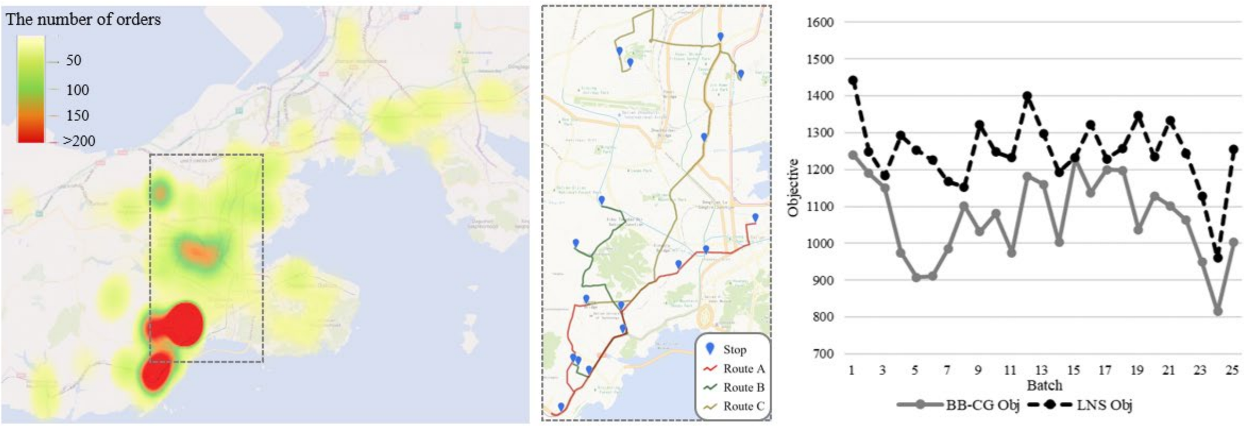


Figure 5: Sensitivity analysis of the influence of the ratio of three types of vehicles

7. Application to a real case

In this section, we validate the effectiveness of the BB-CG nested algorithm in solving large-scale real-world instances using real data from Dalian, a major city in northeast China with an area of about 12,574 km^2 and a population exceeding 6.69 million. We use a dataset containing real-world orders in days, with an average of about 2,200 orders per day. Figure 6 shows a representative instance of the orders in a single day, together with a heat map of the historical demand density. These orders mainly span a planning horizon from 6:00 to 19:00 (i.e., 13 hours). Our BB-CG nested algorithm cannot solve an instance with more than 2,000 orders, and the orders in this dataset do not arrive at the customized bus service platform at one time but rather arrive randomly along the planning horizon. Therefore, we need to handle the arriving orders by batch. In other words, we solve the model for each batch of orders in a rolling horizon manner. In this real case study, the batch size is 20 orders, which accumulate over an average period of about 7 minutes [i.e., $7 \approx (13 \times 60) \div (2200 \div 20)$]. Therefore, we need to solve the model for a batch of 20 orders and then execute the plan for the batch; after about 7 minutes, we must solve the model again for another batch of 20 orders and execute the plan and then repeat this process of solution and execution plan. This approach suggests that in our real case study, we need to solve for 110 batches (i.e., $110 = 2200 \div 20$). The solution time for one batch should be within 7 minutes; otherwise, the proposed model and algorithm cannot be applied in this realistic context.

When applying our proposed methodology to this rolling horizon-based real case, we also must extend the original model to consider the ongoing executed plan when making the new plan. Specifically, when we plan the customized bus routes for one batch of orders, some of the customized buses scheduled for previous batches could also be used to serve customers in the current batch. Although a new batch of orders accumulates every 7 minutes, the duration to fulfil a batch of orders (calculated from the time when the first bus departs from its start point to the time when the last bus arrives at its end point) is much longer than 7 minutes. Details of this model extension are elaborated in Appendix E due to space limitations.



