Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 6599089, 20 pages
https://doi.org/10.1155/2022/6599089

Research Article

Truck and Unmanned Vehicle Routing Problem with Time Windows: A Satellite Synchronization Perspective

Hongqi Li, Jiamin Zhao, and Zhuopeng Zhan

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

Correspondence should be addressed to Hongqi Li; lihongqi@buaa.edu.cn

Received 21 September 2021; Revised 2 March 2022; Accepted 5 April 2022; Published 12 May 2022

Academic Editor: Vincent F. Yu

Copyright © 2022 Hongqi Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider an important feature of satellite synchronization in the practical scenario of using unmanned vehicles (UVs) carried by trucks for "last-meter" delivery and introduce the truck and UV routing problem with time windows (TUVRP-TW) for optimizing the routes of a homogeneous fleet of truck-UV combinations. A UV that has been dispatched from its truck must be picked up by the same truck or must return by itself to the depot. Customers with time windows are classified into two types: truck-UV customers (TUCs) and UV customers (UCs). The TUCs where trucks dispatch or pick up the carried UVs are regarded as satellites. Fleet coordination and satellite synchronization are essential for modelling the TUVRP-TW. We classify satellite synchronization into inner-satellite synchronization and intersatellite synchronization. The inner-satellite synchronization generally considered in the literature focuses on synchronization operations at the same satellite. Intersatellite synchronization, which focuses on synchronization operations at various satellites, allows UVs to not return to the dispatched locations, if necessary. In the mixed-integer linear programming model of the TUVRP-TW, both binary variables for identifying the appointed satellites and continuous variables for time continuity constraints are introduced to ensure the interaction between truck routes and UV routes. A hybrid algorithm based on a greedy randomized adaptive search procedure (GRASP) and a variable neighborhood search (VNS) is provided. Based on generated instances and benchmark instances, computational experiments are conducted to evaluate the performance of the intersatellite synchronization, the performance of the developed formulation, and the applicability of the hybrid algorithm.

1. Introduction

Through developing and adopting new technologies to improve operations, logistics enterprises can offer high-quality express services to customers. Great advancements in the area of unmanned vehicle (UV) technologies are bringing many new possibilities for the utilization of UVs in the logistics sectors. A number of large enterprises are investing in the delivery modes that take advantage of the utilization of UVs. At a small scale, some enterprises have tested or attempted "last-meter" small parcel deliveries through using UVs.

We strive to develop an innovative methodological approach for tackling the practical scenarios that logistics and supply-chain enterprises are presently facing in the course of using UVs that are carried by trucks for "last-meter" delivery. Many enterprises are now racing toward delivery by UVs. The project "Vans & Robots," which was launched at Mercedes-Benz Vans in the Future Transportation unit, aims at developing the "mothership approach," which is expected to be an ideal solution for ground transport of parcels via the cooperation between vans and carried UVs (Figure 1(a)). The UVs are developed as ground-based autonomous delivery robots for parcel deliveries. Compared with a single UV for "last-meter" delivery, the "mothership approach" realizes more efficient and more flexible delivery operations: on an ideal route, the van drops off one or more UVs with parcels at each stop. Each UV rolls autonomously through the last meters of the delivery route. When the delivery has been completed, the UV

(a) (b)

Figure 1: UV examples that are developed by companies. (a) The "mothership approach" in the project "Vans & Robots," which was launched at Mercedes-Benz Vans in the Future Transportation unit (https://www.daimler.com/innovation/specials/future-transportation-vans/parcel-carrier-2-0.html). (b) A UV that was developed by JD e-business (http://tech.qq.com/a/20160901/039363.htm).

finds its way back to the mothership (the van) by itself. Returned UVs can be reloaded for the next delivery missions. Although many technical issues must be overcome, Daimler is not alone in developing delivery by UV. For example, JD e-business employs UVs for campus delivery in some cities in China (Figure 1(b)).

