ADAPTIVE LARGE NEIGHBORHOOD SEARCH ALGORITHM FOR ROUTE PLANNING OF FREIGHT BUSES WITH PICKUP AND DELIVERY

ZHENG CHANG
Research Institute of Highway Ministry of Transport
Beijing 100070, China

HAOXUN CHEN AND FAROUK YALAOUI
University of Technology of Troyes
TROYES 10004, France

BO DAI
Hunan University of Technology and Business
ChangSha 410205, Hunan, China

(Communicated by Aviv Gibali)

Abstract. Freight bus is a new public transportation means for city logistics, and each freight bus can deliver and pick up goods at each customer/supplier location it passes. In this paper, we study the route planning problem of freight buses in an urban distribution system. Since each freight bus makes a tour visiting a set of pickup/delivery locations once at every given time interval in each day following a fixed route, the route planning problem can be considered a new variant of periodic vehicle routing problem with pickup and delivery. In order to solve the problem, a Mixed-Integer Linear Programming (MILP) model is formulated and an Adaptive Large Neighborhood Search (ALNS) algorithm is developed. The development of our algorithm takes into consideration specific characteristics of this problem, such as fixed route for each freight bus, possibly serving a demand in a later period but with a late service penalty, etc. The relevance of the mathematical model and the effectiveness of the proposed ALNS algorithm are proved by numerical experiments.

1. Introduction. The rapid development of e-commerce has been making urban logistics flows more and more intensive. Under the influence of market demand, more and more city freighters operated by different private third-party logistics companies were born and circulate in the centers of cities. These have caused serious traffic congestion and environmental pollution problems in large cities. To reduce traffic congestions and improve the efficiency and time accuracy of delivery, it needs more collaboration among third-party logistics companies (carriers) in urban logistics [1].

In 1973, Japanese scholar Shize [2] first put forward the joint distribution which has been proved to be an effective way for city logistics. Joint distribution promotes
enterprises with similar functions to use common facilities and equipment such as warehouses, logistics platforms, and vehicles, through which small orders of goods for delivery can be consolidated into a large-volume distribution order to achieve the economics of scale in transportation and other logistics services. Gill and Allerheiligen [3] pointed out that members of a distribution channel should cooperate with each other through joint distribution, and illustrated the effectiveness of joint distribution, and proposed several principles for implementing it. Hao and Su [3] discussed the basic concepts and operation models of joint distribution in city logistics. Xu and Yang [4] proposed a model for cost sharing among small companies implementing joint distribution.

Motivated by joint distribution, in our previous work(Model and Algorithm for route planning of freight buses), we put forward the concept of freight bus, which is a new public transportation means for city logistics that can replace city freighters belonging to different private third-party logistics companies in the center of a city. Freight bus has some advantages compared with city freighter. Firstly, freight bus can realize joint distribution of different third-party logistics companies, and can thus save city logistics costs and reduce the air pollution; Secondly, because of having a regular schedule, freight bus can improve the timeliness and accuracy of logistics services [5]; Thirdly, replacing private city freighters by freight buses can facilitate the traffic control in a city and reduce the traffic congestion. Finally, freight bus can improve the utilization rate of special lanes reserved for buses. In that article, we did not consider both pickup and delivery of goods at each customer/supplier location a freight bus passes at the same time. However, in practice as a new public transportation means for city logistics, freight buses should perform both pickup and delivery of goods at every customer/supplier location they visit [6, 7].

In this paper, we study the route planning problem of freight buses with pickup and delivery in an urban distribution system. In this system, each freight bus makes a tour visiting a set of pickup/delivery locations once at every given time interval in each day following a fixed route in a city, and the route planning problem can be considered a new variant of periodic vehicle routing problem with pickup and delivery. To the best of our knowledge, this problem was rarely studied in the literature. In order to solve the problem, a Mixed-Integer Linear Programming (MILP) mathematical model is formulated and an Adaptive Large Neighborhood Search (ALNS) algorithm is proposed in this paper.

The periodic vehicle routing problem (PVRP) was first introduced by Beltrami and Bodin [3] in the networks and vehicle routing for municipal waste collection. This is the first time that the periodicity of customer deliveries was specifically addressed in combination with the consideration of vehicle routing costs. Russell and Igo [8] named the periodic routing problem as the assignment routing problem. Christofides and Beasley [9], which is well cited by periodic routing papers, named the problem as period routing problem and provided the first mathematical formulation for the problem. The first article that uses the term “periodic vehicle routing” appears to be Gaudioso and Paletta [10]. In the paper entitled “Forty Years of Periodic Vehicle Routing”, Ann and Jill [11] discussed a wide range of circumstances and settings in which the PVRP has been applied and reviewed models and solution methods developed for the PVRP, including both exact and heuristic methods.

Beltrami and Bodin [12] designed the first two heuristics to solve the PVRP. The first one used a Clarke and Wright procedure to route customers, then assign
routes to periods. The second one is to randomly assign customers to these periods and create routes for each period. Chao [13] proposed a two-stage method to solve the PVRP, firstly, the set of customers is divided into several subsets, and then the vehicle’s route for serving each subset of customers is optimized. Francis [14] proposed an improved LNS (large neighborhood search) heuristic algorithm to solve the PVRP, which takes customer service frequency as a decision variable. Pourghaderi [15] designed a simple and effective line building algorithm through an embedded improvement program, and used this algorithm to solve the PVRP. Matos and Oliverira [15] adopted a pheromone update strategy to improve the performance of an ant colony algorithm, and used an improved ant colony algorithm to solve the PVRP. Drummond [16] presented an asynchronous parallel metaheuristic, which is based on concepts used in parallel genetic algorithms and local search, for the PVRP.

For the vehicle routing problem with pickup and delivery, Bent and Hentenryck [17] apply Variable Neighborhood Search (VNS) to the PDPTW and their computational results show promising performance of their algorithm, compared with the previous PDPTW metaheuristics. Ropke and Pisinger [18] design an ALNS algorithm which is probably the most effective meta-heuristic for the PDPTW so far, with results reported for up to 1000 customer nodes. ALNS uses several destruction and repair operators to improve the current solution. The neighborhood of a solution can be obtained by deleting few customers from the current solution and re-inserting these customers [19]. In ALNS, the deletion operator and the insertion operator used in each iteration are dynamically selected according to their past performance; each operator is associated with a weight (a fraction). If the operator improves the current solution, its weight increases; Meanwhile, if the newly generated solution conforms to the acceptance criteria defined by ALNS, it is accepted as the current solution for the next iteration. ALNS has been successfully applied to solve various vehicle routing problems [20].

The rest of this paper is organized in the following way. In section 2, we describe the operations of freight buses in city logistics. A mathematical formulation for the route planning problem of freight buses is provided in section 3. In section 4, we present an ALNS algorithm to solve this special periodic vehicle routing problem with pickup and delivery. In order to verify the model and evaluate the performance of the algorithm, we conduct numerical experiments on instances generated based on benchmark instances and analyze the numerical results in section 5. In section 6, we summarize the main work of this paper, and discuss further research subjects for implementing freight buses in city logistics.

2. Freight bus in city logistics. Freight is a new public transportation means for city logistics that can replace city freighters belonging to different private third-party logistics companies in the center of a city [21].

As a public service for third-party logistics companies and customers, the use of freight buses can achieve joint distribution. One important feature of freight buses is that they are standardized vehicles and have fixed time schedules. With the fixed schedules, both shippers and customers can well arrange their order delivery and delivery times [22]. As we can see from Fig. 2, each depot in such distribution system is a freight bus station, which may be a smart cabinet (depot) for temporary storage of goods. According to the schedules, customers can pick up their ordered products by self-service at the freight bus stations (depots), or ask the last-mile
3. Mathemathic model for freight bus routing. In this section, we establish a mathematical model for the vehicle routing problem of freight bus.

Each freight bus is operated (run) between a distribution center and multiple depots. Compared with the capacitated vehicle routing problem, the vehicle routing problem of freight bus has the following new characteristics: 1) Each freight bus makes a tour visiting a set of pickup/delivery locations once at every given time.
interval (period) in each day. 2Each freight bus has a fixed vehicle route in these peri-
ods. 3Both delivery and pick up can be operated at each depot (customer/supplier location). 4delivery and pickup of goods can be delayed but with penalties.

Therefore, the freight bus routing problem considered in this paper is a multi-
period vehicle routing problem with pickup and delivery, fixed routes, and late
pickup/delivery penalties. To the best of our knowledge, this problem was rarely
studied in the literature. In order to solve the problem, in this section we first for-
mulate a Mixed-Integer Linear Programming (MILP) model for the route planning
problem of freight buses.

