Transportation Research Part B 138 (2020) 179–201

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Two-echelon vehicle routing problem with time windows and mobile satellites

Hongqi Li*, Haotian Wang, Jun Chen, Ming Bai

School of Transportation Science and Engineering, Beihang University. No. 37 Xueyuan Road, Haidian District, Beijing, 100191, China

ARTICLE INFO

Article history:
Received 17 October 2019
Revised 11 May 2020
Accepted 15 May 2020

Keywords:
Two-echelon vehicle routing problem
Mobile satellite
Time windows
Van and unmanned-aerial-vehicles
Vehicle-flow formulation
Adaptive large neighborhood search

ABSTRACT

To tackle the logistics challenges faced by enterprises using unmanned aerial vehicles (UAV) with human-driven vans for parcel deliveries, we introduce the two-echelon vehicle routing problem with time windows and mobile satellites (2E-VRP-TM), which, when solved, optimizes delivery routes for a fleet of van-UAV combinations. Typically, one van carries several UAVs. The first echelon involves time-window-driven parcel deliveries using vans from a distribution center (DC) to customers. The second echelon involves UAVs being dispatched from mobile-satellite vans to serve customers with time windows and directly delivering parcels from the DC. When the first-echelon vehicles park at customer locations and wait for second-echelon vehicle departures and returns, the first-echelon vehicles are used as mobile satellites. We develop a vehicle-flow formulation, in which the mobile-satellite synchronization constraints are included to ensure the echelon interaction. We provide an adaptive large neighborhood search heuristic. Computational experiments evaluate the validity of the 2E-VRP-TM formulation and the effectiveness of the heuristic.

1. Introduction

Multi-echelon distribution systems are commonplace. Currently, nearly all mathematical methods for solving routing problems of multi-echelon networks are based on studies of the two-echelon vehicle routing problem (2E-VRP), the two-echelon location routing problem (2E-LRP), and the truck and trailer routing problem (TTRP). Routing problems of two-echelon networks involve various types of vehicles for first- and second-echelon routes and intermediate platforms. These intermediate platforms are called “satellites” in the literature and are used to consolidate and transship cargoes. Crainic and Sgalambro (2014) stated that echelon interactions (e.g., fleet coordination between echelons and satellites) are key challenges when modeling the two-echelon scheme. From the methodological perspective of constructing mathematical models, the assignment decisions between satellites and customers or/and satellite synchronization constraints were introduced.

Despite this background, many important realistic features remain to be considered. To the best of our knowledge, there is no literature that targets the routing problem of the two-echelon network using mobile satellites. Thus, we strove to find a suitable method of tackling the practical situation that express enterprises face over the course of using unmanned-aerial-vehicles (UAV) carried by vans for parcel deliveries.

Great advancements in the area of the ground vehicles and UAV technologies have brought many new utilization possibilities for UAVs in the logistics sector. For example, the Mercedes-Benz Vision Van features a fully automated cargo space

* Corresponding author.
E-mail address: lihongqi@buaa.edu.cn (H. Li).

https://doi.org/10.1016/j.trb.2020.05.010
0191-2615/

(a) (b)

Fig. 1. Mercedes-Benz Vision Van concept in the "Vans & Drones" project lodged at Mercedes-Benz Vans in the Future Transportation Systems unit. (a) https://www.netcarshow.com/mercedes-benz/2016-vision_van_concept/. (b) https://www.daimler.com/innovation/specials/future-transportation-vans/vans-drones-2.html

and integrated UAV for autonomous air deliveries (Fig. 1). The Vision Van serves as the central intelligence element in a fully connected delivery-chain system. The system supplies two UAVs, each with consignments for delivery. The Vision Van is expected to provide the perfect mobile platform for UAVs, enabling efficient operations.

Some enterprises are now racing toward UAV-delivery modes. At the small scale, several enterprises, such as Amazon, UPS, JD.com, and SF Technology[1], have tested last-mile small-parcel deliveries via UAVs. Amazon developed the Prime Air UAV, which has a limited flight range and capacity. Amazon stated that 86% of its deliveries weighed less than the payload capacity of the UAV[2], and there is a percentage of deliveries exceeding UAV capacity, requiring traditional means (e.g., vans and trucks). Several strategies have been proposed to deal with limited flight ranges and capacities, including approaches that combine UAVs with vans. UAVs can serve customers distributed across a large region using mobile platforms provided by the vans, and the vans can serve customers with demands exceeding UAV capacity.

Considering the commercial interest in UAV delivery modes, only a few studies have tackled operational challenges. However, research addressing the technological barriers to introducing UAVs into delivery operations is abundant (Ham, 2018). We are already aware of the operational challenges of optimizing delivery routes for a homogeneous fleet of van-UAV combinations.

In this study, the UAV combination involves pairing several UAVs with one van. A van that carries several UAVs and parcels departs from a distribution center (DC). At a selected location, the van stops, and the UAVs take off with their assigned parcels. Each UAV returns to the host van after delivery. From the perspective of a network with two echelons, van routes originating from the DC are on one echelon, and UAV routes originating from selected customer locations or the DC are on the other. At each selected customer location where UAVs are dispatched, the arrival time of one van needs to be earlier than or equal to the take-off time of the carried UAVs. At the selected customer location for dispatching UAVs, the time difference between the arrival time of the van and the return time of the dispatched UAVs should be considered and is of influence for the penalty cost of the waiting van. When all dispatched UAVs return to their paired van, the van-UAV combination moves to new customer locations to make more deliveries.

We thus introduce the two-echelon vehicle routing problem with time windows and mobile satellites (2E-VRP-TM) for the first time. The 2E-VRP-TM involves mobile satellites of first-echelon routes using a homogeneous fleet of second-echelon vehicles, when possible. Only when first-echelon vehicles park at certain locations and wait for second-echelon vehicles' departure and return can they be used as mobile satellites. Compared with the satellite concept of the 2E-VRP and the 2E-LRP, the mobile-satellite concept indicates a type of temporary moving infrastructure for parcel transshipment and consolidation. The location of a mobile satellite is selected for the carried UAV's departure and return. When en route, each van-UAV combination moves to assigned customer areas to make deliveries. Thus, the locations of mobile satellites vary along with each van's movement.

Fig. 2 illustrates route examples of one 2E-VRP-TM instance. The customers are numbered from 1 to 20. Customers 1 to 10 are served by vans, whereas the other ten are served by UAVs dispatched from the DC (denoted as 0) or from vans at selected customer locations. At these customer locations, one or several UAVs may take off from their paired van, deliver their cargo, and return. The parked van at the customer location is a mobile satellite. In Fig. 2, customers 2, 3 and 4 are mobile satellite locations. In the route travelled by the No.3 van with UAVs, the van departs from the DC at 8:45, arrives at customer 2 at 9:02 and parks. The No.3-1 UAV takes off at customer 2 and arrives at customers 8 and 11 at 9:03 and 9:13, respectively. It returns to the No.3 van at 9:20. The No.3 van with UAVs returns to the DC at 10:08. In the route travelled by the No.2 van with carried UAVs, the van departs from the DC at 7:47, arrives at customer 3 at 8:32 and parks. The No.2-1

[1] http://www.sf-tech.com.cn/.
[2] http://www.cnn.com/2013/12/02/tech/innovation/amazon-drones-questions/.

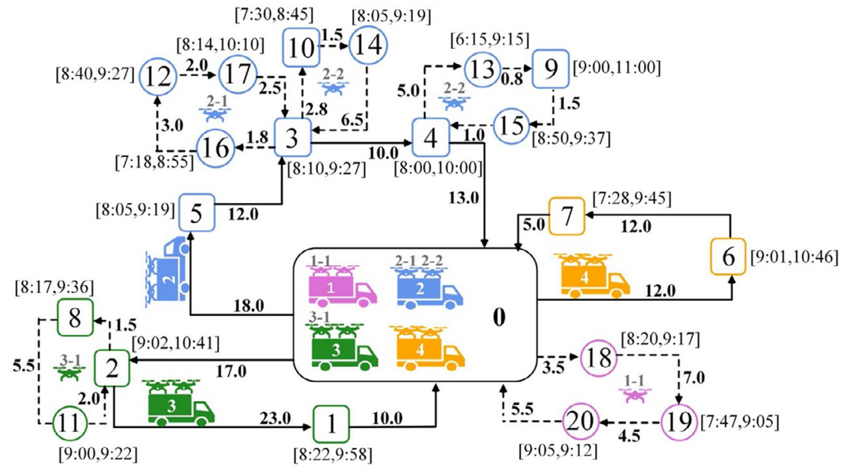


Fig. 2. Illustrative example of the two-echelon process.

Table 1
Abbreviations and meanings.

Abbreviations	Meanings
TSP	Traveling salesman problem
VRP	Vehicle routing problem
VRPTW	Vehicle routing problem with time windows
2E-VRP	Two-echelon vehicle routing problem
2E-LRP	Two-echelon location routing problem
TTRP	Truck and trailer routing problem
2E-VRP-TM	Two-echelon vehicle routing problem with time windows and mobile satellites
FSTSP	Flying sidekick traveling salesman problem
TSP-D	Traveling salesman problem with drones
mTSPD	Multiple traveling salesman problem with drones
VRPD	Vehicle routing problem with drones
MIP	Mix integer programming
MILP	Mix integer linear programming
ALNS	Adaptive large neighborhood search
DC	Distribution center
UAV	Unmanned aerial vehicle
VUAC	Van-UAV customer. In the 2E-VRP-TM, each VUAC can be served by one van and can be served by one UAV if the UAV capacity is respected
UAC	UAV customer. In the 2E-VRP-TM, each UAC can only be served by one UAV

UAV takes off at customer 3 and arrives at customers 16, 12 and 17 at 8:34, 8:42 and 8:49, respectively. It returns to the No.2 van at 8:56. The No.2-2 UAV takes off at customer 3 and arrives at customers 10 and 14 at 8:35 and 8:41, respectively. It returns to the No.2 van at 8:52. Then, the No.2 van with carried UAVs arrives at customer 4 at 9:06 and parks. The No.2-2 UAV takes off at customer 4 and arrives at customers 13, 9 and 15 at 9:11, 9:17 and 9:23, respectively. It returns to the No.2 van at 9:29. The No. 2 van with its UAVs returns to the DC at 9:42. In the route travelled by the No.4 van (i.e., 0-6-7-0), the van departs from the DC at 8:49, arrives at customers 6 and 7 at 9:01 and 9:28, respectively, and returns to the DC at 9:48. The No.1-1 UAV takes off at 8:41 from the DC, arrives at customer 18, 19 and 20 at 8:44, 8:56 and 9:05, respectively, and returns to the DC at 9:15.

The contributions of this paper include the introduction and formal definition of the 2E-VRP-TM, which caters to the delivery applicability of a fleet of van-UAV combinations with respect to customer time windows. We propose a vehicle-flow formulation in which mobile-satellite synchronization constraints provide a new modeling mechanism connecting the two echelons. We provide an adaptive large neighborhood search (ALNS) heuristic, and we experimentally evaluate the validity of the 2E-VRP-TM mathematical formulation and the effectiveness and applicability of the proposed heuristic. We also briefly discuss management implications.

For reader convenience, we list the abbreviations used in this paper in Table 1.

The paper is organized as follows. Section 2 reviews studies on UAV-routing problems involving mobile-satellite synchronization and those on the 2E-VRP, 2E-LRP, and TTRP models. Section 3 formally defines the 2E-VRP-TM and introduces a mixed-integer linear programming model. Section 4 presents an ALNS heuristic. Section 5 describes the test instances, their exact and heuristic results, and corresponding analyses. We conclude with Section 6.

2. Literature review

2E-VRP and 2E-LRP are the two general types of routing problems using two-echelon networks. The TTRP is as a special variant of 2E-VRP (Cuda et al., 2015). In the literature that developed 2E-VRP formulations, e.g., Perboli et al. (2011), Perboli et al. (2018), and Dellaert et al. (2019), the assignment decision between satellites and customers, and/or constraints that keep balance between the quantity delivered by first-echelon routes to a satellite and the customer demands supplied from the satellite are key to ensuring the echelon interaction. To solve the 2E-VRP, Perboli and Tadei (2010), Santos et al. (2013), Jepsen et al. (2013), Sitek and Wikarek (2014), Perboli et al. (2018), and Liu et al. (2018) proposed exact algorithms. Crainic et al. (2011), Wang et al. (2017b), Belgin et al. (2018), and Zhou et al. (2018) presented heuristics. To solve the 2E-LRP, Dalfard et al. (2013), Breunig et al. (2016), and Pichka et al. (2018) provided heuristics. For the TTRP, vehicles servicing customers could be a truck-and-trailer or just a single truck. Chao (2002), Scheuerer (2006), Lin et al. (2009, 2011), Villegas et al. (2013), Belenguer et al. (2015), Parragh and Cordeau (2017), and Rothenbächer et al. (2018) presented exact and heuristic algorithms to solve the TTRP. For satellite synchronization of the 2E-VRP variants, Grangier et al. (2016) presented the two-echelon multiple-trip vehicle routing problem (VRP) with satellite synchronization. The first- and second-echelon vehicles must be present at the same satellite at the appointed time. Li et al. (2016) proposed the two-echelon time-constrained VRP (2E-TVRP), with which time constraints ensured the echelon interaction. Li et al. (2018) took satellites' real-time transshipment capacity as varying using transshipment and consolidation operations.

