Research Article

An Improved Adaptive Large Neighborhood Search Algorithm for the Heterogeneous Customized Bus Service with Multiple Pickup and Delivery Candidate Locations

Shouqiang Xue (✉), Rui Song (✉), Shiwei He (✉), Jiuyu An, and Youmiao Wang

Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Rui Song; rsong@bjtu.edu.cn

Received 17 September 2021; Accepted 17 March 2022; Published 27 May 2022

Academic Editor: Yajie Zou

In order to tackle the congestion and environmental issues, customized bus services are proposed and deployed in metropolitan areas. As emerging public transportation services, customized bus services bring passengers more convenience and accessibility. Besides, conventional customized bus services generally organize homogeneous fleet and single location selection to passengers. In this paper, to enhance the mobility and flexibility of customized buses and increase companies’ profit, we propose a new form of customized bus service with heterogeneous fleets and multiple candidate locations. First, a mixed-integer programming model (MIP) is developed to describe the customized bus problem. Compared with the conventional model, the proposed MIP is involved in the case of one passenger with multiple candidate pickup or delivery locations and can be solved by GUROBI on small scale, quickly and efficiently. Second, an improved adaptive large neighborhood search algorithm (ALNSip) is utilized to address the large-scale problem more efficiently. Time slack calculation method is then designed to optimize vehicle timetables, which provides stable and excellent performance for searching feasible solutions. In addition, we propose two inserting operators to deal with the problem with multiple candidate locations and analyse its influence on the results. Finally, we test the performance of the proposed model and algorithm on the numerical experiments. And they are verified the effectiveness and implication in a small-scale case on a simplified Sioux waterfall network and a large-scale problem in Beijing, China. The result shows that ALNSip outperforms other algorithms in searching for more satisfying solutions with higher efficiency. However, the GUROBI solver can obtain the solution to small-scale problems within a shorter time than ALNSip. Furthermore, it can be suggested that the heterogeneous fleets service with multiple candidate locations is helpful to facilitate collaboration among vehicles and optimize pickup and delivery routes in consequence.

1. Introduction

The increase in the number of private cars has imposed an impact on potential traffic jams and environmental issues, and in the meanwhile, the increasing number of people has grown interest in enjoying personalized travel services. Therefore, many transportation service providers are gradually focusing their attention on seeking new services to attract more public transport users and increase passengers’ accessibility. In recent years, internet-based customized bus services have become more popular in the market due to its higher accessibility and quicker response, especially in congested metropolitan areas. Since Kirby and Bhatt analysed and discussed in detail about subscribing to the bus service in 1974 [1], the customized bus system (CB) has been continuously developed into a demand-responsive transportation service. Nowadays, some transportation companies have constructed mobile application platforms for customized bus services to collect detailed travel information [2, 3]. With the spatial and temporal characteristics of travel requests taken into considerable consideration, the customized bus service platform can aggregate similar requests and design bus routes. To the best of our knowledge, some customized bus services have been deployed in physical road networks of metropolitan areas such as Beijing.

According to different characteristics of travel requests, customized bus services can be usually divided into four major
categories: “Customized commuter bus,” “Customized school bus,” “Customized business bus,” and “Customized community bus,” and other kinds of customized feeder or shuttle bus systems [2]. This paper focuses on the customized commuter bus, which offers commute shuttle service and takes customers from the residential community to the workplace. This is also the most important CB system. It is noted that there are some unique service features about commuters. They usually book relatively fixed demands and change them in different weeks or months. Therefore, the customized bus services will be updated according to new requests in each optimization time (e.g., one month and one week) [4].

The detailed operating procedure of the customized bus service is presented in Figure 1. Each passenger needs to input his/her request through the network and platform to complete the service-design process. This paper mainly focuses on the process to plan and deploy new routes based on fail-to-match requests in the requests pool. If requests cannot be matched with the existing routes, these requests will be put into the requests pool. When the number of booked people meets the minimum load requirement, the platform will plan new routes deployed in the physical road network based on the requests in the requests pool.

There are several notable features of customized bus problems. As can be seen from the above operating procedure, compared to the traditional bus routing problem, customized bus problems pay more attention to individual passenger request information. Bus route and specific timetables are updated and designed by considering more about companies’ profit and various characteristics of passenger demands, including origin-destination pairs and time windows. In the field of operations research, the problem regarding passengers’ pickup and delivery locations with time windows can be classified as a special vehicle planning problem [5]. With the consideration of all constraints, it is worthwhile but challenging to design reasonable routes and specific timetables. Furthermore, it is also of great significance to enrich the personalized public transportation services and provide highly reliable and accessible trip services for commuters living in congested metropolitan areas.

The remainder of this paper is organized as follows. The next section describes an overview of earlier research. Section 3 describes the MIP model. In Section 4, we describe ALNS$_p$, in detail. Section 5 reports the test of the MIP and ALNS$_p$, on numerical experiments. Finally, Section 6 summarizes the research content and future research directions of the paper.

2. Literature Review

Customized bus service can be regarded as a request-oriented public transportation service based on its unique service features. In earlier research, on-demand transportation was described as a set of vehicle routing problems with pickup and deliveries (VRPPD). Nagy et al. [6] divided the previous studies about the problem into three main categories: simultaneous pickups and deliveries, mixed pickups and deliveries, and delivery-first, pickup-second. In subsequent research, VRPPD is gradually evolved into different versions due to the specific problems and traffic conditions. When the pickup and delivery locations with time windows are determined, VRPPD can be seen as pickup and delivery problems with time windows (PDPTW). In general, the exact algorithm is a major way to solve multivehicle pickup and delivery problems in most studies [7–9]. Ropke et al. [10] proposed a new branch-and-cut-and-price algorithm to tackle the problem, which dynamically introduced valid inequalities and heuristic algorithm to enhance algorithm performance. Wang et al. [11–14] constructed related optimization models and proposed algorithms based on state-space-time network to effectively solve some pickup and delivery problems.

With the development of mobile technology, passengers can book vehicles and pay for trips through smartphones or platforms. In the operations research, this type of transportation service can be classified as dial-a-ride problems (DARP). Compared to VRPPD, DARP offers more unique characteristics, including depots, time window, vehicles capacity, ride time, and route duration. Stein [15] proposed the first DARP model and gave the bounds of both static and dynamic models. For static and dynamic DARP versions, Psaraftis et al. [9] developed a dynamic programming algorithm for the problem with a single vehicle. For the multiobjective problem in DARP, Jørgensen et al. [16] considered the total weight of multiple objects and proposed a Granular Tabu Search algorithm to solve the static DARP. In addition, Garaix et al. [17] and Schilde et al. [18] considered optimizing multiobjective problems by priority. And Paquette et al. [19] used the Pareto boundary to solve it.

The original application of DARP was a nonprofit dial-a-ride system service (DAR) for the elderly and the disabled. Relevant constraints include ride and waiting time, pickup or delivery with time windows, vehicles capacity, and fleet systems [20–23]. However, due to market demand and emerging communication technology, DARP has been applied in the field of public transportation. Based on the on-demand transportation system developed in rural France, Garaix et al. [17] designed a free-riding system that maximized customer occupancy and used the column generation way to deal with it. The problem where not all passengers are collected in advance is classified as dynamic on-demand transportation services [5]. Pillac et al. [24] introduced the concept of dynamic problem and reviewed the solution methods about it. Alonso-Mora et al. [25] paid attention to the dynamic ride-sharing problem with high-capacity and proposed a pairwise shareability graph to compute feasible trips. Studies about customized bus service, one type of demand-responsive transport service, are also gradually increasing. Liu et al. [2] introduced the development and application of customized bus services in China. To solve customized bus service-design problems, Tong et al. [4] proposed a multimmodity optimization model and provided a Lagrangian decomposition-based solution algorithm to solve the routing problems.

Furthermore, in terms of solving route planning problems, the neighborhood search algorithm always has a high performance [26–31]. To improve the performance of neighborhood search algorithm, Ropke and Pisinger [32] proposed an adaptive large neighborhood search algorithm for the first time (ALNS). In the subsequent studies, it is found that ALNS has an outstanding performance in solving large-scale route planning problems with NP-hard, and it has been applied to different fields. To stack with the pollution routing problem (PRP), Demir et al. [33] improved the ALNS algorithm and updated some
mechanisms, applicable to the problem. Polat et al. [34] proposed a mixed-integer mathematical optimization model and design a perturbation-based variable neighborhood search heuristic to solve the vehicle routing problem with simultaneous pickup and delivery with time limit (VRPSPDTL). To study the share-a-ride problem, Li et al. [35] proposed an adaptive large neighborhood search method with time slack. Yu et al. [36] and Yu et al. [37] developed an adaptive large neighborhood search heuristic to solve green vehicle routing problems in large-scale instances. In addition, ALNS is also widely used in the vehicle routing problem with drones, a supply chain, carsharing service, train stop planning, timetabling, and other fields [38–41]. Furthermore, there are also some hybridization and improvement from ALNS and other algorithms [42–44].

In the paper, to further improve passengers’ accessibility and increase bus operators’ profits, we consider adding two strategies (Strategy 1 and Strategy 2) into customized bus services.

(I) **Strategy 1:** the heterogeneous fleets with different operating cost and vehicle capacity can be dispatched to offer customized bus service

(II) **Strategy 2:** passengers can choose multiple candidate locations as their pickup and delivery locations

The existing literature still lacks relative research on the heterogeneous customized bus service problem with multiple pickup and delivery candidate locations. With the comprehensive consideration about the new service form, this paper builds a new MIP model based on the classical DARP model proposed by Cordeau [20] to address the practical and theoretical challenges. At the same time, to effectively solve large-scale problems, we introduce and improve ALNS (ALNSp) proposed by Li et al. [35] by designing corresponding operators (M1.1 and M1.2) to address new challenges with multiple candidate locations. In addition, given the complexity of the problem with time window, a time slack calculation method is utilized to adjust vehicle arriving time during searching feasible solutions.