3.1. Problem description. In the model, each freight bus runs between a Distribu-
tion Center and a set of depots. For simplicity, we don’t consider the interactions
of the freight buses with the electro-tricycles which perform last-mile deliveries from
depots to customers. Key features of the model are first introduced in the following.

1Multiple periods: We consider a time horizon (e.g. one day) that is divided into
M periods ($M > 1$) and assume that each freight bus visits its served depots once
in each period, and the demand of delivery and pickup of goods to each depot in
each period is known.

2Fixed routes: As passenger buses we assume that each freight bus has a fixed
route in the time horizon, and every depot must be served by one freight bus in
each period.

3Both delivery and pick up: Each freight bus can perform pickup and deliver
goods at each depot (customer/supplier location). The freight bus arrives at each
customer/supplier location (station), unload goods first and load goods later. Dur-
ing the whole tour, the total amount of goods in the freight bus should not exceed
its capacity. So it is possible that the delivery or the pickup demand of a depot in
a period is only partially met in this period because of the capacity limitation of a
freight bus, in this case, the unmet demand of the period can be only met in later
periods. In other words, the goods loaded or unloaded by the freight bus must be
the pickup or delivery demand of this period or the previous periods that were not
met due to the limitation of vehicle capacity.

4Penalty of delivery/pickup delay. Delivery and pickup of goods at a cus-
tomer/supplier location can be delayed but with penalties. There are two types
of penalty introduced in this model. One is the penalty caused by the delay in the
time horizon (e.g. one day) of M periods, which linearly depending on the quantity
of the late delivery or pickup demands and the number of periods delayed with the
penalty per period and per unit of demand given by a coefficient $\alpha$ (In this paper,
we assume late pickup and delivery have the same penalty coefficient). The other is
the penalty for all unmet demands at the end of the time horizon (e.g. one day) of
M periods, which linearly depending on the quantity of the late delivery or pickup
demands, with the penalty per unit of demand given by another coefficient $\beta$.

What’s more, Because of these two types of penalty, we can assume that the
operation of the freight bus has two characteristics: 1,When the freight bus arrives
at a depot, it unloads/delivers first and then loads/pickup the goods. (In order to
free up more capacity for pickup). 2, For delivery, the freight bus will give priority to
meeting the needs of the previous visiting depots according to the order of visiting;
for pickup, the freight bus will try its best to meet the loading needs of the depots
according to its maximum remaining capacity (Because late pickup and delivery
have the same penalty coefficient).

The main parameters of the model are defined as follows:
The distribution center where each freight bus leaves from and returns to.

- $V$ Set of freight buses.
- $G$ Set of depots.
- $U$ The capacity of each freight bus.
- $C_{ij}$ The operating cost for a freight bus when it travels from node $i$ to node $j$ ($i,j \in \{o\} \cup G$).
- $M$ The number of time periods we consider in the route planning problem.
- $d_i(k)$ The demand of delivery of depot $i$ in the $k$-th period, $i \in G, k \in \{1,2,...M\}$
- $p_i(k)$ The demand of pickup of depot $i$ in the $k$-th period, $i \in G, k \in \{1,2,...M\}$
- $\alpha$ The per period and per unit late delivery/pickup penalty cost for goods delivered/picked up in the time horizon of $M$ periods.
- $\beta$ The per unit late delivery/pickup penalty cost for goods delivered/picked up beyond the time horizon.

We assume that the Distribution Center $o$ serves all depots $G$ in the distribution system considered. The distance between node $i$ and node $j$ is denoted by $D_{ij}$. The operating cost of a freight bus from node $i$ to node $j$ is calculated as $C_{ij} = \gamma * D_{ij}$, where $\gamma$ is the unit distance operating cost of each freight bus. There are $N$ ($N$ is an integer) freight buses operated for the Distribution Center $o$, and the capacity of each freight bus is $U$.

In each period, each freight bus leaves from the Distribution Center $o$, visits its served depots and returns to the Distribution Center. What’s more, for one freight bus, every period, all goods loaded at the DC must be unloaded at its served depots before it returns to the Distribution Center; and all goods loaded at its served depots must be unloaded at the Distribution Center when it returns to the DC. It is assumed that the demand $d_i(k)$ and $p_i(k)$ of each depot $i$ in each period $k \in \{1,2,...M\}$ is known.

We need to plan the vehicle route for each freight bus $v$, and the delivery and pickup quantity of each freight bus at each depot in each period. The objective is to minimize the operating costs of all freight buses plus the late delivery and pickup penalty costs.

3.2. Mathematical model. In this subsection, we propose a mathematic model for the route planning of freight buses by considering its all characteristics. With this mathematic model, we can optimize the total cost of freight buses composed of their operating costs and penalty costs for the late delivery and pickup of goods in the planning horizon. At the same time, we can also get the optimal routes of freight buses by solving the model.

The detailed mathematical model for the route planning of freight buses is given as follows:

**Decision Variables**

- $x^v_{ij}$ A binary variable which is equal to 1 if the freight bus $v \in V$ goes from node $i$ to $j$ ($i,j \in \{o\} \cup G$); 0 otherwise.
- $y^v_i$ A binary variable which is equal to 1 if and only if the depot $i \in G$ is served by the freighter bus $v \in V$; 0 otherwise.
- $d^v_i(k)$ The unloaded quantity of the freighter bus $v \in V$ at the depot $i \in G$ in the $k$-th visit $k \in \{1,2,...,M\}$; 0 otherwise.
• $p^i_v(k)$: The loaded quantity of the freighter bus $v \in V$ at the depot $i \in G$ in the $k$-th visit $k \in \{1, 2, \ldots, M\}$; 0 otherwise.
• $Q^i_v(k)$: The quantity of all goods remaining to deliver in the freighter bus $v \in V$ when it just arrives at node $i \in \{0\} \cup G$ during the $k$-th visit, $k \in \{1, 2, \ldots, M\}$.
• $W^i_v(k)$: The quantity of all goods picked up by the freighter bus $v \in V$ when it just arrives at node $i \in \{0\} \cup G$ during the $k$-th visit, $k \in \{1, 2, \ldots, M\}$.

**Objective Function** The objective is to minimize the total cost including the operating costs of the freight buses and the penalty costs for the late delivery and pickup of goods in the planning time horizon of $M$ periods.

$$\text{Min } Z = M \sum_{i \in \{0\} \cup G} \sum_{j \in \{0\} \cup G} \sum_{v \in V} c_{ij} x_{ij}^v$$

$$+ \alpha \sum_{i \in G} \sum_{n=1}^{M-1} \sum_{k=1}^{n} \left( d_i(k) - \sum_{v \in V} d^i_v(k) \right)$$

$$+ \beta \sum_{i \in G} \sum_{n=1}^{M} \left( d_i(k) - \sum_{v \in V} d^i_v(k) \right)$$

$$+ \alpha \sum_{i \in G} \sum_{n=1}^{M-1} \sum_{k=1}^{n} \left( p_i(k) - \sum_{v \in V} p^i_v(k) \right)$$

$$+ \beta \sum_{i \in G} \sum_{n=1}^{M} \left( p_i(k) - \sum_{v \in V} p^i_v(k) \right)$$

**Constraints**

$$\sum_{j \in G} x_{oj}^v = \sum_{j \in G} x_{j0}^v \quad \forall v \in V$$

(2)

$$\sum_{i \in \{0\} \cup G} x_{ij}^v = \sum_{i \in \{0\} \cup G} x_{ji}^v \quad \forall j \in G, \forall v \in V$$

(3)

$$\sum_{i \in \{0\} \cup G} x_{ij}^v = y_j^v \quad \forall j \in G, \forall v \in V$$

(4)

$$\sum_{v \in \{0\} \cup G} y_j^v = 1 \quad \forall j \in G$$

(5)

$$Q^v_{ij}(k) \leq Q^v_r(k) - d^i_v(k) + U(1 - x_{ij}^v)$$

\forall i \in G, \forall j \in \{0\} \cup G, \forall v \in V, \forall k \in \{1, 2, \ldots, M\}$$

(6)

$$Q^v_{ij}(k) = 0 \quad \forall v \in V, \forall k \in \{1, 2, \ldots, M\}$$

(7)

$$W^v_{ij}(k) \geq W^v_r(k) + p^i_v(k) - U(1 - x_{ij}^v)$$

\forall i \in G, \forall j \in \{0\} \cup G, \forall v \in V, \forall k \in \{1, 2, \ldots, M\}$$