Savelsbergh and Van Woensel (2016) stated that adequate decision support for multi-echelon models for mobile satellites was missing in the literature and offered an interesting path for future work. The mobile-satellite concept was occasionally mentioned in the literature. Considering the situation in which a UAV works in collaboration with a truck to distribute parcels, and the truck is used as the mobile depot, some variants of the traveling salesman problem (TSP) (e.g., flying sidekick TSP (FSTSP) and TSP with drone (TSP-D)) were introduced. Considering the situation in which some UAVs work in collaboration with trucks, and selected parking locations for trucks are used as mobile satellites, variants of the VRP (e.g., VRP with drones (VRPD)) were introduced.

In the general forms of the TSP variants involving the use of a UAV, one truck paired with one UAV has been used. The truck and UAV depart from and return to the DC exactly once. Both the truck and the UAV can serve customers. The truck travels along its route, and the UAV can land on and depart from the truck only while it is parked at a customer location or the DC. Each customer should be served exactly once, and customer time windows are not considered. The UAV's flight endurance and capacity are usually considered. The objective is to minimize time or costs. Randomly-generated instances are used to test these models and algorithms. Murray and Chu (2015) introduced the FSTSP, in which two vehicles departed and returned either in tandem or independently. UAVs could make multiple flights. A mathematical programming model was introduced for this. A route and re-assign heuristic was proposed, which solves a TSP that assigns the truck to visit all customers then UAV sub routes are constructed by a savings-based heuristic. To solve the FSTSP, de Freitas and Penna (2018) created an initial solution using the optimal TSP solution obtained by the Concorde solver. Then, the randomized variable neighborhood descent heuristic was used. Jeong et al. (2019) introduced an FSTSP variant that implemented energy consumption and a no-fly zone, which aimed to minimize the arrival time at the ending depot after serving all customers. A mixed-integer linear programming model and an evolutionary-based heuristic were proposed. Agatz et al. (2018) introduced TSP-D, in which a truck could travel to serve customers and remain stationary at the origin until the UAV returned. The objective was to find the shortest tour in terms of time. An integer program and several route-first, cluster-second heuristics, based on local search and dynamic programming, were presented. Es Yurek and Ozmutlu (2018) presented an iterative algorithm to minimize delivery completion time of the TSP-D. Ha et al. (2018) introduced a TSP-D variant to minimize operational costs, including total transportation cost and another created by wasted time when a vehicle waits. A greedy randomized adaptive search procedure was also proposed.

In the extensions of the FSTSP and TSP-D, some constraints (e.g., using one truck paired with one UAV and the UAV capacity limit) were relaxed. Kitjacharoenchai et al. (2019) introduced the multiple TSP with drones (mTSPD) model. The capacity of trucks and the demand quantity of customers were not considered. A drone could be retrieved by any nearby truck. It could also land on a recharging station to extend its range. Mixed-integer programming and an algorithm based on greedy insertion heuristics were developed. The problem introduced by Chang and Lee (2018) involved a truck carrying delivery items departing from and returning to the depot, travelling the shifted centers of clusters bundled by several delivery locations. In each cluster, several drones left the truck and returned after delivery. The solution approach included three steps: clustering delivery locations, routing centers of clusters, and finding the shift weights. Ferrandez et al. (2016) investigated the reduced overall delivery time and energy for a truck-drone network by comparing the in-tandem system with a stand-alone delivery effort. One or more drones and a single truck worked in tandem. The proposed algorithm determined the optimal number of launch sites and locations and drones per truck to minimize the costs associated with the parabolic convex cost function. Karak and Abdelghany (2019) introduced the hybrid vehicle-drone routing problem, in which customers were served only by drones. Three heuristic approaches were developed. Luo et al. (2017) presented a 0-1 programming model for the two-echelon cooperated routing problem. It used a number of rendezvous points where the ground-vehicle stopped to allow the UAV to take off and land. Gambella et al. (2018) proposed a mixed-integer second-order conic programming formula for the carrier-vehicle TSP, in which the carrier transported the faster vehicle and deployed, recovered, and served it. A ranking-based algorithm was proposed for this.

Table 2
Studies related to TSP and VRP variants involving UAVs.

Problem	Reference	Truck	Truck and UAV	Duration		UAV capacity	Time window	Objective and model
				Truck	UAV			
FSTSP	Murray and Chu (2015); de Freitas and Penna (2018); Jeong et al. (2019)	1	1:1		√	one		Min time; MILP
TSP-D	Agatz et al. (2018); Es Yurek and Ozmutlu (2018)	1	1:1		√	one		Min time; MILP
	Ha et al. (2018)	1	1:1		√	one		Min costs; MILP
mTSPD	Kitjacharoenchai et al. (2019)	m	1:1			one		Min time; MIP
TSP variant	Ferrandez et al. (2016)	1	1:n			one		Min costs
TSP variant	Karak and Abdelghany (2019)	1	1:n		√	many		Min costs; MIP
VRPD	Wang et al. (2017a)	m	1:n	√		one		Min time;
	Wang and Sheu (2019)	m	1:n		√	many		Min costs; MIP
2E-VRP variant	This paper	m	1:n	√	√	many	√	Min costs; MILP

Note: √ denotes that the factor was considered.

The literature that investigated the VRP variants with UAVs were not as plentiful as the literature that investigated the TSP variants with UAVs. Wang et al. (2017a) introduced the VRPD, where a fleet of trucks equipped with drones delivered packages. The objective was to minimize the completion time from when trucks were dispatched from the depot to the time when the last truck or the last drone returned to the depot. A number of questions were posed to study the maximum savings, and several worst-case theorems were proven. Wang et al. (2017a) stated that smart exact and heuristic approaches were needed to solve VRPD. Wang and Sheu (2019) investigated a VRPD variant in which a drone could travel with a truck, take off from its stop location to serve customers, and land at a service hub to travel with another truck. On the network, there were a set of transfer stations for drones. A mixed-integer programming model and a branch-and-price algorithm were proposed for this. Wang and Sheu (2019) stated that the problem with customer time windows was one of the promising and challenging new topics.

Studies on TSP and VRP variants involving UAVs in collaboration with ground vehicles, in which the mobile-satellite concept was indicated, have become popular since 2015. However, the mobile-satellite concept was indicated for specific and simplified situations (e.g., one UAV paired with one truck and a model without time-window constraints). There still exists a need for methodological innovations to formulate a 2E-VRP variant with mobile-satellite synchronization constraints to cater to more general and practical situations involving customer time windows, a fleet of van-UAV combinations, and UAV-direct delivery. In Table 2, we compare previous works with this work to show the main differences.

3. The 2E-VRP-TM definition and mathematical formulation

3.1. The 2E-VRP-TM definition

The reviewed literature shows that researchers are striving to find suitable methods to tackle various practical situations of van-UAV delivery modes. Mathematical models are usually unable to capture all features of practical situations. Since there are many new utilization possibilities for UAVs combined with vans for parcel deliveries, some features are defined in the paper, so that several assumptions of the 2E-VRP-TM are reasonable and the developed model can generalize the UAV routing problems in the literature.

The 2E-VRP-TM has the following characteristics.

i) On the 2E-VRP-TM network, there is one DC where a homogeneous fleet of van-UAV combinations are available. The DC is the only parcel source on the network, and all vans and UAVs in use should depart from and return to the DC.

ii) The van-UAV combination includes one van paired with several UAVs. It is assumed that a van and its UAVs work in a pair mode. Pair mode means at a mobile satellite a UAV departs from its van, serves the assigned customers and then lands on its van; and a UAV is not allowed to depart itself at the DC and then be picked up by its van at a mobile satellite. Considering the homogeneous fleet feature, the number of UAVs carried by each van is assumed to be the same and remains constant. The capacity, maximum working time, variable cost, and average velocity of each van remain the same. The capacity, maximum flying time of one takeoff, operating cost, and average velocity of each UAV remain the same.

iii) It is assumed that the UAVs carried by each van do not occupy van capacity. The total of parcels needed by the intended customers on a route travelled by the van-UAV combination should not exceed van capacity. A UAV in use can combine in one route the load of several customers.
iv) UAVs or vans can be used for direct delivery from the DC. In the scenario of one UAV being used for direct delivery from the DC, the UAV takes off from its paired van and lands on the same van while the van is not in use. In the scenario of one van being used for direct delivery from the DC, the carried UAVs are not in use.
v) Considering the accessing capability of UAVs or vans, customers are classified into two types: the van-UAV customer (VUAC) and the UAV customer (UAC). Each VUAC can be served by one van and can be served by one UAV if the UAV capacity is respected. Each UAC can only be served by one UAV. Customer demand cannot be split. All customer demand must be satisfied. The known time windows of customers must be respected.
vi) Considering the situation of UAVs needing intervention from their vans, it is assumed that a van can not drive while the dispatched UAVs are making deliveries. At selected VUAC locations, one or several UAVs may take off from a paired van to fly and serve UACs or VUACs. Then, they must land back on the paired van. If the UAVs are dispatched, the paired van cannot visit other VUACs until all dispatched UAVs return. The selected VUAC location with the parked van is regarded as a mobile satellite.
vii) At a mobile satellite, a UAV in use can be dispatched exactly once. Each carried UAV can be dispatched once at each of its mobile satellites, and the time needed to transship parcels and fresh energy-units to UAVs is not considered.

The 2E-VRP-TM objective is to minimize integrated costs. Because time is of great importance for estimating vehicle occupations, and timeliness is crucial for parcel delivery, the penalty cost of a van or UAV waiting at customer locations is transformed in terms of money and integrated into the objective function.

The two echelons of the 2E-VRP-TM network are distinguished by referring to the vehicle types. The first echelon involves vans delivering parcels from the DC to VUACs. The second echelon involves UAVs taking off from mobile satellites to visit UACs or VUACs, and involves UAVs delivering parcels directly from the DC. There are three types of routes in the 2E-VRP-TM solution. The first type is called “pure UAV route”. A pure UAV route originates and terminates at the DC and is travelled by one UAV for direct delivery. The second type is called “van route”. In a van route, the van travels with the carried UAVs not being in use. The third type is called “combination route”. In a combination route, there are one or several mobile satellites. A combination route comprises one main-tour and one or many sub-tours. The main-tour is travelled by the van and the carried UAVs, and sub-tours are travelled by dispatched UAVs departing from the mobile satellites. The origin and destination of one sub-tour are the same mobile satellite. There may be more than one customer en one pure UAV route or en one sub-tour, because it is assumed that a UAV in use can combine in its route the load of several customers.

3.2. Mathematical formulation

The 2E-VRP-TM is presented on a directed complete graph, $G = (V, A)$. The definitions of parameters and variables are summarized in Table 3. In the V^{tuac} set, each node except 0 can be used as a mobile satellite. Considering the assumption that the UAVs carried by each van do not occupy van capacity, the capacity of each van-UAV combination is Q^{t}. The time variables should be greater than or equal to 0.

3.2.1. The objective function

The objective function is described by Eq. (1).

$$\begin{aligned} Min z = & \sum_{k\in K}\sum_{i\in V^{\mathrm{tuac}}}\sum_{j\in V^{\mathrm{tuac}}} c^{\mathrm{t}} \cdot d_{ij} \cdot x_{kij} + \sum_{k\in K}\sum_{i\in V^{\mathrm{tuac}}} c^{\mathrm{t}} \cdot \left(wt^{\mathrm{t}}_{ki} + wt^{\mathrm{tu}}_{ki}\right) \cdot v^{\mathrm{t}} \cdot \tau \\ & + \sum_{k\in K}\sum_{u\in U}\sum_{r\in V^{\mathrm{tuac}}} c^{\mathrm{u}} \cdot w_{kur} + \sum_{k\in K}\sum_{u\in U}\sum_{i\in V} c^{\mathrm{u}} \cdot wtu_{kui}/UL \cdot \tau \end{aligned} \tag{1}$$

The objective is to minimize the integrated cost. The function includes four parts: the variable cost of vans; transformed cost of possible waiting-times of vans at customers waiting for time windows or waiting for dispatched UAV returns; operating cost of UAVs; and transformed cost of possible waiting-time of UAVs at customers waiting for time windows. Each of the four parts is estimated using money units.

3.2.2. Constraints on mobile satellites

A mobile satellite is actually one customer in a main-tour. Considering that customer demand cannot be split, it is assumed that one van (with carried UAVs) arrives once at and leaves once from the VUAC used as the mobile satellite. A mobile satellite is used as the origin and the destination of a sub-tour of combination routes. At a mobile satellite, one dispatched UAV leaves once from the mobile satellite and arrives once at the mobile satellite, considering the assumptions. With the carried UAVs taking off from the paired van at a mobile satellite to serve UACs, the van serves the mobile satellite while it waits for the dispatched UAV's return. Only if all dispatched UAVs return and land on the paired van, can the van-UAV combination continue to the next node. Constraints on mobile satellites should identify their locations, identify the use of UAVs, and ensure time continuity.

$$w_{kur} = \sum_{i\in V^{\mathrm{c}}} y_{kurri}, k \in K, u \in U, r \in V^{\mathrm{tuac}} \tag{2}$$

Table 3
Parameters and variables.

