AN ADAPTIVE LARGE NEIGHBORHOOD SEARCH HEURISTIC FOR MULTI-COMMODITY TWO-ECHelon VEHICLE ROUTing PROBLEM WITH SATELlITE SYNCHRONIZATION

SHENGYANG JIA\textsuperscript{1,3}, LEI DENG\textsuperscript{1,3}\textsuperscript{*}, QUANWU ZHAO\textsuperscript{2,3} AND YUNKAI CHEN\textsuperscript{1,3}

1.College of Mechanical and Vehicle Engineering, Chongqing University, 400030, China
2.School of Economics and Business Administration, Chongqing University, 400030, China
3.Chongqing Key Laboratory of Logistics, Chongqing University, Chongqing, China

(Communicated by Changzhi Wu)

ABSTRACT. In considering route optimization from multiple distribution centers called depots via some intermediate facilities called satellites to final customers with multiple commodities request, we introduce the Multi-Commodity Two-Echelon Vehicle Routing Problem with Satellite Synchronization (MC-2E-VRPSS). The MC-2E-VRPSS involves the transportation from multiple depots to satellites on the first echelon and the deliveries from satellites to final customers on the second echelon. The MC-2E-VRPSS integrates satellite synchronization constraints and time window constraints for satellites on the two-echelon network and aims to determine cost-minimizing routes for the two echelons. The satellite synchronization constraints which trucks from the multiple depots to some satellites need to be coordinated guarantee the efficiency of the second echelon network. In this study, we develop a mixed-integer programming model for the MC-2E-VRPSS. For validating the model formulation, we conduct the computational experiments on a set of small-scale instances using GUROBI and an adaptive large neighborhood search (ALNS) heuristic which we develop for the problem. Furthermore, the computation experiments for evaluating the applicability of the ALNS heuristic compared with large neighborhood search (LNS) on a set of large-scale instances are also conducted, which proved the effectiveness of the ALNS.

1. Introduction. With the rapid development of China’s e-commerce industry the logistics demand, especially express service, has experienced explosive growth. The total number of packages in China delivered during 2020 was 83.4 billion increasing by 31.2\% compared to 2019. For an urban city, large amounts of freight from all over the country are transported to it via long-haul tracks. However, deliver the freight to the final customers requires some more innovative, effective, and environmentally-friendly ways due to traffic congestion, environmental issues\cite{5}. Thus, the last mile distribution plays an important role in serving the final customers, which the two-echelon distribution systems have introduced the effective way to execute it \cite{11}.

In the two-echelon distribution system, goods are first delivered from a central depot to a set of intermediate facilities called satellites where consolidation and transshipment operations take place. Those goods are then delivered from the
satellites to their destination. However, the routing problems arising in the network cannot be directly solved by separating it into two sub-problems because the flow of freight in one echelon must be coordinated with that in the other echelon. This problem is known in the literature as the two-echelon vehicle routing problem (2E-VRP).

The two-echelon vehicle routing problem (2E-VRP) consists of delivering goods from a single central depot to customers through a set of satellites. The depot is an intermodal logistics facility called the distribution center (DC). It consolidates all goods delivered to customers serviced by the distribution center. Satellites have limited storage capacity but are located closer to customers. Two fleets of vehicles are involved: first-level vehicles (FLVs) carry goods from the DC to the satellites, and second-level vehicles (SLVs) carry goods from the satellites to customers. FLVs are usually significantly larger than SLVs. The objective is to route vehicles operating at both levels to service customers with minimizing the total routing cost of the system.

In practice, for timeliness and efficiency, many logistics suppliers have more than one distribution center in an urban city to serve customers better. For example, one express company has several distribution centers in an urban city, such as SF, JD, and China Post, etc., each distribution center focuses on the operation of a kind of commodity such that the two-echelon distribution network is more efficient. In recent years, the concept of joint distribution has been discussed in the industry, but the application of the concept hasn’t been performed well. There are a lot of reasons for that, one of the reasons is that each cooperative company have not a great coordination relationship with each other about the use of the network. Formally, each cooperative company has its own distribution centers in the urban city, but all the satellites can be used by every company. Each satellite has limited resources such as vehicles, staff, equipment, etc. Thus, from the operation perspective, the problem is how to use the limited resource of the network to meet the demand of customers and minimize the cost of the network.

In this paper, we address a new variant of 2E-VEP called the Multi-Commodity Two-Echelon Vehicle Routing Problem with Satellite Synchronization (MC-2E-VRPSS). In the problem, each customer has two commodities request from two distinct depots. The deliveries to the customers from the depots are transferred at the satellites. There exists a homogenous fleet of vehicles, which the capacity is smaller than FLVs, can be used by the satellites and haven’t been assigned to any satellite. Each depot has a homogenous fleet of vehicles we call both the vehicles as FLVs.

In the two-echelon distribution network, the first echelon consists of vehicle routes that start and end at each depot and transfer customers’ requests at a subset of satellites. Moreover, each satellite has a hard time window for the FLVs. Because the customers serviced by a satellite have two commodities demand must be met by the two distinct depots, the arrival time of the vehicles depart from the two different depots possibly have a gap. There are three results of the gap if it happens. Before we explain the results, we define the arrival and departure time of the FLVs at satellite \( j \) depart from the depot \( o_1, o_2 \) as \( at_{o_1}^j, dt_{o_1}^j \) and \( at_{o_2}^j, dt_{o_2}^j \). Each FLV needs to spend time to unload goods at the satellite, which is related to the number of freights to be unloaded, so we defined the unloading time as \( ut_{o_1}^j, ut_{o_2}^j \). Thus, the departure time of the FLVs \( dt_{o_1}^j = at_{o_1}^j + ut_{o_1}^j, dt_{o_2}^j = at_{o_2}^j + ut_{o_2}^j \) respectively. One of the results is \( dt_{o_1}^j < at_{o_2}^j \) or \( at_{o_1}^j > dt_{o_2}^j \) shown in Fig 1. (c), which means the satellite has a penalty time for the late FLV. The result incurs that the freights
from the first arriving FLV will be delayed until idle SLVs at the satellite, which is unacceptable in actual operation. Or assigning some of the SLVs to deliver the freights for the timeliness guarantee. However, the freights from the late FLV also need to be delivered by the SLVs, which means that the satellite needs more SLVs to finish its job while the number of the SLVs is limit. Hence we need to coordinate the arrival time of the FLVs at the satellites, which we called this coordination as soft synchronization. We defined this gap as penalty time. The other two results both have no negative influence on the operation of satellites. Fig 1.(a) and (b) present the other two results that the satellite has no penalty time for any vehicle.

![Figure 1. Three results of the gap](image)

On the second echelon, the SLVs loading the two commodities goods together at the satellite and deliver them to customers. We assumed that only both the two FLVs have finished the unloading operation the SLVs start loading the goods delivered by them and the two commodities goods of each customer must be met by just one SLV together with timeliness restriction.

To the best of our knowledge, the MC-2E-VRPSS hasn’t been proposed. We have developed a mixed-integer programming model for the problem. To solve the problem, an adaptive large neighborhood search heuristic is proposed. To test our algorithm, the computational experiments with the heuristic and GUROBI solver are conducted on small-scale instances. The contributions of this paper are as follows: First, the MC-2E-VRPSS integrates satellite soft synchronization, time windows constraints on the satellites, and timeliness constraints on the customers. Second, we proposed a mixed-integer programming model for the MC-2E-VRPSS which can be directly solved by the GUROBI solver. Third, we provide a heuristic algorithm to solve the MC-2E-VRPSS and also can be used for solving the 2E-VRPTW which hasn’t been solved with heuristics. Finally, we experimentally evaluate the validation of the MC-2E-VRPSS formulation and the effectiveness of the proposed algorithm for various instances.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature. In Section 3, we formally describe the problem and introduce the formulation of MC-2E-VRPSS. The proposed algorithm is detailed in section 4. Computational experiments are conducted and the results are reported and analyzed in Section 5, followed by conclusions and suggestions for future research in Section 6.
2. Related work. To the best of our knowledge, the MC-2E-VRPSS has never been addressed in the literature. However, many closely related problems have been proposed and studied, which include multi-commodity vehicle routing problem, vehicle routing problem with synchronization constraint, two-echelon vehicle routing problem (2E-VRP), and its variants such as two-echelon vehicle routing problem with satellite synchronization (2E-VRPSS) and two-echelon vehicle routing problem with time windows (2E-VRPTW).

The multi-commodity vehicle routing problem consists of delivering multiple commodities from a depot or multiple depots to a set of final customers, while the commodities are incompatible which cannot be transported in the same vehicle [6]. Jörn Schönberger[34] presented the problem of having to offer two different products to a group of customers. Specifically, each commodity is delivered by a particular type of vehicle. More recently, Dellaert[14] studies a multi-commodity two-echelon capacitated vehicle routing problem with time windows for customers and proposed an exact solution approach for the problem. In particular, they take the customer-specific demand into account which has non-substitute demand.