Historical demand density heat map | Solved routes for one batch | Results of two methods in rush hours

Figure 6: Demonstration of a real-world case in Dalian, China

Experiments are conducted to address the following three aspects. The first is a comparison of the BB-CG nested algorithm and LNS algorithm when applied to the real case of Dalian. As mentioned, we need to solve the model for 110 batches. The middle panel in Figure 6 demonstrates the locations of 20 orders in one batch, the locations of 17 candidate bus stops, and the three routes determined as a solution for the batch. The right panel in Figure 6 shows the results of a comparison between two solution methods for 25 batches of orders corresponding to the morning rush period, demonstrating that the BB-CG nested algorithm outperforms the LNS algorithm for all batches. We conduct 10 series of experiments using data from 10 days. Table 7 records the average objective value and average solution time across the 110 batches as computed by the model in each series of experiments. Each value (Obj_B, Obj_L, t_B, t_L) in Table 7 is the average value of the corresponding values from the 110 batches. The results of our experiments covering 10 days validate that our BB-CG nested algorithm outperforms the LNS algorithm; the average gap value is about 8.97%. All of the instances can be solved by our BB-CG nested algorithm within 7 minutes, which also implies the applicability of our algorithm in a real-world context. The LNS also has merit in terms of its solution time, although its solution quality is inferior to that of our BB-CG nested algorithm. The LNS algorithm may also be useful for some extremely large-scale applications (elaborated in Appendix D).

Table 7: Comparison between the BB-CG nested algorithm and LNS algorithm on real-world instances

Instances number	**BB-CG** Obj_B	t_B	**LNS** Obj_L	t_L	**Gap** $gap(\%)$
1	1059.71	93.75	1232.44	0.56	16.30
2	1699.38	123.73	1824.36	18.03	7.35
3	1491.59	94.61	1612.28	4.92	8.09
4	1401.17	79.50	1535.77	3.25	9.61
5	1612.22	29.45	1803.48	2.76	11.86
6	1601.29	29.52	1700.52	27.90	6.20
7	1729.81	61.19	1823.36	9.69	5.41
8	1666.06	27.19	1770.15	3.05	6.25

9	1480.28	41.85	1612.26	8.06	8.92
10	1506.80	47.27	1647.93	6.79	9.37
Average					8.97

Notes: $gap = (Obj_L - Obj_B)/Obj_B$

We conduct the second series of experiments to investigate the benefit of considering scheduled routes in the previous batch. As mentioned, we conduct the first series of experiments using an extended model to adapt to the context of a rolling horizon. However, our original, unchanged model can also be applied directly to the batches in the planning horizon; specifically, we can regard the 110 batches as 110 independent cases, implying that buses that are scheduled in previous batches and remain in use are not available for the order assignment of the batch that is being scheduled. We then can compare the extended and original models in the context of a rolling horizon and use the gap between them to reflect the benefit of considering routes scheduled in the previous batch. Table 9 presents the average values of the objectives in batches, which were calculated using either the extended model or the original model. The results in Table 9 validate the advantage of considering the routes scheduled in the previous batch: this brings a benefit of 22.98% in the average objective value per batch.

Table 8: Evaluating the benefit of considering scheduled routes in the previous batch

Instances number	**Original model**		**Extend model**		**Gap**
	Obj_O	t_O	Obj_E	t_E	gap(%)
1	1781.66	154.16	1407.28	27.39	21.01
2	1680.94	145.31	1335.64	24.54	20.54
3	1743.33	79.63	1400.00	43.14	19.69
4	1617.35	69.91	1243.82	36.29	23.09
5	1729.67	45.61	1435.59	14.18	17.00
6	1770.39	12.57	1435.66	6.33	18.91
7	1901.11	114.41	1433.61	87.51	24.59
8	1797.65	57.58	1132.11	38.27	37.02
9	1959.86	53.01	1402.96	28.09	28.42
10	1687.43	115.98	1358.53	84.81	19.49
Average					22.98

Notes:(1) Obj_O and obj_E denote the objective value for solving the original model and the expanded model using the BB-CG nest algorithm, respectively. (2) t_O and t_E denote the time for solving the original model and the expanded model using the BB-CG nest algorithm, respectively. (3) $gap = (Obj_O - Obj_E)/Obj_O$.