(8)

$$W^v_{ij}(k) = \sum_{i \in G} p^i_v(k) \quad \forall v \in V, \forall k \in \{1, 2, \ldots, M\}$$

(9)

$$0 \leq Q^v_{ij}(k) + W^v_{ij}(k) \leq U \quad i \in \{0\} \cup G, \forall v \in V$$

(10)

$$\sum_{k=1}^{n} d^v_{ij}(k) \leq \sum_{k=1}^{n} d^i_v(k) * y^v_i$$

\forall i \in G, \forall v \in V, \forall n \in \{1, 2, \ldots, M\}$$

(11)

$$\sum_{k=1}^{n} p^v_i(k) \leq \sum_{k=1}^{n} p_i(k) * y^v_i$$

\forall i \in G, \forall v \in V, \forall n \in \{1, 2, \ldots, M\}$$

(12)
Constraints (2) indicate that each freight bus leaves from and returns to the DC. Constraints (3) ensure that each freight bus arriving at a depot has to leave it. Constraints (4) and (5) guarantee that all depots must be served and each depot is served by at most one freight bus. Constraints (6) (7) and (8) (9) (10) formulate vehicle capacity constraints. Constraints (11) (12) indicate that in each period, the delivery and pickup of freight bus v at each depot i can only be the demand of that period or earlier periods but cannot be the demand of later periods. Finally, constraints (13) define the domains of all decision variables.

4. Adaptive large neighborhood search. We tried to use CPLEX to solve the proposed mathematical model in section 3, but only small instances could be solved to optimality in a reasonable computation time. For this reason, we propose an adaptive large neighborhood search algorithm to solve the vehicle routing problem of freight buses with both delivery and pickup.

4.1. Procedure of the ALNS. The large-scale neighborhood search algorithm was first proposed by Shaw [23]. This is an iterative algorithm. The idea of the algorithm is to improve the current solution in each iteration by using a destruction operator that removes some customer nodes from the current routes and a repair operator that reinserts these customer nodes to the routes. If the new solution is better than the current solution, the former is accepted as the current solution for the next iteration.

ALNS uses multiple destruction and repair operators to improve the current solution in each iteration. The neighborhood of a solution can be obtained by deleting several customer nodes from the current routes (solution) and re-inserting into them the customer nodes. In ALNS, a deletion operator and a re-insertion operator are dynamically selected in each iteration according to their past performance [24]; each operator is associated with a probability. If the operator improves the current solution, the probability will increase, otherwise the probability may decrease; The newly generated solution is accepted if it improves the current solution, otherwise it will be accepted with a probability depending on a temperature and defined according to a Simulation Annealing (SA) rule, the temperature will be gradually decreased with the progress of the algorithm; If the new generated solution is accepted, it will update the current solution for the next iteration. ALNS has been successfully applied to solve various vehicle routing problems [25, 18].

The procedure framework of our ALNS is given in Fig.4.

4.2. ALNS design. Taking into account the specific characteristics of our freight buses routing problem with both delivery and pickup, in this section, we design an ALNS to solve the problem. The most special features of this design are about the method of evaluation of a solution, the method of destroy operators and the method of repair operators.

4.2.1. Initial solution construction. Firstly, we need to generate the initial route for each freight bus. Cordeau, Gendreau, and Laporte [26, 27] analyzed and reviewed methods for generating initial solutions for vehicle routing problems, and classified these methods based on four different aspects, which are generated accuracy, speed,
simplicity and elasticity of solutions. After comparison, it is found that the method of Saving Algorithm [27] has advantages in generating an initial solution quickly and is simple and easy to implement. For this reason, firstly, we use Saving Algorithm to generate a single giant tour (long route), which contains all the nodes in the distribution system. Next, we randomly divide the long route into k segments (K is the maximum number of freight buses predefined for the system), and each segment corresponds to the initial route of a freight bus. The procedure of construction of the initial solution is given as follows:

Step 1: Apply the Saving Algorithm to construct a giant tour containing all nodes in the distribution system.

Step 2: The giant tour formed in the first step is randomly divided into K-segments (K is the number of freight buses predefined for the distribution system), and each segment corresponds to the initial route of a freight bus.

4.2.2. Evaluation of a solution. When we obtain a solution represented by given values of \( \{x_{ij}\} \) for the freight bus routing problem in ALNS, we need to evaluate its quality, i.e., the cost of the solution. Although this cost can be calculated by assigning the values to variables \( \{x_{ij}^v\} \) in the model of Section 3.2 and then solving the derived linear programing model, it is time consuming because such cost evaluation must be done a large number of times in ALNS. In the following, we provide a much more efficient way to calculate the cost of a solution given the values of \( \{x_{ij}^v\} \) under some reasonable assumptions about the operation of the freight buses.

We assume all freight buses operate with the two rules: 1) When a freight bus arrives at a depot, it unloads/delivers goods first and then loads/pickup goods from the depot. This order of delivery and pickup can free more capacity for pickup. 2) For delivery, when a freight bus loads goods to be delivered on its route before its departure from the CDC, it gives a higher priority to the demand of a depot to be visited earlier, i.e., it loads goods to be delivered according to the order of depots to be visited on the route; On the other hand, for pickup, every time when a freight bus arrives a depot with pickup demand, it will pick up goods as much as possible, i.e., use as much as possible its remaining capacity to meet the pickup demand. Note that this rule of pickup and delivery will not affect the cost of a solution since we assume late pickup (resp. late delivery) at each depot node has the same penalty coefficient.

With the operation rules, if we know the vehicle route of each freight bus, i.e., the values of \( x_{ij}^v \), in a solution, the delivery demand \( d_i(k) \) of depot \( i \) in the \( k \)-th period, the pickup demand \( p_i(k) \) of depot \( i \) in the \( k \)-th period, we can calculate the
values of variables $d^v_i(k)$ and $p^v_i(k)$ in the model in Section 3.2 and then the cost of the solution by the following formulas (14) - (19).

Calculation of $d^v_i(k)$
Define $d^v_i(k)$ as the sum of the delivery demand of depot $i$ in the $k$-th period (i.e., $d_i(k)$) and the unmet delivery demands of depot $i$ in all previous periods (periods 1 to $k-1$), $k \in \{1, 2, \ldots, m\}, d^v_i(k)$, which is referred to as the cumulative unmet delivery demand of depot $i$ in the $k$-th period, can be calculated by the formula:

$$d^v_i(k) = \begin{cases} 
    d_i(k) - d_i(k-1) & k \geq 2 \\
    d_i(1) & k = 1 
\end{cases}$$

(14)

Define $CD^v_i(k)$ as the remaining goods to be delivered in freighter bus $v$ when it arrives at node $i \in G$ on its route in the $k$-th visit, $k \in \{1, 2, \ldots, M\}, CD^v_i(k)$ can be recursively calculated from the first depot node to the last depot node visited by the freight bus $v$ applying the following formula.

$$CD^v_i(k) = \begin{cases} 
    CD^v_j(k) - d^v_i(k) & j \in G, x^v_{ji} = 1, j \in G' \\
    \min \left\{ U, S(k) \right\} & i \in G, x^v_{ii} = 1 
\end{cases}$$

where $U$ is the capacity of the freight bus and $S(k) = \sum_{i \in G} d^v_i(k) \ast y^v_i$ is the total cumulative undelivered demand of all depots to be visited by freight bus $v$ in the $k$-th period. In this formula, node $j$ is visited by the freight bus $v$ just before node $i$. Since each depot can only be visited once by a freight bus in each period, the depot node $j$ in the formula is unique for any given depot node $i$.