In summary, this study has several contributions in theory and application: (I) constructing a new bus service with heterogeneous fleets and multiple candidate locations to improve conventional service quality and facilitate collaboration among different buses, (2) formulating relative optimization model with considering the different time windows, capacities, passenger assignment, and location selection to maximize operator’s profit, (3) designing an improved heuristic algorithm based on ALNS, including a time slack calculation method and two inserting operators to address the large-scale problem with high efficiency.

### 3. Mathematical Model

The PDPTW and the DARP can be modelled using classical formulation (see, e.g., Cordeau [20] and Dumas et al. [7]). It is usually utilized to optimize the problem where each request is represented by a set of unique pickup location and delivery location. In this paper, to further tackle the challenges with multiple depots, heterogeneous fleets and multiple candidate locations, we propose a MIP model to deal with the problem, where the state node concept is utilized to describe the relationship between every passenger and their pickup and delivery locations.

#### 3.1. Problem Statement

We define the problem on a direct graph $G = (S, \Lambda)$, where $S$ is the set of all state nodes and $\Lambda$ is the set of all arcs connecting state nodes. Each state node $(p, i)$ in $S = \{(p, i)|p \in P, i \in V\}$ means the passenger $p$ is picked up and delivered at the physical location $i$. $P$, $V$, and $K$ mean, respectively, the set of passenger requests, the set of physical locations, and the set of customized buses. For each passenger $p_0$, $(p_0, i) \in S_p$ is the set of state nodes for passenger $p_0$ picked up, and the same for $(p_0, i) \in Sq$ is the set of state nodes for passenger $p_0$ delivered. $\Lambda = \{(p, i), (q, j)|(p, i) \in S, (q, j) \in S\}$ indicates one directed arc from one state node $(p, i)$ to another state node $(q, j)$, where bus moves from location $i$ to location $j$ to offer service for passenger $p$ and passenger $q$. Although one passenger may have multiple pickup and delivery locations, he/she must be picked up or delivered at a unique location.
To guarantee bus operators’ profit, the number of passengers the bus serves must meet the minimum load constraint. As a result, if the loading capacity of bus \( k \) is \( \text{Cap}_k \), the number of passengers on the bus must be more than or equal to \( \text{Cap}_k \). Given a number of service features from customized buses, some unreasonable passenger requests may be rejected. The rejected passengers in the model are served by the vehicle \( k \in K^V \), where \( K^V \) is the set of virtual vehicles. Expensive cost from virtual vehicles is the penalty cost for unserved customers. The remaining passengers are served by the buses \( k \in K^L \cup K^S \), where \( K^L \) is the bus set with large capacity.

\[
\text{Max} E - C^d - C^f - C^p.
\]

\[
E = \sum_{k \in K^L \cup K^S} \sum_{(p,j) \in S} \sum_{q \in S} t_{p0} \cdot y_{(p,j),(q,j)},
\]

\[
C^d = \sum_{k \in K^V} \sum_{v \in V_O} \sum_{(p,j) \in S} c^d_k \cdot x_{v,(p,j)}^k,
\]

\[
C^p = \sum_{k \in K^V} \sum_{v \in V_O} \sum_{(p,j) \in S} c^p_k \cdot x_{v,(p,j)}^k,
\]

\[
C^f = \sum_{k \in K^L \cup K^S} \sum_{v \in V_O} \sum_{(p,j) \in S} c^f_k \cdot \cdot t_{(v,d)} \cdot x_{v,(p,j)}^k + \sum_{k \in K^L \cup K^S} \sum_{(q,j) \in S} \sum_{p \in V_O} c^f_k \cdot \cdot t_{(i,j)} \cdot y_{(p,j),(q,j)}^k.
\]

\[
\sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k \leq 1, \quad \text{for } k \in K.
\]

\[
\sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k = \sum_{v \in V_O} \sum_{(q,j) \in S} x_{v,(q,j)}^k, \quad \text{for } k \in K.
\]

\[
\sum_{k \in K} \sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k + \sum_{k \in K^L \cup K^S} \sum_{(p,j) \in S} y_{(p,j),(p_0,j)}^k = 1, \quad \text{for } p_0 \in P.
\]

\[
\sum_{k \in K} \sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k + \sum_{k \in K^L \cup K^S} \sum_{(q,j) \in S} y_{(p,j),(q,j)}^k = 1, \quad \text{for } p_0 \in P.
\]

\[
\sum_{(p_0,i) \in S} \sum_{(q,j) \in S} y_{(p_0,i),(p_0,j)}^k = \sum_{(p_0,i) \in S} \sum_{(q,j) \in S} y_{(p_0,i),(q,j)}^k, \quad \text{for } p_0 \in P, \text{ for } k \in K.
\]

\[
\sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k + \sum_{(q,j) \in S} \sum_{(p,j) \in S} y_{(p,j),(q,j)}^k = \sum_{(q,j) \in S} \sum_{(q,j) \in S} y_{(p,j),(q,j)}^k, \quad \text{for } k \in K, \text{ for } (p_0,i) \in S_p.
\]

\[
\sum_{v \in V_D} x_{v,(q,j)}^k + \sum_{(p,j) \in S} y_{(p,j),(p_0,j)}^k = \sum_{(p,j) \in S} y_{(p,j),(p_0,j)}^k, \quad \text{for } k \in K, \text{ for } (q_0,j) \in S_d.
\]

\[
\sum_{(q,j) \in S(p,j) \in S} y_{(p,j),(q,j)}^k \leq \sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k \cdot (2 \cdot \text{Cap}_k^u - 1), \quad \text{for } k \in K.
\]

\[
\sum_{(q,j) \in S(p,j) \in S} y_{(p,j),(q,j)}^k \geq \sum_{v \in V_O} \sum_{(p,j) \in S} x_{v,(p,j)}^k \cdot (2 \cdot \text{Cap}_k^l - 1), \quad \text{for } k \in K.
\]
Table 1: Notation and definition in the model.

| Set                | Definition                                                                 |
|--------------------|-----------------------------------------------------------------------------|
| $P$                | Set of all passengers                                                      |
| $S$                | Set of all state nodes, $S = S_P \cup S_d$                                  |
| $\Lambda$          | Set of all arcs connecting state nodes; $\Lambda = \{(p, i), (q, j)\mid (p, i) \in S, (q, j) \in S\}$ indicates one directed state arc from one state node $(p, i)$ to another state node $(q, j)$, where bus moves from location $i$ to location $j$ to offer service for passenger $p$ and passenger $q$ |
| $K$                | Set of all customized bus: $K = K_L \cup K_S \cup K_V$                      |
| $K_L$              | Set of customized bus with large capacity                                   |
| $K_S$              | Set of customized bus with small capacity                                   |
| $K_V$              | Set of virtual customized bus                                               |
| $V$                | Set of all physical locations: $V = V_P \cup V_d \cup V_O \cup V_D$         |
| $V_P$              | Set of pickup locations in the transportation network                       |
| $V_d$              | Set of delivery locations in the transportation network                     |
| $V_O$              | Set of start depots of customized bus in the transportation network         |
| $S_P$              | State node means the passenger is picked up at the location: $(p, i) \in S_P$, $p \in P, i \in V_P$ |
| $S_d$              | State node means the passenger is delivered at the location, $(p, i) \in S_d$, $p \in P, i \in V_d$ |

| Parameters         |                                                                 |
|--------------------|----------------------------------------------------------------|
| $E$                | Total profits                                                  |
| $C_d$              | Total departure cost                                           |
| $C_f$              | Total fuel cost                                                |
| $C_p$              | Total penalty cost                                             |
| $t_{P_0}$          | The ticket price                                               |
| $c_d$              | Departure cost of bus $k \in K$                               |
| $c_f$              | The fuel cost per unit time of bus $k \in K$                   |
| $\text{Cap}_u^k$  | The maximum load capacity of customized bus $k \in K$          |
| $\text{Cap}_l^k$  | The minimum load capacity of customized bus $k \in K$          |
| $[\varepsilon^l(p), \varepsilon^u(p)]$ | Time window when passenger $p$ is picked up, where $\varepsilon^l(p)$ and $\varepsilon^u(p)$ represent the earliest and latest time, respectively |
| $[\ell^l(p), \ell^u(p)]$ | Time window when passenger $p$ is delivered, where $\ell^l(p)$ and $\ell^u(p)$ represent the earliest and latest time, respectively |

| $(p, i)$           | State node passenger $p$ is picked up or delivered at location $i$, $(p, i) \in S$ |
| $T_{(p,i)}$        | The time of passenger $p$ is picked up or delivered at location $i$ |
| $t_{(i,j)}$        | Travel time between location $i$ and $j$, $i, j \in V$          |

| Decision variable  |                                                                 |
|--------------------|----------------------------------------------------------------|
| $\mu_{(p,i)}$      | An auxiliary variable: it indicates the order of state node $(p, i)$ in the route |
| $x^k_{v,(p,i)}$     | Customized bus $k$ departs from depots or return depots |
| $y^k_{(p,i),(q,j)}$ | The bus $k$ moves from the state node $(p, i)$ to the state node $(q, j)$ |

\begin{align*}
T_{(q,j)} & \geq T_{(p,i)} - M \cdot \left( 1 - \sum_{k \in K} y^k_{(p,i),(q,j)} \right), \quad \text{for} \ (p, i), (q, j) \in S, \\
T_{(p,i)} & = \begin{cases} 
[\varepsilon^l(p), \varepsilon^u(p)], & \text{for} \ (p, i) \in S_P, \\
[\ell^l(p), \ell^u(p)], & \text{for} \ (p, i) \in S_d,
\end{cases} \\
\mu_{(q,j)} & \geq \mu_{(p,i)} + 1 - M \cdot \left( 1 - \sum_{k \in K} y^k_{(p,i),(q,j)} \right), \quad \text{for} \ (p, i), (q, j) \in S, \\
x^k_{v,(p,i)} & \in \{0, 1\}, \quad \text{for} \ k \in K, \text{for} \ v \in V_O \cup V_D, \text{for} \ (p, i) \in S, \\
y^k_{(p,i),(q,j)} & \in \{0, 1\}, \quad \text{for} \ k \in K, \text{for} \ (p, i), (q, j) \in S.
\end{align*}
The objective function (1) maximizes total profits. Constraints (2)–(5) are ticket income, total departure cost, penalty cost, and fuel cost, respectively. And constraints (3) and (4) can be represented by constraint (20).