Parameters	
V	Node set, $V = V^{\text{tuac}} \cup V^{\text{uac}} = \{0, 1, 2, ..., n^{\text{tuac}}, ..., n\}$, and 0 denotes the DC
V^{c}	Customer set, $V^{\text{c}} = V \setminus \{0\}$
V^{tuac}	DC and VUAC set, $V^{\text{tuac}} = \{0, 1, 2, ..., n^{\text{tuac}}\}$
V^{uac}	UAC set, $V^{\text{uac}} = \{n^{\text{tuac}} + 1, ..., n\}$
A	Arc set, $A = \{(i, j) \mid i, j \in V\}$
d_{ij}	Distance of arc (i, j) $(i, j \in V)$, supposing $d_{ij} = d_{ji}$
q_i	Weight of parcels required by customer i
$[e_i, l_i]$	Time window of customer i, where e_i and l_i represent the earliest and latest service-starting times
s_i	Service time at customer i
$Q^{\text{u}}\&Q^{\text{t}}$	UAV capacity and van capacity
$v^{\text{t}}\&v^{\text{u}}$	Average velocity of a van and a UAV
$c^{\text{t}}\&c^{u}$	Variable cost of a van and the operating cost of one UAV takeoff
K	Set of vans available at the DC
U	Set of UAVs paired with each van
TL	Maximum working time of each van
UL	Maximum flying time of one takeoff for each UAV
τ	Penalty cost of a van or a UAV waiting for a unit time at customer locations
M	A sufficiently large positive integer
Variables	
x_{kij}	A binary variable that is 1 iff van k $(k \in K)$ (with the carried UAVs) covers arc (i, j), $(i, j \in V^{tuac},\ i \neq j)$
dt_k^t	Departure time of van k at the DC
at_{kj}^t	Arrival time of van k at the DC or customer j $(j \in V^{tuac} \setminus \{0\})$
wt_{kj}^t	Possible waiting-time of van k for customer time-window opening at customer j $(j \in V^{tuac} \setminus \{0\})$
wt_{kj}^{tu}	Waiting time of van k at mobile satellite j $(j \in V^{tuac} \setminus \{0\})$ for waiting for the dispatched UAV's return
$dt_{ki}^{\max}$	Leaving time of van k at mobile satellite i $(i \in V^{tuac} \setminus \{0\})$
y_{kurij}	A binary variable that is 1 iff UAV u $(u \in U)$ carried by van k takes off at the DC or at the mobile satellite, r $(r \in V^{tuac}, r \neq i \neq j)$, and visits nodes i, j $(i, j \in V, i \neq j)$ in turn
w_{kur}	A binary variable that is 1 iff UAV u carried by van k takes off at mobile satellite r
dtu_{ku}	Take-off time of UAV u carried by van k at the DC
atu_{kui}	Arrival time of UAV u paired with van k at node i $(i \in V)$
wtu_{kui}	Possible waiting-time of UAV u paired with van k at customer i $(i \in V^{\text{c}})$ for customer time-window opening
$at_{ki}^{\text{u-max}}$	Latest returning-time of dispatched UAVs (paired with van k) at mobile satellite i $(i \in V^{tuac} \setminus \{0\})$

$$\sum_{j \in V^{\text{c}}} y_{kurrj} \leq \sum_{i \in V^{\text{tuac}}} x_{kir}, k \in K, u \in U, r \in V^{\text{tuac}} \setminus \{0\} \tag{3}$$

$$\sum_{j \in V^{\text{c}}} y_{kurrj} \leq 1, k \in K, u \in U, r \in V^{\text{tuac}} \tag{4}$$

$$\sum_{j \in V^{\text{tuac}} \setminus \{0\}} x_{k0j} + \sum_{j \in V^{\text{c}}} y_{ku00j} \leq 1, k \in K, u \in U \tag{5}$$

The constraint identified by Eq. (2) classifies mobile-satellite locations. Only if UAV u, carried by van k, takes off at node r, node r is used as a mobile satellite. The constraint identified by Eq. (3) indicates that mobile satellites are included in the main tours of combination routes. The constraint identified by Eq. (4) guarantees that, at a mobile satellite or DC, each UAV can be used at most once. The constraint identified by Eq. (5) ensures that a van departs from the DC at most once.

$$wt_{ki}^{\text{tu}} \geq \max l\left(0, at_{ki}^{\text{u-max}} - \left(at_{ki}^{\text{t}} + wt_{ki}^{\text{t}} + s_i\right)\right), k \in K, u \in U, i \in V^{\text{tuac}} \setminus \{0\} \tag{6}$$

$$at_{kr}^{\text{u-max}} \geq atu_{kur}, k \in K, u \in U, r \in V^{\text{tuac}} \setminus \{0\} \tag{7}$$

$$at_{kr}^{t} + d_{ri}/v^{u} + M \cdot (1 - y_{kurri}) \geq atu_{kui}, k \in K, u \in U, r \in V^{\text{tuac}} \setminus \{0\}, i \in V^{\text{c}} \tag{8}$$

$$at_{kr}^{t} + d_{ri}/v^{u} - M \cdot (1 - y_{kurri}) \leq atu_{kui}, k \in K, u \in U, r \in V^{\text{tuac}} \setminus \{0\}, i \in V^{\text{c}} \tag{9}$$

$$dt_{ki}^{\max} = \max l(at_{ki}^{u-max}, at_{ki}^{t} + wt_{ki}^{t} + s_i), k \in K, u \in U, i \in V^{tuac} \setminus \{0\} \tag{10}$$

$$dt_{ki}^{\max} + d_{ij}/v^{\text{t}} - M \cdot (1 - x_{kij}) \leq at_{kj}^{\text{t}}, k \in K, i \in V^{\text{tuac}} \setminus \{0\}, j \in V^{\text{tuac}} \tag{11}$$

$$dt_{ki}^{\max} + d_{ij}/v^{t} + M \cdot (1 - x_{kij}) \geq at_{kj}^{t}, k \in K, i \in V^{\mathrm{tuac}} \backslash \{0\}, j \in V^{\mathrm{tuac}} \tag{12}$$

The constraint identified by Eqs. (6) indicates that the waiting time of a van at a mobile satellite for UAV returns is decided by the returning time of the last UAV and the van service time at the mobile satellite. If the last UAV returns before the van finishes the service at the mobile satellite, the waiting time of the van at the mobile satellite is assumed to be 0. " max*l* ", which is defined by CPLEX solver, is a function to return the maximal value from a list of integers or floats. The constraint identified by Eqs. (7) defines the latest returning-time of dispatched UAVs at a mobile satellite. The constraints identified by Eqs. (8) and (9) indicate the time relationship between the UAV taking-off time at a mobile satellite and the UAV arrival time at the first UAC on the sub-tour. The UAV taking-off time at a mobile satellite is assumed to be the same as the van-UAV combination arrival time at the mobile satellite. The constraint identified by Eqs. (10) identifies the leaving time of a van at a mobile satellite. If the time of a van finishing the customer service is later than the returning-time of the last UAV at a mobile satellite, the leaving time of the van is assumed to be the same as the time of the van finishing the customer service; otherwise, the leaving time of the van is assumed to be the same as the returning-time of the last UAV. When node i is a mobile-satellite, the constraints identified by Eqs. (11) and (12) indicate the time relationship between the leaving time of a van-UAV combination at a mobile satellite and the arrival time of the van-UAV combination at the next node on a main-tour. When node i is not a mobile-satellite, the constraints identified by Eqs. (11) and (12) indicate the time relationship between the arrival time of a van at a node i and that at the next node j.

3.2.3. Constraints on vehicle capacity and route duration

$$\sum_{i \in V^{c}, i \neq r} \sum_{j \in V} q_i y_{kurij} \leq Q^{\mathrm{u}}, k \in K, u \in U, r \in V^{\mathrm{tuac}} \tag{13}$$

$$\sum_{i \in V^{\mathrm{tuac}} \backslash \{0\}} \sum_{j \in V^{\mathrm{tuac}}} q_i x_{kij} + \sum_{r \in V^{\mathrm{tuac}} \backslash \{0\}} \sum_{i \in V^{c}, i \neq r} \sum_{j \in V} \sum_{u \in U} q_i y_{kurij} \leq Q^{\mathrm{t}}, k \in K \tag{14}$$

$$at_{k0}^{t} - dt_{k}^{t} \leq TL, k \in K \tag{15}$$

$$atu_{ku0} - dtu_{ku} \leq UL, k \in K, u \in U \tag{16}$$

$$atu_{kur} - at_{kr}^{t} \leq UL, k \in K, u \in U, r \in V^{\mathrm{tuac}} \backslash \{0\} \tag{17}$$

The constraints identified by Eqs. (13) and (14) acknowledge the UAV and van capacities, respectively. The constraint identified by Eq. (15) respects the maximum working time of each van. The constraints identified by Eqs. (16) and (17) respect the maximum flying time of one takeoff of each UAV in a situation where the UAV travels on a pure UAV route or a sub-tour.

3.2.4. Constraints on vehicle-visiting nodes

$$\sum_{k \in K} \sum_{i \in V^{\mathrm{tuac}}} x_{kij} + \sum_{k \in K} \sum_{u \in U} \sum_{r \in V^{\mathrm{tuac}}} \sum_{i \in V} y_{kurij} \geq 1, j \in V^{\mathrm{tuac}} \backslash \{0\} \tag{18}$$

$$\sum_{k \in K} \sum_{i \in V^{\mathrm{tuac}}} x_{kij} \leq 1, j \in V^{\mathrm{tuac}} \backslash \{0\} \tag{19}$$

$$\sum_{k \in K} \sum_{u \in U} \sum_{r \in V^{\mathrm{tuac}}} \sum_{i \in V} y_{kurij} = 1, j \in V^{\mathrm{uac}} \tag{20}$$

$$\sum_{i \in V^{\mathrm{tuac}}} x_{kij} = \sum_{a \in V^{\mathrm{tuac}}} x_{kja}, k \in K, j \in V^{\mathrm{tuac}} \tag{21}$$

$$\sum_{i \in V} y_{kurij} = \sum_{a \in V} y_{kurja}, k \in K, u \in U, r \in V^{\mathrm{tuac}}, j \in V \tag{22}$$

If a VUAC is not used as a mobile satellite, the VUAC should be served by a van since it is assumed that all customer demand must be satisfied. If a VUAC is used as a mobile satellite, the VUAC should be served by a van, furthermore the VUAC is used as the origin and the destination of a sub-tour. Each VUAC is visited at least once by vans and UAVs, which is indicated by the constraint identified by Eq. (18). It is assumed that customer demand cannot be split, the constraint identified by Eq. (19) indicates that each VUAC is served at most once by vans, and the constraint identified by Eq. (20) indicates that each UAC is visited exactly once by one used UAV. The constraint identified by Eq. (21) ensures that, for each van, the number of arrivals at a node equals the number of departures from the same node. The constraint identified by Eq. (22) ensures that, for each UAV, the number of arrivals at a node equals the number of departures from the same node.

3.2.5. Constraints on time continuity

$$dt_k^t + d_{0i}/v^t + M \cdot (1 - x_{k0i}) \geq at_{ki}^t, k \in K, i \in V^{\mathrm{tuac}} \backslash \{0\} \tag{23}$$

$$dt_k^t + d_{0i}/v^t - M \cdot (1 - x_{k0i}) \leq at_{ki}^t, k \in K, i \in V^{\mathrm{tuac}} \backslash \{0\} \tag{24}$$

$$e_i \leq at_{ki}^t + wt_{ki}^t \leq l_i, k \in K, i \in V^{\mathrm{tuac}} \backslash \{0\} \tag{25}$$

The constraints identified by Eqs. (23) and (24) indicate that a van's arrival time at its first VUAC is decided by the van's departure time at the DC and the travelling time on the arc connecting the DC and the first VUAC. The constraint identified by Eqs. (25) guarantees that a van should visit VUACs on condition of respecting VUAC time windows. A van is permitted to arrive before VUAC time window opening and the van waiting-time should be considered.

$$dtu_{ku} + d_{0i}/v^u + M \cdot (1 - y_{ku00i}) \geq atu_{kui}, k \in K, u \in U, i \in V^c \tag{26}$$

$$dtu_{ku} + d_{0i}/v^u - M \cdot (1 - y_{ku00i}) \leq atu_{kui}, k \in K, u \in U, i \in V^c \tag{27}$$

$$\begin{aligned} &atu_{kui} + wtu_{kui} + s_i + d_{ij}/v^u + M \cdot (1 - y_{kurij}) \geq atu_{kuj}, k \in K, u \in U, \\ &\quad r \in V^{\mathrm{tuac}}, i \in V^c, i \neq r, j \in V \end{aligned} \tag{28}$$

$$\begin{aligned} &atu_{kui} + wtu_{kui} + s_i + d_{ij}/v^u - M \cdot (1 - y_{kurij}) \leq atu_{kuj}, k \in K, u \in U, \\ &\quad r \in V^{\mathrm{tuac}}, i \in V^c, i \neq r, j \in V \end{aligned} \tag{29}$$

$$e_j \leq atu_{kuj} + wtu_{kuj} \leq l_j, k \in K, u \in U, r \in V^{\mathrm{tuac}}, j \in V^c, j \neq r \tag{30}$$

The constraints identified by Eqs. (26) and (27) indicate that in the direct delivery a UAV's arrival time at its first customer is decided by the UAV departure time at the DC and the flying time on the arc connecting the DC and the first customer. The constraints identified by Eqs. (28) and (29) indicate the time relationship between the arrival time of a UAV at node i and that at the next node j. The constraint identified by Eqs. (30) guarantees that a UAV should visit customers by respecting customer time windows. A UAV is permitted to arrive before customer time window opening and the UAV waiting-time should be considered.