We conduct a third series of experiments to investigate the benefit of the zoning strategy, which is a common approach used to apply a model/algorithm to a larger context. The map shown in the left part of Figure 6 depicts just a central district of Dalian. If we apply our model to a wider territory, for example the entire city, the zoning strategy should be adopted. We note that zone planning is a strategic-level decision, whereas this study focuses on a real-time operational level decision; accordingly, the zones have been determined when making decisions in this study. If each order's origin and destination are located in the same zone, the orders could be allocated to zones according to the locations of their origins (or destinations); this would allow our proposed model to be solved for each zone independently. However, if there exists an order whose origin and destination are located in two zones, as shown in Figure 7, we would need to establish a rule to classify that order into a zone so that our proposed model could be applied to solve the routes in each zone.

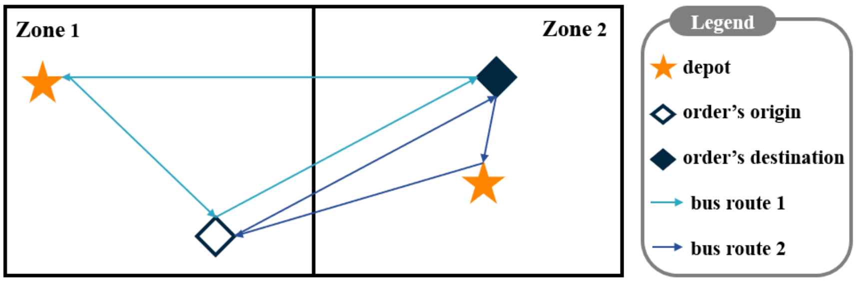


Figure 7: Example of an order whose origin and destination are located in two zones

Three intuitive rules can be adopted to identify the belongingness of the orders whose origins and destinations are located in different zones.

Rule 1: Each order is assigned to a zone according to its origin.

Rule 2: Each order is assigned to a zone according to its destination.

Rule 3: Each order is assigned to a zone according to the length of the bus route. For the example shown in Figure 7, if the length of bus route 2 is less than that of bus route 1, the order is assigned to Zone 2.

We then divide the municipal area of Dalian into four zones and experimentally compare the three zoning strategy rules. These comparative experiments are performed using two groups of five instances each; the percentage of orders that cross multiple zones is 20% in one group and 30% in the other group. Figure 9 illustrates the comparative results, which show that Rule 1 is the best of the three rules, while Rule 2 is the worst. We present the following possible reason and examples for these findings. Under Rule 1, a bus in a given zone serves orders whose origins are in that zone; the bus can also pick up passengers under other orders on the way to the order's origin. Under Rule 2 or Rule 3, a bus in Zone 1 serves an order whose origin is in Zone 2; the bus probably cannot pick up other passengers under other orders while traveling from Zone 1 to Zone 2 because the travel distance is relatively long and the start times of the time windows for the orders in the batch are usually similar to each other. Another finding from Table 7 is that the objective values in the instance group with 20% cross-zone orders are relatively smaller than those in the group with 30% cross-zone orders. This finding implies that as the proportion of cross-zone orders increases, the cost incurred by the whole customized bus service may also increase.