We are aware of the operational challenges in optimizing delivery routes for truck-UV combinations. In the delivery application of pairing one UV with a truck, a truck departs from the depot carrying one UV and cargoes. At a specified customer, the truck drops off the carried UV with the assigned cargoes. En route, the UV requires no intervention from the truck; however, the UV must travel along a predetermined route to return to the paired truck, which may continuously make deliveries and move to a new customer. From the two-echelon network perspective, truck routes that originate at the depot are on one echelon and UV routes that originate at customers or the depot are on the other echelon. Specified customers for UV dropping or return are regarded as satellites. At a satellite for UV dropping, the arrival time of the truck should be earlier than or equal to the dispatch time of the carried UV. At a satellite for UV return, the time difference between the arrival time of the truck and that of the UV should be considered. To address the interdependence, we introduce intersatellite synchronization, which focuses on synchronization operations at satellites.

The main contributions of the paper are as follows. First, we introduce and define the truck and UV routing problem with time windows (TUVRP-TW) that caters to the delivery applicability of a fleet of truck-UV combinations with respect to customer time windows. Second, we propose a mixed-integer linear programming model, in which intersatellite synchronization constraints are specially developed. Third, we provide a hybrid algorithm based on a greedy randomized adaptive search procedure (GRASP) and a variable neighborhood search (VNS). Fourth, we experimentally evaluate the intersatellite synchronization performance, the effectiveness of the developed TUVRP-TW mathematical formulation, and the applicability of the hybrid algorithm for large-scale instances.

The remainder of this paper is organized as follows. Section 2 reviews the literature on routing problems that involve the satellite synchronization in specified and simplified situations. Section 3 defines the TUVRP-TW and introduces the mixed-integer linear programming model, which is expressed in the vehicle flow formulation. Section 4 presents a hybrid algorithm. Section 5 describes the generated instances, the exact and heuristic results. We present the conclusions of this work in Section 6.

2. Literature Review

Almost all the mathematical methods for solving routing problems involving satellite synchronization are based on studies on the two-echelon vehicle routing problem (2E-VRP), the truck and trailer routing problem (TTRP), and the two-echelon location routing problem (2E-LRP). According to Crainic and Sgalambro [1], the echelon interactions, such as the fleet coordination and satellite synchronization operations, are the main challenges in modelling the routing variants. In the 2E-VRP, the TTRP, and the 2E-LRP literature, satellite synchronization constraints require that first-echelon vehicles and second-echelon vehicles meet at the same satellite. The satellite synchronization, which is called inner-satellite synchronization in this paper, focuses on synchronization operations at the same satellite. Despite the wide application backgrounds of routing problems involving satellite synchronization, many important realistic features remain to be considered. The scenario in which an autonomous delivery robot or a drone works in collaboration with a truck to distribute parcels brings new and interesting topics of routing problems involving satellite synchronization.

In the literature, many forms of the travelling salesman problem (TSP) variants (e.g., the FSTSP or the TSPD) are introduced considering truck-drone delivery modes. For an overview of these variants, we refer to Chung et al. [2]; Macrina et al. [3]; Rojas Viloria et al. [4]; and Li et al. [5]. The FSTSP or the TSPD has some basic characters, for example, one truck paired with one drone is used. The truck and drone depart from and return to the depot exactly once. The drone can launch from and land on the truck only while the truck is parked at a node. One drone route is assumed to cover exactly one customer. Murray and Chu [6] introduced the FSTSP. de Freitas and Penna [7, 8], Dell'Amico et al. [9], and Jeong et al. [10] provided various methods for the FSTSP and variants. Agatz et al. [11] introduced the TSPD. Es Yurek and Ozmutlu [12]; Ha et al. [13, 14]; and Marinelli et al. [15]