From the model construction perspective, the 2E-VRP-TM is a new variant, compared with the 2E-VRP and variants in the literature. In the 2E-VRP-TM formulation, constraints on mobile-satellite synchronization ensure the echelon interaction, which provide a new methodological way to construct the formulations for routing problems of two-echelon networks. On a combination route, the main-tour should respect the van capacity constraint, and sub-tour(s) should respect the UAV capacity constraint. The mobile satellite capacity is limited by the van capacity. Vehicles flow from the DC to mobile satellites, then to customers, and finally to return to the DC. These constraints ensure the route interaction between the two echelons. Some constraints are introduced to identify mobile satellite locations, i.e., only if a UAV carried by a van takes off at a node, the node is used as a mobile satellite. Mobile satellites should be included in main tours of combination routes.

The exact method of the commercial tool, CPLEX 12.9, was used to directly solve the mathematical formulation. Our computational trials indicated that the exact method of CPLEX 12.9 could obtain exact solutions of some small-scale instances. There were two aims for attaining exact solutions of small-scale instances. First, exact solutions of small-scale instances imply the validity of the 2E-VRP-TM formulation. Second, exact solutions of small-scale instances are used to test and calibrate the heuristic. For practical applications, a commercial tool is seldom used to attain exact solutions, because practical-size instances are usually large-scale and use too much computer memory and computation time. To find satisfactory solutions of large-scale instances, we propose an ALNS. Once the effectiveness of the ALNS is addressed by comparing exact solutions with heuristic solutions of some small-scale instances, some large-scale instances can be solved using the heuristic.

4. An ALNS heuristic

Ropke and Pisinger (2006) introduced an ALNS that used multiple destroy-and-repair operators within the same search process. At each iteration of the large-neighborhood search, part of the current solution was destroyed and then repaired in a different way, so that a new solution could be generated. The new solution was accepted according to a predetermined criterion. A pair of destroy-and-repair operators were selected according to an adaptive probabilistic mechanism. The probability of selecting a given pair of operators depended on how well it performed in past iterations.

The 2E-VRP-TM formulation seems very complex, because mobile satellite synchronization and time constraints are especially considered. Thus, we adopted an ALNS to solve the 2E-VRP-TM. In the literature, the ALNS was one of the most effective heuristic tools for solving many VRP variants (Ropke and Pisinger, 2006; Ribeiro and Laporte, 2012; Demir et al., 2012; Hemmelmayr et al., 2012; Azi et al., 2014; Alinaghian and Shokouhi, 2018; Chen et al.,2018; and Liu et al., 2019).

4.1. Constructing the initial solution

From a methodological perspective, constructing pure UAV routes or van routes for direct delivery from the DC means solving the VRP with time windows (VRPTW) on the condition that customers served by pure UAVs or pure vans are identified. It seems complex to construct combination routes, because the mobile-satellite location problem and synchronization constraints are involved. The phase of constructing the initial solution includes three steps: classifying customers; constructing pure UAV routes and van routes; and seeking mobile satellites and constructing sub-tours.

4.1.1. Classifying customers

The operation of classifying customers aims to identify customers of UAV and van routes and customers of sub-tours. For customer i in set V^c, the demand and travelling distance between the DC and customer are used to judge whether customer i should be selected. If the demand of customer i is not more than the UAV capacity, and the UAV travelling between the DC and customer i exceeds the maximum flying distance, customer i is removed from V^c and added to the set of sub-tour customers (*SubCustomerList*). The maximum flying distance of one takeoff is the product of UL and v^u. Customer i is selected, removed from V^c, and added to *SubCustomerList* only if $q_i \le Q^u$ and $d_{0i} > 0.5 \cdot UL \cdot v^u$. When the operation of classifying customers finishes, the two sets, V^c and *SubCustomerList*, are renewed.

4.1.2. Constructing pure UAV routes and van routes

When the renewed V^c is not empty, pure UAV routes are first constructed one by one. Each constructed pure UAV route is denoted as *URoute*. All *URoutes* are included in the set of routes in the solution (denoted as *RoutePlan*). In one *URoute*, the first customer is selected as follows. If the demand of customer i in the renewed V^c exceeds the UAV capacity, customer i is removed from V^c and added into the set of van-route customers (denoted as *VanCustomerList*). If each customer in V^c can be served via UAV direct delivery from the DC, the customer in V^c with the minimum latest service-starting time of customer time windows is inserted as the first customer in *URoute*. Then, the customer with the maximum savings is added, step-by-step, on the condition that the UAV's capacity and UL are respected. For inserting a customer, the savings is estimated as $S_{ij} = (d_{i0} + d_{0j} - d_{ij})/v^u - wt_{(URoute)j}$, where the last customer of *URoute* and the inserting customer are denoted as customers i and j, respectively. If the UAV leaves from customers i, flies on arc (i, j), and arrives at customer j at the time window of customer j, $wt_{(URoute)j}$ equals 0; otherwise, $wt_{(URoute)j}$ equals the UAV's waiting time at customer j for the time window of customer j opening.

When the set, *VanCustomerList*, is not empty, van routes are constructed, one by one. Each constructed van route is denoted as *TRoute*. All *TRoutes* are included in *RoutePlan*. To construct one *TRoute*, the customer in *VanCustomerList* with the minimum latest service-starting time of customer time windows is inserted as the first customer. Then, the customer with the maximum savings is added to *TRoute*, step by step, on the condition that the van's capacity and TL are respected. When inserting a customer, the savings is estimated as $S_{ij} = (d_{i0} + d_{0j} - d_{ij})/v^t - wt_{(TRoute)j}$, where the last customer of *TRoute* and the inserting customer are denoted as customers i and j, respectively. If the van leaves from customers i, runs on arc (i, j), and arrives at customer j at the time window of customer j, $wt_{(URoute)j}$ equals 0; otherwise, $wt_{(URoute)j}$ equals the van's waiting time at customer j for the time window of customer j opening.

4.1.3. Seeking mobile satellites and constructing sub-tours

Constructing one sub-tour involves selecting a mobile satellite as the origin and destination of the sub-tour and assigning customers to it. We introduce two parameters, denoted RT and TS. If the energy unit of a UAV is not exhausted after it finishes its assigned route, the remaining energy can be used to loiter, and that time is called RT. En route, RT is actually the maximum time, ahead of which the UAV may take off on the condition that it returns to its origin. During a route, for each customer (e.g., customer i) there is the difference (TS_i) between the van or UAV arrival time at customer i and the latest service-starting time of the time window of customer i. TS is the minimum TS_i of all customers of the same route. For a route, TS is actually the maximum time by which the assigned van or UAV can postpone its departure from its origin.

To select a mobile satellite, an assumption of regarding one VUAC included in van routes as the mobile satellite is made. On the condition that the van capacity and the maximum flying time of one UAV takeoff are respected, time continuity constraints are used to test this assumption. Fig. 3 shows examples of selecting a mobile satellite. Each example includes one van route (denoted "DC → customer 1 → customer 2 → DC"), and one UAC (customer 3) for constructing a sub-tour. It is assumed that customer 2 is selected as the mobile satellite to serve customer 3. There are three ways to construct a sub-tour, considering time continuity constraints.

The first way is illustrated in Fig. 3(a). An assigned UAV takes off from customers 2, flies on arc (customer 2, customer 3), and arrives at customer 3 at the time window of customer 3. Customer 2 is used as the mobile satellite, and a sub-tour denoted "customer 2 → customer 3 → customer 2" is constructed. The second way is illustrated in Fig. 3(b). An assigned UAV takes off from customers 2, flies on arc (customer 2, customer 3), and arrives at customer 3 before the time window of customer 3 opens. The UAV has to wait, while the waiting-time is included in the maximum flying time of one takeoff for the UAV. If the waiting-time is not more than RT (where $RT = UL - (2 \cdot d_{23}/v^u + s_i)$), customer 2 is used as the mobile satellite; otherwise, the departure time of the van route is postponed, so that the time of UAV taking off at customer 2 is also postponed to avoid some waiting-time at customer 3. The difference between the van arrival time at customer 1 or 2 and the latest service-starting time of the time window of customer 1 or 2 are denoted TS_1 and TS_2, respectively. The

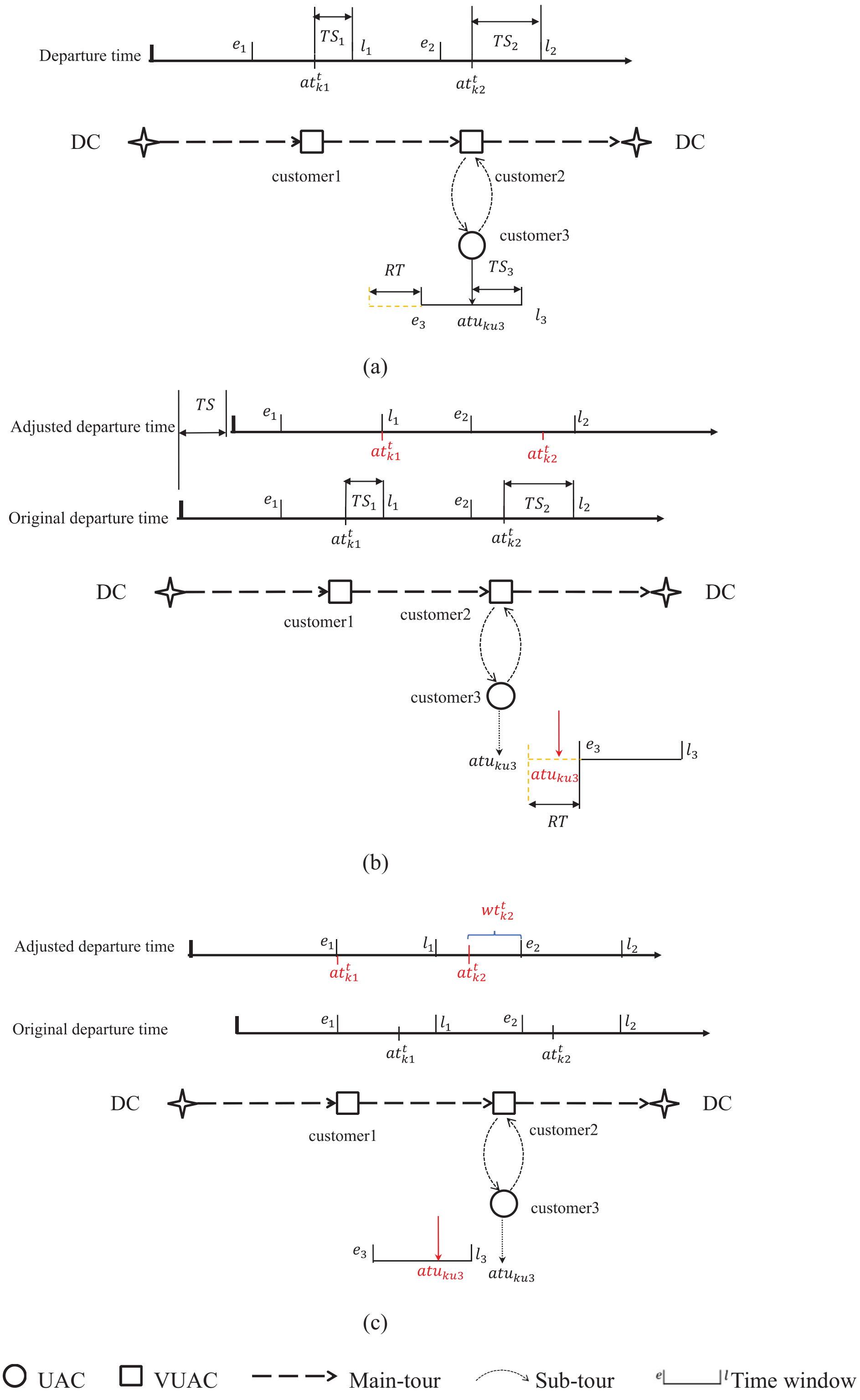


Fig. 3. Illustrative examples of selecting a mobile satellite.