Table 9: Comparison of different rules of zoning strategy

Instances		**Rule 1**	**Rule 2**	**Rule 3**		**Gap**	
Percentage of cross-zone orders	ID	Obj_1	Obj_2	Obj_3	gap_1	gap_2	gap_3
20%	1	2361.09	2436.01	2324.54	1.57%	4.80%	0.00%

	2	2346.32	2716.63	2332.14	0.61%	16.49%	0.00%
	3	2350.99	2686.93	2419.81	0.00%	14.29%	2.93%
	4	2322.27	2663.24	2383.32	0.00%	14.68%	2.63%
	5	2394.82	2726.59	2397.63	0.00%	13.85%	0.12%
30%	1	2684.79	3312.24	2826.42	0.00%	23.37%	5.28%
	2	2650.73	3053.83	2639.45	0.43%	15.70%	0.00%
	3	2370.71	2778.02	2552.41	0.00%	17.18%	7.66%
	4	2646.42	3030.03	2655.24	0.00%	14.50%	0.33%
	5	2496.82	2780.20	2686.71	0.00%	11.35%	7.61%
Ave.					0.26%	14.62%	2.66%

Notes: Obj_1, Obj_2, and Obj_3 denote the sum of objective values of all zones according to rules 1, 2, and 3, respectively. Define $Obj_{min} = \min\{Obj_1, Obj_2, Obj_3\}$, and $gap_1 = (Obj_1 - Obj_{min})/Obj_{min}$, $gap_2 = (Obj_2 - Obj_{min})/Obj_{min}$, $gap_3 = (Obj_3 - Obj_{min})/Obj_{min}$.

8. Conclusions

This paper investigates a VRP variant for a customized bus service. Given a set of orders containing passengers' origins, destinations, and time windows, the routes of buses to serve these orders and the pick-up and drop-off locations for each order are decided to minimize the total costs of bus usage and the travel routes. The methodology proposed in this study can pave the way for developing more intelligent software for customized bus service platforms. The main contributions of the study are as follows.

From a modeling perspective, the proposed MILP model considers a comprehensive panel of factors affecting customized bus services. For example, it considers decisions on the pick-up and drop-off locations for each order, which contains information about the number of passengers and the earliest (latest) time when they can depart from their origin (should arrive at their destination). The walking times of passengers from their origins to the pick-up locations (or from the drop-off locations to their destinations) and the passenger capacities of different bus types are also taken into account. The proposed model thus can be regarded as a new VRP variant that considers floating targets. Our model may contribute to the VRP literature in terms of increasing the consideration of coordination issues between customers and vehicles.

From an application perspective, we propose a novel solution method with an embedded acceleration technique (i.e., a multi-labelling algorithm) to solve our model rapidly. Accordingly, our proposed methodology is applicable in a real-world context. The proposed BB-CG nested solution method can be used to solve much larger-scale instances than the CPLEX solver, in a much shorter time. Our real data-based experiments demonstrate the applicability of our methodology to Dalian, where the customized bus service receives more than 2,000 orders per day, in the scenario of batch-based rolling horizon decision-making. We additionally provide managerial insights to support decisions on, for example, the appropriate number of candidate stops and the numbers of different types of buses. We further investigate extensions of our model to account for aspects such as zoning strategy to determine whether our methodology is applicable in more generic contexts.

This study also has some limitations. Although the optimality gaps are zero for the majority of the instances tested in our experiments, the proposed BB-CG nested algorithm is a heuristic and thus differs from the exact branch-and-price algorithm; the optimality loss in our algorithm lies in the stage of obtaining feasible solutions within the CG. Besides the limitation on the heuristic, this study does not consider uncertain demands; and some realistic factors may not be considered in the current model. Future studies could be conducted according to the following directions.

- We can explore the development of an exact algorithm to solve the problem considered herein.
- Stochastic models can be developed to make decisions for a current batch while simultaneously considering the uncertain demand in the next batch.
- Multi-objective model could be formulated to consider both the traditional objective and some environment related objectives (Dulebenets 2022).

All of these directions for future studies could provide interesting information about this problem.

Funding

This research is supported by the National Natural Science Foundation of China (Grant numbers 72025103, 71831008, 72071173, 71871043).