introduced the TSPD variants and solution methods. Poikonen and Golden [16] and Gonzalez-R et al. [17] relaxed the constraint of one drone route covering exactly one customer. Murray and Raj [18]; Moshref-Javadi et al. [19, 20]; Salama and Srinivas [21]; and Dell'Amico et al. [22] relaxed the constraint of using one truck paired with one drone. Poikonen and Golden [23] introduced the k-multi-visit drone routing problem that considered a tandem between a truck and *k* drones. Each drone could launch from the truck with one or more packages to deliver to customers. There is literature that investigated the vehicle routing problem (VRP) variants with drones, called the VRPD. Wang et al. [24] introduced the VRPD. Schermer et al. [25]; Sacramento et al. [26]; and Euchi and Sadok [27] proposed suitable methods for the VRPD and variants. Di Puglia Pugliese and Guerriero [28] and Das et al. [29] introduced the VRPD with time windows. Kitjacharoenchai et al. [30] and Li et al. [31] studied VRPD variants from a two-echelon perspective.

Especially, Boysen et al. [32] introduced scheduling procedures for a truck-based robot delivery, which aims at minimizing the weighted number of late customer deliveries after the announced delivery time. Customers were exclusively serviced by robots. A truck loaded the shipments for a set of customers at a central depot, and a fixed part of the truck's loading capacity was reserved to also load autonomous robots on board. Once the truck reached a drop-off point, one or multiple robots were loaded with shipments and launched to autonomously deliver goods to customers. The truck exclusively acted as a mobile satellite, and it did not access customer locations to also service customers directly. A multistart local search procedure was proposed. A tailor-made local search procedure was provided for determining an optimal assignment of customers to drop-off points and robot depots for a predetermined truck route. Two instances with 16 and 86 nodes were randomly generated. The computational study revealed that the truck fleet could considerably be reduced if autonomous robots supported the delivery process.

According to the literature, intersatellite synchronization constraints have been imposed in specified and simplified scenarios. Methodological innovation is necessary for formulating the routing variant with intersatellite synchronization constraints to cater to more general and practical scenarios that involve, e.g., customer time windows and a fleet of truck-UV combinations. In many promotional videos and other material distributed by companies developing UV or drone delivery, a customer often must be present to receive delivery. The use of time windows and intersatellite synchronization is a key point of distinction of the introduced variant in this paper. In Table 1, we compare the literature with this paper to highlight the main differences. Although the body of literature on truck-drone routing variants is presently relatively large, time windows are seldom considered by the majority of the truck-drone routing variants. The variants introduced by Di Puglia Pugliese and Guerriero [28] and Das et al. [29] involve time windows, while exactly one customer is covered by each drone route. Li et al. [31] studied a variant with time windows from a two-echelon perspective, while intersatellite synchronization is forbidden.

3. The TUVRP-TW Definition and Mathematical Formulation

3.1. Problem Definition. From the methodological innovation perspective, the TUVRP-TW utilizes intersatellite synchronization to realize more efficient and more flexible use of UVs, and customer time windows are respected. The intersatellite synchronization permits the meeting satellites for a truck and its UV to vary, which is expected to provide more possibilities in determining the optimized routes. From the perspective of model constructions, both binary variables for identifying the appointed satellites and continuous variables for time continuity constraints ensure the interaction between truck routes and UV routes.

The TUVRP-TW network includes one depot and a specified number of customers. Customers are classified into two types: truck-UV customers (denoted as TUCs) and UV customers (denoted as UCs). Each TUC can be served by one truck-UV combination or one UV, and each UC can only be served by one UV. The customer demand with time windows is known in advance and cannot be divided. At TUCs, a UV may be dispatched from the paired truck to run by itself to visit UCs or TUCs. The TUCs where trucks drop off or pick up carried UVs are regarded as satellites. Each truck can drop off and pick up the carried UV only at TUCs. The truck can continue to visit TUCs after the carried UV has been dispatched and the truck may pick up the UV at another TUC, if possible. The UV or the paired truck that arrives first at the pick-up TUC should wait for the other vehicle.