Algorithm 1 Mobile-satellite selection

Input: *TRoute*, the first customer, i, in *SRoute*

Output: The mobile satellite, r_{ms}

The demand of all customers of *TRoute* is estimated and denoted as Q_{TRoute}

for a randomly-selected $r \in TRoute$ **do**

  if $Q_{TRoute} + q_i < Q^{\mathrm{t}}$ **then**

    if $2 \cdot d_{ri} / v^{\mathrm{u}} + s_i < UL$ **then**

      $RT = UL - (2 \cdot d_{ri} / v^{\mathrm{u}} + s_i)$

      $edt^{\mathrm{u}} = e_i - RT - d_{ri} / v^{\mathrm{u}}, ldt^{\mathrm{u}} = l_i - d_{ri} / v^{\mathrm{u}}$

      Estimate the van arriving time window $[eat_r^{\mathrm{t}}, lat_r^{\mathrm{t}}]$ at r, and adjust $[eat_r^{\mathrm{t}}, lat_r^{\mathrm{t}}]$, if necessary

      if $[edt^{\mathrm{u}}, ldt^{\mathrm{u}}] \cap [eat_r^{\mathrm{t}}, lat_r^{\mathrm{t}}] \neq \varnothing$ **then**

        customer r is used as the mobile satellite

        $r_{ms} = r$

        $at_{kr}^{\mathrm{t}} = \min\{ldt^{\mathrm{u}}, lat_r^{\mathrm{t}}\}$

      end if

    end if

  end if

end for

Return r_{ms}

Fig. 4. Pseudocode for selecting a mobile satellite.

departure time of the van route is postponed by TS=min{TS_1, TS_2}. On condition that the UAV waiting-time at customer 3 is not more than RT after the departure time of the van route is postponed, customer 2 is used as the mobile satellite. The third way is illustrated in Fig. 3(c). An assigned UAV takes off from customers 2, flies on arc (customer 2, customer 3), and arrives at customer 3 after the time window of customer 3 closes. The departure time of the van route is advanced, so that the time of UAV taking off at customer 2 is also advanced to arrive at customer 3 on time. On condition that the UAV arrives at customer 3 at its time window after the departure time of the van route is advanced, customer 2 is used as the mobile satellite. If customer 2 cannot be used as a mobile satellite by any one of the three ways, customer 2 in the assumption is substituted with another VUAC included in van routes.

Sub-tours of combination routes are constructed one by one, and each sub-tour is denoted as *SRoute*. The process of seeking mobile satellites and constructing sub-tours is as follows. First, in the set, *SubCustomerList*, the customer (i) with the minimum latest service-starting time of time windows is inserted as the first customer of *SRoute*. Second, in each *TRoute* included in *RoutePlan*, one VUAC (customer r) is selected as the mobile satellite to serve customer i, on the condition that the van capacity and the maximum flying time of one UAV takeoff and time continuity constraints are respected. Fig. 4 shows the algorithm (Algorithm 1) for selecting the mobile satellite. Third, based on *SRoute* including the only customer, the customer with the maximum savings is added to *SRoute*, step by step, on the condition that the constraints on van or UAV capacity, constraints on customer time windows, and constraints on the maximum flying time of one takeoff of each

Algorithm 2 Solving the 2E-VRP-TM

1: Construct an initial solution, s_0, in which direct services are permitted;

2: Initialize best solution $s^* = s_0$, current solution $s' = s_0$;

3: Employ an ALNS (i.e., **Algorithm 3**) to improve the routes to obtain a new solution, s'', whose objective value is the most minimized. Update $s^* = s''$;

4: Return the final best solution, s^*.

Fig. 5. Heuristic for solving the 2E-VRP-TM.

Table 4
The values of σ_1, σ_2 and σ_3.

ValuesPercentage gaps	σ_1	σ_2	σ_3
$\geq$ 0.1	90	30	5
$\geq$ 0.01	80	25	5
$\geq$ 0.001	70	20	5
$\geq$ 0.0001	60	15	5
$<$ 0.0001	50	10	5

UAV are respected. For inserting a customer, the savings is estimated as $S_{ij} = (d_{ir} + d_{rj} - d_{ij})/v^{\mathrm{u}} - wt_{(URoute)j}$, where the last customer of *SRoute* and the inserting customer are denoted as customers *i* and *j*, respectively.

4.2. The ALNS

There is probably a situation in which no mobile satellite can be found for serving certain customers in *SubCustomerList*, because of tight van-route time constraints. To guarantee all customers are included in the initial solution, the constraint on the maximum flying time of one takeoff for each UAV is relaxed. The initial solution includes some direct services. In a direct service, a UAV departs from the DC, covers a customer, and returns directly to the DC. All constraints except that of the maximum flying time of one takeoff for each UAV are respected by direct services. Corresponding to each direct service, a penalty cost, denoted by a large positive integer, is added to the objective value.

The process of the whole heuristic for solving the 2E-VRP-TM is briefly described in the pseudocode of Algorithm 2 (Fig. 5). The heuristic starts from an initial solution that probably includes direct services. Then, the solution is improved with an ALNS (i.e., Algorithm 3).

In Algorithm 3 (Fig. 6), γ is the number of optimization phases, and φ is the number of iterations for each optimization phase. In earlier optimization phases, a best solution is easily attained by certain destroy-and-repair operator, and each pair of destroy-and-repair operators may have an effect. In later optimization phases, most pairs of destroy-and-repair operator have similar probabilities to be selected for pursuing a best solution. To avoid the situation that there is no difference between the adaptive weights of the best and worst pairs selected during an optimization phase, the number of iterations for each optimization phase is not fixed but changes according to the adaptive weights. In our computational experiments, between the difference between the adaptive weights of the best and worst pairs selected and the number of iterations for one optimization phase, a negative correlation relationship is indicated. During the first phase, $\phi = 100$; otherwise, $\varphi = 100 + \frac{1}{\Delta P}$, where ΔP is the difference between the adaptive weights of the best and worst pairs selected during the last optimization phase. If there is no difference between the adaptive weights of the best and worst pairs, we assume that $\Delta P = 0.02$.

After ϕ iterations, the weight of a pair of destroy-and-repair operators referred for selection is updated according to a score. Referring to the quality of the newly obtained solutions, the score is increased by σ_1, σ_2, or σ_3. If a pair of destroy-and-repair operators finds a new best solution, the score is increased by σ_1. If a pair of destroy-and-repair operators finds a solution that is better than the current one, the score is increased by σ_2. If a pair of destroy-and-repair operators finds a non-improving solution that is accepted according to the simulated annealing criterion, the score is increased by σ_3. All scores are set to zero in the next optimization phase. We classify σ_1 according to the percentage gap between the new and the current best solutions, and classify σ_2 according to the percentage gap between the new and current solutions. Based on computational trials for some instances, the values of σ_1, σ_2, and σ_3 are estimated in Table 4.

We introduce π_i to denote the score of the *i*-th pair of destroy-and-repair operators and O_{ij} to denote the number of times the *i*-th pair of destroy-and-repair operators are selected during the last optimization phase *j*. The adaptive weights are estimated as $\omega_{i,j+1} = \omega_{i,j}$ (when $O_{ij} = 0$), and $\omega_{i,j+1} = (1-\eta)\omega_{ij} + \frac{\eta\pi_i}{O_{ij}}$, (when $O_{ij} \neq 0$). Parameter η controls the inertia in the weight-update operation. When η is close to 0, the history prevails. When η is close to 1, the update operation is driven by the most recent score (Azi et al., 2014). Computational trials on some instances show that, during earlier optimization phases, each pair of destroy-and-repair operators may have effectiveness, whereas, in later optimization phases, only some pairs can improve the solution. To control the weight adjustment, we define $\eta = \min\{(0.002 \times \gamma - 0.001), 0.999\}$.

Algorithm 3. ALNS for improving the routes

Initialize ψ=0, $\gamma = 0$;
Repeat
  Initialize $\omega_{i,0}$=1 $(i = 1,2,\ldots,6)$, $\gamma := \gamma + 1$, $n = 0$;
  While $n < \varphi$, **do**
    Using the roulette-wheel mechanism to select a destroy-and-repair pair;

$$\frac{\omega_{i,n}}{\sum_{j=1}^{6} \omega_{j,n}}$$

    Apply the pairs of destroy-and-repair operators to get a new solution, s'';
    If $f(s'') < f(s^*)$ then
      $s^* = s' = s''$;
      Update the score with σ_1;
    Else if $f(s'') < f(s')$ then
      $s' = s''$;
      Update the score with σ_2;
    Else if s'' is accepted by the simulated annealing criterion, then
      $s' = s''$;
      Update the score with σ_3;
    End if
    $n := n + 1$, adjust weights $\omega_{i,n}$ $(i = 1,2,\ldots,6)$;
    If the number of iterations reaches φ;
      Reset the weights, $\omega_{i,n}$=1 $(i = 1,2,\ldots,6)$;
      Modify φ of the next phase according to ΔP;
    $\psi := \psi + 1$;
Until the stopping criterion is met;
Return best solution, s^*;

Fig. 6. Pseudocode of the ALNS.

In the simulated annealing criterion, if $f(s'') > f(s')$, s'' is accepted with a probability of $e^{\frac{-(f(s'')-f(s'))}{T}}$, where $T_{start} = 0.07f(s_0)$, $T = T_{start} \times c^{\psi}$, and $c = 0.99975$. The temperature starts at T_{start} and is multiplied by c (i.e., cooling rate) at every iteration.

The destroy operators include Shaw removal, worst removal, and random removal. The Shaw removal operator, which was introduced in Shaw (1998), selects and removes customers that are similar to each other. To estimate the degree of similarity between two customers, the distance between them is used. Because the 2E-VRP-TM considers the maximum flying time of one takeoff for each UAV, customers en a sub-tour are generally near to the mobile satellite. The Shaw removal operator is used to bring about sub-tour changes. The worst removal operator removes one customer whose removal results in the greatest savings. The customer with the largest difference of objective values of the two types of solutions is removed. Because the 2E-VRP-TM considers the transformed cost of possible waiting-times of vehicles at customers, the worst removal operator should involve both distance and time estimation. The random-removal operator randomly selects and removes a predetermined number of customers from the current solution. The random-removal operator does not consider route types and customer types, and we use the random-removal operator to generate large enough neighborhoods.

In the literature (e.g., Ropke and Pisinger, 2006; Hemmelmayr et al., 2012; Ribeiro and Laporte, 2012), the repair operators generally used by the ALNS include greedy and regret-k insertions. In the paper, the repair operators include deep-greedy and regret-2 insertions. The deep-greedy insertion operator involves estimating the cost change of the solution incurred by inserting each removed customer to any feasible position. When one customer is inserted to a position that increases the objective function the least, the customer insertion is accepted. There may be various types of routes in a solution, the best insertion of one node can affect the insertion operations of other nodes. The regret-2 insertion decides where to insert one customer with the largest cost difference between the best and the second-best positions. We use the regret-2 insertion to avoid local optimum. To employ insertion operators, customer types are identified first. UACs, who should be inserted into pure UAV routes or sub-tours of combination routes, are especially considered. Moreover, there are different ways to insert a VUAC, i.e., constructing a new pure UAV route, inserting into a constructed pure UAV route, constructing a new van route,

Table 5
Values of involved parameters.

Parameters	Value (unit)	Parameters	Value (unit)
Number of UAVs carried by a van	2 (UAV)	c^u	15 (Chinese Yuan per takeoff)
Q^t	1000 (kg)	v^t	60 (km/h)
Q^u	5 (kg)	v^u	65 (km/h)
TL	10 (h)	τ	1
UL	0.67 (h)	M	100
c^t	0.8 (Chinese Yuan per km)		

inserting into a constructed van route, constructing a new sub-tour, inserting into a constructed sub-tour, and constructing a new combination route with the VUAC as a mobile satellite.

We define 10 consecutive optimization phases as an optimization group. When the difference between the objective values obtained by two adjacent optimization groups is 0, or the allowed number of global iterations (i.e., ψ) is exhausted, the iteration stops. In the computational experiments, ψ=20000.

5. Computational experiments

We adopted computational experiments to evaluate the validity of the 2E-VRP-TM formulation and the effectiveness of the heuristic. In some instances, we used CPLEX 12.9 solver to solve directly the mathematical formulation. The ALNS provided in Section 4 was coded in C++. The codes for the heuristic and the exact method were run on a Windows 8 computer configured with an Intel(R) Core(TM) 3.2-GHz processor with 8-GB memory.

5.1. Test instances

There were no test instances for the direct applicability of the 2E-VRP-TM model. We chose some VRPTW test problems provided by Solomon (1987) and converted them to 2E-VRP-TM instances. We used 21 small-scale instances to evaluate the validity of the 2E-VRP-TM formulation and the effectiveness of the heuristic. We used 27 large-scale instances to show the applicability of the heuristic.