The MC-2E-VRPSS also relates to vehicle routing problems with synchronization constraints because the arrival time of the FLVs at a satellite needs to be coordinated. Rousseau et al.[31, 30] studied a synchronized VRP with time windows for customers, in which the customers need to be visited twice such that the two visits are imposed with precedence constraints. Eveborn et al.[18] firstly incorporated synchronization constraints into the Home Health Care (HHC) caregivers scheduling and routing model. The synchronization as defined in this study requires two caregivers to visit and serve some patients at the same time. Similarly, the synchronization constraint in our problem is coordinating the visit of FLVs from the two depots to satellites, but the visit may not occur at exactly the same time, which we defined the coordination as soft synchronization. Bredström and Rönqvist[3] extended the synchronization constraints to temporal precedence constraints between customer visits. In the paper, the mathematical formulation of the problem is given, and a heuristic algorithm based on optimization is proposed. Dohn et al.[15] also proposed a branch-and-price framework to solve the VRP with synchronization constraints. Rabeh et al.[28] addressed the planning and scheduling of two caregivers in the context of HHC services, where a patient may require two caregivers to visit separately, but in a predetermined order. The problem is formulated as a mixed-integer programming model and solved by the Lingo solver. Dohn et al.[16] proposed a generalized vehicle routing problem with time windows and synchronization, VRPTWSyn, which is called the vehicle routing problem with time windows and temporal dependencies. In addition to the standard synchronization, more general temporal dependencies are considered, such as the maximum and minimum difference between the starting time or ending time of two visits. Afifi et al.[1] also studied the VRPTWSyn defined in the paper. The authors proposed a Simulated Annealing (SA) algorithm to the problem and test their algorithm on the standard instances of Bredström and Rönqvist[3]. More recently, Minh Hoàng Hà a et al.[20] proposed a new constraint programming (CP) model for VRPSyn and an ALNS based on solving linear programming for the design of insertion operators. Drexl[17] presents an exhaustive survey of applications of VRP with various synchronization constraints.

The 2E-VRP as the main framework of the MC-2E-VRPSS relates to our works. The 2E-VRP was officially introduced by Perboli and Tadei[25], who proposed a
AN ADAPTIVE LARGE NEIGHBORHOOD SEARCH HEURISTIC

mathematical model. Since then several algorithms have been developed: math-based heuristics[25, 26] , greedy randomized adaptive search procedure (GRASP)[8, 9, 36] , adaptive large neighborhood search (ALNS)[21] , and a large neighborhood search combined with a local search[4]. Exact methods include [22, 32, 33]. Crainic, Perboli, Mancini, and Tadei[10] study the impact of satellite location on the cost of a 2E-VRP solution compared to that of a VRP. Cuda, Guastaroba, and Speranza[12] published a survey on two-echelon routing problems.

The 2E-VRP variants considering satellite synchronization constraints are challenging owing to the interdependency among the routes on different echelons and have received relatively little attention. In the 2E-VRPSS literature, the synchronization constraints are different from the soft synchronization. The synchronization constraints that arose in 2E-VRP in the literature usually indicate the coordination between first-level vehicle and second-level vehicle at a satellite, whereas the soft synchronization indicates the coordination of the two first-level vehicles depart from the two different depots at a satellite. To our knowledge, the kind of synchronization constraint has never been proposed in the 2E-VRP context. Grangier, Gendreau, Lehuéde, and Rousseau[19] presented a two-echelon multiple-trip VRP with satellite synchronization that considered synchronization constraints and multiple trips at the second echelon. Li, Wang, Chen, Bai[23] presented a two-echelon vehicle routing problem with satellite bi-synchronization involving the inter-satellite line-haul on the first echelon. The authors considered pairwise synchronization constraints in the pickup and delivery stage respectively. Anderluh, Larsen, Hemmelmayr, Nolz[2] considered a 2E-VRP with spatial and temporal synchronization and proposed a two-stage greedy randomized adaptive search procedure with path relinking.

The 2E-VRPTW also relates to our problem since the customers have a timeliness constraint. Dellaert, Saridarq, Van Woensel, Crainic[13] proposed a branch-and-price-based algorithm for the 2E-VRPTW, while each satellite also has a time window for the first-level vehicles in our problem. The same authors[14] presented a multi-commodity two-echelon capacitated vehicle routing problem with time windows. They have proposed a number of model formulations for the problem and developed a branch-and-price-based solution approach for the real-life problem. To the best of our knowledge, there is no heuristics to solve the 2E-VRPTW, so the ALNS we proposed may filled the gap and attracts more interest to develop more effective heuristics to solve the problem.

3. Problem formulation.

3.1. Problem description. We introduce the multi-commodity two-echelon vehicle routing problem with satellite synchronization (MC-2E-VRPSS). We consider two distribution centers called depots $V_D = \{o_1, o_2\}$, a set of satellites $V_s$, a set of customers $V_c$, and two heterogeneous fleets of vehicles $K_1$ and $K_2$ of capacity $Q_1$ and $Q_2$, located at depots and satellites respectively, but they are homogenous at each echelon. The MC-2E-VRPSS is defined on a directed graph $G = (V, A)$, where $V$ is the set of nodes and $A$ is the set of arcs. The first level is defined by $G_1 = (V_1, A_1)$ with $V_1 = V_D \cup V_S$ and $A_1 = \{(o_1, i) \mid i \in V_s\} \cup \{(o_2, i) \mid i \in V_s\} \cup \{(i, j) \mid i, j \in V_s, i \neq j\} \cup \{(i, o_1) \mid i \in V_s\} \cup \{(i, o_2) \mid i \in V_s\}$. The second level is defined by $G_2 = (V_2, A_2)$ with $V_2 = V_s \cup V_c$ and $A_2 = \{(i, j) \mid i \in V_s, j \in V_c\} \cup \{(i, j) \mid i, j \in V_c, i \neq j\} \cup \{(i, j) \mid i \in V_c, j \in V_s\}$. With each arc $(i, j) \in A = A_1 \cup A_2$ is associated a travel time $t_{ij}$ and a travel cost $c_{ij}$.
On the first echelon, each depot has a homogeneous fleet of trucks \( K_1^{o_1}, K_1^{o_2} \) and \( K_1 = K_1^{o_1} \cup K_1^{o_2} \). FLVs travel from the depot to satellite locations and return to the same depot. Each satellite has a hard time window \([c_s, l_s]\) for FLVs that has to be respected. Different from many of the 2E-VRP literature, we assumed that the demand of satellites cannot be split. On the second echelon, each SLV can be assigned to any satellite, travels from the satellite location, and return to the same location. FLVs and SLVs have a quantity limit \( m_1^h, m_2^h, h \in V_D \) respectively and a working duration time limit \( t_1^1, t_2^2 \) respectively.

Each \( i \in V_c \) customer has a deadline \( l_i \) and two commodities demand \( d_i^h, h \in V_D \) from two distinct depots, which cannot be split, and must be met simultaneously by just one SLV. The demand cannot be delivered by direct shipping from the depot but must be consolidated in a satellite. All goods are located at the depots at the beginning of the time horizon.

Fig 2 illustrates a multi-commodity two-echelon network that includes two depots where the two commodities are located, three satellites, and customers. The FLVs transport the two commodities from the two distinct depots to the satellites. After consolidation and transfer at satellites, the two commodities are delivered to final customers. Each customer has a demand of two commodities and we assumed that the demand must be satisfied together by just one vehicle, which means that the vehicle must simultaneously load the two commodities freight for the customer. Thus, the SLVs depart from satellites only if it has loaded the two commodities freight delivered by it. Moreover, for the convenience of modeling, we assumed that the SLVs start to load the frights only after the two FLVs have unloaded the two commodities freight.

![Figure 2. An illustrative example of the MC-2E-VRPSS](image-url)
3.2. **Mathematical formulation.** We develop a mixed-integer programming formulation for the MC-2E-VRPSS. The objective of the MC-2E-VRPSS is to find the cost-minimizing two echelons routes of vehicles. The notation used in the model has presented in Table 1.