We use an illustrative example in Figure 2 to show the route forms in the TUVRP-TW. Other characters of the TUVRP-TW, e.g., time windows, are omitted from Figure 2, for simplicity. Figure 2 involves a network of one depot and eleven customers. The eleven customers are classified into six TUCs and five UCs. Among the six TUCs, TUC1, TUC2, and TUC5 are used as satellites. Routes travelled by trucks or truck-UV combinations are represented by solid lines and routes travelled by UVs are represented by dotted lines. In the route travelled on by a UV, the UV (which is paired with truck C) with cargoes departs by itself from the depot, visits UC4 and UC5 in order, and returns to the depot. In the route travelled by a truck-UV combination, the truck-UV combination (of truck A and UV a) with cargoes departs from the depot to TUC1. At TUC1, UV a is dispatched to serve UC1 and UC2. After serving TUC1, truck A leaves and travels to serve TUC2. The truck and the UV meet at TUC2. The truck-UV combination leaves from TUC2, serves TUC3, and returns to the depot. In the route travelled by another truck-UV combination, the truck-UV combination (of truck B and UV b) with cargoes departs from the depot to TUC5. At TUC5, UV b is dispatched to serve UC3. Then, UV b returns by itself to the depot. After serving TUC5, truck B leaves and travels to serve TUC4 and TUC6. Truck B leaves from TUC6 and returns to the depot.

The following are some of the key components that constitute the TUVRP-TW.

First, the TUVRP-TW is defined in a directed complete graph, in which the nodes represent the depot and

Table 1: Studies that are related to intersatellite synchronization.

References	Truck	Truck and UV	UV capacity	Time window	Objective and model
Murray and Chu [6]; de Freitas and Penna [7, 8]; Dell'Amico et al. [9]; Agatz et al. [11]; Es Yurek and Ozmutlu [12]; Ha et al. [13, 14]; Jeong et al. [10]; and Marinelli et al. [15]	1	1 : 1	One		Min makespan, time, or costs; MILP
Poikonen and Golden [16] and Gonzalez-R et al. [17]	1	1 : 1	Many		Min makespan; MIP
Poikonen and Golden [23]	1	1 : n	Many		Min makespan; MILP
Murray and Raj [18]; Moshref-Javadi et al. [19, 20]; Salama and Srinivas [21]; and Dell'Amico et al. [22]	1	1 : n	One		Min makespan, time, or costs; MILP
Sacramento et al. [26] and Euchi and Sadok [27]	m	1 : 1	One		Min time or costs; MILP
Di Puglia Pugliese and Guerriero [28] and Das et al. [29]	m	1 : 1	One	✓	Min distance or costs; MIP
Wang et al. [24] and Schermer et al. [25]	m	1 : n	One		Min makespan; MILP
Kitjacharoenchai et al. [30]	m	1 : n	Many		Min makespan; MILP
Li et al. [31]	m	1 : n	Many	✓	Min costs; MILP
This paper	m	1 : 1	Many	✓	Min costs; MILP

Note. "✓" denotes that the factor is considered. "MIP" denotes mixed-integer programming. "MILP" denotes mixed-integer linear programming.

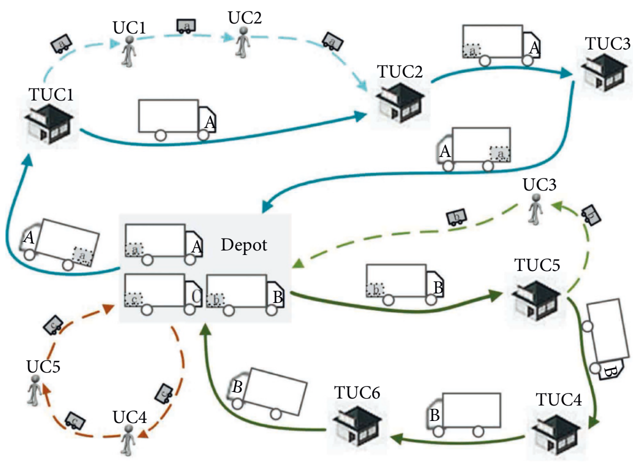

Figure 2: Route examples of the TUVRP-TW.

customers, and the arcs represent the possible moves of trucks or UVs. At the depot, a homogeneous fleet of truck-UV combinations is available. The depot is the only source of cargoes. UVs can be used for direct delivery from the depot.