We derived the test instances from the Solomon VRPTW benchmark problems with 100 customers. Solomon (1987) benchmark instances were classified into three categories and six sets, among which, C1 includes clustered customers with a narrow scheduling horizon. We selected the nine instances with 100 customers from C1 to generate test instances. The demand of a VUAC or a UAC is a random number in the range of $[0, Q^t/20]$ or $[0, Q^u/2]$, respectively. Customer time windows are at the range of [0, 18]. The center of the time window of a customer is estimated by the method presented in Solomon (1987). The range of a customer time window is a random number. The service time at a VUAC is 15 min, and the service time at a UAC is 5 min. The parameters involved in the 2E-VRP-TM formulation are evaluated in Table 5.

i) Large-scale instances. We modified the VRPTW benchmark instances by classifying customers into VUACs and UACs. Each of the nine VRPTW instances was converted into three types of 2E-VRP-TM test instances using a method similar to what Chao (2002) used. For each customer, i, in a VRPTW benchmark instance, the distance between customer i and the neighboring customer is denoted by A_i. In the first type of 2E-VRP-TM instance, 25% of customers with the smallest A_i were specified as UACs. This percentage increased to 50% or 75% for the second and third types of 2E-VRP-TM instances, respectively. Each large-scale instance is denoted by C10*j*-*p*, where j=1, 2, …, 9, and p is the percentage of UACs and p=0.25, 0.50, or 0.75.
ii) Small-scale instances. A number of customers (nc) included in the VRPTW benchmark instances were randomly selected to generate small-scale instances. The nc customers are classified into VUACs and UACs using the same method of classifying customers for generating large-scale instances. Each small-scale instance is denoted by C1-*nc*-*p*, where nc=6, 7, …, 12.

5.2. The effectiveness of the formulation and the ALNS

For every small-scale instance, CPLEX 12.9 runs with default settings until it finds the exact solution or until it stops, because computer memory or the predetermined computation time is exhausted. We set the predetermined computation time 14400 s. Exact results of the small-scale instances are shown in Table A1 in the Appendix. Heuristic results of the small-scale instances are presented in Table A2 in the Appendix. The computational results of 21 small-scale instances imply that the mathematical formulation performs well while the ALNS is effective.

i) Compared with the exact method by CPLEX 12.9, the ALNS can output the satisfactory solution very quickly. The ALNS can solve each of the 21 small-scale instances in less than 10.0 s. The average computational time is 2.3 s. Of the 21 instances, the computational time is less than 2.3 s in 13 cases; the longest computational time is 9.0 s. Although the numbers of variables and constraints included in each of the 21 small-scale instances vary remarkably, the computational time of the ALNS does not fluctuate obviously.

ii) From the perspective of solution quality, the proposed ALNS performs well. Of the 17 small-scale instances having exact solutions, the percentage gap between the objective value of the ALNS solution and that of the exact solution of each instance, i.e., Gap0=100%×(Obj^{ALNS}-Obj^{cplex})/Obj^{ALNS}, is 0.00% in all the 17 cases, indicating that the 17 instances can be solved optimally with the ALNS.

iii) CPLEX12.9 requires the full four-hour limit for 4 instances with up to 11 or 12 customers, which indicates that none of the solutions to the formulation are provably optimal. Of the 4 instances that CPLEX 12.9 may not find the exact solutions in the limited computation time, Gap0 is 0.00% in 3 cases, and the average equals 0.04%. From the objective value perspective, the ALNS solution is very close to the solution found by CPLEX 12.9, while the computation time of the ALNS is obviously less than the computation time of CPLEX 12.9 (the comparison may be unfair for CPLEX solver).

iv) For the heuristic results, the gap (i.e., Gap1=100%×(Obj^{I}-Obj^{ALNS})/Obj^{I}) between the objective value of the initial solution and that of the ALNS solution is large. Of all the 21 instances, Gap1 is more than 11.00%. The average Gap1 equals to 25.70%. The largest Gap1 is 52.41%. Gap1 is less than 25% in 11 cases, and Gap1 is more than 30% in 8 cases. It implies that the ALNS is necessary and important.

v) For several instances (i.e., instance C1-6-0.75, C1-10-0.75, and C1-11-0.75), N^{t}, N^{u}, N^{mr}, and N^{sr} of the exact solution are different from N^{t}, N^{u}, N^{mr}, and N^{sr} of the ALNS solution, although the objective value of the exact solution equals the objective value of the ALNS solution. It indicates that some routes can be adjusted on condition of ensuring the optimized objective value, which is useful for the tactical or operational decision.

vi) Heuristic results of the 21 small-scale instances indicated that the average of the integrated cost per customer was 14.95 (unit: Chinese Yuan), and the standard deviation of the integrated cost per customer was 3.75 (unit: Chinese Yuan). Heuristic results imply that the integrated cost per customer decreases with the number of customers increases.

vii) Heuristic results indicated that for 81 percentage of the 21 small-scale instances, $N^{u} \geq N^{sr}$, that is, in 17 cases the number of pure UAV routes for the direct delivery was more than the number of sub-tours.

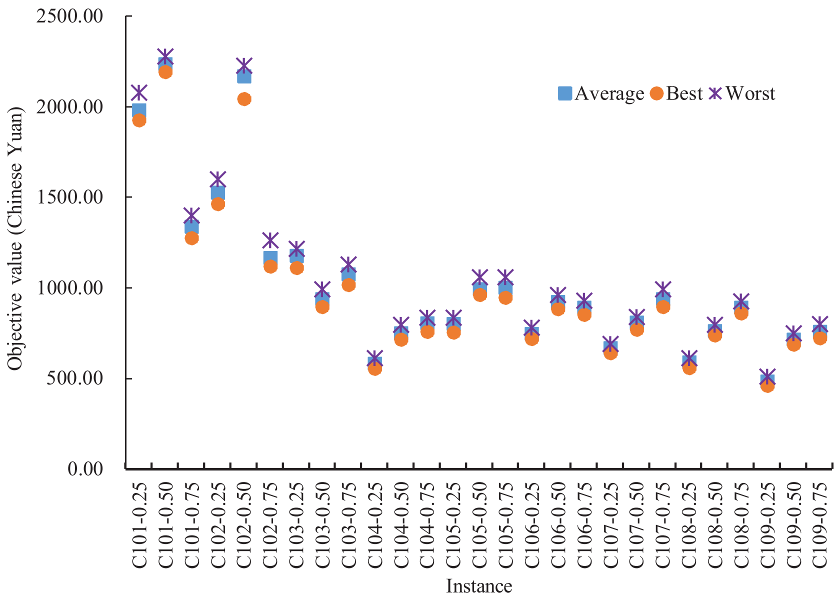
By comparing heuristic solutions with exact solutions of small-scale instances, the effectiveness of the ALNS, indicated by the high quality of heuristic solutions, is addressed. Based on the validity of the 2E-VRP-TM formulation and the effectiveness of the heuristic, the ALNS can be adopted to solve large-scale instances.

5.3. The heuristic applicability for large-scale instances

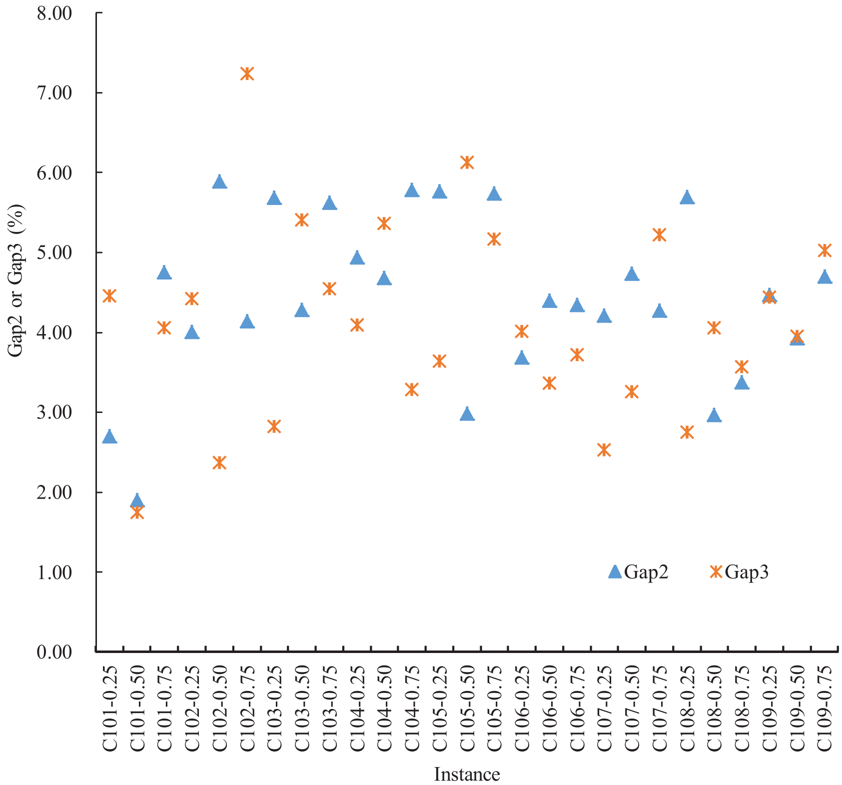
For each large-scale instance, the ALNS is repeated 10 times. The heuristic objective values of the best, the worst or the average solution in the ten repetitions are reported in Fig. 7(a). For each large-scale instance (Fig. 7(b)), the gap between the heuristic objective value of the best solution (denoted as OBS) and the average objective value of solutions of the ten repetitions (denoted as OAS), which is described as Gap2=100%×(OAS-OBS)/OAS, is between 1.90% and 5.89% and the average equals to 4.44%. The gap between the heuristic objective value of the worst solution (denoted as OWS) and the average objective value of solutions of the ten repetitions (i.e., OAS), which is described as Gap3=100%×(OWS-OAS)/OWS, is between 1.75% and 7.24% and the average equals to 4.10%. The heuristic running times of the best, the worst or the average solution in the ten repetitions are reported in Fig. 8(a). For each large-scale instance (Fig. 8(b)), the gap between the heuristic running time of the best solution (denoted as TBS) and the average time of the ten repetitions (denoted as TAS), which is described as Gap4=100%×(TAS-TBS)/TAS, is between -17.63% and 38.78% and the average equals to 3.66%. The gap between the heuristic running time of the worst solution (denoted as TWS) and the average time of the ten repetitions (i.e., TAS), which is described as Gap5=100%×(TAS-TWS)/TAS, is between -19.97% and 29.18% and the average equals to -0.75%. It implies that the ALNS is reliable while the randomness plays a role in the ALNS. Besides, when the percentage of UACs (i.e., p) is 25%, i.e., there are 75 VUACs that can be selected as mobile satellites, the average running time of the ALNS is relatively long. When p is 75%, i.e., there are 25 VUACs that can be selected as mobile satellites, the average running time of the ALNS is relatively short.

The initial and final results of the best solution of the 10 results are reported in Table 6. Column "Instance" indicates the instance. Column "Obj^{I}" or "Obj^{ALNS}" shows the objective value of the initial solution or that of the ALNS solution. Columns "N^{t}", "N^{u}", "N^{mr}", and "N^{sr}" show the number of van routes, the number of pure UAV routes, the number of main-tours, and the number of sub-tours of combination routes in the ALNS solution, respectively. Column "CN^{t}" shows the number of van-route and main-tour customers. Column "CN^{u}" shows the number of UAV-route and sub-tour customers. Column "T^{ALNS} (s)" shows the computation time (s) of the ALNS. Column "Gap1" shows the percentage gap between the objective value of the initial solution and that of the ALNS solution of each instance. The proposed heuristic can solve large-scale instances relatively quickly. Referring to the computation times of the best solution, the ALNS can solve any one of the 27 large-scale instances in less than 1,500 seconds. Of the 27 instances, the computational time was less than 750 s in 16 cases. The longest computational time was 1,426.3 s, and the average computational time was 750 s. The largest Gap1 was 62.05%, and the average Gap1 was 31.51%. Of the large-scale instances, the average Gap1 was larger than the average Gap1 of the small-scale instances. The scale of each large-scale instance was larger than that of any small-scale instance, and the initial solution of a large-scale instance can have more neighborhoods than small-scale instances.

Our computational experiments on the 27 large-scale instances indicated that the instance data considerably affected the heuristic solutions. We present specific conclusions as follows.



(a)



(b)

Fig. 7. The heuristic objective values for the best, worst or average solution in the ten repetitions.