### Table 1. Sets, parameters, and variables

| Notation | Description |
|----------|-------------|
| **Sets** | |
| $V_D$ | The set of depots, $V_D = \{o_1, o_2\}$ |
| $V_s$ | The set of satellites |
| $V_c$ | The set of customers |
| $K_1^h$ | The set of first echelon vehicles belong to the depot $h \in V_D$ |
| $K_1$ | The set of first echelon vehicles, $K_1 = K_1^{o_1} \cup K_1^{o_2}$ |
| $K_2$ | The set of second echelon vehicles |
| **Parameters** | |
| $m_1^h$ | The quantity of available first echelon vehicles belong to depot $h \in V_D$ |
| $m_2$ | The quantity of available second echelon vehicles |
| $Q_1$ | The capacity of the first level vehicle |
| $Q_2$ | The capacity of the second level vehicle |
| $Q_S$ | The capacity of satellite $S$ accommodating goods from depot $h \in V_D$ |
| $p_{S}$ | The unsynchronized penalty cost per unit time at satellite $S$ |
| $c_{i,j}$ | The arc cost of arc $(i,j)$ |
| $t_{s}^{i,j}$ | The travel time of arc $(i,j)$ |
| $d_l^h$ | The demand from the depot $h \in V_D$ of customer $i$ |
| $w$ | The time of loading or unloading at a satellite per unit |
| $u$ | The service time of the second level vehicle at a customer per unit |
| $t_{1h}$ | The maximum working duration time for first echelon vehicles of depot $h$ |
| $t_{2s}$ | The maximum working duration time for second echelon vehicles of satellite $s$ |
| $e_{i,s,l}$ | The time window of satellite $s$ |
| $l_{ij}$ | The deadline for customer $j$ |
| $M$ | A sufficient large positive number |
| **Variables** | |
| $y_{1h}^{i}$ | The load from the depot $h$ of the first level vehicle $k$ after visiting node $i$ |
| $y_{2s}^{i}$ | The load from the depot $h$ of the second level vehicle $k$ after visiting node $i$ |
| $w_{S}$ | The punishment time at the satellite $S$ |
| $D_S^h$ | The demand from the depot $h$ of satellite $S$ |
| $t_{1h}^{i}$ | The departure time of first level vehicle $k$ at node $i$ |
| $t_{2h}^{i}$ | The arrival time of first level vehicle $k$ at node $i$ |
| $t_{2h}^{i}$ | The departure time of second level vehicle $k$ at node $i$ |
| $t_{2h}^{i}$ | The arrival time of second level vehicle $k$ at node $i$ |
| $z_{j}$ | Binary variable that takes the value 1 only if the customer $j$ assigned to satellite $s$ |
| $x_{ij}^{i}$ | Binary variable that takes the value 1 only if the first echelon vehicle $k$ travels from $i$ to $j$ |
| $y_{ij}^{i}$ | Binary variable that takes the value 1 only if the second echelon vehicle $k$ travels from $i$ to $j$ |

The model to minimize the total cost of the MD-2E-VRPSS is formally formulated as follows:

\[
\begin{align}
\text{Min} & \sum_{k \in K^1} \sum_{i,j \in V_D \cup V_s, i \neq j} c_{ij} x_{ij}^{k} + \sum_{k \in K^2} \sum_{j \in V_s \cup V_c, i \neq j} c_{ij} y_{ij}^{k} + \sum_{s \in V_s} p_s w_{t_s} \\
\text{Subject to:} & \sum_{k \in K^1} x_{ij}^{k} \leq m_{1h}^i, \quad \forall h \in V_D \\
& \sum_{k \in K^2} y_{ij}^{k} \leq m^2 \\
& \sum_{i \in h \cup V_s, i \neq j} x_{ij}^{k} = \sum_{t \in h \cup V_s, j \neq t} x_{jt}^{k}, \forall j \in V_s, k \in K^1, h \in V_D
\end{align}
\]
\[ \sum_{i \in V_s \cup V_c, i \neq j} y_{ij}^k = \sum_{i \in V_s \cup V_c, i \neq j} y_{ij}^k, \forall j \in V_c, k \in K^2 \quad (5) \]
\[ q_{jk}^{kh} \leq Q^1, \forall k \in K^{1h}, h \in V_D \quad (6) \]
\[ \sum_{h \in V_D} q_{sk}^{kh} \leq Q^2, \forall k \in K^2, s \in V_s \quad (7) \]
\[ D_{sh}^h \leq Q_s^h, \forall s \in V_s, h \in V_D \quad (8) \]
\[ \sum_{i \in V_s} z_{ij} = 1, \forall j \in V_c \quad (9) \]
\[ \sum_{k \in K^2} \sum_{i \in V_s \cup V_c, i \neq j} y_{ij}^k = 1, \forall j \in V_c \quad (10) \]
\[ y_{ij}^k + y_{ij}^k \leq 1, \forall i, j \in V_c, i \neq j, \forall k \in K^2 \quad (11) \]
\[ \sum_{k \in K^{1h}} \sum_{i \in h \cup V_c, i \neq j} x_{ij}^k \leq M \sum_{i \in V_s} z_{ij}, \forall j \in V_s, h \in V_D \quad (12) \]
\[ M \cdot \sum_{k \in K^{1h}} \sum_{i \in h \cup V_c, i \neq j} x_{ij}^k \geq \sum_{i \in V_s} z_{ij}, \forall j \in V_s, h \in V_D \quad (13) \]
\[ \sum_{k \in K^{1h}} \sum_{i \in h \cup V_c, i \neq j} x_{ij}^k \leq 1, \forall j \in V_s, h \in V_D \quad (14) \]
\[ D_{sh}^h = \sum_{j \in V_c} z_{sj} d_{hj}^h, \forall s \in V_s, h \in V_D \quad (15) \]
\[ \sum_{j \in V_s} x_{hj}^k = \sum_{j \in V_s} x_{hj}^k, \forall k \in K^{1h}, h \in V_D \quad (16) \]
\[ \sum_{j \in V_s} y_{hj}^k = \sum_{j \in V_s} y_{hj}^k, \forall s \in V_s, k \in K^2 \quad (17) \]
\[ at_{1h}^k \leq l_{1h} + M \left( 1 - \sum_{j \in V_s} x_{hj}^k \right), \forall k \in K^{1h}, h \in V_D \quad (18) \]
\[ \epsilon_s - M \left( 1 - \sum_{i \in h \cup V_c} x_{ia}^k \right) \leq at_{1s}^k \leq l_s + M \left( 1 - \sum_{i \in h \cup V_c} x_{ia}^k \right), \forall k \in K^{1h}, s \in V_s, h \in V_D \quad (19) \]
\[ at_{2s}^k \leq l^2_s + M \left( 1 - \sum_{j \in V_c} y_{sj}^k \right), \forall k \in K^2, s \in V_s \quad (20) \]
\[ at_{2k}^k \leq l_j + M \left( 1 - \sum_{i \in V_s \cup V_c, i \neq j} y_{ij}^k \right), \forall k \in K^2, j \in V_c \quad (21) \]
\[ dt_{1j}^k + t_{ij} - M \left( 1 - x_{ij}^k \right) \leq at_{1j}^k, \forall i, j \in h \cup V_c, k \in K^{1h}, h \in V_D \quad (22) \]
\[ at_{1j}^k + wD_{ij}^h - M \left( 1 - \sum_{i \in h \cup V_c, i \neq j} x_{ij}^k \right) \leq dt_{1j}^k, \forall j \in V_s, k \in K^{1h}, h \in V_D \quad (23) \]
\[ dt_{2j}^k + t_{ij} - M \left( 1 - y_{ij}^k \right) \leq at_{2j}^k, \forall i, j \in V_s \cup V_c, k \in K^2 \quad (24) \]
\[ at_{2j}^k + s \sum_{h \in V_D} d_{hj}^k - M \left( 1 - \sum_{i \in V_s \cup V_c} y_{ij}^k \right) \leq dt_{2j}^k, \forall j \in V_c, k \in K^2 \quad (25) \]
\[
\begin{align*}
\sum_{h \in V_D} q_{ij}^{kh} - M \left( 2 - \sum_{i \in V_s \cup V_c} x_{ij}^{kh} - \sum_{t \in V_c} y_{jt}^{k} \right) & \leq dt_{ij}^{2k}, \\
\forall j \in V_s, k \in K^2, k_{o1} \in K^{1_{s1}} \\
\sum_{h \in V_D} q_{ij}^{kh} - M \left( \sum_{i \in V_s \cup V_c} y_{ij}^{k} + z_{sj} - 2 \right) & \leq y_{st}^k, \forall s \in V_s, j \in V_c, k \in K^2 \\
\sum_{j \in V_c} y_{sj}^k & \geq M \left( \sum_{i \in V_s \cup V_c} y_{ij}^k + z_{sj} - 2 \right) + 1, \forall s \in V_s, j \in V_c, k \in K^2 \\
q_{ij}^{2kh} - d_{ij}^{h} + M \left( 1 - y_{ij}^k \right) & \geq q_{ij}^{kh}, \forall k \in K^2, \forall i \in V_s \cup V_c, \forall j \in V_c, h \in V_D \\
q_{ij}^{2kh} - d_{ij}^{h} - M \left( 1 - y_{ij}^k \right) & \leq q_{ij}^{kh}, \forall k \in K^2, \forall i \in V_s \cup V_c, j \in V_c, h \in V_D \\
\sum_{h \in V_D} q_{ij}^{kh} - M \left( 1 - y_{ij}^k \right) & \leq 0, \forall k \in K^2, \forall i \in V_c, j \in V_s \\
q_{ij}^{kh} - d_{ij}^{h} - M \left( 1 - x_{ij}^k \right) & \leq q_{ij}^{kh}, \forall k \in K^{1h}, \forall i \in V_s \cup h, j \in V_s, h \in V_D \\
q_{ij}^{kh} - d_{ij}^{h} + M \left( 1 - x_{ij}^k \right) & \geq q_{ij}^{kh}, \forall k \in K^{1h}, \forall i \in V_s \cup h, j \in V_s, h \in V_D \\
q_{ij}^{kh} - M \left( 1 - x_{ij}^k \right) & \leq 0, \forall k \in K^{1h}, \forall i \in V_s, h \in V_D \\
x_{ij}^k & \in \{0,1\}, \forall i, j \in V_D \cup V_s, k \in K_1 \\
y_{sj}^k & \in \{0,1\}, \forall i, j \in V_s \cup V_c, k \in K_2 \\
z_{sj} & \in \{0,1\}, \forall j \in V_c, s \in V_s
\end{align*}
\]

The objective function (1) includes three parts: routes cost of FLVs; routes cost of SLVs; the penalty cost owe to the non-synchronization of the two FLVs at satellites.