Because goods loaded or unloaded by a freight bus must be the pickup or delivery demand of the current period or those of the previous periods that were not met due to the limited capacity of the freight bus. So $d^v_i(k)$ the quantity unloaded by freighter bus $v \in V$ at the depot $i \in G$ in the $k$-th visit must take the smaller value between $d^v_i(k)$ and $CD^v_i(k)$.

$$d^v_i(k) = \min \left\{ d^v_i(k), CD^v_i(k) \right\}$$

(16)

Calculation of $p^v_i(k)$:
Define $p^v_i(k)$ as the sum of the pickup demand of depot $i$ in the $k$-th period (i.e., $d_i(k)$) and the unmet pickup demands of depot $i$ in all previous periods (periods 1 to $k-1$), $k \in \{1, 2, \ldots, m\}, p^v_i(k)$, which is referred to as the cumulative unmet pickup demand of depot $i$ in the $k$-th period, can be calculated by the formula:

$$p^v_i(k) = \begin{cases} 
    p_i(k) - p_i(k-1) & k \geq 2 \\
    p_i(1) & k = 1 
\end{cases}$$

(17)

Define $CP^v_i(k)$ as the remaining capacity for pickup of the freighter bus $v \in V$ when it arrives at node $i \in G$ on its route in the $k$-th visit, $k \in \{1, 2, \ldots, M\}, M^v_i(k)$ can be recursively calculated from the first depot node to the last depot node visited by the freight bus $v$ applying the following formula:

$$CP^v_i(k) = \begin{cases} 
    CP^v_j(k) - d^v_i(k) + d^v_i(k) & j \in G, x^v_{ji} = 1, j \in G' \\
    U - S^v(k) + d^v_i(k); & i \in G, x^v_{ii} = 1 
\end{cases}$$

where $U$ is the capacity of the freight bus and $S^v(k) = \sum_{i \in G} d^v_i(k) \ast y^v_i$ is the cumulative unmet delivery demand of all depots to be visited by freight bus $v$ in the $k$-th period. Similarly, in this formula, node $j$ is visited by the freight bus $v$ just before node $i$, and node $j$ is unique for any given node $i$. 


So \( p_v^i(k) \) (the quantity loaded by the freighter bus \( v \in V \) at the depot \( i \in G \) in the \( k \)-th visit) must also take the smaller value between \( p_v^i(k) \) and \( CP_v^i(k) \), i.e.,

\[
p_v^i(k) = \min \{ p_v^i(k), CP_v^i(k) \}
\]  

On the whole, we can evaluate the cost of a solution by the following steps:

Step1: Apply the formulas (14) – (16) to calculate the unloaded quantity \( d_v^i(k) \) of the freighter bus \( v \in V \) at the depot \( i \in G \) in its \( k \)-th visit \( k \in \{1, 2 \ldots m\} \);

Step2: Apply the formulas (17)-(19) to calculate the loaded quantity \( p_v^i(k) \) of the freighter bus \( v \in V \) at the depot \( i \in G \) in its \( k \)-th visit \( k \in \{1, 2 \ldots m\} \);

Step3: Calculate the cost of the solution by applying the formula (1) in section 3.2.

4.2.3. Adaptive selection of destroy/repair operators. Our ALNS chooses a removal operator and an insertion operator in each iteration by applying the roulette-wheel mechanism. At the beginning all removal operators and all insertion operators are selected with the same probability. In the ALNS algorithm, 8 removal operators and 3 insertion operators are used. Therefore, the probability of initial selection of each removal operator and each insertion operator is 1/8 and 1/3 respectively. In the process of algorithm execution, the probability of each operator being selected is updated according to the following formula.

\[
p_{t+1}^i = p_t^i (1 - r) + r \beta_t^i / \alpha_t^i
\]  

\( p_t^i \) is the probability of operator \( i \) being selected in the \( t \)-th iteration;
\( r \in [0, 1] \) is the Roulette parameter predefined.
\( \alpha_t^i \) is the number of times the operator \( i \) was selected in the past \( t \) iterations.
\( \beta_t^i \) is the resulting score of the operator \( i \) in the \( t \)-th iteration.

The score of an operator is used to measure its performance in each iteration. If a new best solution is found, the score of the operator will increase by \( Q_1 \) if the solution found is better than the current solution, the score of this operator will increase by \( Q_2 \) if the solution found is worse than the incumbent solution, the score of this operator will increase by \( Q_3 \) [28].

4.2.4. Destroy operators. Our ALNS contains two classes of destroy operators. The first class of destroy operator is to remove a certain number of depots from different routes, which involves 4 destroy operators. The second class of destroy operator chooses one route and then removes all the depots on this route which involves 4 destroy operators too.

What’s more, for the first class of destroy operators, we will give the number \( n \) of all the depots in system and a removal fraction \( \rho \in [0, 1] \), each of operators applies a strategy to select \( \rho \times n \) depots to remove.

Only one destroy operate is executed in each ALNS iteration, the details of those 8 destroy operators are described as following:

1. Random depot Removal
   This operator randomly chooses \( k \) depots at a time, \( (k = \rho \times n, \) where \( \rho \) is a removal fraction, \( n \) is the number of depots), removes them from the current vehicle routes, and places them in the removal list LR. This operator helps increase the diversity of solutions.

2. Worst-distance depot Removal
   This operator removes \( \rho \times n \) depots with largest travel distances from different routes. The travel distance of each depot \( i \) is defined as \( L_{ji} + L_{ik} \), where node \( j \) and \( k \) are the depot or the DC visited before depot \( i \) and the depot or the DC visited
after depot \( i \) in the same route, respectively. The depots with the largest travel distances will be removed from the current routes in turn.

3 Proximity-based Removal

The goal of this operator is to find out a set of depots that have some connection in terms of distance and remove them from the current solution. We use the way of Shaw removal [30] to define the relationship between depots. Firstly, a depot is chosen randomly and we place it in the removal list as the first depot to be removed. Then the next depot to be removed is the depot closest (Minimum distance) to the last removed depot. After repeating \( n \) times Shaw removal, we get all the depots to removal in this operator.

4 Historical depot Removal

The operator records the distance information of every depot in the past iterations. The distance of a depot is defined as the sum of distances from the depot before and after the depot (The distance for a depot is \( \frac{L_{i-1,i} + L_{i,i+1}}{L_i} \). During the execution of our algorithm, the optimal distance of depot \( i \) is minimum distance obtained in all past iterations. This operator removes \( \rho \times n \) depots with the greatest distance difference between the current distance and its optimal distance in each iteration.

5 Random Route Removal

This operator removes an entire route from the current solution. The route to remove is randomly chosen from all routes in the current solution.

6 Largest demand Route Removal

This operator first calculates the sum of the delivery and pickup demands for all depots in each route in the current solution, then remove the route which has the largest total demand.

7 Largest distance Route Removal

This operator calculates the distance traveled by a freight bus in each route, then removes the route which has the largest travel distance.

8 Least depots Route Removal

This operator calculates the number of depots visited by each route, then removes the route which has the least number of depots served.

4.2.5. Repair operators. Our ALNS algorithm contains three kinds of insertion operators, which are mainly used to re-insert the depots in the removed list into the current solution to generate a new feasible solution. The following describes the function of each insertion operator in detail:

1 Basic Greedy Insertion

The idea of this operator is to insert each depot in the removal list into the best possible route and position. Firstly the insertion distance of each depot \( i \) is calculated for each possible route as \( L_i = L_{ji} + L_{ik} - L_{jk} \), where \( j \) and \( k \) are the node preceding and to node following depot \( i \) in the route if \( i \) is inserted. then the depot with the lowest insertion distance is selected and inserted into the best route and the best insertion position [28].

2 Greedy Insertion with Noise function

This operator is an extension of the basic greedy algorithm, but with a degree of freedom in selecting the insertion route and the insertion position for each depot to insert. The degree of freedom is realized by appropriately changing the insertion cost (distance) of each depot by adding a noise. The insertion cost of each node \( i \) becomes \( L'_i = L_i + L_{\max} \times u \times \epsilon (\text{Noise}) \), where \( L_{\max} \) is the maximum distance between any two depots; \( u \) is the noise parameter set to be
0.1 in our numerical experiments, and $\varepsilon$ is a random value between $-1$ and $1$. This operator calculates the modified insertion costs of all depots in the removal list, and then selects the depot with the smallest insertion cost and insert it into the best route and the best position [19].

3 Greedy Insertion with new route generated

This operator is that combines basic greedy insertion to generate a new route. It consists of two steps: the first step is to randomly select a destroyed depot to connect with the Distribution Centre to form a new route (That is, we first generate a path that only serves one depot); The second step is to insert other destroyed depots into the existing routes according to the basic greedy insertion method introduced in repair operator 1.

4.2.6. Acceptance and stop criterion. The ALNS algorithm proposed in this paper adopts simulated annealing as the external local search framework. The algorithm framework of the ALNS is given in Figure 5.2. The algorithm stops when a specified number of ALNS iterations is reached.