\[ C^d + C^p = \sum_{k \in K} \sum_{v \in V} \sum_{e \in E_p (p,j) \in S_p} k^d \cdot x^k_{v,(p,j)} + \sum_{k \in K} \sum_{v \in V} \sum_{e \in E_p (p,j) \in S_p} k^d \cdot x^k_{v,(p,j)} = \sum_{k \in K} \sum_{e \in E_p (p,j) \in S_p} \sum_{k \in K} \sum_{e \in E_p (p,j) \in S_p} c^k \cdot x^k_{v,(p,j)}. \] (20)

For vehicles, constraints (6) and (7) ensure that each vehicle travels at most once, and each one leaves start depots and enters start depots. For some customers offering multiple locations as their candidate pickup or delivery locations, he/she can only be picked up or delivered at one location in the end, and the platform also will choose one location from their multiple pickup or delivery candidate locations. Therefore, constraints (8) and (9) ensure that all passengers are served and each passenger is picked up or delivered at a unique location. In addition, constraint (10) indicates that one person is just picked up and delivered by the same vehicle. For network flow, constraints (11) and (12) enforce traffic flow balance. If vehicle \( k \) arrives in location \( i \) to serve one person, it will leave from location \( i \) to location \( j \) to serve another one. Constraint (13) guarantees that the total number of served passengers cannot exceed the capacity of vehicle. To achieve the profit goal, constraint (14) ensures a minimum number of passengers in the vehicle. Constraints (15) and (16) enforce each passenger to be picked up or delivered within he/her time windows. Constraint (17) is subtour elimination constraint. Finally, constraints (18) and (19) define the binary decision variables. If the vehicle \( k \) moves from the state node \((p,i)\) to the state node \((q,j)\), then \( x^k_{(p,i)\rightarrow(q,j)} = 1 \); otherwise, \( x^k_{(p,i)\rightarrow(q,j)} = 0 \). If one vehicle departs from start depots or returns to end depots, then \( x^k_{v,(p,i)} = 1 \); otherwise, \( x^k_{v,(p,i)} = 0 \).

4. An Adaptive Large Neighborhood Search Algorithm for Customized Bus Service System

In this section, we describe ALNS\(_p\) and its application to the problem in great detail. The ALNS\(_p\) constructs the initial feasible solution using the greedy algorithm represented in Section 4.1. It uses several competing removal and insertion algorithms and chooses a reasonable set of operators by the score of each operator during the search. The removal algorithms are described in Section 4.3, and the insertion algorithms are described in Section 4.4. Furthermore, we propose a time slack calculation strategy to adjust timetables for searching more feasible solutions.

4.1. Initial Solution Construction. A greedy insertion algorithm is used to search for a feasible initial solution. Requests in the set \( P \) are inserted one by one in a random order into routes. The entire process can be shown in the pseudocode of Algorithm 1.

4.2. ALNS\(_p\) Process. To diversify the search, the ALNS\(_p\) selects suitable removal and insertion operators by a roulette wheel mechanism at every iteration. We define every \( U/p \) times as a cycle. The probabilities of choosing operator \( i \) in the \( k^{th} \) cycle, \( p^i_k \), is updated as formulation \( p^i_k = w^i_k \Delta f^i_k \sum w^i_k \), where \( w^i_k \) is the weight of the operator \( i \) in the \( k^{th} \) cycle. For every \( U/p \) iteration, weights are recalculated using the scores obtained as formulation \( w^i_k = w^{k-1}_{i}(1-\gamma) + \gamma \sigma^{k-1}_{i} m^{k-1}_{i} \), where \( \gamma \) is the weight coefficient and \( m^{k-1}_{i} \) are the overall score and selected times of the operator \( i \) in \( k^{th} \) cycle. If the operator \( i \) is chosen, its score is increased by \( \lambda_p(q = 1, 2, 3 \ or \ 4) \), as shown in Table 2, according to its performance at the current iteration. For every \( U/p \) iteration, all scores are reset to zero. Finally, the entire process of ALNS\(_p\) is completed under \( U \) iterations, as shown in Algorithm 2.

Some parameters are shown in Table 2 by using the tuning strategy proposed by Ropke and Pisinger [32] and Li et al. [35] to obtain values. When just one parameter is adjusted, the rest will be fixed. Finally, the setting with the best average behaviour is chosen.

The acceptance criterion adopts the simulated annealing criterion shown as Algorithm 2, where \( f^{\text{best}}(S) \) is the objective value of the new solution, \( f^{\text{best current}}(S) \) is the objective value of the current solution, and \( f^{\text{best global}}(S) \) is the objective value of the global optimal solution. The probability that we accept new solution is \( \rho = e^{-\phi (f^{\text{best current}}(S)) - f^{\text{best current}}(S))} T_k \), where \( T = T_0 e^\epsilon \) is the current temperature, \( \epsilon \) is the cooling rate, and \( T_0 \) is the initial temperature.

4.3. Removal Algorithms (RA). In each iteration, a removal operator is selected from RA and removes the total number of passenger requests \( n_p = u \cdot P \) in the solution to break the current solution, where \( u \) is the random removal parameter. These requests removed are added to the perturbation set \( R_P \). The section has designed five operators, the first three are chosen from Ropke and Pisinger [32], the fourth operator is adapted from Demir et al. [33], and the last one is motivated by Cordeau and Laporte [5].

R1-Random Removal: the operator randomly removes \( n_p \) requests from the current solution put them into the set \( P \).

R2-Worst-Removal: the operator calculates the \( \Delta f^p \) value of all passengers, where \( \Delta f^p \) denotes the change in objective value incurred by removing passenger request \( p \) from current position. Then, \( n_p \) requests with the greatest value \( \Delta f^p \) are removed and put into the set \( R_P \).

R3-Shaw Removal: to diversify the search, the operator measures the similarity level \( R_{ij} \) between passenger \( i \) and passenger \( j \) through the correlation function \( R_{ij} = \phi_1 \cdot (d^p_{ij} + d^m_{ij}) + \phi_2 \cdot (l^p_{ij} - l^m_{ij}) + \phi_3 \cdot l_{ij} \), where \( \phi_1 - \phi_3 \) are the Shaw coefficients, \( d^p_{ij} \) is the
(1) Distance between pickup locations of passengers $i$ and $j$, $d_{ij}$ is the distance between delivery locations, $t_{p_i}$ is the boarding time, $t_{d_i}$ is the alighting time, $l_{ij}$ indicates whether passengers $i$ and $j$ are in the same vehicle ($l_{ij} = -1$) or not ($l_{ij} = 1$). The operator randomly chooses one customer and calculates all $R_{ij}$ between the

Algorithm 1: Initial solution.

(1) Input initial solution $S$, the objective values of initial solution $S$ as initial values of $f_{\text{best}}^\text{current}(S)$ and $f_{\text{best}}^\text{global}(S)$;
(2) $i \leftarrow 1$
(3) While $i < U$
(4) Select a couple of removal and insertion operators using a roulette wheel mechanism to generate new feasible solution $S'$;
(5) If $f' (S') > f_{\text{best}}^\text{global}(S)$ then
(6) Score of removal and insertion operators is increased by $\lambda_1$; $f_{\text{best}}^\text{global}(S) \leftarrow f' (S')$; $f_{\text{best}}^\text{current}(S) \leftarrow f' (S')$; $S \leftarrow S'$;
(7) Else if $f' (S') \geq f_{\text{best}}^\text{current}(S)$ and the solution has not been accepted before then
(8) Score of removal and insertion operators is increased by $\lambda_2$; $f_{\text{best}}^\text{current}(S) \leftarrow f' (S')$; $S \leftarrow S'$;
(9) Else if $f' (S') \geq f_{\text{best}}^\text{current}(S)$ and the solution has been accepted before then
(10) Score of removal and insertion operators is increased by $\lambda_3$; $f_{\text{best}}^\text{current}(S) \leftarrow f' (S')$; $S \leftarrow S'$;
(11) Else if random value $\rho' \leq \rho$ then
(12) Score of removal and insertion operators is increased by $\lambda_4$; $f_{\text{best}}^\text{current}(S) \leftarrow f' (S')$; $S \leftarrow S'$;
(13) End if
(14) if mod($i$, UpI) $\neq 0$ then
(15) Update weight of each operator;
(16) End if
(17) $i \leftarrow i + 1$;
(18) Get the efficient feasible solution $S$;

Algorithm 2: ALNS$_p$ algorithm procedure.

| Parameters | Definition | Value |
|------------|------------|-------|
| $U$ | Total iterations | 5000 |
| UpI | Cycle | 500 |
| $\lambda_1$ | A new global optimal solution | 17 |
| $\lambda_2$ | A new and better than the current solution | 13 |
| $\lambda_3$ | Accepted before and better than the current solution | 5 |
| $\lambda_4$ | A worse solution | 4 |
| $T_0$ | Initial temperature | 100 |
| $\gamma$ | The weight coefficient | 0.3 |
| $\phi_1$ | The Shaw coefficients | 3 |
| $\phi_2$ | The Shaw coefficients | 2 |
| $\phi_3$ | The Shaw coefficients | 2 |
| $u$ | Delete parameter | 0.15–0.3 |

Table 2: Parameters used in ALNS$_p$. 

distance between pickup locations of passengers $i$ and $j$, $d_{ij}$ is the distance between delivery locations, $t_{p_i}$ is the boarding time, $t_{d_i}$ is the alighting time, $l_{ij}$ indicates whether passengers $i$ and $j$ are in the same vehicle ($l_{ij} = -1$) or not ($l_{ij} = 1$). The operator randomly chooses one customer and calculates all $R_{ij}$ between the
After a number of passenger delivery locations in the paper.