Constraints (2) and (3) ensure that the number of routes at each level must not exceed the number of vehicles for that level. Constraints (4) and (5) are flow conservation constraints for the FLVs and SLVs respectively. Constraints (6) and (7) ensure that the capacity of FLVs and SLVs are satisfied. The limit on the satellite capacity of the two commodities freight is satisfied by constraint (8). Constraint (9) assigns each customer to only one satellite, while constraints (10) and (11) indicate that there is only one 2nd-level route passing through each customer. Constraints (12)-(14) indicate that the FLVs departed from each depot can only visit the satellite once if there is a demand for it, otherwise the visit is forbidden. Constraint (15) indicates the two commodities demand of satellites. Constraint (16) ensure that if the FLV is used, it should depart from its depot once and ultimately return to the same depot, while the same restriction for SLVs is satisfied by constraint (17). Constraint (18) and (20) respects the working duration for FLVs and SLVs respectively.
Constraints (19) indicate the time window restriction for FLVs visiting a satellite. Constraint (21) ensure that each customer’s deadline is satisfied. Constraints (22) and (23) indicate the departure and arrival time of a FLV at its depot and the satellites visited by it. Constraints (24)-(27) indicate the departure and arrival time of a SLV at its satellite and the customers visited by it. Constraints (28)-(30) indicate the punishment time at the satellite. Constraints (31) and (32) ensure that a SLV departs from a satellite to deliver goods to a customer if the customer is assigned to that satellite and only in such a case. Constraints (33) and (34) indicate the load of a SLV, while constraint (35) indicates that the load of the SLV has to equal 0 when it returns to its satellite. Constraints (36) and (37) indicate the load of a FLV, while constraint (38) indicates that the load of the FLV has to equal 0 when it returns to its depot.

Finally, (39)-(41) specify the domains of the variables.

4. Solution method. From the model constitution perspective, the MC-2E-VRPSS is made up of two single-depot two-echelon vehicle routing problems with time windows, which are both NP-Hard problems and interact with each other. So the formulation for the MC-2E-VRPSS is unlikely to develop an exact method for large-scale instances to solve it. To solve the MC-2E-VRPSS, we develop the adaptive large neighborhood search (ALNS) heuristics which has been used for the two-echelon capacitated vehicle routing problem (2E-CVRP), which is the special case of the MC-2E-VRPSS, and obtained some excellent results[21].

Ropke, Pisinger[29] proposed ALNS based on extending the large neighborhood search framework of Shaw[35]. ALNS starts with an initial solution and iteratively improves the quality of the current solution by using a pair of operators (the destroy and repair operators). For solving the routing problem, the destroy operator removes a part of customers usually q, and then the repair operator reinserts the removed customers into the destroyed solution to rebuild a new solution.

We refer to Ropke, Pisinger[29] and Pisinger, Ropke[27] for a more detailed introduction of ALNS. For the MC-2E-VRPSS, the current solution changes a lot when the customers are removed or reinserted. In other words, when we modify the structure of the second echelon the solution may be infeasible and the penalty time may be varied. For example, when we insert a customer into a 2nd-level route starting from a satellite the unloading time of the two FLVs at the satellite will increase, which causes that the departure time of the two FLVs and all the SLVs at the satellite to be delayed. In this case, the deadline for all the customers in the satellite and time window for the FLVs in their remaining route may be violated. Furthermore, the deadline for the customers serviced by the satellites in the remaining influenced 1st-level route may also be violated. Apparently, because the arrival time gap has been varied the penalty time will increase or decrease. The problem is defined as the interaction problem in the literature. Considering the difficulty of solving the MC-2E-VRPSS, we design a special structure for ALNS to tackle the interaction.

The general outline of the ALNS procedure is described in Algorithm 1. The heuristic starts from an initial solution s. Next, the second-echelon destroy and repair operators are chosen and applied to create a temporary solution. Then, the reconstruction and improvement for the 1st-level routes of the temporary solution are conducted by the reconstruction and improvement heuristics together. A new solution is accepted based on simulated annealing criteria. We update the scores,
which are used for quantifying the performance of the operators in the last period, of the second-echelon destroy and repair operators based on their last performance. The score of an operator is increased by either $\sigma_1$, $\sigma_2$ or $\sigma_3$. The weights of the destroy and repair operators are adjusted after a number of iterations called a segment that we set as 200 iterations. The score of all the operators is set to zero at the start of each segment. The formula of the weights adjustment is followed by Ropke, Pisinger[29].

In the following, we describe the components of our method, namely the construction of the initial solution, the second-echelon destroy and repair operators, reconstruction and improvement heuristics in detail.

Algorithm 1: ALNS heuristic for the MC-2E-VRPSS

1. Construct an initial solution $s$; set $S_{best}, S_{cur} := S$
2. while the stopping criterion is not met do
3. Choose a second-echelon destroy operator and a second-echelon repair operator using roulette wheel selection based on previously obtained scores;
4. Generate a temporary solution from using the two operators;
5. Reconstruct the 1st-level routes of the $s'$;
6. Improve the 1st-level routes using the pair of heuristics and generate a new solution;
7. if $S_{new}$ can be accepted then
8. if $f(S_{new}) < f(S_{best})$ then
9. Set $S_{best} = S_{new}, S_{cur} = S_{new}$;
10. end
11. else
12. Set $S_{cur} = S_{new}$
13. end
14. end
15. Update scores $\pi_i$ of the destroy and repair operators
16. end
17. Return $S_{best}$

4.1. Initial solution. In our problem, all customers have not to be assigned to any satellite in advance. To construct an initial solution, every customer is first assigned to a satellite by a greedy insertion procedure based on the distance to the customer. After that, the two commodities demand of satellites is known, the two VRPTWs for the first echelon are solved by means of the savings algorithm proposed by Clarke, Wright[7]. Finally, with the given arrival time of FLVs at satellites, the VRPs for the second echelon are solved by means of the savings algorithm again.

4.2. Second-echelon destroy operators. In the following, we describe the second-echelon destroy operators used in the ALNS algorithm. All of them are used in the destruction of the 2nd level routes and have been used in the[29, 24, 21], and we just slightly modified some of them to suit our problem. In each destroy operator, $q$ customers are removed from the current solution. The number $q$ is important for performance of the ALNS, thus we have performed some preliminary tests to set properly the parameter.
4.2.1. Random removal. The random operator randomly chooses $q$ customers and puts them into the customer pool, then removes them from the incumbent solution.

4.2.2. Worst removal. The worst removal method removes the $q$ customers with the greatest savings. More precisely, the savings are defined as the difference between the cost when the customer is in the solution and the cost when it is removed. In particular, we referred to the cost in the process of the operator as 2nd-level route cost. In our problem, the time of the SLV arriving and departing satellites will be changed when the demand of the satellite is changed, which influences the soft synchronization at those are visited by the SLV. Hence, we just consider the 2nd-level route cost in the operator, while taking the time aspect into account in the reconstruction heuristic. In addition, this worst removal method has a parameter $p > 1$ that determines the degree of randomization to avoid situations where the same customer is removed over and over again.

4.2.3. Related removal. The related removal method removes some customers that are similar to each other. The relatedness measure of two customers $i$ and $j$ in this paper depends on the distance and their two commodities demand. The relatedness measure is given by $R(i, j) = \varphi d_{i,j} + \lambda \left( \theta |q_{1,i}^1 - q_{1,j}^1| + (1 - \theta) |q_{2,i}^1 - q_{2,j}^1| \right)$. The higher $R(i, j)$ is, the less related are the two customers; $d_{i,j}$ is the travel distance between the customers $i$ and $j$; $q_{1,i}^1$ and $q_{2,i}^1$ denote the first and second commodity demand of customer respectively; $\varphi$, $\lambda$ and $\theta$ are weight parameters. The key idea of the measure is that we want to reschedule the closed customers with little demand difference to get a better reschedule plan. In our problem, the customers are distributed in every district of the city and they have been serviced by the satellites with empirical planning. Thus, the rescheduling of the customers possibly improves the quality of the solution. The outline of the related removal method is shown in Algorithm 2. The operator firstly selects a seed customer and puts it into the customer pool. Then, the related removal method randomly selects the next seed customer from the customer pool and ranks the relatedness of the customers not in the customer pool. After that, we select the highest relatedness customer from the ranking sequence and put it into the customer pool until the $q$ customers have been selected. We also introduce the parameter $p > 1$ into the operator for the same reason in the worst removal operator.