Second, the truck-UV combination includes one truck that is carrying one UV. Each truck that is in use departs from the depot carrying one UV and cargoes for the intended customers. The UVs are autonomous vehicles for cargo delivery. A truck and its UV work in a pair mode. The pair mode means a UV that has been dispatched from its truck must be picked up by the paired truck or return by itself to the depot. A UV is not allowed to depart itself and then be picked up by its truck at a satellite, while the opposite case is allowed.

Third, routes are classified into three types: (i) UV routes (UV routes, which are travelled by UVs for direct delivery, originate and terminate at the depot); (ii) truck routes (truck routes are travelled by trucks to visit TUCs); and (iii) combination routes (combination routes consist of main tours and subtours). Each combination route includes one main tour and one or several subtours and is travelled by one truck-UV combination. The main tour, which includes TUCs and the depot, is travelled by the truck-UV combination or the truck. The subtour is travelled by the dispatched UV. The origin of each subtour is a satellite, namely, a specified TUC on the main tour. The destination of each subtour is a satellite (namely, a TUC on the main tour) or the depot.

Fourth, the objective of the TUVRP-TW is to minimize the integrated distance, which includes the total travelling distance of the vehicles and the transformed distance of the probable waiting times of vehicles at customers. The probable waiting time is that the vehicle waits for the time window to open at a customer.

To formulate the TUVRP-TW, the following assumptions are considered. First, the UV that is carried by each truck does not occupy the truck's capacity. Second, a running range limitation related to battery-powered UVs is not considered. Third, there are enough vehicles available at the depot.

3.2. Mathematical Formulation

3.2. Mathematical Formulation. We develop a mixed-integer linear programming model for the TUVRP-TW. The mixed-integer linear programming model, which includes the objective function and constraints, is expressed by the vehicle flow formulation.

The TUVRP-TW is presented on a directed complete graph $G = (V, A)$ that includes vertices and arcs. The vertex set is denoted as $V = \{0\} \cup V^{\text{tuc}} \cup V^{\text{uc}} = \{0, 1, 2, \ldots, n^{\text{tuc}}, \ldots, n\}$, in which vertex 0 is the depot and vertex subset $V^{\text{c}} = V \backslash \{0\}$ is the customer set. The customer set consists of the TUC subset $(V^{\text{tuc}} = \{1, 2, \ldots, n^{\text{tuc}}\})$ and the UC subset $(V^{\text{uc}} = \{n^{\text{tuc}} + 1, \ldots, n\})$. The origins and the destinations of routes or subtours that are travelled by UVs should be the depot or satellites. We introduce the vertex set (denoted as $V^{\text{tu}} = \{0, 1, 2, \ldots, n^{\text{tuc}}\}$) of all vertices that can act simultaneously as the origin and the destination of UV routes or subtours. The arc set is denoted as $A = \{(i, j)|i,\ j \in V\}$. d_{ij} denotes the travelling distance of arc (i, j) $(i, j \in V)$. We introduce set K to denote the truck-UV combination set available at the depot initially, and $|K| = n$.

The weight of the cargoes that are required by customer i $(i \in V^{\mathrm{c}})$ is denoted as q_i. $[e_i, l_i]$ denotes the time window of customer i, where e_i and l_i represent the earliest and latest service starting times, respectively, of customer i. s_i is the service time at customer i.

The capacities of each truck and of each UV are denoted as Q^{t} and Q^{u}, respectively. The total amount of cargoes that is needed by the intended customers in a route that is travelled by the truck-UV combination should not exceed $Q^{\mathrm{t}} + Q^{\mathrm{u}}$, and the capacity of each truck-UV combination is $Q^{\mathrm{tu}} = Q^{\mathrm{t}} + Q^{\mathrm{u}}$. v^{t} and v^{u} denote the average velocities of a truck-UV combination and of a UV. The waiting cost of a vehicle for a unit of time at a customer is τ.