4.4. Insertion Algorithms (IA). After a number of passenger requests are removed from the vehicle routing solution, insertion operators are used to repair and generate new solution. The first three operators are adapted from Ropke and Pisinger [32], the fourth one is motivated by Cordeau and Laporte [5], the fifth is inspired by Demir et al. [33], while the last two are the insertion operators designed for the problem with multiple pickup and delivery locations using I1–I5 separately and selects the candidate location with the best performance as the current pickup or delivery location of the passenger and inserts it.

4.5. Time Slack Calculation Strategy. As seen in Figure 2, if we break and repair the solution using RA and IA operators without adjusting timetables, it is possible to fail to search more feasible insertion solutions. Therefore, we design the instance to simulate the issue, where the maximum ride time of each passenger is 40 and the service time is 0. Time window, pickup, and delivery locations of two passengers are presented in Figure 2.

If we just use Algorithm 3 without any ways to adjust timetables, it can be found that the ride time of passenger 1 is more than 40 and passenger 2 cannot be inserted into the route as shown in Figure 2. After using Algorithm 4 to change the timetables, passenger 2 can be inserted in the position and the insertion solution is feasible, as shown in Figure 3. Therefore, it is very essential to adjust timetables to further improve the solution during the search. The time slack calculation strategy we proposed can test the feasibility of any insertion cases in time constraints. The symbols that may be used in Section 4.5 are shown in Table 3.

As shown in Figure 2, the greedy algorithm just guarantees time window constraint but cannot consider the impact of the maximum ride time. In addition, it may also cause more waiting time. $T_{i}^{eaw}$ is the adjustable time bus can delay its arrival at the location $i$ according to the timetable and time window:

$$\tau_{i}^{eaw} = LT_{i} - AT_{i}. \quad (21)$$

$T_{i}^{ed}$ is the adjustable time bus can delay its arrival at the location $i$ within the maximum ride time constraint:

$$\tau_{i}^{ed} = M_{rd} - RD_{i}. \quad (22)$$

MTS means the adjustable time vehicle can delay its arrival at the location $i$, where its value may affect time constraints at other locations.

**Algorithm 3**: Greedy time calculate strategy.

(1) Input initial information $AT_i = Ed_{i}$, $DT_i = AT_i + ST$ and $WT_i = 0$;
(2) $u \leftarrow 2$;
(3) While $u \leq n$

\[
\begin{align*}
AT_u &= \max \{DT_{u-1} + t_{u-1, u}, Ed_u\}; \\
DT_u &= AT_u + ST; \\
WT_u &= \min \{0, Ed_u - AT_u\}; \\
u &\leftarrow u + 1;
\end{align*}
\]

(4) End

For the passenger with multiple locations, we need to combine one of I1–I5 and one of the last two operators to complete the insertion of the passenger. For this, one of the following two operators is randomly selected:

**MI.1**-Random insertion based on multiple locations: MI.1 principle randomly selects a candidate location as the current pickup or delivery location of the passenger and then inserts the passenger using I1–I5.

**MI.2**-One-by-one insertion based on multiple locations: MI.2 principle compares the performance of all candidate locations using I1–I5 separately and selects the candidate location with the best performance as the current pickup or delivery location of the passenger and inserts it.
Note:

(1) If both boarding and alighting locations of one passenger are behind the location \( u \), \( MTS_i \) on pickup location or delivery location of the passenger is \( T_{tw}^u \).

(2) If only delivery location \( i \) of one passenger is behind the location \( u \) and \( T_{rd}^i < 0 \), \( MTS_i \) of the location \( i \) is \( T_{tw}^i \).

\[ MTS_i = \min \{ T_{tw}^u, T_{rd}^i \}. \]  

\[ MTS_i = \min \{ T_{tw}^u, T_{rd}^i \}. \]  

\[ MTS_i = \min \{ T_{tw}^u, T_{rd}^i \}. \]

(2) If only delivery location \( i \) of one passenger is behind the location \( u \) and \( T_{rd}^i < 0 \), \( MTS_i \) of the location \( i \) is \( T_{tw}^i \).

\[ MTS_i \] means the adjustable time of the vehicle at location \( i \) based on waiting time, where its value may affect time constraints of other locations.

\[ MTS_i \] means the maximum adjustable time of the vehicle at the location \( i \), where its value cannot affect time constraints at other locations.
(1) Calculate and remark arrival time, departure time, and waiting time of all locations on one route by Algorithm 3;
(2) Begin with first location of the route and set \( u \leftarrow 1 \);
(3) While \( u < n \) and \( v < n \);
   (4) Calculate \( T_{uv}^d \), \( T_{vd}^d \) and \( MTS_v^u \) at the location \( v \) behind \( u \);
   (5) \( MTS_v^u = MTS_v^u + \sum_{u \in \{1, v \}} WT_w \)
   (6) \( u \leftarrow u + 1 \)
   (7) \( v \leftarrow v + 1 \)
(8) \( MTS_u^v = \min_{v \in \{u \}} [MTS_v^u] \) and obtain \( \overline{v} \); \( \overline{v} \) is the location where \( MTS_v^u \) is the minimum value;
(9) \( AT_u = AT_u + MTS_u^v \)
(10) Update arrival time, departure time, and waiting time of locations from the location \( u \) to \( n \) based on Algorithm 3;
(11) \( u \leftarrow \overline{v} \)
(12) End

Algorithm 4: Time slack calculation strategy to one route.

The relationship among \( MTS_v^u \), \( MTS_v^d \), and \( MTS_v \) and the entire process of time slack calculation strategy is shown as Algorithm 4.

4.6. Stopping Criteria. There are two types of stopping criteria used to avoid local optimization:

(i) When the current iteration exceeds the maximum iteration, the algorithm operation ends and the result is output

(ii) When the temperature \( T = T_0 \varepsilon \) is less than 0.5, the algorithm operation ends and the result is output

5. Numerical Experiments

We first verify the accuracy of the above proposed MIP model by solving an instance with small-scale requests in the simplified Sioux waterfall network proposed by Tong et al. [3]. The experiments in the paper were performed on the Intel(R) Core(TM) i5-7300HQ CPU @ 2.50 GHz 8G RAM computer. The MIP model was solved by the commercial software GUROBI.

5.1. Small Numerical Experiments on a Simplified Sioux Waterfall Network. To verify the proposed model, we test and compare it with the proposed algorithm by one example with 20 passenger requests on the simplified Sioux Falls network proposed by Tong et al. [4]. Some nodes meaning candidate stop locations (denoted by the blue and green dots), other nodes (denoted by the white dots), and transportation links are shown in Figure 4(a). The number on links means travel time.

Two kinds of buses can be originally located at depot nodes, namely, nodes 1, 2, and 13 (denoted by the yellow dots), which have different load capacity, departure costs, and fuel costs. The maximum and minimum load capacity, \( \text{Cap}_v^L \) and \( \text{Cap}_v^M \), of one with large capacity is 10 and 7, and the other with small capacity is 6 and 3. The departure cost \( c^d_j \) of is, respectively, 22 and 17, and the fuel cost per unit time \( c^f_j \) is 1 and 0.7. The ticket price is set to 10, and the penalty cost for each unserved passenger is half of the ticket price in this example. Note that, in this case, we assume that the end depot of the customized bus is virtual depots, of which distance to any point is zero.

Table 4 lists related information of all passenger requests. Take passenger 1 as an example. Passenger 1 can be picked up at node 3 between time 11 and 15 and be delivered at node 7 between time 26 and 30. In addition, there are some passengers (passengers 4, 15, and 19) with more than one candidate pickup location in requests information. For passenger 4, he/she can select only one node as the final candidate pickup location, namely, 4 or 5.

The commercial software GUROBI and ALNS\(_p\) are used to solve this instance. Finally, the route is generated as shown in Figure 4(b).

5.1.1. Optimal Solution for the Example. Computational results of the example are found using, respectively, the commercial software GUROBI and algorithm ALNS\(_p\), to assess their respective performance. The computational results show that the total income is 62.6. Four passengers cannot be served, namely, passengers 5, 6, 9, and 16. Other passengers are assigned to two buses. The computational result is the optimal solution (Gap value is 0), as shown in Table 5. Although the optimal solution from the solver and ALNS\(_p\) are the same, shorter CPU time indicates the MIP solver has better performance in terms of the small-scale problems.

Table 6 lists detailed vehicle routes and passenger assignments. Ten passengers are assigned to the bus with large capacity, and the rest six passengers to the bus with small one. Their routes are presented by red and orange lines in Figure 4(b).

5.1.2. Sensitive Analysis for Strategy 1. By the example on Sioux Waterfall Network, we further analyse the impact of Strategy 1 on the result. The computational results are shown in Table 7.

As can be seen from Table 7, the heterogeneous fleet achieves better profit in both experiments. Furthermore, the
Figure 4: The simplifying Sioux Waterfall Network. (a) The basic information of the road network and customers. (b) The final solution and commuting routes.