**Algorithm 2:** Related removal

1. Randomly select a seed customer $i$ and put it into the customer pool $D$;
2. while $|D| < q$ do
   3. Randomly select a seed customer $j$ from $D$;
   4. Generate the ranking sequence of the customers not in the pool $L$ by calculating the relatedness;
   5. Sort $L$ in ascending order;
   6. Choose a random number $y$ from the interval $[0, 1)$;
   7. Select the customer $L[y^p|L|]$ and put it into $D$
9. Remove all the customers in $D$ from the current solution
4.2.4. Route removal. The route removal method randomly selects a 2nd-level route and removes all the customers in the route from the current solution and puts them into the customer pool. When the number of customers in the customers pool less than $q$, an additional route is randomly selected and we execute the same operation for the customers in the route. This procedure is repeated until the number of removed customers is greater than or equal to $q$.

4.3. Second-echelon repair operators. We present two second-echelon repair operators in ALNS. After a removal operator is applied, the requests of the customers in the customer pool haven’t been satisfied yet. So we need to reinsert them into the destroyed solution. Moreover, the 1st-level routes are possibly infeasible, which in the case that some satellites have no demand after the destroy operators have removed all customers serviced by the satellite, but we save the correction of 1st-level routes for the reconstruction of the 1st-level routes.

4.3.1. Greedy insertion. The greedy insertion method performs $q$ iterations because it inserts one customer for each iteration. The operator inserts one customer into the position that minimizes the insertion cost. Every time we try to insert one customer in the customer pool into any position the demand of the satellite will be changed, the departure time of SLVs will be increased because the unloading time of the FLVs will be increased. Hence, the insertion may be infeasible but we do not check the feasibility in the temporal aspect, we simply checked the capacity constraint of the satellite and SLV and calculated the cost difference of the 2nd-level route including the customer. We also try to create a new route in all satellites for one customer if it is feasible and calculate the cost difference. In each iteration we choose the customer that the insertion cost is minimum, then we insert it into the minimum cost position referred to Ropke, Pisinger [29] and remove it from the customer pool. This procedure is repeated until all the customers in the customer pool are inserted into the incumbent solution.

4.3.2. Regret insertion. In the operator, a customer is reinserted based on the regret value. The regret value of a customer is the cost difference between the best insertion and the second best. That is to say, the operator selects the insertion that will be regretted mostly if it is not inserted currently. More precisely, a regret-$k$ heuristic chooses to insert customer $i$ among the set $D$ of customer pool according to

$$i := \arg \max_{i \in D} \left( \sum_{h=2}^{k} \Delta f^h_i - \Delta f^1_i \right),$$

where $\Delta f^h_i$ is the cost of inserting customer $i$ at the $h$th best position. In this paper, we use the regret-3 heuristic. The difference with the literature exists in how to choose the reinsertion position. For one customer, e.g., $i$, we create a new 2nd-level route in each satellite for the customer and we consider all the feasible positions that just satisfy the capacity restriction of SLVs and satellites. If there are multiple feasible positions for customer $i$, the cost difference between its best and second best insertion is estimated as the regret value. If there is only one feasible position, which the position must belong to a new route, the regret value is estimated as a sufficiently small number just like the setting in Li, Wang, Chen Bai [23]. Like the greedy insertion, this operator also does not compute the 1st-level cost and the penalty cost change, but we focus on the 2nd-level route insertion cost only, which includes the customer. Finally, the customer with the maximum regret value is inserted in the minimum cost position.
4.4. **Reconstruction of the 1st-level routes.** After the application of a pair of second-echelon removal and repair operators, the 2nd-level routes have been planned but the 1st-level routes possibly infeasible. For example, a satellite visited by the FLVs has already no demand because the customers that used to be assigned to it have been reinserted to the routes of the other satellites. For the reconstruction of the 1st-level routes, we propose a simple procedure that generates a giant tour containing all the satellites with demand in a greedy way and then we split the giant tour into a set of routes with no violation of the constraints.

The process of the reconstruction for the 1st-level routes starting from the two depots is the same, so we just describe the steps of the procedure for one of the two depots. The steps of the procedure are as follows:

**Step 1.** Insert the satellite that is farthest away from the depot into an empty tour \( R_1 \).

**Step 2.** Calculate the cost difference of the remaining unplanned satellites when it is inserted into its cheapest position in \( R_1 \).

**Step 3.** Choose the satellite with the minimum cost difference and insert it into its cheapest position in \( R_1 \).

**Step 4.** Back to step 2 if there are still unplanned satellites; else go to step 5.

**Step 5.** Split \( R_1 \) into a set of routes with satisfying the capacity and working duration constraints of FLVs, time windows for the satellites, the deadline for customers, and the working duration constraint for SLVs.

**Step 6.** Return the reconstructed solution.

Noticeably, we have to ensure the feasibility of a satellite constructing a giant tour. Otherwise, the constraints of the satellite in the two giant tours probably be violated so that the problem is infeasible. In the process of splitting the giant tour, we take a fast feasibility check for the feasibility of the 2nd-level routes about the temporal aspect. After the application of a second-echelon insertion operator, for each 2nd-level route, we set the departure time of SLVs to the maximum departure time of the FLVs directly traveling from the depots to the satellite including the route and calculate the gap between the working duration of the SLV and the time spent by the SLV and the gap between the deadline of each customer visited by the SLV and the arrival time of the SLV at the customer. Thus, when we attempt to add a satellite to a 1st-level route, the departure time of the SLVs is known and we just need to check the departure time of every used SLVs if or not exceeds the min gap. So the feasibility of 2nd-level routes is assured.

4.5. **Improvement of the 1st-level routes.** After reconstructing the 1st-level routes, we develop two heuristics to improve the routes, namely the synchronization-based satellite removal and basic greedy insertion heuristics. The basic idea of the improvement is that firstly remove some satellites and then reinsert them into the routes. The synchronization-based satellite removal heuristic removes all the satellites that have penalty cost because the arrival of the FLVs visiting them is not be coordinated from the current solution and puts them into the satellite pool. In particular, all the 2nd-level routes that start from the satellites are saved in the satellite pool simultaneously. Afterwards, the basic greedy insertion heuristic reinserts the satellites with the 2nd-level routes in the satellites pool into the destroyed solution and respects all the constraints. To be noticed, the 2nd-level routes
remain unchanged for each satellite for the reason that we want to speed the ALNS algorithm.

After the application of the heuristics, we perform sequentially swap and 2-opt operators for each depot to execute a local search for improving 1st-level routes. To be noticed, the synchronization-based satellite removal and basic greedy insertion heuristics may not be performed when all the satellites have no penalty cost, i.e., all of used FLVs have been synchronized. However, the swap and 2-opt operators are executed whether the two heuristics are used or not. Anyway, the process of the improvement has to ensure the current solution is feasible.

4.6. Acceptance and stopping criteria and adaptive weight adjustment. The acceptance criterion determines whether a new solution \( s_{\text{new}} \) is accepted as the current solution \( s_{\text{cur}} \) or the best solution \( s_{\text{best}} \). If \( s_{\text{new}} \) has a lower objective value than \( s_{\text{cur}} \), whether the objective value \( s_{\text{new}} \) of lower than the incumbent best value or not, we set \( s_{\text{cur}} \) as \( s_{\text{new}} \). If the objective value of \( s_{\text{new}} \) better than the incumbent best value, we set \( s_{\text{best}} \) as \( s_{\text{new}} \). Else, we apply a simulated annealing criterion for determining the probability of accepting the new solution as a new current solution. The score of the second-echelon removal and insertion operators is \( \sigma_1, \sigma_2, \) and \( \sigma_3 \) based on the above three situations. The probability of acceptance is equal to \( e^{-\left(f(s_{\text{new}}) - f(s_{\text{cur}})\right)/T_{\text{cur}}} \), where \( T_{\text{cur}} > 0 \) is the current temperature at the current iteration and \( f(S_{\text{new}}) \) is the objective value of the new solution \( s_{\text{new}} \).

At the beginning of ALNS, \( T_{\text{cur}} \) is initialized as \( T_{\text{start}} \) and linearly cooled after each iteration. The cooling equation is \( T'_{\text{cur}} = T_{\text{cur}} \times \rho \) where \( \rho \) is the cooling rate with a domain \( 0 < \rho < 1 \). The ALNS algorithm is terminated when \( T_{\text{cur}} \) is lower than or equal to \( T_{\text{end}} \).