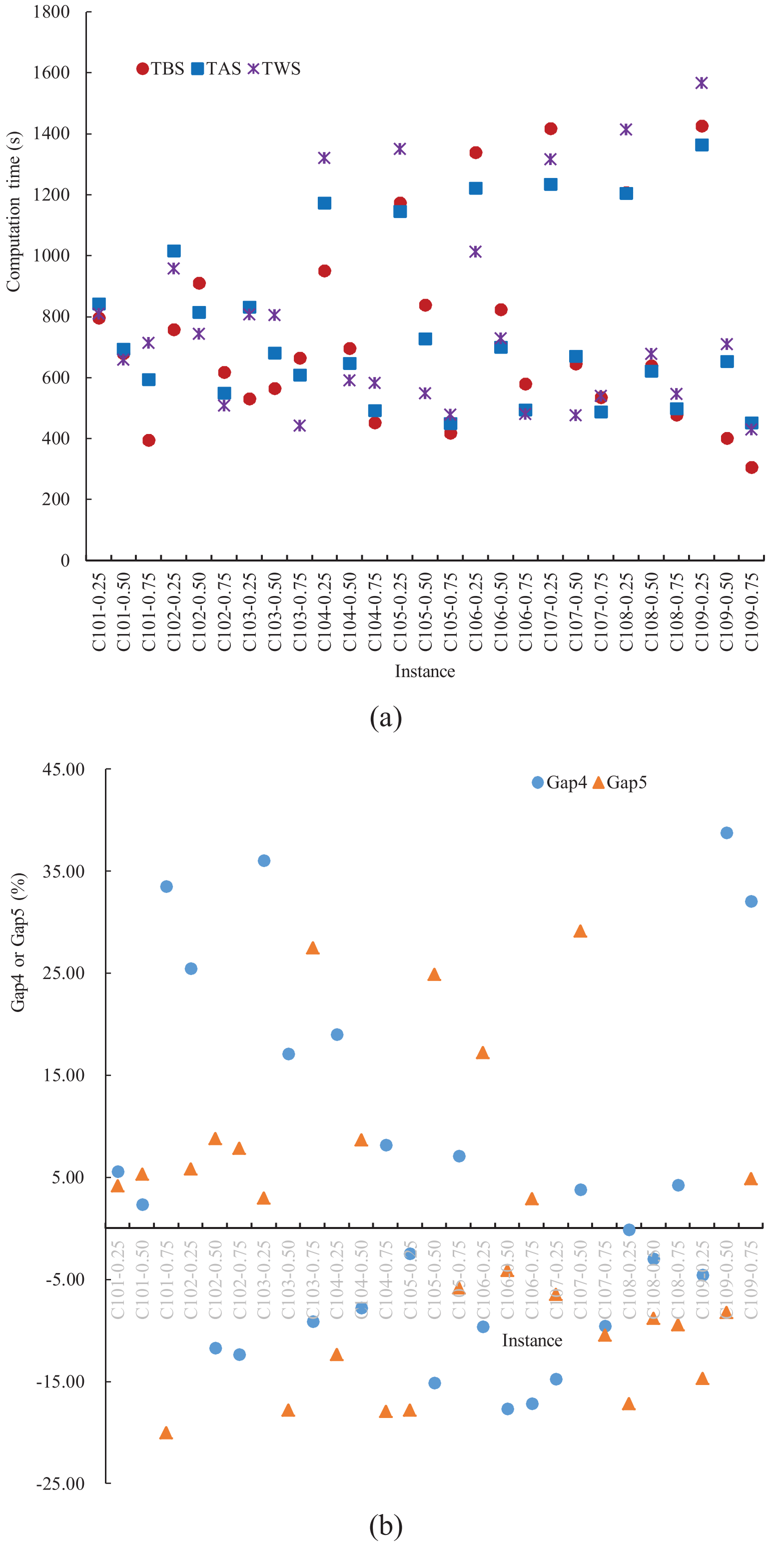


Fig. 8. The heuristic running times for the best, worst or average solution in the ten repetitions.

Table 6
Heuristic results of large-scale instances.

Instance	The initial solution	The ALNS solution								Gap1
	Obj^{I}	Obj^{ALNS}	N^{t}	N^{u}	N^{mr}	N^{sr}	CN^{t}	CN^{u}	T^{ALNS} (s)	
C101-0.25	2773.92	1928.51	5	18	5	5	73	27	796.4	30.48
C101-0.50	2595.12	2193.28	2	33	9	11	49	51	678.2	15.48
C101-0.75	1660.76	1276.15	1	39	6	8	25	75	395.6	23.16
C102-0.25	1787.60	1464.34	2	11	5	6	74	26	758.0	18.08
C102-0.50	2620.76	2043.65	1	25	10	11	50	50	910.0	22.02
C102-0.75	1458.48	1119.61	0	31	6	10	25	75	618.5	23.23
C103-0.25	1571.36	1110.67	2	8	3	4	74	26	531.7	29.32
C103-0.50	1172.92	897.88	0	18	4	7	50	50	565.0	23.45
C103-0.75	1483.08	1016.50	0	31	5	7	25	75	664.1	31.46
C104-0.25	1139.52	554.22	3	7	2	2	75	25	950.5	51.36
C104-0.50	992.20	717.65	1	17	4	5	49	51	697.1	27.67
C104-0.75	997.40	758.44	0	23	4	7	25	75	453.0	23.96
C105-0.25	1251.80	756.69	5	9	2	2	75	25	1173.4	39.55
C105-0.50	1423.88	963.60	2	22	3	4	50	50	839.0	32.33
C105-0.75	1128.63	945.55	0	32	3	5	25	75	418.7	16.22
C106-0.25	1123.96	720.16	5	11	1	1	75	25	1339.6	35.93
C106-0.50	1328.32	884.16	3	21	2	3	50	50	823.7	33.44
C106-0.75	1064.79	853.92	2	32	3	4	25	75	579.8	19.80
C107-0.25	1374.80	641.41	3	9	1	1	75	25	1417.6	53.35
C107-0.50	1248.80	771.82	1	19	3	3	50	50	645.7	38.20
C107-0.75	1087.12	897.20	0	32	4	6	25	75	535.6	17.47
C108-0.25	1186.44	558.00	3	8	2	2	75	25	1207.2	52.97
C108-0.50	1208.00	739.96	1	17	3	4	50	50	639.8	38.75
C108-0.75	1060.00	860.55	0	28	3	5	25	75	478.1	18.82
C109-0.25	1219.60	462.80	2	7	2	3	75	25	1426.3	62.05
C109-0.50	1139.80	687.58	0	14	3	5	50	50	400.5	39.68
C109-0.75	1071.64	723.15	0	28	2	4	25	75	306.7	32.52

- Currently, several Chinese enterprises are in the race towards UAV-mode delivery. The practice data from an E-business company (i.e., SJ), was referenced to estimate the parameter values of the test instances. From this, a UAV can carry one, two, or three parcels per takeoff. Heuristic results of the 27 large-scale instances indicated that the average of the integrated cost per customer was 9.83 (unit: Chinese Yuan), and the standard deviation of the integrated cost per customer was 4.37 (unit: Chinese Yuan). The heuristic results provided the reference of the integrated cost per customer for enterprises to demonstrate the financial feasibility of the delivery-by-UAV mode.
- The 2E-VRP-TM permits vans to serve as mobile satellites to extend the UAV service area. The number of sub-tours per combination route indicates the use frequency of carried UAVs. We introduced *MSR* to denote the number of sub-tours per combination route in the solution of an instance. For each instance solution, $MSR = N^{sr} / N^{mr}$. For each of the 27 large-scale instances, the smallest *MSR* was 1.0, the largest *MSR* was 1.8, and the average *MSR* was 1.4.
- For the solution of large-scale instances, when more main tours were included, more sub-tours were included. When using a straight line to fit the relationship between the number of main tours and the number of sub-tours in the solution of large-scale instances, we get y=1.1954x+0.5728, where x denotes the number of main tours, and y denotes the number of sub-tours in an instance solution. The correlation coefficient is 0.9339. We can estimate the number of sub-tours by referring to the quantitative relationships and the number of designed main-tours.
- All routes can be classified two ways: those visited by vans (carrying UAVs) and those visited by UAVs. The number of routes visited by UAVs is the sum of N^{u} and N^{sr}, and the number of routes visited by vans is the sum of N^{t} and N^{mr}. We introduced Num^{u} to denote the number of included customers per route visited by a UAV and Num^{t} to denote the number of included customers per route visited by a van. $Num^{u} = CN^{u} / (N^{u}+N^{sr})$ and $Num^{t} = CN^{t} / (N^{t}+N^{mr})$. Of the 27 large-scale instances, the average Num^{u} was 2.1, and the average Num^{t} was 10.2. The largest Num^{u} was 2.8, and the largest Num^{t} was 18.8. Num^{u} was probably affected by UAV capacity and the maximum flying time of one takeoff each.
- Heuristic results of large-scale instances show that as the percentage of UACs increases, the number of van routes generally decreases while the number of pure UAV routes increases. Compared with N^{u} varying with different percentages of UACs, the change of the number of sub-tours is not substantial. The increase in the percentage of UACs leads to the number of VUACs decreases, which means candidate mobile-satellites decrease and some sub-tours become infeasible. Besides, compared with the construction of pure UAV routes, the construction of sub-tours is affected by more factors, especially those on time continuity constraints.
- The mobile-satellite approach, based on van-UAV combinations, was not always advantageous to decreasing integrated costs. For each of the 27 large-scale instances, $N^{u} \geq N^{sr}$, the number of pure UAV routes for the direct delivery was more than the number of sub-tours. Using UAVs for direct delivery from the DC is necessary to save integrated costs. Both practitioners and researchers are suggested to identify suitable situations to adopt the mobile satellite approach and UAV direct delivery.

6. Conclusion

Considering the important realistic features of adopting UAVs carried by vans for parcel deliveries, we introduced and formally defined the 2E-VRP-TM as an innovative method to tackle the operational challenges of optimizing delivery routes for a fleet of van-UAV combinations. These combinations include one van carrying several UAVs. We also introduced mobile-satellite synchronization. When first-echelon vehicles park at certain customer locations and wait for second-echelon vehicles' departures and returns, the first-echelon vehicles are used as mobile satellites. With the 2E-VRP-TM, the first echelon involves van-UAV combinations delivering parcels from the DC to VUACs. The second echelon involves carried UAVs being dispatched from mobile satellites to visit VUAC or UAC locations. It also involves UAVs directly delivering parcels from the DC. The objective of the 2E-VRP-TM is to minimize integrated costs.

We formulated the 2E-VRP-TM by introducing binary variables for locating mobile satellites and continuous variables for time constraints. We proposed an ALNS heuristic. Based on computational experiments, the validity of the 2E-VRP-TM formulation and the effectiveness of the heuristic were evaluated. This study provides a reference for enterprises to demonstrate the feasibility of delivery-by-UAV modes. Suitable situations to adopt the mobile-satellite approach and UAV direct delivery should be identified next.

For future research, more effective exact and heuristic algorithms are expected. It is interesting to estimate the savings compared with traditional delivery modes. Moreover, some extensions to the 2E-VRP-TM considering the situation in which UAVs cannot be feasible again for similar areas so that the origin and destination of a sub-tour do not use the same mobile satellite, considering vans being equipped with different numbers of UAVs, or considering that a UAV in use can be dispatched more than once at a mobile satellite, are expected.

Statement of contribution/potential impact

We believe the paper may be of particular interest to the readers of Transportation Research Part B, as it proposes the two-echelon vehicle routing problem with time windows and mobile satellites (2E-VRP-TM) from the perspective of modeling routing problems of two-echelon networks. We introduce the 2E-VRP-TM for the first time to tackle the logistics challenges faced by enterprises using unmanned aerial vehicles (UAV) with human-driven vans for parcel deliveries. Typically, one van carries several UAVs. The first echelon involves time-window-driven parcel deliveries using vans from a distribution center (DC) to customers. The second echelon involves UAVs being dispatched from mobile-satellite vans to serve customers with time windows and directly delivering parcels from the DC. When the first-echelon vehicles park at customer locations and wait for second-echelon vehicle departures and returns, the first-echelon vehicles are used as mobile satellites.

We focus on a background that in recent years great advancements in the area of the ground vehicles and UAV technologies have brought many new utilization possibilities for UAVs in the logistics sector. Considering the commercial interest in UAV delivery modes, only a few studies have tackled operational challenges. We are already aware of the operational challenges of optimizing delivery routes for a homogeneous fleet of van-UAV combinations. We have tried to find the solution from the literature (the journals as Transportation Research (part B, C), Transportation Science, EJOR, etc.). Savelsbergh and Van Woensel (2016) stated that adequate decision support for multi-echelon models for mobile satellites was missing in the literature and offered an interesting path for future work.

We thus introduce the 2E-VRP-TM for the first time. The 2E-VRP-TM involves mobile satellites of first-echelon routes using a homogeneous fleet of second-echelon vehicles, when possible. Only when first-echelon vehicles park at certain locations and wait for second-echelon vehicles' departure and return can they be used as mobile satellites. Compared with the satellite concept of the 2E-VRP and the 2E-LRP, the mobile-satellite concept indicates a type of temporary moving infrastructure for parcel transshipment and consolidation. The location of a mobile satellite is selected for the carried UAV's departure and return. When en route, each van-UAV combination moves to assigned customer areas to make deliveries. Thus, the locations of mobile satellites vary along with each van's movement.

The contributions of this paper include the introduction and formal definition of the 2E-VRP-TM, which caters to the delivery applicability of a fleet of van-UAV combinations with respect to customer time windows. We propose a vehicle-flow formulation in which mobile-satellite synchronization constraints for constructing combination routes provide a new connection mechanism of formulating routing problems for two-echelon networks. We provide an adaptive large neighborhood search (ALNS) heuristic, and we experimentally evaluate the validity of the 2E-VRP-TM mathematical formulation and the effectiveness and applicability of the proposed heuristic. We also briefly discuss management implications.

Declaration of Competing Interest

None.