Table 4: Passenger trip requests on the simplifying Sioux Waterfall Network.

| Customers id | Candidate pickup location | Candidate delivery location | Pickup information | Delivery information |
|--------------|----------------------------|----------------------------|--------------------|---------------------|
| 1            | 3                          | 7                          | 3:[11, 15]         | 7:[26, 30]          |
| 2            | 3                          | 8                          | 3:[14, 18]         | 8:[26, 30]          |
| 3            | 4                          | 7                          | 4:[15, 19]         | 7:[26, 30]          |
| 4            | 4, 5                       | 8                          | 4:[20, 24]; 5:[20, 24] | 8:[26, 30]          |
| 5            | 5                          | 18                         | 5:[15, 19]         | 18:[26, 30]         |
| 6            | 11                         | 8                          | 11:[12, 16]        | 8:[26, 30]          |
| 7            | 12                         | 7                          | 12:[7, 11]         | 7:[26, 30]          |
| 8            | 12                         | 8                          | 12:[10, 14]        | 8:[26, 30]          |
| 9            | 12                         | 7                          | 12:[37, 41]        | 7:[56, 60]          |
| 10           | 20                         | 7                          | 20:[50, 54]        | 7:[56, 60]          |
| 11           | 20                         | 7                          | 20:[50, 54]        | 7:[56, 60]          |
| 12           | 20                         | 18                         | 20:[52, 56]        | 18:[56, 60]         |
| 13           | 20                         | 18                         | 20:[52, 56]        | 18:[56, 60]         |
| 14           | 21                         | 7                          | 21:[44, 48]        | 7:[56, 60]          |
| 15           | 21, 22                     | 7                          | 21:[45, 49]; 22:[45, 49] | 7:[56, 60]          |
| 16           | 22                         | 8                          | 22:[39, 43]        | 8:[56, 60]          |
| 17           | 23                         | 7                          | 23:[41, 45]        | 7:[56, 60]          |
| 18           | 23                         | 18                         | 23:[43, 47]        | 18:[56, 60]         |
| 19           | 23, 24                     | 7                          | 23:[41, 45]; 24:[41, 45] | 7:[56, 60]          |
| 20           | 24                         | 18                         | 24:[43, 47]        | 18:[56, 60]         |
result shows that the heterogeneous fleet serves more passengers than homogeneous fleets in the second experiment. It means more passengers can be served by a flexible solution from heterogeneous fleets. Homogeneous fleets cannot make flexible dispatch based on various passenger requests, which may increase cost or make more passengers unserviceable.

5.2. Benchmark Test Results

5.2.1. Benchmark Test with Unique Pickup or Delivery Locations. Because there is no benchmark instance applied to customized bus problem, we choose DARP benchmark instances proposed by Cordeau and Laporte [5] as test instances to evaluate the performance of the ALNS proposed, which has similar properties as customized bus problems. Assuming all requests are serviced, the total distance is minimized as the objective function.

Li et al. [35] tested, respectively, two solution evaluation approaches for ALNS (ALNS$_i$ is to allow only feasible solutions, ALNS$_f$ is to allow infeasible solutions) and found that although the results of ALNS$_i$ are better than ALNS$_f$, ALNS$_i$ takes more time than ALNS$_f$. Table 8 shows the calculation results of ALNS$_i$, ALNS$_f$, and ALNS$_{ip}$ on DARP instances. The first column in Table 8 represents the instance number, and the Gap (%) refers to the percentage deviation of the ALNS$_i$, ALNS$_f$, and ALNS$_{ip}$ compared to the best results from [5, 45]. We test the ALNS$_{ip}$ on DARP benchmark instances and find it provides good results within 5000 iterations. The best result of ALNS$_{ip}$ is selected based on 10 runs. All results are shown in Table 8 and compared with the best results of ALNS$_i$, ALNS$_f$ from Li et al. [35].

Table 8 shows that compared to ALNS$_f$, the proposed ALNS$_{ip}$ can obtain better solutions in most instances where the results have on an average 84.16% decreased gap values with only 23.03% longer CPU time on average. In these instances, the ALNS$_{ip}$ generally gives better results but consumes longer time than ALNS$_f$. But the exception is that for instances R2a, R10a, R8b, R9b, and R10b, ALNS$_{ip}$ searches better solutions but faster than ALNS$_f$. Furthermore, ALNS$_{ip}$ generally gives similar (for instances, R1a, R2a, R4a, R7a, and R7b) or even worse results but consumes less time than ALNS$_i$ in these instances. Table 8 shows that the proposed ALNS$_{ip}$ searches solutions 60.71% higher gap values but 42.75% shorter CPU time on average than ALNS$_f$. While the exception is that for instances R2a, R4a and R9a, ALNS$_{ip}$ obtains better solutions than ALNS$_i$ within a shorter time. As a result, it turned out that ALNS$_{ip}$ could get efficient and high-quality results in a reasonable time. In addition, ALNS$_{ip}$ ensures the feasible solutions at each iteration. It is of great significance to solve the large-scale routing problems with multiple locations and time windows.

5.2.2. Benchmark Test with Multiple Pickup or Delivery Locations. In the section, we improve DARP benchmark instances to further analyse the impact of Strategy 2 and test the performance of the ALNS$_{ip}$ in terms of multiple pickup or delivery locations problem. Some passengers are randomly appended pickup or delivery candidate locations, where new pickup or delivery locations are some locations closest to the passenger’s original pickup or delivery location. Considering different scales, we choose R1a (small-scale problem), R2a (medium-scale problem), and R5a (large-scale problems) as tested instances to illustrate the following analysis.

(i) Analysis for Strategy 2: Table 9 compares the impacts of Strategy 2 or not. It can be seen that, compared with the strategy that does not consider multiple candidate locations, Strategy 2 provides more flexible and higher solution quality for the final solution. The results also show that compared with classic ALNS, ALNS$_{ip}$ can effectively solve the problem with multiple candidate locations in a reasonable time.

(ii) Analysis for the number of candidate locations: to perform a sensitive analysis of the number of candidate locations, we test it from two perspectives, that is, the number of candidate locations of each passenger (2, 3, or 4 candidate locations) and the total number of people chosen to append candidate locations (5%, 10%, and 20% of passengers randomly). Overall, Strategy 2 with more than one location gives more applicable solutions than cases without multiple locations in terms of two perspectives. Table 10 shows that with the increase of candidate locations, it can obtain the better solution in small-scale problems. But for larger-scale instances, the results in Table 10 may be worse if one passenger has more candidate locations. Possibly, the reason is that multiple candidate locations cause more feasible solutions during the search, which expands the search area and need more iterations and time to search optimized solutions. Therefore, given that passengers generally do not have too many candidate locations near their destination in practice, 2 or 3 candidate locations are more appropriate in practical issues.

5.2.3. Comparison of Heuristic Algorithms. To evaluate the effectiveness of the ALNS$_{ip}$ in searching the problem with multiple candidate locations, we compare and analyse its performance with other heuristic algorithms, namely, Genetic Algorithm (GA) [46], Particle Swarm Optimization (PSO) [47], and Artificial Bee Colony Algorithm (ABC) [48]. We randomly choose 10% customers with 2 candidate locations based on the benchmark instances R1b–R3b. These heuristic algorithms continuously find feasible solutions and
| No. | Number | Type | Solution | Route |
|-----|--------|------|----------|-------|
| 1:  | 10     | Large| VO  ⟷ 19 ⟷ 18 ⟷ 20 ⟷ 4 ⟷ 15 ⟷ 3 ⟷ 10 ⟷ 4 ⟷ 12 ⟷ 21 ⟷ 12 ⟷ 4 ⊨ 14 ⟷ 10 ⟷ 18 ⟷ 14 ⟷ 10 ⟷ 19 ⟷ 11 ⟷ 17 | VO  ⟷ 19 ⟷ 18 ⟷ 20 ⟷ 4 ⟷ 15 ⟷ 3 ⟷ 10 ⟷ 4 ⟷ 12 ⟷ 21 ⟷ 12 ⟷ 4 ⊨ 14 ⟷ 10 ⟷ 18 ⟷ 14 ⟷ 10 ⟷ 19 ⟷ 11 ⟷ 17 |
| 2:  | 6      | Small| VO  ⟷ 7 ⟷ 8 ⟷ 2 ⟷ 4 ⟷ 14 ⟷ 2 ⊨ 7 ⟷ 3 ⟷ 1 | VO  ⟷ 7 ⟷ 8 ⟷ 2 ⟷ 4 ⟷ 14 ⟷ 2 ⊨ 7 ⟷ 3 ⟷ 1 |

Note that the number in Solution is customer's id.
### Table 7: Impact of various parameters.

| Experiments | Vehicle type | Capacity | Departure cost | Fuel costs | Cars | Profit | Unserved |
|-------------|--------------|----------|----------------|------------|------|--------|----------|
| No. 1       | Heterogeneous vehicle | (7, 10); 22; 17 | 1.00 | 0.7 | 2 | 62.6 | 4 |
|             | Homogeneous vehicle  | (6, 10) | 22 | 1.0 | 2 | 51.0 | 4 |
| No. 2       | Heterogeneous vehicle | (7, 10); 22; 17 | 0.40; | 3 | 6 | 86.75 | 3 |
|             | Homogeneous vehicle  | (6, 10) | 22 | 0.40 | 2 | 78.0 | 4 |