Notes on contributors

Lu Zhen is a Professor and Dean at School of Management, Shanghai University, Shanghai, China. His research interests include operations management and optimization; mixed-integer linear programming and algorithms; port operations and maritime transportation; urban logistics and supply chain management. He has served as an associate editor or an editorial board member of five journals such as Transportation Research Part B, Journal of the Operational Research Society, Computers & Operations Research; and he is also the Fellow of the Operational Research Society (U.K.). He has been awarded the National Funds for Distinguished Young Scientists, for Outstanding Young Scientists in China, the Changjiang Young Scholar in China, and etc.

Xueting He is a Ph.D. student at School of Management, Shanghai University, Shanghai, China. Her research interests include vehicle routing problem's variants. She has published six papers on SCI international journals.

Shuaian Wang is a Professor at Faculty of Business, The Hong Kong Polytechnic University, Hong Kong, China. His research interests include big data in shipping, green shipping, shipping operations management, port planning and operations, urban transport network modeling, and logistics and supply chain management. He is an editor-in-chief of Cleaner Logistics and Supply Chain and Communications in Transportation Research, an associate editor of Transportation Research Part E, Flexible Services and Manufacturing Journal, Transportmetrica A, and Transportation Letters, a handle editor of Transportation Research Record, an editorial board editor of Transportation Research Part B, and an editorial board member of Maritime Transport Research.

Jingwen Wu is a Ph.D. student at School of Management, Shanghai University, Shanghai, China. Her research interests include courier routing for delivery. She has published four papers on SCI international journals.

Kai Liu is a Professor at School of Transportation and Logistics, Dalian University of Technology, China. His research interests include transportation planning and management, travel behavior modeling, GIS in Transportation, and Transportation Electrification. He is an editorial board member of Transportation Research Part D, an associate editor of Asian Transport Research.

References

Agatz N., Erera A., Savelsbergh M., Wang X. (2012) Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research* 223(2): 295–303

Asghari M., Al-e-Hashem S.M.J.M., Rekik Y. (2022a) Environmental and social implications of incorporating carpooling service on a customized bus system. *Computers & Operations Research* 142: 105724.

Asghari M., Fathollahi-Fard A.M., Al-e-hashem S.M.J.M., Dulebenets M.A. (2022b) Transformation and linearization techniques in optimization: a state-of-the-art survey. *Mathematics* 10(2): 283.

BTDRC (2020) *Beijing Transport Development Annual Report*. Beijing Transportation Development Research Center (BTDRC): Beijing, China, 2020.

Chen X., Wang Y., Ma X. (2021) Integrated optimization for commuting customized bus stop planning, routing design, and timetable development with passenger spatial-temporal accessibility. *IEEE Transactions on Intelligent Transportation Systems* 22 (4): 2060–2075.

Chen X., Wang Y., Wang, Y., Qu X., Ma X. (2021) Customized bus route design with pickup and delivery and time windows: Model, case study and comparative analysis. *Expert Systems with Applications* 168: 114242.

Dou X., Meng Q., Liu K. (2021) Customized bus service design for uncertain commuting travel demand. *Transportmetrica A - Transport Science* 17(4): 1405–1430.

Dulebenets M.A. (2022) Multi-objective collaborative agreements amongst shipping lines and marine terminal operators for sustainable and environmental-friendly ship schedule design. *Journal of Cleaner Production* 342: 130897.

Fielbaum A., Bai X., Alonso-Mora J. (2021) On-demand ridesharing with optimized pick-up and drop-off walking locations. *Transportation Research Part C* 126: 103061.

Gambella C., Naoum-Sawaya J., Ghaddar B. (2018) The vehicle routing problem with floating targets: formulation and solution approaches. *INFORMS Journal on Computing* 30(3): 554–569

Gong M., Hu Y., Chen Z., Li X. (2021) Transfer-based customized modular bus system design with passenger-route assignment optimization. *Transportation Research Part E* 153: 102422

Guan Y., Wang Y., Yan X., Guo H., Zhao Y. (2022) The One E-Ticket Customized Bus Service Mode for Passengers with Multiple Trips and the Routing Problem. *Sustainability* 14(4): 2124.