The process of ALNS is partitioned into many segments, in each segment, the weights of the second-echelon operators remain unchanged but are adjusted based on \( w_{i,i+1} = (1-r)w_{i,i} + r\pi_i/\eta_i \), where \( w_{i,j} \) is the weight of operator \( i \) at the segment \( j \), \( \pi_i \) is the score of operator \( i \), \( \eta_i \) is the number of times which the operator has been used during the last segment, and is the reaction factor referred to Ropke, Pisinger [29].

5. Computational experiments. In this section, the effectiveness of the MC-2E-VRPSS formulation and the ALNS were evaluated, based on small-scale and large-scale instances. To the best of our knowledge, there are no benchmark instances in the literature special for the MC-2E-VRPSS model. We designed a set of small-scale and large-scale instances based on real-world data. The set of small-scale instances with up to three satellites and 12 customers, which can be directly solved by GUROBI 9.1.1. The set of large-scale instances are solved by the ALNS and LNS.

All the algorithms presented in this paper are coded in C++ and tested on an Intel Core i5-9400F CPU clocked at 2.90 GHz and 8 GB memory running on a Windows 10 operating system.

5.1. Instance data.

5.1.1. Small-scale instances. Considering the complexity of our problem and the computing abilities of GUROBI 9.1.1 indicated by some computational trials, there are two depots, two or three satellites. The number of customers in the set is 5, 6, 7, 8, 9, 10, 11, 12. For each kind of the number of customers, we designed five instances for it. Thus, the set of small-scale includes 40 instances. For the instances that the
number of customers is 5, 6, 7, 8, 9, or 10, we set the number of satellites equal to two. For the two others, we set the number of satellites equal to three. Each small-scale instance is denoted by \( \text{num}_s-\text{num}_c-\text{sn} \), where \( \text{sn} \) is the instance sequence for the set, \( \text{num}_s \) and \( \text{num}_c \) denote the number of satellites and customers respectively.

5.1.2. Large-scale instances. We designed the large-scale instances by referring to real-world data provided by an express company in Chongqing city of China. The location of the depots and satellites are all real, while the time windows of satellites and the deadline for customers are randomly generated around the company’s baseline. The locations of the depots and satellites are shown in Fig. 3. We select the eight districts in Chongqing as our experiment sites. Furthermore, we integrate the data of two districts of those as one instance. Thus, the set of large-scale instances includes 28 instances. However, there are some satellites and customers with different quantities in each district, the number of satellites and customers in each instance is different. Generally speaking, we designed 28 large-scale instances that are distinguished by the location and number of the satellites and customers. The large-scale instance is denoted by \( \text{sn}-\text{num}_s-\text{num}_c \), where \( \text{sn} \) is the instance sequence for the set, \( \text{num}_s \) and \( \text{num}_c \) denote the number of satellites and customers respectively.

Figure 3. Layout of the depots and satellites

5.2. Parameter setting. The most parameters in the model and used by the ALNS algorithm in small-scale and large-scale instances are presented in Table 2. For the parameters in simulated annealing criteria, we choose \( T_{\text{start}} \) value such that the first new solution can be accepted with a probability of 40% if its objective value is 8% larger than the initial solution. \( T_{\text{end}} \) is set as \( 0.02\% \times T_{\text{start}} \).

We have conducted a set of preliminary tests to set the parameter \( q \) for obtaining best results by the ALNS algorithm on large-scale instances. The results shown that the average objective function value is lowest when the parameter \( q \) is selected.
randomly in the interval $[0.1n, 0.2n]$ where $n$ is the total number of the customers. Thus, we select randomly the $q$ in the interval in the subsequent computational tests.

Table 2. Parameters setting

| Parameters | Explanation | Value |
|------------|-------------|-------|
| $\sigma_1, \sigma_2, \sigma_3$ | Score of second-echelon removal and insertion methods | 60, 30, 20 |
| $\varphi, \lambda, \theta$ | Weight parameters in related removal method | 0.7, 0.3, 0.4 |
| $\rho_0$ | Randomness parameter in related and worst removal | 5 |
| $\tau$ | Reaction factor in adaptive weight adjustment | 0.55 |
| $\rho$ | Cool rate in simulated annealing criteria | 0.99 |
| $p_S$ | Penalty cost per unit time at satellite $S$ | 20RMB/min |
| $w$ | The time of loading or unloading at a satellite per unit | 0.07min |
| $u$ | The service time of the second level vehicle at a customer per unit | 1min |

5.3. Results on small-scale instances. For each small-scale instance, GUROBI 9.1.1 runs with default settings until finding an exact solution or reaching the computation time limit which is four hours. On each instance, the ALNS runs 5 times and we select the best of them as the counterpart of the result obtained by GUROBI.

We compare the results obtained by ALNS with those of the GUROBI solver in Table 3 and Table 4. The target for the comparison is that assessing the validity of the MC-2E-VRPS model and the ALNS heuristic. In Table 3 and Table 4, columns 1 and 2 indicates the instances and best found solution (BFS) respectively. Columns 3, 4, and 5 show the total cost obtained by Gurobi, computation time spent by Gurobi, and percentage gap between the cost obtained by Gurobi and BFS (denoted as $\text{GAP}^G$), respectively. Columns 6, 7, 8 show the total cost obtained by the ALNS, computation time spent by the ALNS, and percentage gap between the total cost obtained by the ALNS and BFS (denoted as $\text{GAP}^A$), respectively. To evaluate the effectiveness of the ALNS, we introduce two kinds of gap, $\text{GAP}^G$ and $\text{GAP}^A$, one for Gurobi and the other for the ALNS. $\text{GAP}^G = 100\% \times (\text{Cost}^G - \text{BFS})/\text{BFS}, \text{GAP}^A = 100\% \times (\text{Cost}^A - \text{BFS})/\text{BFS}$.

From Table 3 and Table 4, all 40 instances are solved by GUROBI and the ALNS to optimal, which means that the effectiveness of the ALNS and the formulation model are validated. But we noticed that the computation time spent by GUROBI is larger than our ALNS algorithm on all 40 instances. In particular, the computation time of 19 instances, namely 47.5%, spent by GUROBI greater than 600 seconds even reached the computation time limit which is 4 hours. Whereas, the ALNS heuristic solves the 40 instances in 5 seconds.

5.4. Comparison with LNS heuristic. Through the comparison of the exact solutions obtained by GUROBI with the heuristic solutions on a set of small-scale instances, we have verified the effectiveness of the ALNS. We solve a set of large-scale instances using the ALNS to test the algorithm performance and compared with the large neighborhood search (LNS), which was first introduced by Shaw[35], on large-scale data. The operators in the LNS are same as the operators used by the ALNS, we just adapt the LNS framework to suit our problem better.

On each instance, the ALNS and LNS runs five times and we take the best. The computational results are shown in Table 5. We show the results of three parts of the objective function in the model. The $\text{Obj}, \text{Obj}^1, \text{Obj}^2$ and $\text{Obj}^3$ denote the total cost, first-echelon routing cost, second-echelon routing cost, and penalty cost.
Table 3. Results on 30 small-scale instances with 2 satellites

| Instance | BFS | Groubi | ALNS |
|----------|-----|--------|------|
|          | Cost | CPU(s) | GAP  | Cost | CPU(s) | GAP |
| 1-2-5    | 507.89 | 507.89 | 2.62 | 0.00 | 507.89 | 0.47 | 0.00 |
| 2-2-5    | 237.98 | 237.98 | 4.65 | 0.00 | 237.98 | 0.24 | 0.00 |
| 3-2-5    | 397.80 | 397.80 | 2.14 | 0.00 | 397.80 | 0.47 | 0.00 |
| 4-2-5    | 443.54 | 443.54 | 3.77 | 0.00 | 443.54 | 0.18 | 0.00 |
| 5-2-5    | 288.02 | 288.02 | 1.97 | 0.00 | 288.02 | 0.71 | 0.00 |
| 6-2-6    | 810.29 | 810.29 | 960.02 | 0.00 | 810.29 | 0.71 | 0.00 |
| 7-2-6    | 298.21 | 298.21 | 3.64 | 0.00 | 298.21 | 1.17 | 0.00 |
| 8-2-6    | 324.49 | 324.49 | 7.59 | 0.00 | 324.49 | 0.57 | 0.00 |
| 9-2-6    | 483.34 | 483.34 | 18.63 | 0.00 | 483.34 | 0.61 | 0.00 |
| 10-2-6   | 268.89 | 268.89 | 21.43 | 0.00 | 268.89 | 0.31 | 0.00 |
| 11-2-7   | 326.37 | 326.37 | 34.96 | 0.00 | 326.37 | 0.47 | 0.00 |
| 12-2-7   | 503.27 | 503.27 | 18.63 | 0.00 | 503.27 | 0.41 | 0.00 |
| 13-2-7   | 331.86 | 331.86 | 74.96 | 0.00 | 331.86 | 0.82 | 0.00 |
| 14-2-7   | 403.75 | 403.75 | 47.84 | 0.00 | 403.75 | 1.19 | 0.00 |
| 15-2-7   | 485.47 | 485.47 | 163.73 | 0.00 | 485.47 | 0.79 | 0.00 |
| 16-2-8   | 315.66 | 315.66 | 1353.69 | 0.00 | 315.66 | 1.56 | 0.00 |
| 17-2-8   | 209.23 | 209.23 | 790.08 | 0.00 | 209.23 | 5.49 | 0.00 |
| 18-2-8   | 355.16 | 355.16 | 9814.36 | 0.00 | 355.16 | 3.01 | 0.00 |
| 19-2-8   | 265.10 | 265.10 | 1466.87 | 0.00 | 265.10 | 1.36 | 0.00 |
| 20-2-8   | 140.42 | 140.42 | 14400.00 | 0.00 | 140.42 | 1.95 | 0.00 |