### Table 8: Compared results of ALNS<sub>i</sub> on the benchmark DARP instances.

| Instance | Requests | Best profit | Gap (%) | CPU (min) | ALNS<sub>i</sub> | ALNS<sub>f</sub> | ALNS<sub>i</sub> | ALNS<sub>f</sub> | Gap (%) | CPU (min) |
|----------|----------|-------------|---------|-----------|------------------|------------------|------------------|------------------|---------|-----------|
| R1a      | 24       | 190.02      | -1.15   | 1.35      | 1.63             | 0.49             | 0.00             | 1.67             |
| R2a      | 48       | 301.336     | 0.55    | 9.59      | 9.12             | 12.11            | 0.32             | 8.01             |
| R3a      | 72       | 531.996     | 2.01    | 18.14     | 17.67            | 12.11            | 0.74             | 20.40            |
| R4a      | 96       | 570.246     | 4.95    | 25.21     | 56.14            | 18.07            | 4.73             | 39.97            |
| R5a      | 120      | 628.114     | 5.38    | 26.79     | 65.46            | 37.37            | 7.30             | 49.11            |
| R6a      | 144      | 794.061     | 0.92    | 29.53     | 153.57           | 49.84            | 5.27             | 68.94            |
| R7a      | 36       | 291.71      | 0.00    | 6.05      | 4.01             | 1.12             | 0.00             | 2.64             |
| R8a      | 72       | 487.843     | 1.45    | 21.27     | 44.32            | 20.93            | 3.47             | 22.40            |
| R9a      | 108      | 658.312     | 3.66    | 57.32     | 148.41           | 39.78            | 2.34             | 42.79            |
| R10a     | 144      | 857.109     | 2.53    | 29.62     | 242.52           | 128.30           | 7.01             | 72.30            |
| R1b      | 24       | 164.46      | -0.05   | 2.70      | 3.59             | 0.41             | 3.60             | 3.74             |
| R2b      | 48       | 295.664     | 0.18    | 15.05     | 12.35            | 3.76             | 1.06             | 15.41            |
| R3b      | 72       | 486.569     | 2.19    | 24.21     | 29.94            | 28.88            | 4.56             | 32.04            |
| R4b      | 96       | 530.697     | 4.10    | 29.66     | 48.76            | 23.85            | 6.41             | 62.58            |
| R5b      | 120      | 578.607     | 3.11    | 33.12     | 124.12           | 33.55            | 6.11             | 94.23            |
| R6b      | 144      | 740.354     | 3.49    | 41.94     | 145.33           | 48.90            | 6.17             | 109.70           |
| R7b      | 36       | 248.21      | 0.01    | 8.63      | 3.65             | 4.43             | 0.00             | 5.28             |
| R8b      | 72       | 461.389     | 3.68    | 10.95     | 31.49            | 51.79            | 2.11             | 41.42            |
| R9b      | 108      | 597.746     | 3.28    | 25.40     | 134.27           | 75.87            | 4.51             | 72.57            |
| R10b     | 144      | 795.157     | 4.48    | 38.08     | 242.64           | 115.17           | 6.28             | 104.47           |
| Average  | —        | —           | 2.24    | 22.73     | 75.95            | 35.34            | 3.60             | 43.48            |

### Table 9: Results of Strategy 2 or not.

| Instance | Requests | Without strategy 2 | With strategy 2 | CPU (min) |
|----------|----------|---------------------|-----------------|-----------|
| R1a      | 24       | 190.02              | 183.05          | 1.71      |
| R2a      | 48       | 302.31              | 298.75          | 8.14      |
| R5a      | 120      | 673.95              | 664.80          | 50.17     |

### Table 10: Influence of the number of candidate locations.

| (%)  | Instance | Best profit without strategy | 2 candidate locations | 3 candidate locations | 4 candidate locations |
|------|----------|------------------------------|-----------------------|-----------------------|-----------------------|
| 5    | R1a      | 190.02                       | 183.96                | 183.96                | 183.95                |
|      | R2a      | 302.31                       | 300.94                | 297.80                | 296.79                |
|      | R5a      | 673.95                       | 668.26                | 667.06                | 657.15                |
| 10   | R1a      | 190.02                       | 183.96                | 183.96                | 183.95                |
|      | R2a      | 302.31                       | 298.75                | 297.28                | 292.10                |
|      | R5a      | 673.95                       | 664.80                | 666.23                | 667.63                |
| 20   | R1a      | 190.02                       | 178.70                | 177.86                | 174.40                |
|      | R2a      | 302.31                       | 296.38                | 285.02                | 286.45                |
|      | R5a      | 673.95                       | 667.28                | 644.85                | 651.82                |
use the data preprocessing techniques [49] to avoid infeasible results. For multiple candidate locations, one location from multiple candidate locations will be chosen during the iteration. The parameter settings are set as follows: the number of populations $P_{\text{size}} = 20$ and the number of generations $G_{\text{max}} = 5000$, and crossover ratio of GA $P_{c}^{\text{GA}} = 0.8$, crossover ratio of PSO $P_{c}^{\text{PSO}} = 0.6$, and mutation ratio $P_{m} = 0.2$. The comparisons of four heuristic algorithms’ performances, including mean value and the best value, are shown in Table 11.

Table 11: Comparison of ALNS$_{ip}$ with the GA-SA, PSO, and ABC on the benchmark instances.

| No | GA Mean | GA Best | PSO Mean | PSO Best | ABC Mean | ABC Best | ALNS$_{ip}$ Mean | ALNS$_{ip}$ Best |
|----|---------|---------|----------|----------|----------|----------|-----------------|-----------------|
| R1b | 185.9   | 181.7   | 183.3    | 174.9    | 190.7    | 184.6    | 163.8           | 160.4           |
| R2b | 370.9   | 350.7   | 343.9    | 340.1    | 341.9    | 337.3    | 304.7           | 300.9           |
| R3b | 664.6   | 638.8   | 611.6    | 580.8    | 627.1    | 600.4    | 505.3           | 502.2           |

Figure 5: Convergence graph obtained for ALNS$_{ip}$, GA, PSO, and ABC algorithms. (a) R1b. (b) R2b. (c) R3b.

Table 11 shows the best value and mean value computed by each algorithm for each instance. The results of the four algorithms show a significant difference. We can observe in comparison with other heuristic algorithms the results obtained ALNS$_{ip}$ are optimal in terms of the best value and mean value. And with an increase in the number of customers, the ability to find the best optimization solution is higher. In addition, the convergence plot (Figure 5) shows us the convergence of the different algorithms. On observing the plot of them, we can see that ALNS$_{ip}$ performs better than other algorithms in convergence and has a strong ability to jump out of the local optimum to avoid premature convergence. As a result, ALNS$_{ip}$ can obtain better optimization solutions and effectively search the solution space in terms of solution quality and stability.
Figure 6: The distribution of candidate locations. Red and green, respectively, represent the pickup and delivery locations of passengers. (a) Destination: Jinrongjie and (b) Destination: Guomao.

Table 12: Passenger demands information of Group 1.

| Passenger id | Pick up location with candidate location | Delivery location | $c^0(p)$ | $l^0(p)$ | $c^1(p)$ | $l^1(p)$ |
|--------------|------------------------------------------|------------------|---------|---------|---------|---------|
| 1            | 1                                        | 9                | 0       | 10      | 47      | 60      |
| 2            | 1                                        | 3                | 0       | 10      | 47      | 60      |
| 3            | 2                                        | 0                | 10      | 0       | 45      | 60      |
| 4            | 2                                        | 0                | 10      | 0       | 45      | 55      |
| 5            | 2                                        | 4                | 0       | 12      | 47      | 60      |
| 6            | 3                                        | 0                | 10      | 5       | 47      | 60      |
| 7            | 3                                        | 0                | 9       | 5       | 45      | 55      |
| 8            | 3                                        | 0                | 10      | 4       | 16      | 50      | 60      |
| 9            | 4                                        | 0                | 9       | 10      | 20      | 45      | 55      |
| 10           | 4                                        | 0                | 10      | 7       | 20      | 47      | 57      |
| 11           | 4                                        | 0                | 10      | 8       | 20      | 50      | 60      |
| 12           | 5                                        | 0                | 10      | 10      | 20      | 50      | 60      |
| 13           | 5                                        | 0                | 10      | 8       | 20      | 47      | 58      |
| 14           | 6                                        | 0                | 10      | 10      | 20      | 50      | 58      |
| 15           | 6                                        | 0                | 9       | 8       | 20      | 47      | 55      |
| 16           | 6                                        | 0                | 9       | 10      | 20      | 45      | 55      |
| 17           | 7                                        | 0                | 10      | 20      | 32      | 47      | 60      |
| 18           | 7                                        | 0                | 10      | 20      | 30      | 45      | 60      |
| 19           | 8                                        | 7                | 9       | 22      | 33      | 47      | 60      |
| 20           | 8                                        | 0                | 10      | 25      | 33      | 50      | 60      |

Table 13: The routing time between locations of Group 1.

| min | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|
| 1   | 0 | 8 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  |    |
| 2   | 8 | 0 | 6 | 3 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| 3   | 4 | 6 | 0 | 7 | 4 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| 4   | 3 | 7 | 0 | 5 | 7 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| 5   | 4 | 5 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| 6   | 7 | 7 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| 7   | 6 | 0 | 5 | 0 | 25| 30| 4 | 4 | 0 | 0  | 0  | 0  |    |
| 8   | 5 | 0 | 25| 0 | 4 | 0 | 0 | 0 | 0 | 0  | 0  | 0  |    |
| 9   | 0 | 0 | 29| 5 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  |    |
| 10  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
5.3. Customized Buses Instances. In order to test the performance of ALNS\(_p\) in solving practical problems with large scale, we are inspired by Tong et al. [4] and consider the real commuting issue in Beijing. Finally, we design three customized bus service instances containing small-scale problems and large-scale problems, where commuters leave for their workplaces from their houses and two main business areas, namely, Jingrongjie (destination 1) and Guomao (destination 2), as shown in Figure 6.

| Table 14: Passenger demands information of Group 2. |
|---------------------------------------------------|
| Passenger id | Pick up location with candidate location | Delivery location | \(e^0(p)\) | \(l^0(p)\) | \(e^1(p)\) | \(l^1(p)\) |
| 1 | 1 | 0 | 11 | 0 | 5 | 70 | 75 |
| 2 | 1 | 8 | 9 | 0 | 5 | 45 | 55 |
| 3 | 1 | 0 | 11 | 0 | 5 | 55 | 60 |
| 4 | 1 | 0 | 10 | 0 | 5 | 65 | 70 |
| 5 | 2 | 0 | 9 | 5 | 10 | 45 | 55 |
| 6 | 2 | 0 | 11 | 5 | 10 | 55 | 60 |
| 7 | 2 | 0 | 10 | 5 | 10 | 55 | 60 |
| 8 | 2 | 3 | 11 | 5 | 10 | 55 | 60 |
| 9 | 3 | 0 | 9 | 7 | 15 | 60 | 65 |
| 10 | 3 | 0 | 11 | 7 | 15 | 70 | 75 |
| 11 | 3 | 8 | 10 | 7 | 15 | 65 | 70 |
| 12 | 3 | 0 | 11 | 7 | 15 | 70 | 75 |
| 13 | 4 | 0 | 11 | 10 | 15 | 70 | 75 |
| 14 | 4 | 3 | 9 | 12 | 15 | 60 | 65 |
| 15 | 4 | 0 | 11 | 10 | 15 | 70 | 75 |
| 16 | 4 | 3 | 11 | 10 | 15 | 70 | 75 |
| 17 | 5 | 0 | 9 | 15 | 20 | 60 | 65 |
| 18 | 5 | 0 | 11 | 15 | 20 | 70 | 75 |
| 19 | 6 | 0 | 10 | 10 | 15 | 55 | 60 |
| 20 | 6 | 5 | 11 | 10 | 15 | 55 | 60 |
| 21 | 6 | 0 | 11 | 10 | 15 | 55 | 60 |
| 22 | 6 | 0 | 9 | 10 | 15 | 45 | 55 |
| 23 | 6 | 5 | 9 | 10 | 15 | 45 | 55 |
| 24 | 6 | 0 | 11 | 10 | 15 | 55 | 60 |
| 25 | 7 | 0 | 10 | 12 | 20 | 55 | 60 |
| 26 | 7 | 0 | 11 | 10 | 20 | 55 | 60 |
| 27 | 7 | 0 | 10 | 12 | 15 | 55 | 60 |
| 28 | 7 | 0 | 11 | 12 | 20 | 55 | 60 |
| 29 | 8 | 0 | 10 | 2 | 10 | 65 | 70 |
| 30 | 8 | 1 | 11 | 2 | 10 | 70 | 75 |
| 31 | 8 | 0 | 9 | 2 | 10 | 60 | 65 |
| 32 | 8 | 0 | 11 | 2 | 10 | 70 | 75 |