In the figure, $X_{\text{best}}$ refers to the best solution found by the algorithm; $X_{\text{current}}$ refers to the current solution; $X_{\text{new}}$ refers to a new temporary solution found in an iteration, which may be discarded or accepted as the current solution for the next iteration. The objective value of a feasible solution $X$ is denoted by $f(X)$. If $f(X_{\text{new}}) < f(X_{\text{current}})$, then $X_{\text{new}}$ is always accepted, otherwise $X_{\text{new}}$ will be accepted at a probability of $e^{-(f(X_{\text{new}}) - f(X_{\text{current}}))/T}$, where $T$ is the temperature of simulated annealing, set initially as $f(X_{\text{init}}) \cdot P_{\text{init}}$, where $f(X_{\text{init}})$ is the objective value of the initial solution, and $P_{\text{init}}$ is a constant. The cooling rate of the simulated annealing is denoted by $h$, and $h \in (0, 1)$ is a given parameter. The algorithm returns the best solution after reaching the maximum number of iterations [23, 20].

5. Numerical experiments. In order to verify the mathematical model of freight bus routing and evaluate the effectiveness of our ALSN, in this section, we design 70 instances in small, medium and large sizes, and compare the performance of the ALSN (implemented in MATLAB 2014a with Intel Core i5-4210M CPU 2.6GHz) with that of the MILP solver of CPLEX (CPLEX 12.6on the instances. In addition, the impact of joint distribution realized by freight buses on the reduction of transportation costs is also evaluated and analyzed in this section.

5.1. Instance generation. The freight bus routing problem considered in this paper is related to the Vehicle Routing Problem with Pickup and Delivery (VRPPD). So when generating the instances, we use some data from the benchmark instances of VRPPD provided by Breedam at http://neo.lcc.uma.es/vrp/. However, since the freight bus routing problem involves multiple periods, we generate demand data randomly based on the benchmark data.

We designed 70 instances for the problem which are grouped in instances of small size ($N \in \{7, 13\}; M \in \{3, 5\}$), instances of medium size ($N \in \{20, 30, 40\}; M \in \{3, 5\}$) and instances of large size ($N \in \{60, 80\}; M \in \{3, 5\}$) instances (see Table 3). For all instances, the following data are taken from the benchmark instances: the coordinates of all nodes, the number of vehicles, and the capacity of each vehicle. The other data are generated randomly based on the benchmark instances: the delivery demand $d_i(k)$ of each depot in each period, the pickup demand $p_i(k)$ of each depot in each period, the number of periods $M$, the unit distance operating cost $\gamma$, the penalty coefficient $\alpha$ and $\beta$. 
Algorithm 1 - Pseudo-code of ALNS metaheuristic

1: Generate an Initial Solution by Saving Algorithm (Section 4.2.1)
2: Initialize weight and score of each destroy operator d and each repair operator i; Where d D, i I. (Section 4.2.2)
3: Initialize constant P_{init}, cooling rate h, removal fraction
4: X_{current} = X_{init}
5: TT_{beg}
6: for iter 1 to niters do
7: Select a remove operator d from D with the probability P_d^t
8: The remove operator d is applied to X_{current} to obtain a partial solution X_{new}^{'}.
9: Select an insertion operator i from I with the probability P_i^t
10: The insertion operator i is applied to X_{new}^{'} to obtain a new feasible solution X_{new}
11: If f(X_{new}) < f(X_{current}), then
12: X_{current} = X_{new};
13: else ve^{-((f(X_{new}) - f(X_{current}))/T)}
14: Randomly generate a number between 0 and 1
15: If < v, then X_{current} = X_{new}; end if
16: end if
17: If f(X_{current}) < f(X_{best}), then X_{best} = X_{new} end if
18: TT =
19: Update the selection probability of each operator according to the adaptive weight adjustment strategy
20: end for

The number of periods M is set to 3 or 5, since it is assumed that all freight buses have the same unit distance operating cost, we simply set γ to 1 for all instances. In order to further evaluate the impact of the joint distribution realized by freight buses in section 5.5, we generate the demand of each depot in each period by grouping the customer demands of two private third party logistics companies A and B at the depot in the period, where the demand (both the delivery and pickup) of each company at each depot in each period is randomly generated from [20, 24]. For the penalty coefficient α, because the ratio of α to γ and β to γ affect the tradeoff between the operating costs of the freight buses and the penalty costs for late delivery and pickup as well as the service level to customers, i.e., the percentage of customer demands delivered/picked up on-time, we cannot set α and β too big or too small. After some tests with different α and β values, in our numerical experiments we set α to 20, β t 400 for all instances.

5.2. Parameter setting. We used CPLEX to solve the MIP model of freight bus routing, and used Matlab to implement our ALNS.

Since the ALNS algorithm is composed of several procedures and each procedure has its own parameters, parameter setting was tuned by finding a tradeoff between solution quality and CPU time. The values of the parameters used are listed in Table 1. As we can see from the table, the parameters are divided into three groups. The first group consists of the parameters related to the roulette mechanism, and the second group consists of the parameters related to the simulated annealing. The third group consists of the parameters related to the removal operators and insertion operators [29].
| Parameter                                                                 | Small instances | Medium instances | Large instances |
|--------------------------------------------------------------------------|-----------------|------------------|-----------------|
| Maximum iterations number of ALNS (N)                                    | 25000           | 30000            | 35000           |
| The roulette parameter (r)                                               | 0.1             | 0.1              | 0.1             |
| The score increment of generating a new best solution (Q1)              | 5               | 5                | 5               |
| The score increment of generating a better solution (Q2)                 | 3               | 3                | 3               |
| The score increment of generating a worse solution (Q3)                  | 1               | 1                | 1               |
| Initial temperature parameter ($P_{init}$)                               | 100             | 120              | 120             |
| Cooling rate (h)                                                         | 0.992           | 0.994            | 0.996           |
| Removal fraction ($\rho$)                                                | 0.2             | 0.2              | 0.3             |
| Noise parameter (u)                                                      | 0.1             | 0.1              | 0.1             |

Since it is very time consuming for CPLEX to find an optimal solution of the freight bus routing model even for instances of small size, we set its CPU running time to 1800, 5400, 7200, 10800, 14400, 18000 seconds for instances with 7, 13, 20, 30, 40, 60, 80 depot nodes, respectively.

5.3. Experimental results. Three key values were used to evaluate the performance of our ALNS:
1) The lower bound produced by CPLEX, which indicates a lower bound of the optimal objective value of the model.
2) The best feasible solution of the MIP model found by CPLEX, which is an upper bound of the optimal objective value.
3) The best objective value of the model obtained by the ALNS.

The indicators used in the performance evaluation of our ALNS and their definitions are given in Table 2.

The following three tables give the experimental results of small size instances, medium size instances, and large size instances, respectively. Where each instance is identified by the number of depots and the number of periods. For example, instance 7-3 is an instance with 7 depots and 3 periods (7-3a, 7-3b is the different delivery demands instances with 7 depots and 3 periods).

Table 3 compares the solutions of our ALNS with those of CPLEX solver on small size instances. We can see, for the set of instances with $N = 7$, $M = 3$ and $N = 7$, $M = 5(7-5)$, both ALNS and CPLEX can solve the model to optimality, and our ALNS consumed less CPU time than CPLEX. When the number of depot nodes increases to 13, no proven optima was obtained. In this case, we compare near-optimal solutions of the two methods and their running times. Comparing their three indicators ($Gap_{CPLEX}$, $Gap_{ALNS}$, and $Imp_{ALNS-CPLEX}$) we can see that ALNS found better solutions than CPLEX with an average percentage improvement $Imp_{PMA-CPLEX}$ of 20.41% in terms of total cost. Furthermore, we can see our ALNS has a great advantage over CPLEX in terms of running time, the longest CPU time.
Table 2. Performance indicators

| Abbreviation | Definition |
|--------------|-----------|
| CplexObj | The best feasible objective value found by CPLEX in a preset running time |
| LB | The lower bound produced by CPLEX in a preset running time |
| ALNSObj | The best feasible objective value obtained by ALNS after a preset number of iterations |
| GapCplex | The gap between CplexObj and LB, which is defined as \((\text{CplexObj} - \text{LB})/\text{LB} \times 100\) |
| GapALNS | The gap between ALNSObj and LB, which is defined as \((\text{ALNSObj} - \text{LB})/\text{LB} \times 100\) |
| ImpALNS-Cplex | The improvement (reduction) of ALNSObj over CplexObj, which is defined as \((\text{CplexObj} - \text{ALNSObj})/\text{CplexObj} \times 100\) |
| CPUCplex | The CPU time (second) of CPLEX |
| CPUALNS | The CPU time (second) of ALNS |

Table 3. Experimental results of small size instances.