Table 4. Results on 10 small-scale instances with 3 satellites

| Instance | BFS | Groubi | ALNS |
|----------|-----|--------|------|
|          | Cost | CPU(s) | GAP  | Cost | CPU(s) | GAP |
| 1-3-11   | 308.11 | 308.11 | 14400.00 | 0.00 | 308.11 | 1.95 | 0.00 |
| 2-3-11   | 287.45 | 287.45 | 14400.00 | 0.00 | 287.45 | 2.83 | 0.00 |
| 3-3-11   | 336.78 | 336.78 | 14400.00 | 0.00 | 336.78 | 1.94 | 0.00 |
| 4-3-11   | 321.06 | 321.06 | 14400.00 | 0.00 | 321.06 | 2.58 | 0.00 |
| 5-3-11   | 312.86 | 312.86 | 14400.00 | 0.00 | 312.86 | 4.17 | 0.00 |
| 6-3-12   | 319.56 | 319.56 | 14400.00 | 0.00 | 319.56 | 2.96 | 0.00 |
| 7-3-12   | 314.40 | 314.40 | 14400.00 | 0.00 | 314.40 | 3.41 | 0.00 |
| 8-3-12   | 336.92 | 336.92 | 14400.00 | 0.00 | 336.92 | 2.54 | 0.00 |
| 9-3-12   | 274.74 | 274.74 | 14400.00 | 0.00 | 274.74 | 4.27 | 0.00 |
| 10-3-12  | 344.50 | 344.50 | 14400.00 | 0.00 | 344.50 | 3.36 | 0.00 |
respectively. Column 1 indicates the instances. Column 6 and 11 indicate the computation time spent by the ALNS and LNS respectively. The gap in column 12 indicate the difference of the best objective value obtained by ALNS and LNS. 
\[ \text{Gap} = \frac{100\% (\text{Obj}_{\text{ALNS}} - \text{Obj}_{\text{LNS}})}{\text{Obj}_{\text{LNS}}}, \]

where \( \text{Obj}_{\text{ALNS}} \) and \( \text{Obj}_{\text{LNS}} \) is the objective value obtained by ALNS and LNS respectively.

| Instance | ALNS | LNS | GAP |
|----------|------|-----|-----|
| 1-7-86   | 2996.61 | 2550.03 | 446.58 | 0.00 |
| 2-12-172 | 3020.23 | 2466.56 | 553.67 | 0.00 |
| 3-11-131 | 3837.07 | 2966.56 | 976.51 | 0.00 |
| 4-10-113 | 3451.23 | 2854.07 | 597.16 | 0.00 |
| 5-10-128 | 2285.54 | 1784.24 | 501.30 | 0.00 |
| 6-10-243 | 1508.17 | 1240.37 | 267.80 | 0.00 |
| 7-6-109  | 1492.76 | 1234.75 | 258.01 | 0.00 |
| 8-12-233 | 2209.44 | 1992.44 | 193.65 | 0.00 |
| 9-11-130 | 1586.99 | 1234.75 | 258.01 | 0.00 |
| 10-10-103 | 1492.76 | 1234.75 | 258.01 | 0.00 |
| 11-8-118 | 1492.76 | 1234.75 | 258.01 | 0.00 |
| 12-6-233 | 2209.44 | 1992.44 | 193.65 | 0.00 |
| 13-5-103 | 1158.97 | 1019.10 | 139.87 | 0.00 |
| 14-19-131 | 2209.44 | 1992.44 | 193.65 | 0.00 |
| 15-18-151 | 1869.67 | 1629.70 | 239.97 | 0.00 |
| 16-16-149 | 2048.08 | 1736.08 | 312.01 | 0.00 |
| 17-17-281 | 2136.65 | 1665.81 | 470.84 | 0.00 |
| 18-13-151 | 1501.17 | 1318.51 | 182.66 | 0.00 |
| 19-5-110  | 1860.78 | 1341.12 | 516.66 | 0.00 |
| 20-15-125 | 2125.10 | 1879.57 | 245.53 | 0.00 |
| 21-16-240 | 2282.91 | 1859.31 | 423.60 | 0.00 |
| 22-12-110 | 1623.49 | 1493.46 | 130.04 | 0.00 |
| 23-14-145 | 1996.53 | 1694.82 | 301.72 | 0.00 |
| 24-15-260 | 2113.04 | 1624.00 | 487.04 | 0.00 |
| 25-11-130 | 1495.12 | 1327.46 | 167.66 | 0.00 |
| 26-13-275 | 2187.32 | 1691.48 | 495.85 | 0.00 |
| 27-9-145  | 1536.04 | 1308.93 | 227.10 | 0.00 |
| 28-10-260 | 1613.69 | 1228.41 | 385.27 | 0.00 |

The total cost in the objective function includes three parts: first-echelon routing cost (\( \text{Obj}^1 \)), second-echelon routing cost (\( \text{Obj}^2 \)), and penalty cost (\( \text{Obj}^3 \)). From Table 5, we can observe that the penalty cost of the 26 instances and 23 instances of the 28 large-scale instances solved by the ALNS and LNS respectively equals 0, while the penalty cost of the other instances are less than 80 RMB, namely the penalty time at some satellites is less than 4 minutes that is acceptable in the setting which \( p_s = 20 \) RMB/min. That is to say, the target to coordinate the arrival of the FLVs is achieved.

In the 28 instances, the number of satellites and customers up to 19 and 275 respectively. Table 5 shows that the computation time solving all the instances by the ALNS is less than the computation time spend by the LNS.

Table 5 shows that the total cost of the 22 instances of the 28 instances solved by the ALNS is lower than the result obtained by the LNS, which the maximum gap is 5.55%. The result shows that the performance of the ALNS is better than the LNS in the MC-2E-VRPSS. Fig. 4 (A) and (B) show that the first-echelon routing cost is the most costly part. Thus, the logistics suppliers can focus on the optimization of the first echelon routes.

5.5. Sensitivity analysis. The total cost of the MC-2E-VRPSS consists of routing cost, including first echelon routing and second echelon routing, and penalty cost
from satellites synchronization constraint is violated. We note that the two types
of costs interact with each other for the reason of finding the minimum cost of the
logistics network. Hence, we want to see how are they interact with each other to
give some insights into the coordination by setting several different values of the
time of loading or unloading at a satellite per unit $w$.

Therefore, we conduct sensitivity analysis using a set of varied $w$, from 0 to 0.1.
The analysis is done on one of the case (Instance 14) by our ALNS. The analysis
results are shown in Figure 5. Fig. 5 (A), (B) and (C) show the trend of the total
cost, routing cost and penalty cost with the varying of the parameter $w$ respectively.

From the Fig. 5 (A), we can observe that the total cost is insensitive to the par-
parameter $w$, which indicate that the stability of our ALNS algorithm on coordinating
the interaction about the two types of costs. From Fig. 5 (B) and (C), we can ob-
serve that both the routing cost and penalty cost are all sensitive to the parameter.
Specifically, we note that when the parameter $w$ is equal to 0.01 the routing cost
has a steep increase in Fig. 5 (B). At the same time, the penalty cost has a steep
decrease when the parameter $w$ is equal to 0.01. This indicate that if the operating time is extremely short the synchronization constraint is difficult to be satisfied, which cause enormous penalty cost. Thus, the ALNS algorithm sacrifices routing cost for synchronizing the arrival at satellites, which is the main goal of operating management.

In summary, the sensitivity analysis shows that the satellite synchronization is crucial to the logistics network from the operating management. In practice, the enterprises have the flexibility of adjusting their resource on satellites depending on the management ability at satellites to planning their daily routes.

5.6. Managerial analysis. In this section, we conduct managerial analysis from operation management for satellites. In practice, the satellites have equipped with sufficient resource including staff and equipment which have been planned at the establishment of the logistics network, based on the customers' requests can be satisfied properly. Hence, the enterprises need to increase the investment of the resource to tackle with the difficulty which the customers’ requests of multi-commodity satisfied by multi-depot if the satellites synchronization are not dealt with it properly.