| Table 15: The routing time between locations of Group 2. |
|-------------------------------------------------------|
| min | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 0 | 5 | 4 | 3 | 5 | 0 | 4 | 1 | 52 | 44 | 44 | 0 |
| 2 | 5 | 0 | 4 | 3 | 5 | 0 | 4 | 3 | 52 | 44 | 44 | 0 |
| 3 | 4 | 0 | 5 | 4 | 3 | 0 | 4 | 3 | 52 | 44 | 44 | 0 |
| 4 | 3 | 5 | 0 | 4 | 4 | 0 | 4 | 4 | 52 | 44 | 44 | 0 |
| 5 | 4 | 0 | 4 | 0 | 4 | 4 | 0 | 4 | 52 | 44 | 44 | 0 |
| 6 | 8 | 4 | 5 | 0 | 2 | 2 | 0 | 2 | 37 | 52 | 52 | 0 |
| 7 | 4 | 2 | 0 | 0 | 37 | 52 | 0 | 0 | 0 | 0 | 52 | 52 |
| 8 | 4 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 52 | 44 | 37 | 0 | 7 | 0 | 3 | 0 | 55 | 52 | 3 | 0 |
| 10 | 52 | 44 | 37 | 0 | 7 | 0 | 3 | 0 | 55 | 52 | 3 | 0 |
| 11 | 55 | 52 | 3 | 0 | 55 | 52 | 3 | 0 | 55 | 52 | 3 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
with large capacity is 30, 25 for the small one. The fuel cost per unit time is 0.4 for those buses with large capacity and 0.25 for the small. The maximum route duration of buses is determined according to the shortest path of the home-to-work. If it is more than 60 minutes, the ticket will be set to 14, and the penalty cost of unserved passengers is 7. If it is less than or equal to 60 minutes, the ticket is set to 10, and the penalty cost is 5. With respect to their proximity in time and space, the requests are divided into 3 groups, and the results of the actual case are finally obtained as follows.

Tables 12 and 13 show the passengers’ information and routing time between some locations (measured by map software) of Group 1. Similarly, Tables 14 and 15 are for Group 2, and Tables 16 and 17 are for Group 3.

Given the data information in Tables 12-15, the results using MIP solver and ALNS are shown in Table 18.

### Table 16: Passenger demands information of Group 3.

| Passenger id | Pick up location | Delivery location | \(c^0(p)\) | \(l^0(p)\) | \(c^1(p)\) | \(l^1(p)\) |
|--------------|-----------------|------------------|----------|----------|----------|----------|
| 1            | 1               | 14               | 0        | 10       | 45       | 55       |
| 2            | 1 5             | 16               | 0        | 10       | 45       | 50       |
| 3            | 1 0             | 14               | 5        | 15       | 45       | 55       |
| 4            | 1 8             | 15               | 5        | 15       | 45       | 55       |
| 5            | 2 0             | 14               | 0        | 10       | 45       | 55       |
| 6            | 2 0             | 15               | 5        | 15       | 45       | 55       |
| 7            | 2 0             | 14               | 5        | 15       | 45       | 55       |
| 8            | 2 0             | 15               | 5        | 15       | 40       | 50       |
| 9            | 2 0             | 16               | 0        | 10       | 40       | 50       |
| 10           | 3 0             | 14               | 0        | 10       | 40       | 50       |
| 11           | 3 0             | 14               | 5        | 15       | 40       | 55       |
| 12           | 3 0             | 16               | 5        | 15       | 40       | 50       |
| 13           | 3 0             | 16               | 5        | 15       | 40       | 50       |
| 14           | 4 0             | 14               | 0        | 10       | 45       | 55       |
| 15           | 4 6             | 14               | 0        | 15       | 45       | 55       |
| 16           | 4 0             | 14               | 7        | 20       | 45       | 55       |
| 17           | 5 0             | 14               | 5        | 20       | 45       | 55       |
| 18           | 5 6             | 14               | 5        | 17       | 45       | 55       |
| 19           | 5 0             | 16               | 5        | 20       | 40       | 50       |
| 20           | 5 0             | 16               | 7        | 20       | 45       | 50       |
| 21           | 6 0             | 15               | 5        | 15       | 45       | 55       |
| 22           | 6 0             | 14               | 7        | 17       | 45       | 55       |
| 23           | 6 0             | 16               | 5        | 15       | 40       | 50       |
| 24           | 6 0             | 16               | 2        | 15       | 40       | 50       |
| 25           | 7 0             | 16               | 5        | 15       | 40       | 50       |
| 26           | 7 5             | 14               | 5        | 15       | 45       | 55       |
| 27           | 7 0             | 15               | 5        | 15       | 45       | 55       |
| 28           | 7 12            | 14               | 2        | 17       | 45       | 55       |
| 29           | 8 0             | 16               | 0        | 10       | 40       | 50       |
| 30           | 8 0             | 14               | 5        | 15       | 45       | 55       |
| 31           | 8 0             | 14               | 5        | 20       | 45       | 55       |
| 32           | 9 0             | 15               | 10       | 20       | 45       | 55       |
| 33           | 9 12            | 15               | 5        | 20       | 45       | 55       |
| 34           | 9 0             | 14               | 5        | 20       | 45       | 55       |
| 35           | 9 0             | 16               | 7        | 20       | 40       | 50       |
| 36           | 10 0            | 14                | 10       | 20       | 45       | 55       |
| 37           | 10 0            | 14                | 7        | 20       | 45       | 55       |
| 38           | 1 0             | 14                | 7        | 20       | 45       | 55       |
| 39           | 1 5             | 14                | 10       | 20       | 40       | 50       |
| 40           | 1 0             | 16                | 12       | 20       | 40       | 50       |
| 41           | 1 8             | 14                | 12       | 25       | 45       | 55       |
| 42           | 2 0             | 16                | 12       | 25       | 40       | 50       |
| 43           | 2 0             | 14                | 5        | 20       | 45       | 55       |
| 44           | 2 0             | 16                | 10       | 20       | 40       | 50       |
| 45           | 2 0             | 14                | 10       | 20       | 45       | 55       |
| 46           | 2 0             | 16                | 10       | 20       | 40       | 50       |
| 47           | 3 0             | 15                | 15       | 25       | 45       | 55       |
| 48           | 3 0             | 14                | 10       | 20       | 45       | 55       |
| 49           | 3 0             | 14                | 12       | 25       | 45       | 55       |
We set 1000s as the termination condition of the MIP solver. In the first two cases, GROUBI returns feasible solutions with 2.33% and 0% of gap value in the allowed time. In the last large-scale case, there are no feasible solutions in the allowed time for MIP solver. But for ALNS, as shown in Table 18, it can efficiently solve routing problems with different scales. Multiple experiments in Sections 5.2 and 5.3 also demonstrate that it is effective for the proposed ALNS to solve practice problems with large scale. Additionally, to further verify the effectiveness and advancement of heterogeneous customized bus service with multiple candidate locations, we compare the performance among the two strategies, the homogeneous fleet, as well as the result without candidate locations in Table 19. It can be compared with other cases that the heterogeneous fleets and candidate locations strengthen the cooperation among vehicles and flexibility of routes planning. Therefore, both strategies provide the highest profits (relative increase of 12.3%, 6.2%, and 13.7% in total profits, respectively), no unserved requests, and fewer vehicles to reduce lane occupation. It is of great practical significance for increasing operators’ profit and improving service quality.