Guo R., Zhang W., Guan W., Ran B. (2021) Time-dependent urban customized bus routing with path flexibility. *IEEE Transactions on Intelligent Transportation Systems* 22(4): 2381–2390.

Huang D., Gu Y., Wang S., Liu Z., Zhang W. (2020) A two-phase optimization model for the demand-responsive customized bus network design. *Transportation Research Part C* 111: 1–21.

Huang K., Xu L., Chen Y., Cheng Q., An K. (2020) Customized bus route optimization with the real-time data. *Journal of Advanced Transportation* 2020: 8838994.

Kim M., Levy J., Schonfeld P. (2019) Optimal zone sizes and headways for flexible-route bus services. *Transportation Research Part B* 130: 67–81.

Lee E., Cen X., Lo H.K., Ng K.F. (2021a) Designing zonal-based flexible bus services under stochastic demand. *Transportation Science* 55(6): 1280–1299.

Lee E., Cen X., Lo H.K. (2021b) Zonal-based flexible bus service under elastic stochastic demand. *Transportation Research Part E* 152: 102367.

Li Y., Li X., Zhang S. (2021) Optimal pricing of customized bus services and ride-sharing based on a competitive game model. *Omega* 103: 102413.

Liu J., Zhao H., Li J., Yue X. (2021) Operational strategy of customized bus considering customers' variety seeking behavior and service level. *International Journal of Production Economics* 231: 107856.

Lyu Y., Chow C.-Y., Lee V.C.S., Ng J.K.Y., Li Y., Zeng J. (2019) CB-Planner: a bus line planning framework for customized bus systems. *Transportation Research Part C* 101: 233–253.

Ma C., Wang C., Xu X. (2021) A multi-objective robust optimization model for customized bus routes. *IEEE Transactions on Intelligent Transportation Systems* 22(4): 2359–2370.

Matin-Moghaddam N., Sefair J.A. (2021) Route assignment and scheduling with trajectory coordination. *IISE Transactions* 53(2): 164–181.

Montufar J., Arango J., Porter M., Nakagawa S. (2007) Pedestrians' normal walking speed and speed when crossing a street. *Transportation Research Record* 2002: 90–97.

Pasha J., Nwodu A.L., Fathollahi-Fard, A.M., Tian G., Li Z., Wang H., Dulebenets M.A. (2022) Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings. *Advanced Engineering Informatics* 52: 101623.

Shu W., Li Y. (2022) A novel demand-responsive customized bus based on improved ant colony optimization and clustering algorithms. *IEEE Transactions on Intelligent Transportation Systems*. In press.

Tong L., Zhou L., Liu J., Zhou, X. (2017) Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transportation Research Part C* 85: 451–475.

Wang W., Wu Y. (2021) Is uncertainty always bad for the performance of transportation systems? *Communications in Transportation Research* 1: 100021.

Wang J., Yamamoto T., Liu K. (2019) Role of customized bus services in the transportation system: insight from actual performance. *Journal of Advanced Transportation* 2019: 6171532.

Wang J., Yamamoto T., Liu K. (2020) Key determinants and heterogeneous frailties in passenger loyalty toward customized buses: An empirical investigation of the subscription termination hazard of users. *Transportation Research Part C* 115: 102636.

Xia D., Zheng L., Cai X., Liu W., Sun D. (2022) Urban customized bus design for private car commuters. *IEEE Internet of Things Journal* 9(21): 21723–21735.

Xue S., Song R., He S., An J., Wang Y. (2022) An improved adaptive large neighborhood search algorithm for the heterogeneous customized bus service with multiple pickup and delivery candidate locations. *Journal of Advanced Transportation* 2022: 1679469.

Yu M., Shen S. (2020) An integrated car-and-ride sharing system for mobilizing heterogeneous travelers with application in underserved communities. *IISE Transactions* 52(2): 151–165.