For instance, the one of the multi commodities from the depot have been transported to a certain satellite while the other commodity is still stored at the one of the depot, which both the commodities for many customers have a same due time, so how we deliver the first arrival products is the problem that we need to more staff to sort and deliver the products if we deliver the products when them come.

From our study, we should consider the satellites synchronization in planning their daily routes for satisfying the customers' requests when the logistics network involve with multi commodities, which the enterprises will gain more benefits from this.

6. Conclusions. With the explosive growth of logistics demand, especially in the express industry, many logistics suppliers establish more and more distribution centers called depots in this paper to meet customers’ requests better. For the express companies, the two-echelon distribution network is usually an effective way to perform the deliveries. However, how to coordinate the schedule of each depot and the interaction between the first and second echelon is a complicated problem.

In this paper, we introduce a multi-commodity two-echelon vehicle routing problem with satellite synchronization (MC-2E-VRPSS). The MC-2E-VRPSS involves the transport from two depots to a set of satellites on the first echelon, and the deliveries from the satellites to final customers with two commodities demand on the second echelon. Owing to the limited resources, such as staff, equipment, and the timeliness guarantee for customers, soft synchronization constraints and time windows for the satellites should be respected simultaneously. The objective of the MC-2E-VRPSS is to determine the cost-minimizing routes of the two echelons. We develop a mixed-integer programming model for the problem, in which the time windows for the satellites, the deadline for customers, and the soft synchronization constraints at the satellites are considered. For solving the problem, we develop an adaptive large neighborhood search (ALNS) heuristic and we experimentally evaluate the applicability of the heuristic on a set of large-scale instances. We also validate the model formulation using GUROBI 9.1.1 and test the performance of the heuristic on a set of small-scale instances.

Future research can follow several directions. First, more kinds of commodities should be considered. In this study, we just consider two commodities, while there
Figure 5. Sensitivity analysis on the parameter $w$
are more than two commodities in actual practice. Second, more effective heuristics for the MC-2E-VRPSS are expected. Finally, we notice that a similar problem has been solved by an exact algorithm in the literature, so an exact algorithm for the MC-2E-VRPSS is also expected.

REFERENCES

[1] S. Afifi, D.-C. Dang and A. Moukrim, Heuristic solutions for the vehicle routing problem with time windows and synchronized visits, Optim. Lett., 10 (2016), 511–525.
[2] A. Anderluh, R. Larsen, V. C. Hemmelmaer and P. C. Nolz, Impact of travel time uncertainties on the solution cost of a two-echelon vehicle routing problem with synchronization, Flexible Services and Manufacturing Journal, 32 (2020), 806–828.
[3] D. Bredstrom and M. Ronnqvist, Combined vehicle routing and scheduling with temporal precedence and synchronization constraints, European Journal of Operational Research, 191 (2008), 19–31.
[4] U. Breunig, V. Schmid, R. F. Hartl and T. Vidal, A large neighbourhood based heuristic for two-echelon routing problems, Comput. Oper. Res., 76 (2016), 208–225.
[5] D. Cattaruzza, N. Absi, D. Feillet and J. Gonzalez-Feliu, Vehicle routing problems for city logistics, Euro Journal on Transportation and Logistics, 6 (2017), 51–79.
[6] D. Cattaruzza, N. Absi, D. Feillet and D. Vigo, An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows, Computers & Operations Research, 51 (2014), 257–267.
[7] G. Clarke and J. W. Wright, Scheduling of vehicles from central depot to number of delivery points, Operations Research, 12 (1964).
[8] T. G. Crainic, S. Mancini, G. Perboli and R. Tadei, Multi-start heuristics for the two-echelon vehicle routing problem, European Conference on Evolutionary Computation in Combinatorial Optimization, 6622 (2011), 179–190.
[9] T. G. Crainic, S. Mancini, G. Perboli and R. Tadei, GRASP with path relinking for the two-echelon vehicle routing problem, Advances in Metaheuristics, 53 (2013), 113–125.
[10] T. G. Crainic, G. Perboli, S. Mancini and R. Tadei, Two-echelon vehicle routing problem: A satellite location analysis, Procedia Social and Behavioral Sciences, 2 (2010), 5944–5955.
[11] T. G. Crainic, N. Ricciardi and G. Storchi, Models for evaluating and planning city logistics systems, Transportation Science, 43 (2009), 432–454.
[12] R. Cuda, G. Guastaroba and M. G. Speranza, A survey on two-echelon routing problems, Comput. Oper. Res., 55 (2015), 185–199.
[13] N. Dellaert, F. D. Saridarq, T. Van Woensel and T. G. Crainic, Branch-and-price-based algorithms for the two-echelon vehicle routing problem with time windows, Transportation Science, 53 (2019), 463–479.
[14] N. Dellaert, T. V. Woensel, T. G. Crainic and F. D. Saridarq, A multi-commodity two-Echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach, Comput. Oper. Res., 127 (2021), 105154, 12pp.
[15] A. Dohn, E. Kolund and J. Clausen, The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach, Comput. Oper. Res., 36 (2009), 1145–1157.
[16] A. Dohn, M. S. Rasmussen and J. Larsen, The vehicle routing problem with time windows and temporal dependencies, Networks, 58 (2011), 273–289.
[17] M. Drexl, Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints, Transportation Science, 46 (2012), 297–316.
[18] P. Eveborn, P. Flisberg and M. Ronnqvist, LAPS CARE—an operational system for staff planning of home care, European Journal of Operational Research, 171 (2006), 962–970.
[19] P. Grangier, M. Gendreau, F. Lehude and L.-M. Rousseau, An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization, European J. Oper. Res., 254 (2016), 80–91.
[20] M. H. Há, T. D. Nguyen, T. N. Duy, H. G. Pham, T. Do and L.-M. Rousseau, A new constraint programming model and a linear programming-based adaptive large neighborhood search for the vehicle routing problem with synchronization constraints, Comput. Oper. Res., 124 (2020), 105085.
[21] V. C. Hemmelmayr, J.-F. Cordeau and T. G. Crainic, An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics, *Comput. Oper. Res.*, **39** (2012), 3215–3228.

[22] M. Jepsen, S. Spoorendonk and S. Ropke, A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem, *Transportation Science*, **47** (2013), 23–37.

[23] H. Li, H. Wang, J. Chen and M. Bai, Two-echelon vehicle routing problem with satellite bi-synchronization, *European J. Oper. Res.*, **288** (2021), 775–793.

[24] R. Liu, Y. Tao and X. Xie, An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits, *Comput. Oper. Res.*, **101** (2019), 250–262.

[25] G. Perboli, R. Tadei and R. Tadei, New families of valid inequalities for the two-echelon vehicle routing problem, *Electronic Notes in Discrete Mathematics*, **36** (2010), 639–646.

[26] G. Perboli, R. Tadei and D. Vigo, The two-echelon capacitated vehicle routing problem: Models and math-based heuristics, *Transportation Science*, **45** (2011), 364–380.

[27] D. Pisinger and S. Ropke, A general heuristic for vehicle routing problems, *Comput. Oper. Res.*, **34** (2007), 2403–2435.

[28] R. Rabeh, K. SaïD and M. Eric, Collaborative model for planning and scheduling caregivers activities in homecare, *IFAC Proceedings Volumes*, **44** (2011), 2877–2882.

[29] S. Ropke and D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transportation Science*, **40** (2006), 455–472.

[30] Rousseau, The synchronized dynamic vehicle dispatching problem, *INFOR*, **51**.

[31] L. M. Rousseau, M. Gendreau and G. Pesant, The synchronized vehicle dispatching problem.

[32] F. A. Santos, A. S. da Cunha and G. R. Mateus, Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem, *Optim. Lett.*, **7** (2013), 1537–1547.

[33] F. A. Santos, G. R. Mateus and A. S. da Cunha, A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem, *Transportation Science*, **49** (2015), 355–368.

[34] J. Schoenberger, The two-commodity capacitated vehicle routing problem with synchronization, *IFAC-PapersOnLine*, **48** (2015), 168–173.

[35] P. Shaw, Using constraint programming and local search methods to solve vehicle routing problems, *Principles and Practice of Constraint Programming - Cp98*, (eds. M. Maher and J. F. Puget), Lecture Notes in Computer Science, **1520** (1998), 417–431.

[36] Z. Y. Zeng, W. S. Xu, Z. Y. Xu and W. H. Shao, A hybrid GRASP+VND heuristic for the two-echelon vehicle routing problem arising in city logistics, *Math. Probl. Eng.*, **2014** (2014), 517467, 1–11.

Received April 2021; revised October 2021; early access January 2022.

E-mail address: cuqijashengyang@163.com
E-mail address: denglei@cqu.edu.cn
E-mail address: zhaoquanwu@126.com
E-mail address: 984293065@qq.com