Acknowledgment

This research work was supported by the Research Grant from the National Natural Science Foundation of China (grant number 71972007, 71672005). The authors would like to thank the associate editor and two anonymous referees for their insightful comments and suggestions.

Appendix

In Table A1, Column "Instance" shows the instance. Column "Obj^{cplex}" shows the objective value of the exact solution. Obj^{cplex} is estimated by the Chinese Yuan money unit. Column "N^{t}" shows the number of van routes in the solution. Column "N^{u}" shows the number of pure UAV routes in the solution. Column "N^{mr}" shows the number of main-tours of combination routes in the solution. Column "N^{sr}" shows the number of sub-tours of combination routes in the solution. Column "T^{cplex} (s)" shows the computation time (s). Column "Gap^{CPLEX} (%)" shows the gap output by CPLEX 12.9, and Gap^{CPLEX}=100%×(upper boundary-lower boundary)/upper boundary.

Table A1
Exact results of small-scale instances.

Instance	Obj^{cplex}	N^{t}	N^{u}	N^{mr}	N^{sr}	T^{cplex} (s)	Gap^{CPLEX} (%)
C1-6-0.25	110.16	2	2	0	0	2.1	10^{-4}
C1-6-0.50	116.20	1	2	1	1	0.8	10^{-4}
C1-6-0.75	107.00	0	4	1	1	0.7	10^{-4}
C1-7-0.25	110.00	1	1	1	1	462.9	10^{-4}
C1-7-0.50	91.40	0	2	1	1	21.4	10^{-4}
C1-7-0.75	84.72	0	0	1	2	96.0	10^{-4}
C1-8-0.25	161.20	1	1	1	1	159.6	10^{-4}
C1-8-0.50	110.61	0	1	2	2	259.5	10^{-4}
C1-8-0.75	125.84	0	4	1	2	198.3	10^{-4}
C1-9-0.25	159.96	1	0	1	1	686.1	10^{-4}
C1-9-0.50	231.40	1	2	1	1	5633.0	10^{-4}
C1-9-0.75	99.26	0	1	1	2	2670.9	10^{-4}
C1-10-0.25	105.60	1	1	1	1	506.1	10^{-4}
C1-10-0.50	135.84	1	3	1	1	524.5	10^{-4}
C1-10-0.75	115.00	0	4	1	1	11832.4	10^{-4}
C1-11-0.25	113.95	3	1	1	1	14400.0	33.56
C1-11-0.50	165.60	3	3	1	1	2923.6	10^{-4}
C1-11-0.75	158.2	1	4	1	1	4788.7	10^{-4}
C1-12-0.25	134.0	2	1	1	1	14400.0	25.90
C1-12-0.50	148.0	1	3	1	1	14400.0	35.23
C1-12-0.75	172.19	1	3	1	2	14400.0	54.70

Table A2
Heuristic results of small-scale instances.

Instance	The initial solution	The ALNS solution						Gap0	Gap1
	Obj^{I}	Obj^{ALNS}	N^{t}	N^{u}	N^{mr}	N^{sr}	T^{ALNS} (s)		
C1-6-0.25	150.00	110.16	2	2	0	0	0.8	0.00	26.56
C1-6-0.50	133.00	116.20	1	2	1	1	1.0	0.00	12.63
C1-6-0.75	137.24	107.00	1	5	0	0	0.6	0.00	22.03
C1-7-0.25	139.38	110.00	1	1	1	1	2.8	0.00	21.08
C1-7-0.50	132.00	91.40	0	2	1	1	1.8	0.00	30.76
C1-7-0.75	131.29	84.72	0	0	1	2	2.7	0.00	35.47
C1-8-0.25	188.98	161.20	1	1	1	1	2.2	0.00	14.70
C1-8-0.50	146.22	110.61	0	1	2	2	9.0	0.00	24.35
C1-8-0.75	159.40	125.84	0	4	1	2	2.2	0.00	21.05
C1-9-0.25	226.96	159.96	1	0	1	1	3.3	0.00	29.52
C1-9-0.50	261.39	231.40	1	2	1	1	2.8	0.00	11.47
C1-9-0.75	112.20	99.26	0	1	1	2	3.5	0.00	11.53
C1-10-0.25	182.80	105.60	1	1	1	1	3.8	0.00	42.23
C1-10-0.50	213.80	135.84	1	3	1	1	0.6	0.00	36.46
C1-10-0.75	167.60	115.00	1	5	0	0	0.4	0.00	31.38
C1-11-0.25	135.60	114.12	3	2	1	1	1.1	0.15	15.84
C1-11-0.50	200.37	165.60	3	3	1	1	5.6	0.00	17.35
C1-11-0.75	184.72	158.20	2	5	0	0	0.6	0.00	14.36
C1-12-0.25	281.60	134.00	2	1	1	1	1.1	0.00	52.41
C1-12-0.50	229.00	148.00	1	3	1	1	0.7	0.00	35.37
C1-12-0.75	257.19	172.19	1	3	1	2	0.7	0.00	33.05

In Table A2, Column "Instance" indicates the instance. Column "Obj^{I}" indicates the objective value of the initial solution. Column "Obj^{ALNS}" indicates the objective value of the ALNS solution. Columns "N^{t}", "N^{u}", "N^{mr}", and "N^{sr}" represent the number of van routes, number of pure UAV routes, number of main-tours, and number of sub-tours in the ALNS solution, respectively. Column "T^{ALNS} (s)" indicates the computation time (s) of the ALNS. To compare the heuristic solution with the exact solution, we define the percentage gap between the objective value of the ALNS solution and that of the exact solution of each instance, i.e., Gap0=100%×(Obj^{ALNS}-Obj^{cplex})/Obj^{ALNS}. To evaluate the performance of the ALNS, we define the percentage gap between the objective value of the initial solution and that of the ALNS solution of each instance, i.e., Gap1=100%×(Obj^{I}-Obj^{ALNS})/Obj^{I}.

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 52 (4), 965–981.
Alinaghian, M., Shokouhi, N., 2018. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. Omega 76, 85–99.
Azi, N., Gendreau, M., Potvin, J.Y., 2014. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Comput. Op. Res. 41, 167–173.
Belenguer, J.M., Benavent, E., Martínez, A., Prins, C., Prodhon, C., Villegas, J.G., 2015. A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. Transp. Sci. 50 (2), 735–749.
Belgin, O., Karaoglan, I., Altiparmak, F., 2018. Two-echelon vehicle routing problem with simultaneous pickup and delivery: mathematical model and heuristic approach. Comput. Ind. Eng. 115, 1–16.
Breunig, U., Schmid, V., Hartl, R.F., Vidal, T., 2016. A large neighbourhood based heuristic for two-echelon routing problems. Comput. Op. Res. 76, 208–225.
Chang, Y.S., Lee, H.J., 2018. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst. Appl. 104, 307–317.
Chao, I.M., 2002. A tabu search method for the truck and trailer routing problem. Comput. Op. Res. 29 (1), 33–51.
Chen, S., Chen, R., Wang, G.G., Gao, J., Sangaiah, A.K., 2018. An adaptive large neighborhood search heuristic for dynamic vehicle routing problems. Comput. Electr. Eng. 67, 596–607.
Crainic, T.G., Mancini, S., Perboli, G., Tadei, R., 2011. Multi-start heuristics for the two-echelon vehicle routing problem. In: Merz, P., Hao, JK. (Eds.). In: Lecture Notes in Computer Science, 6622. Springer, Berlin, Heidelberg, pp. 179–190.
Crainic, T.G., Sgalambro, A., 2014. Service network design models for two-tier city logistics. Optim. Lett. 8 (4), 1375–1387.
Cuda, R., Guastaroba, G., Speranza, M.G., 2015. A survey on two-echelon routing problems. Comput. Op. Res. 55, 185–199.
Dalfard, V.M., Kaveh, M., Nosratian, N.E., 2013. Two meta-heuristic algorithms for two-echelon location-routing problem with vehicle fleet capacity and maximum route length constraints. Neural Comput. Appl. 23 (7–8), 2341–2349.
de Freitas, J.C., Penna, P.H.V., 2018. A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. Electron. Notes Discret. Math. 66, 95–102.
Dellaert, N., Saridarq, F.D., Woensel, T.V., Crainic, T.G., 2019. Branch-and-price-based algorithms for the two-echelon vehicle routing problem with time windows. Transp. Sci. 53 (2), 463–479.
Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. Eur. J. Op. Res. 223 (2), 346–359.
Es Yurek, E., Ozmutlu, H.C., 2018. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. Transp. Res. Part C: Emerg. Technol. 91, 249–262.
Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R., Rich, R., 2016. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J. Ind. Eng. Manag. 9 (2), 374–388.
Gambella, C., Lodi, A., Vigo, D., 2018. Exact solutions for the carrier-vehicle traveling salesman problem. Transp. Sci. 52 (2), 320–330.
Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Eur. J. Op. Res. 254 (1), 80–91.
Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. Transp. Res. Part C: Emerg. Technol. 86, 597–621.
Ham, A.M., 2018. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. Transp. Res. Part C: Emerg. Technol. 91, 1–14.
Hemmelmayr, V.C., Cordeau, J.F., Crainic, T.G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput. Op. Res. 39 (12), 3215–3228.
Jeong, H.Y., Song, B.D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones. Int. J. Prod. Econ. 214, 220–233.
Jepsen, M., Spoorendonk, S., Ropke, S., 2013. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. Transp. Sci. 47 (1), 23–37.
Karak, A., Abdelghany, K., 2019. The hybrid vehicle-drone routing problem for pick-up and delivery services. Transp. Res. Part C: Emerg. Technol. 102, 427–449.
Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M.A., Brunese, P.A., 2019. Multiple traveling salesman problem with drones: mathematical model and heuristic approach. Comput. Ind. Eng. 129, 14–30.
Li, H., Liu, Y., Jian, X., Lu, Y., 2018. The two-echelon distribution system considering the real-time transshipment capacity varying. Transpo. Res. Part B: Methodol. 110, 239–260.
Li, H., Zhang, L., Lv, T., Chang, X., 2016. The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems. Transpo. Res. Part B: Methodol. 94, 169–188.
Lin, S.W., Vincent, F.Y., Chou, S.Y., 2009. Solving the truck and trailer routing problem based on a simulated annealing heuristic. Comput. Op. Res. 36 (5), 1683–1692.
Lin, S.W., Vincent, F.Y., Lu, C.C., 2011. A simulated annealing heuristic for the truck and trailer routing problem with time windows. Expert Syst. Appl. 38 (12), 15244–15252.
Liu, T., Luo, Z., Qin, H., Lim, A., 2018. A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints. Eur. J. Op. Res. 266 (2), 487–497.
Liu, R., Tao, Y., Xie, X., 2019. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. Comput. Op. Res. 101, 250–262.
Luo, Z., Liu, Z., Shi, J., 2017. A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. Sensors 17 (5), 1144.
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp. Res. Part C: Emerg. Technol. 54, 86–109.
Parragh, S.N., Cordeau, J.F., 2017. Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. Comput. Op. Res. 83, 28–44.
Perboli, G., Tadei, R., 2010. New families of valid inequalities for the two-echelon vehicle routing problem. Electron. Notes Discret. Math. 36, 639–646.
Perboli, G., Tadei, R., Fadda, E., 2018. New valid inequalities for the two-echelon capacitated vehicle routing problem. Electron. Notes Discret. Math. 64, 75–84.
Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transp. Sci. 45 (3), 364–380.
Pichka, K., Bajgiran, A.H., Petering, M.E., Jang, J., Yue, X., 2018. The two echelon open location routing problem: Mathematical model and hybrid heuristic. Comput. Ind. Eng. 121, 97–112.
Ribeiro, G.M., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Comput. Op. Res. 39 (3), 728–735.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.

Rothenbächer, A.K., Drexl, M., Irnich, S., 2018. Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. Transp. Sci. 52 (5), 1174–1190.
Santos, F.A., da Cunha, A.S., Mateus, G.R., 2013. Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. Optim. Lett. 7 (7), 1537–1547.
Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited article-city logistics: Challenges and opportunities. Transp. Sci. 50 (2), 579–590.
Scheuerer, S., 2006. A tabu search heuristic for the truck and trailer routing problem. Comput. Op. Res. 33 (4), 894–909.
Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: In International Conference on Principles and Practice of Constraint Programming. Springer, Berlin, Heidelberg, pp. 417–431.
Sitek, P., Wikarek, J., 2014. A novel integrated approach to the modelling and solving of the two-echelon capacitated vehicle routing problem. Prod. Manuf. Res. 2 (1), 326–340.
Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.
Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2013. A matheuristic for the truck and trailer routing problem. Eur. J. Op. Res. 230 (2), 231–244.
Wang, X., Poikonen, S., Golden, B., 2017a. The vehicle routing problem with drones: Several worst-case results. Optim. Lett. 11 (4), 679–697.
Wang, K., Shao, Y., Zhou, W., 2017b. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. Transp. Res. Part D: Transp. Environ. 57, 262–276.
Wang, Z., Sheu, J.B., 2019. Vehicle routing problem with drones. Transpo. Res. Part B: Methodol. 122, 350–364.
Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. Eur. J. Op. Res. 265 (2), 765–778.