| Instances | CplexObj | LB | ALNSObj | GapCplex | GapALNS | ImpALNS-Cplex | CPUCplex | CPUALNS |
|-----------|----------|----|---------|----------|---------|----------------|----------|---------|
| 7-3a | 1067.1 | 1067.1 | 1067.1 | 0.00 | 0.00 | 0.00 | 86.3 | 7.9 |
| 7-3b | 986.6 | 986.6 | 986.6 | 0.00 | 0.00 | 0.00 | 89.6 | 7.6 |
| 7-3c | 1076.9 | 1076.9 | 1076.9 | 0.00 | 0.00 | 0.00 | 87.9 | 8.2 |
| 7-3d | 962.0 | 962.0 | 962.0 | 0.00 | 0.00 | 0.00 | 87.6 | 8.7 |
| 7-3e | 1005.8 | 1005.8 | 1005.8 | 0.00 | 0.00 | 0.00 | 87.5 | 7.9 |
| 7-5a | 1984.1 | 1984.1 | 1984.1 | 0.00 | 0.00 | 0.00 | 257.3 | 19.7 |
| 7-5b | 1945.1 | 1945.1 | 1945.1 | 0.00 | 0.00 | 0.00 | 262.1 | 20.3 |
| 7-5c | 1887.0 | 1887.0 | 1887.0 | 0.00 | 0.00 | 0.00 | 258.2 | 21.9 |
| 7-5d | 1940.4 | 1940.4 | 1940.4 | 0.00 | 0.00 | 0.00 | 259.3 | 19.8 |
| 7-5e | 1896.6 | 1896.6 | 1896.6 | 0.00 | 0.00 | 0.00 | 260.3 | 20.3 |
| 13-3a | 1459.9 | 1074.8 | 1197.0 | 35.83 | 11.37 | 18.01 | 3600 | 26.9 |
| 13-3b | 1519.1 | 1089.3 | 1207.8 | 39.46 | 10.88 | 20.49 | 3600 | 26.6 |
| 13-3c | 1800.1 | 1342.1 | 1530.6 | 34.13 | 14.05 | 14.97 | 3600 | 26.9 |
| 13-3d | 1397.4 | 1007.7 | 1123.6 | 38.67 | 11.50 | 19.59 | 3600 | 26.3 |
| 13-3e | 2878.7 | 1848.4 | 2086.5 | 55.74 | 12.88 | 27.52 | 3600 | 27.1 |
| 13-5a | 3761.2 | 2527.9 | 2847.0 | 48.79 | 12.62 | 24.30 | 3600 | 40.3 |
| 13-5b | 3232.8 | 2284.2 | 2673.5 | 42.78 | 18.08 | 14.70 | 3600 | 39.7 |
| 13-5c | 3329.6 | 2220.5 | 2607.1 | 49.95 | 17.41 | 21.70 | 3600 | 42.3 |
| 13-5d | 3402.7 | 2238.7 | 2581.8 | 51.99 | 15.33 | 24.13 | 3600 | 41.2 |
| 13-5e | 3405.6 | 2427.8 | 2856.6 | 40.28 | 17.66 | 16.12 | 3600 | 40.3 |

For ALNS is only 42.3 seconds compared with the limit of 3600 seconds reached by CPLEX.

Table 4 compares the experimental results of the two solution methods for medium size instances. With the increase of the number of depot nodes, the gap of CPLEX GapCplex, has a rapid growth from 43.45% to 125.58%, while our ALNS maintains a relatively smaller change from 7.14% to 24.88%. As a result, there is a significant increase in the improvement ImpALNS-Cplex from 24.00% to 47.22%. For the sets of instances with \( N = 30, M = 5(30 - 5) \), \( N = 40, M = 5(40 - 5) \),
Table 4. Experimental results of medium size instances

| Instances | CplexObj | LB | ALNSObj | Gap_{\text{Cplex}} | Gap_{\text{ALNS}} | Imp_{\text{ALNS} - \text{Cplex}} | CPU_{\text{Cplex}} | CPU_{\text{ALNS}} |
|-----------|----------|----|---------|-------------------|-----------------|------------------------|----------------|----------------|
| 20-3a     | 2884.5   | 1966.6 | 2161.4  | 46.67             | 9.91            | 25.07                  | 5400           | 135.7          |
| 20-3b     | 3014.2   | 2101.2 | 2251.3  | 43.45             | 7.14            | 25.31                  | 5400           | 140.7          |
| 20-3c     | 3058.2   | 2089.5 | 2268.2  | 47.77             | 9.60            | 25.83                  | 5400           | 133.2          |
| 20-3d     | 2649.0   | 1807.7 | 2013.2  | 46.54             | 11.37           | 24.00                  | 5400           | 135.7          |
| 20-3e     | 2904.7   | 1949.3 | 2141.2  | 49.01             | 9.84            | 26.29                  | 5400           | 134.2          |
| 20-5a     | 5161.4   | 2951.8 | 3365.4  | 74.86             | 14.01           | 34.80                  | 5400           | 216.4          |
| 20-5b     | 4527.3   | 2627.2 | 2992.6  | 72.32             | 13.91           | 33.90                  | 5400           | 203.5          |
| 20-5c     | 4758.4   | 2608.5 | 3065.1  | 76.34             | 13.59           | 35.59                  | 5400           | 218.8          |
| 20-5d     | 4474.2   | 2556.1 | 2852.5  | 75.04             | 11.60           | 36.25                  | 5400           | 199.5          |
| 20-5e     | 4522.8   | 2573.0 | 2927.5  | 75.78             | 13.78           | 35.27                  | 5400           | 220.3          |
| 30-3a     | 4964.2   | 2664.8 | 3003.9  | 86.29             | 12.73           | 39.49                  | 7200           | 289.3          |
| 30-3b     | 5401.6   | 2922.8 | 3321.7  | 84.81             | 13.65           | 38.50                  | 7200           | 276.8          |
| 30-3c     | 4688.7   | 2520.6 | 2860.1  | 86.02             | 13.47           | 39.00                  | 7200           | 296.3          |
| 30-3d     | 4540.4   | 2381.9 | 2728.1  | 80.62             | 14.53           | 39.92                  | 7200           | 295.3          |
| 30-3e     | 4654.2   | 2502.6 | 2844.2  | 85.97             | 13.65           | 38.89                  | 7200           | 287.6          |
| 30-5a     | 4325.6   | 5167.3 | -      | 19.46             | -              | 356.7                  | 7200           | 320.7          |
| 30-5b     | 8140.4   | 4859.6 | 99.32   | 18.99             | 40.30           | 378.9                  | 7200           | 378.9          |
| 30-5c     | 4390.0   | 5249.4 | -      | 19.58             | -              | 364.3                  | 7200           | 378.9          |
| 30-5d     | 3973.0   | 4733.7 | -      | 19.15             | -              | 378.9                  | 7200           | 378.9          |
| 30-5e     | 4154.9   | 4938.6 | -      | 18.86             | -              | 352.1                  | 7200           | 378.9          |
| 40-3a     | 6214.3   | 3469.4 | 119.15  | 18.89             | 45.75           | 347.9                  | 10800          | 478.7          |
| 40-3b     | 6389.9   | 3282.7 | 125.58  | 19.06             | 47.22           | 480.3                  | 10800          | 480.3          |
| 40-3c     | 2917.5   | 3469.4 | -      | 18.92             | -              | 437.9                  | 10800          | 437.9          |
| 40-5a     | 6072.1   | 3381.9 | 114.41  | 19.42             | 44.30           | 509.3                  | 10800          | 509.3          |
| 40-5b     | 6112.3   | 3590.5 | 103.17  | 19.35             | 41.26           | 469.5                  | 10800          | 469.5          |
| 40-5c     | 4824.4   | 6020.8 | -      | 24.80             | -              | 597.3                  | 10800          | 597.3          |
| 40-5d     | 5065.7   | 6325.8 | -      | 24.88             | -              | 600.1                  | 10800          | 600.1          |
| 40-5e     | 4944.1   | 6088.1 | -      | 23.14             | -              | 623.5                  | 10800          | 623.5          |
| 40-5f     | 4823.4   | 5940.9 | -      | 23.17             | -              | 591.2                  | 10800          | 591.2          |
| 40-5g     | 4664.0   | 5764.5 | -      | 23.60             | -              | 589.7                  | 10800          | 589.7          |

and $N = 40, M = 5(40 - 5)$, we can see sometimes CPLEX failed to find a feasible solution, even after a long running of 2h or 3h. By contrast, our ALNS could always find high quality feasible solutions. Although the running time of the ALNS increases from 133.2 seconds to 623.5 seconds, its computation time is much smaller than that of CPLEX for all instances.