5.4. Implications. The customized bus service based on heterogeneous fleets and multiple candidate locations provides a good organizational form and service framework. The proposed mathematical model and algorithm also verify its effectiveness and offer a theoretical

### Table 17: The routing time between locations of Group 3.

| min | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|
| 1   | 0 | 5 | 3 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2   | 5 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 3   | 0 | 4 | 2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 4   | 4 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 5   | 3 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 6   | 2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 7   | 3 | 2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 8   | 4 | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 9   | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 10  | 4 | 4 | 3 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 11  | 7 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 12  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 13  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 14  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 15  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 16  | 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 17  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 18  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 19  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

### Table 18: Results for 3 Group instances.

| Group No. | Number of requests | \(ALNS_{IP}\) | MIP solver | Gap (%) | Number of served requests | Number of unserved requests | Big cars | Small cars |
|-----------|--------------------|---------------|------------|---------|---------------------------|-----------------------------|-----------|------------|
| 1         | 20                 | 111.25        | 111.25     | 2.33    | 20                        | 0                           | 1         | 1          |
| 2         | 32                 | 336.4         | 336.4      | 0.00    | 32                        | 0                           | 2         | 0          |
| 3         | 49                 | 291.75        | —          | —       | 49                        | 0                           | 1         | 4          |
| Total     | 101                | 739.4         | —          | —       | 101                       | 0                           | 4         | 5          |

### Table 19: Results comparison in different cases.

| Group No. | Without candidate locations | Homogeneous fleet with big capacity | Homogeneous fleet with small capacity | Heterogeneous fleets and candidate locations |
|-----------|-----------------------------|-------------------------------------|--------------------------------------|-------------------------------------------|
|           | Profits | NU | NC | Profits | NU | NC      | Profits | NU | NC | Profits | NU | NC      |
| 1         | 111.25  | 0  | 2  | 98.0    | 0  | 2       | 94.75   | 2  | 2  | 111.25  | 0  | 2       |
| 2         | 269.9   | 2  | 3  | 336.4   | 0  | 2       | 287.75  | 0  | 4  | 336.4   | 0  | 2       |
| 3         | 277.3   | 1  | 5  | 261.4   | 5  | 3       | 268.0   | 0  | 6  | 291.75  | 0  | 5       |
| Total     | 658.45  | 3  | 10 | 695.8   | 5  | 7       | 650.5   | 2  | 12 | 739.4   | 0  | 9       |

NU: number of unserved requests; NC: number of customized buses.
basis for further research and implications in practice. The bus operators can improve fleets management and route planning by introducing the new mechanism and strategy. The detailed managerial implications of the proposed methodology can be summarized as follows:

1. The design of heterogeneous fleets can make customized bus companies construct the fleets more flexibly, save costs, and increase the occupancy rate. Heterogeneous fleets have different load capacities and different transportation costs. The selection of customized buses with different capacities improves scheduling and collaboration among buses and enhances the operational efficiency of bus system. Therefore, effective collaboration among heterogeneous vehicles can contribute to avoiding empty seats and increasing operator profitability.

2. More candidate locations chosen by passengers in the platform make a breakthrough in the limitations of a single candidate location. From the passengers’ perspective, it increases the probability for the passengers to be served. From the operators’ perspective, it means more flexible route planning. Consequently, multiple candidate locations can be utilized to optimize pickup and delivery routes and serve as many passengers as possible to improve service quality.

6. Conclusion

For passengers living in congested metropolitan areas, the customized bus system offers flexible and high-accessibility bus service in the existing transportation network services. Compared with the general customized bus problem, this study constructs a new service form with heterogeneous fleets and multiple candidate locations to deal with practical issues. The two strategies are then analysed and evaluated their effects. To optimize the operator’s profits based on the new service form proposed, a mixed-integer programming model is developed to describe the problem. It is not only suitable for solving small-scale problems but also simple to operate by GUROBI, a commercial solver. An improved ALNSp is then designed to address a large-scale challenge. The initial ALNS algorithm has good performance in terms of routing problem. The time slack calculation method is employed to intelligently adjust vehicle timetables to optimize the wait and duration time. Two inserting operators are utilized to enhance the ability to search solution space, considering some passengers with multiple candidate locations.

Numerical experiments are constructed to evaluate the effectiveness of the proposed model and algorithm. The sensitive analysis on small-scale instances indicates that the mechanism with heterogeneous fleets and multiple candidate locations can serve more passengers and obtain more profits. A comparison among ALNSp and ALNSpf indicates ALNSp can be efficient in searching for better solutions within a reasonable time. In addition, ALNSp also outperform other heuristic algorithms, like GA, PSO, and ABC, in search ability and avoiding premature. Finally, this study tests the customized bus service in real life to verify its theoretical research and practical implication.

This study aims to propose a new form of customized bus service based on heterogeneous fleets and multiple candidate locations. And the designed model and algorithm can offer robust solutions in different scales. In future research, our work can be conducted in the following three directions: (1) researching other models for passengers with random characteristics, who usually offer information in real time and hope to obtain dynamic service, (2) considering the cooperation mechanism between customized bus service and the other transportation systems, and (3) improving the solution algorithm through introducing distributed parallel computing to shorten computing time.

Data Availability

The data of Sections 5.1 and 5.3 used to support the findings of this study are included within the article. The data of Section 5.2 are from [5].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

All the authors gratefully acknowledge the supporting of Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport. This paper was supported by National Natural Science Foundation of China (No. 62076023).

References

[1] R. F. Kirby and K. U. Bhatt, "Guidelines on the Operation of Subscription Bus Services," Report No. UI-5021-5024, Urban Institute and Urban Mass Transportation Administration, Washington, DC, USA, 1974.
[2] T. Liu and A. Ceder, "Analysis of a new public-transport-service concept: customized bus in China," Transport Policy, vol. 39, pp. 63–76, 2015.
[3] L. Yan, C. Y. Chow, V. Lee, Y. Li, and J. Zeng, "T2CBS: mining taxi trajectories for customized bus systems," in Proceedings of the IEEE INFOCOM 2016 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) IEEE, pp. 441–446, San Francisco, CA, USA, April 2016.
[4] L. Tong, L. Zhou, J. Liu, and X. Zhou, "Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing," Transportation Research Part C: Emerging Technologies, vol. 85, pp. 451–475, 2017.
[5] J.-F. Cordeau and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," Transportation Research Part B: Methodological, vol. 37, no. 6, pp. 579–594, 2003.
[6] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," European Journal of Operational Research, vol. 162, no. 1, pp. 126–141, 2005.
[7] Y. Dumas, J. Desrosiers, and F. Soumis, “The pickup and delivery problem with time windows,” *European Journal of Operational Research*, vol. 54, no. 1, pp. 7–22, 1991.

[8] Q. Lu and M. Dessouky, “An exact algorithm for the multiple vehicle pickup and delivery problem,” *Transportation Science*, vol. 38, no. 4, pp. 503–514, 2004.

[9] H. N. Psarafitis, “A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem,” *Transportation Science*, vol. 14, no. 2, pp. 130–154, 1980.

[10] S. Ropke, J.-F. Cordeau, and G. Laporte, “Models and branch-and-cut algorithms for pickup and delivery problems with time windows,” *Networks*, vol. 49, no. 4, pp. 258–272, 2007.

[11] Y. Wang, S. G. Peng, X. S. Zhou, M. Mahmoudi, and L. Zhen, “Green logistics location-routing problem with eco-packages,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 143, 2020.

[12] Y. Wang, Y. Y. Yuan, X. Y. Guan et al., “Collaborative two-echelon multicenter vehicle routing optimization based on state–space–time network representation,” *Journal of Cleaner Production*, vol. 258, 2020.

[13] Y. Wang, S. G. Peng, and M. Xu, “Emergency logistics network design based on space–time–resource configuration,” *Knowledge-Based Systems*, vol. 223, 2021.

[14] Y. Wang, S. L. Zhang, X. Y. Guan, J. X. Fan, H. Z. Wang, and Y. Liu, “Cooperation and profit allocation for two-echelon logistics pickup and delivery problems with state–space–time networks,” *Applied Soft Computing*, vol. 109, 2021.

[15] D. M. Stein, “Scheduling dial-a-ride transportation systems,” *Transportation Science*, vol. 12, no. 3, pp. 232–249, 1978.

[16] R. M. Jørgensen, J. Larsen, and K. B. Bergvinsdottir, “Solving the dial-a-ride problem using genetic algorithms,” *Journal of the Operational Research Society*, vol. 58, no. 10, pp. 1321–1331, 2007.

[17] T. Garaix, C. Artigues, D. Feillet, and D. Josselin, “Vehicle routing problems with alternative paths: an application to on-demand transportation,” *European Journal of Operational Research*, vol. 204, no. 1, pp. 62–75, 2010.

[18] D. S. V. J. H. H¨all, J. T. Lundgren, and V. Stefan, “Evaluating the performance of a dial-a-ride service using simulation,” *Public Transport*, vol. 7, no. 2, pp. 139–157, 2015.

[19] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.
A. Gunawan, A. T. Widjaja, P. Vansteenwegen, and V. F. Yu, “Adaptive large neighborhood search for vehicle routing problem with cross-docking,” in Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC) IEEE, Glasgow, UK, July 2020.

M. P. Lippenberger, “An adaptive large neighborhood search for routing and scheduling carsharing service requests,” Operations Research Proceedings, Springer, Cham, New York, NY, USA, pp. 11–15, 2019.

X. Dong, D. Li, Y. Yin, S. Ding, and Z. Cao, “Integrated optimization of train stop planning and timetabling for commuter railways with an extended adaptive large neighborhood search metaheuristic approach,” Transportation Research Part C: Emerging Technologies, vol. 117, 2020.

L. He, M. de Weerdt, and N. Yorke-Smith, “Time/sequence-dependent scheduling: the design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm,” Journal of Intelligent Manufacturing, vol. 31, no. 4, pp. 1051–1078, 2020.

H. Ben Ticha, N. Absi, D. Feillet, and A. Quilliot, “Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows,” Computers & Operations Research, vol. 104, pp. 113–126, 2019.

M. A. Masmoudi, M. Hosny, E. Demir, and E. Pesch, “Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem,” Journal of Heuristics, vol. 26, no. 1, pp. 83–118, 2020.

S. N. Parragh, K. F. Doerner, and R. F. Hartl, “Variable neighborhood search for the dial-a-ride problem,” Computers & Operations Research, vol. 37, no. 6, pp. 1129–1138, 2010.

R. M. Jorgensen, J. Larsen, and K. B. Bergvinsdottir, “Solving the Dial-a-Ride problem using genetic algorithms,” Journal of the Operational Research Society, vol. 58, no. 10, pp. 1321–1331, 2007.

F. P. Goksal, I. Karaoglan, and F. Altiparmak, “A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery,” Computers & Industrial Engineering, vol. 65, no. 1, 2013.

D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical Report-TR06, Erciyes University, Kayseri, Turkey, 2005.

C. Cubillos, E. Urra, and N. Rodriguez, “Application of genetic algorithms for the DARPTW problem,” International Journal of Computers, Communications & Control, vol. 4, no. 2, pp. 127–136, 2009.