Table 5 compares the experimental results for large size instances. We can see CPLEX failed to find a feasible solution for 17 out of 20 instances with a preset running time. For most of the instances, CPLEX stopped due to out of memory. In this case, we can only compare the lower bound LB produced by CPLEX and the upper bound found by the ALNS ($\text{ALNS}_{\text{Obj}}$). For large size instances, we can see our ALNS produced solutions with an average gap $\text{Gap}_{\text{ALNS}}$ of 20.63%, and the best improvement $\text{Imp}_{\text{ALNS} - \text{Cplex}}$ of the ALNS over CPLEX could reach 53.84%.

Fig.3, Fig.4 and Table 6 summarizes the average performances of our ALNS and the MIP solver of CPLEX on all instances tested. These experimental results show that our ALNS is effective for solving the freight bus routing problem studied in
Table 5. Experimental results of large size instances.

| Instances | CplexObj | LB | ALNSObj | GapCplex | GapALNS | ImpALNS - Cplex | CPU_Cplex | CPU_ALNS |
|-----------|----------|----|---------|----------|---------|-----------------|-----------|----------|
| 60-3a     | 11510.5  | 4550.1 | 5313.7 | 152.97   | 16.78   | -               | 14400     | 979.3    |
| 60-3b     | -        | 5140.4 | 5971.6 | -        | 16.17   | -               | 14400     | 1000.1   |
| 60-3c     | 11918.4  | 4742.3 | 5507.3 | 151.32   | 16.13   | 53.79           | 14400     | 989.3    |
| 60-3d     | -        | 4340.2 | 5106.7 | -        | 17.66   | -               | 14400     | 966.1    |
| 60-3e     | 11509.7  | 4694.8 | 5494.9 | 145.16   | 17.04   | 52.26           | 14400     | 996.1    |
| 60-5a     | -        | 8123.9 | 9972.9 | -        | 22.76   | -               | 14400     | 1421.6   |
| 60-5b     | -        | 7669.4 | 9493.1 | -        | 23.78   | -               | 14400     | 1432.3   |
| 60-5c     | -        | 8041.2 | 9809.9 | -        | 22.00   | -               | 14400     | 1410.5   |
| 60-5d     | -        | 8318.1 | 10172.5| -        | 22.29   | -               | 14400     | 1396.3   |
| 60-5e     | -        | 8372.7 | 10135.2| -        | 21.05   | -               | 14400     | 1389.3   |
| 80-3a     | -        | 6288.6 | 7419.3 | -        | 17.98   | -               | 18000     | 1523.4   |
| 80-3b     | -        | 5992.0 | 7135.8 | -        | 19.09   | -               | 18000     | 1612.3   |
| 80-3c     | -        | 6431.5 | 7683.1 | -        | 19.46   | -               | 18000     | 1496.8   |
| 80-3d     | -        | 5785.2 | 6836.7 | -        | 18.18   | -               | 18000     | 1527.4   |
| 80-3e     | -        | 6104.0 | 7264.1 | -        | 19.01   | -               | 18000     | 1559.3   |
| 80-5a     | -        | 8670.0 | 10889.2| -        | 25.60   | -               | 18000     | 1963.2   |
| 80-5b     | -        | 8260.9 | 10207.5| -        | 23.56   | -               | 18000     | 2001.3   |
| 80-5c     | -        | 8855.8 | 11163.3| -        | 26.06   | -               | 18000     | 1989.7   |
| 80-5d     | -        | 8611.5 | 10627.6| -        | 23.41   | -               | 18000     | 1967.3   |
| 80-5e     | -        | 8596.7 | 10703.1| -        | 24.50   | -               | 18000     | 1995.3   |

Figure 5. The average performances of CPLEX and ALNS (M = 3).

This paper. And Through these comparisons, we can also find that the larger the size of the instances, the more obvious the advantages of our ALNS.

5.4. Impact of the joint distribution realized by freight buses. In order to evaluate the impact of the joint distribution realized by freight buses, we compare a system with freight buses with its corresponding system without freight bus.
Figure 6. The average performances of CPLEX and ALNS ($M = 5$).

Table 6. The average performances of the two solution methods.

| Instances | Gap_{Cplex} | Gap_{ALNS} | Imp_{ALNS-Cplex} | CPU_{Cplex} | CPU_{ALNS} |
|-----------|-------------|------------|------------------|------------|------------|
| 7-3       | 0           | 0          | 0                | 87.78      | 8.06       |
| 7-5       | 0           | 0          | 0                | 259.44     | 20.4       |
| 13-3      | 40.76       | 12.14      | 20.12            | 3600       | 26.3       |
| 13-5      | 46.76       | 16.22      | 20.71            | 3600       | 40.76      |
| 20-3      | 46.69       | 9.57       | 25.30            | 5400       | 135.9      |
| 20-5      | 74.87       | 13.38      | 35.16            | 5400       | 211.7      |
| 30-3      | 86.74       | 13.61      | 39.16            | 7200       | 289.1      |
| 30-5      | 99.32       | 19.21      | 40.30            | 7200       | 354.5      |
| 40-3      | 115.58      | 19.13      | 44.63            | 10800      | 475.1      |
| 40-5      | -           | 23.92      | -                | 10800      | 600.4      |
| 60-3      | 149.82      | 16.76      | 53.30            | 14400      | 994.0      |
| 60-5      | -           | 22.38      | -                | 14400      | 1410.0     |
| 80-3      | -           | 18.74      | -                | 18000      | 1543.8     |
| 80-5      | -           | 24.63      | -                | 18000      | 1983.4     |

In the system without freight bus, we assume that there are city freighters operated by two private third party logistics companies A and B, which separately deliver and pickup their customers’ demands from a distribution center to multiple depots. Each city freighter of company A or B also visits its served depots once during each period $k \in \{1, 2 \cdots M\}$, and the demand of each company’s customers at each depot $i$ in each period must be served by its own city freighters. In each period, each city freighter also begins and ends its travel at the distribution center. What’s more, it is also possible that the demand of a depot in a period is totally or partially served in later periods because of the capacity limitation of a city freighter, and there are also two kinds of penalty costs for late delivery and pickup.

To simplify the comparison of the two systems, we assume that all city freighters operated by company A and company B have the same capacity $U$, the same unit distance operating cost $\gamma$, the same penalty coefficient $\alpha$ and $\beta$, the same number
of periods $M$ in the planning time horizon as those for the freight buses, and all city freighters also have fixed vehicle routes. The objective of each company is to minimize its total cost which includes the operating costs and the penalty costs of its own city freighters. With this assumption, we can use the ALNS proposed in this paper to optimize the vehicle routes of the city freighters of each company and get its total cost. The total cost of the system without freight bus is thus the sum of the total costs of company A and B. The following table 7 compares the average costs of the two systems.

| Instances        | System without Freight bus | System with Freight bus | Cost Savings in percentage |
|------------------|----------------------------|-------------------------|-----------------------------|
| Small size instances |                            |                         |                             |
| 7-3              | 1237.50                    | 1019.7                  | 17.6                        |
| 7-5              | 2363.04                    | 1930.6                  | 18.3                        |
| 13-3             | 1779.70                    | 1429.1                  | 19.7                        |
| 13-5             | 3400.00                    | 2713.2                  | 20.2                        |
| Medium size instances |                            |                         |                             |
| 20-3             | 2818.08                    | 2167.1                  | 23.1                        |
|                  |                            |                         |                             |
| 20-5             | 3938.60                    | 3040.6                  | 22.8                        |
| 30-3             | 3909.40                    | 2951.6                  | 24.5                        |
| 30-5             | 6635.24                    | 4989.7                  | 24.8                        |
| 40-3             | 4625.98                    | 3437.1                  | 25.7                        |
| 40-5             | 8168.02                    | 6028                    | 26.2                        |
| Large size instances |                            |                         |                             |
| 60-3             | 7749.36                    | 5478.8                  | 29.3                        |
| 60-5             | 14186.98                   | 9916.7                  | 30.1                        |
| 80-3             | 10945.48                   | 7267.8                  | 33.6                        |
| 80-5             | 16565.84                   | 10718.1                 | 35.3                        |

From the results in Table 8, we can see that if we use the proposed freight bus system, the cost saving in percentage compared with the corresponding system without freight bus is ranged from 17.6% to 35.3% with the average cost saving 25.1%. Moreover, we can see that the larger the size of an instance, the more the cost savings of the freight bus system. The experimental results show that the system with freight bus can significantly reduce transportation costs compared with the system without freight bus.

6. Conclusions. In this paper, we have studied the route planning problem of freight buses with both pickup and delivery in an urban distribution system. At first, we have described the operations of freight buses in such a system, where each freight bus makes a tour visiting a set of pickup/delivery locations once at every given time interval in each day following a fixed route in a city, and the route planning problem can be considered a new variant of periodic vehicle routing problem with pickup and delivery. To the best of our knowledge, this problem was rarely studied in the literature.

In order to solve the problem, a Mixed-Integer Linear Programming (MILP) mathematical model is formulated and an Adaptive Large Neighborhood Search (ALNS) algorithm is proposed in this paper. In the design of the algorithm, we take account of some specific characteristics of the freight bus system, such as fixed
route for each freight bus, demand can be served in a later period but with a late service penalty cost, etc. Through numerical experiments, we have verified the mathematical model of the problem and evaluated the performance of the proposed ALNS. Moreover, we have also assessed the impact of the joint distribution realized by freight buses.

However, this paper only analyzes the economic benefits (joint distribution) of freight buses versus city freighters in terms of transportation cost reduction. To prove other potential advantages of freight buses such as schedule regularity, traffic congestion reduction, and use of fast lanes, more quantitative analysis is required. In addition, since the route planning of freight buses we study in this paper is a new variant of periodic vehicle routing problem, we could not compare our algorithms with other algorithms in the literature on benchmark instances, because such algorithms and benchmark instances do not exist. So in the future, we should either develop more heuristic algorithms, evaluate and compare them with real data, or improve the algorithms proposed in this paper. And in practice, before make the routing planning for freight buses, the delivery demand and pickup demand at each station in each time period of each day are not known, although we can obtain the probabilistic distributions of these demands by statistical analysis of their historical data. So in the future, we will also extend the model and algorithm by considering stochastic demands at each depot.

Acknowledgments. This work is supported by the French National Research Agency (ANR) under the project ANR-14-CE22-0017 entitled “Collaborative Transportation in Urban Distribution”.

REFERENCES

[1] D. Aksen, O. Kaya, F. S. Salman and Ô. Tuncel, An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem, European Journal of Operational Research, 239 (2014), 413–426.
[2] E. J. Beltrami and L. D. Bodin, Networks and vehicle routing for municipal waste collection, Networks, 4 (1974), 65–94.
[3] R. Bent and P. V. Hentenryck, A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows, Computers & Operations Research, 33 (2006), 875–893.
[4] N. Christofides and J. E. Beasley, The period routing problem, Networks, 14 (1984), 237–256.
[5] A. M. Campbell and H. W. Jill, Forty years of periodic vehicle routing, Networks, 63 (2014), 2–15.
[6] I.-M. Chao, B. L. Golden and E. Wasil, An improved heuristic for the period vehicle-routing problem, Networks, 26 (1995), 25–44.
[7] Z. Chang and H. Chen, Freight buses in three-tiered distribution systems for city logistics: Modeling and evaluation, 7th IESM Conference, (2017).
[8] J. F. Cordeau, M. Gendreau and G. Laporte, A guide to vehicle routing heuristics, Journal of the Operational Research Society, 53 (2002), 512–522.
[9] G. Clarke and J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Operations Research, 12 (1964), 568–581.
[10] B. Dai and H. X. Chen, Proportional egalitarian core solution for profit allocation games with an application to collaborative transportation planning, European Journal of Industrial Engineering, 9 (2015), 53–76.
[11] L. M. A. Drummond, L. S. Ochi and D. S. Vianna, An asynchronous parallel metaheuristic for the period vehicle routing problem, Future Generation Computer System, 17 (2001), 379–386.
[12] E. Demir, T. Bektas and G. Laporte, An adaptive large neighborhood search heuristic for the Pollution-Routing Problem, European Journal of Operational Research, 223 (2012), 346–359.
[13] B. Dai, H. X. Chen and G. K. Yang, Price-setting based combinatorial auction approach for carrier collaboration with pickup and delivery requests, Operational Research, 14 (2014), 361–386.
[14] P. Francis, K. Smilowitz and M. Tzur, The period vehicle routing problem with service choice, *Transportation Science*, 40 (2006), 439–454.
[15] L. E. Gill and R. P. Allerheiligen, Co-operation in channels of distribution: Physical distribution leads the way, *International Journal of Physical Distribution & Logistics Management*, 26 (1996), 49–63.
[16] M. Gaudioso and G. Paletta, A heuristic for the periodic vehicle routing problem, *Transport Science*, 26 (1992), 86–92.
[17] D. Goeke and M. Schneider, Routing a mixed fleet of electric and conventional vehicles, *European Journal of Operational Research*, 245 (2015), 81–99.
[18] Y. Hao and Y. Su, The research on joint distribution mode of the chain retail enterprises, *International Conference on Mechatronics, Electronic, Industrial and Control Engineering, MEIC*, (2014), 1694–1697.
[19] M. Y. Lai, H. M. Yang, S. P. Yang, J. H. Zhao and Y. Xu, Cyber-physical logistics system-based vehicle routing optimization, *Journal of Industrial and Management Optimization*, 10 (2014), 701–715.
[20] L. L. Lv, Z. Zhang, L. Zhang and W. S. Wang, An iterative algorithm for periodic Sylvester matrix equations, *Journal of Industrial and Management Optimization*, 14 (2018), 413–425.
[21] N. Labadie, R. Mansini, J. Melechovský and R. Wolfle-Calvo, The team orienteering problem with time windows: An LP-based granular variable neighborhood search, *European Journal of Operational Research*, 220 (2012), 15–27.
[22] Y. Li, H. X. Chen and C. Prins, Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests, *European Journal of Operational Research*, 252 (2016), 27–38.
[23] S. Majidi, S. M. Hosseini-Motlagh and J. Ignatius, Adaptive large neighborhood search heuristic for pollution-routing problem with simultaneous pickup and delivery, *Soft Computing*, 22 (2018), 2851–2865.
[24] A. C. Matos and R. C. Oliverira, An experimental study of the ant colony system for the period vehicle routing problem, *Lecture Notes in Computer Science*, 3172 (2004), 286–293.
[25] A. R. Pourghaderi, R. Tavakkoli-Moghaddam, M. Alinaghian and B. Beheshti-Pour, A simple and effective heuristic for periodic vehicle routing problem, *Proceedings of the 2008 IEEE International Conference on Industrial Engineering Management*, (2008), 133–137.
[26] D. Pisinger and S. Ropker, A general heuristic for vehicle routing problems, *Computers and Operations Research*, 34 (2007), 2403–2435.
[27] R. Russell and W. Igo, An assignment routing problem, *Networks*, 9 (1979), 1–17.
[28] S. Roker and D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transportation Science*, 40 (2006), 455–472.
[29] I. Roozbeh, M. Ozlen and J. W. Hearne, An Adaptive Large Neighborhood Search for asset protection during escaped wildfires, *Computers and Operations Research*, 97 (2018), 125–134.
[30] X. Shize, Introductions of joint distribution, *Circulation and economic study*, 8 (1973), 87–94.
[31] P. Shaw, Using constraint programming and local search methods to solve vehicle routing-problems, *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming Springer New York*, (1998), 417–431.
[32] A. Trentini, A. Campi, N. Malhene and F. Boscacci, Shared passengers & goods urban transport solutions: New ideas for milan through en international comparison, *Territorio*, 59 (2011), 38–44.
[33] L. Verdonck, A. Caris, K. Ramaekers and G. K. Janssens, Collaborative logistics from the perspective of road transportation companies, *Transport Review*, 33 (2013), 700–719.
[34] L. Xu and D. Yang, Research on joint distribution cost allocation model, *Boletin Tecnico/Technical Bulletin*, 55 (2017), 291–297.
[35] N. Yalaoui, L. Amodeo, F. Yalaoui and H. Mahdi, Efficient methods to schedule reentrant flow shop system, *Journal of Intelligent and Fuzzy Systems*, 26 (2014), 1113–1121.
[36] S. Zhou, Logistics bottleneck of online retail industry in China, *Journal of Supply Chain and Operations Management*, 11 (2013), 1–11.

Received November 2018; revised July 2019.

E-mail address: changzheng@itsc.cn
E-mail address: haoxun.chen@utt.fr
E-mail address: farouk.yalaoui@utt.fr
E-mail address: dr_bo_dai@163.com