Several types of binary variables and continuous variables are introduced. x^{t}_{kij} is a binary variable that is 1 only if truck k $(k \in K)$ covers arc (i, j) $(i, j \in V^{\mathrm{tu}})$. x^{u}_{krcij} is a binary variable that is 1 only if UV k that is carried by truck k is dispatched at vertex r $(r \in V^{\mathrm{tuc}})$ and is picked up at vertex c by covering arc (i, j) $(i, j \in V)$ on a subtour. dt^{t}_k and dt^{u}_k denote the departure time of truck k and UV k from the depot. at^{t}_{ki} $(i \in V^{\mathrm{tu}})$ and at^{u}_{ki} $(i \in V)$ denote the arrival time of truck k and UV k at vertex i. wt^{t}_{ki} $(i \in V^{\mathrm{tuc}})$ and wt^{u}_{ki} $(i \in V^{\mathrm{c}})$ denote the possible waiting time of truck k and UV k at customer i. The continuous variables for time are nonnegative. In various constraints, M denotes a sufficiently large positive integer.

3.2.1. Objective Function. The objective function consists of four parts: the travelling distance of trucks, the transformed distance of possible waiting times of trucks at customers, the travelling distance of UVs, and the transformed distance of possible waiting times of UVs at customers. The objective function is expressed as

$$\begin{aligned}\min \sum_{k\in K}\sum_{i\in V^{\mathrm{tu}}}\sum_{j\in V^{\mathrm{tu}}} d_{ij}\cdot x^{\mathrm{t}}_{kij}\varsigma + \sum_{k\in K}\sum_{i\in V^{\mathrm{c}}} wt^{\mathrm{t}}_{ki}\cdot v^{\mathrm{t}}\cdot\tau \\ + \sum_{k\in K}\sum_{r\in V^{\mathrm{tu}}}\sum_{c\in V^{\mathrm{tu}}}\sum_{i\in V}\sum_{j\in V} d_{ij}\cdot x^{\mathrm{u}}_{krcij} \\ + \sum_{k\in K}\sum_{i\in V^{\mathrm{c}}} wt^{\mathrm{u}}_{ki}\cdot v^{\mathrm{u}}\cdot\tau.\end{aligned} \tag{1}$$

3.2.2. Constraints. We divide the constraints into four categories: constraints on the vehicle capacity, constraints on the time continuity, constraints on the vehicle visiting vertices, and constraints on the intersatellite synchronization.

Constraint (2) restricts the capacity of each UV in UV routes for direct delivery or in subtours by UVs that have been dispatched from trucks at satellites. Constraint (3) restricts the capacity of each truck-UV combination in combination routes or truck routes.

$$\sum_{i\in V^{\mathrm{c}}, i\neq r, i\neq c}\ \sum_{j\in V} q_i x^{\mathrm{u}}_{krcij} \le Q^{\mathrm{u}},\ k\in K,\ r\in V^{\mathrm{tu}},\ c\in V^{\mathrm{tu}}, \tag{2}$$

$$\begin{aligned}\sum_{i\in V^{\mathrm{tuc}}}\sum_{j\in V^{\mathrm{tu}}} q_i x^{\mathrm{t}}_{kij} \\ + \sum_{r\in V^{\mathrm{tuc}}}\sum_{c\in V^{\mathrm{tu}}}\sum_{i\in V^{\mathrm{c}}, i\neq r, i\neq c}\sum_{j\in V} q_i x^{\mathrm{u}}_{krcij} \le Q^{\mathrm{tu}},\ k\in K.\end{aligned} \tag{3}$$

Constraints on the time continuity aim to decide the arrival, leaving, and waiting time for vehicles in use. According to constraints (4) and (5), the truck arrival time at the first TUC is determined by the truck departure time from the depot and the running time on the arc that connects the depot and the first TUC. According to constraints (6) and (7), the truck arrival time at a specified TUC or at the depot is determined by the truck arrival time at previous vertex, the possible waiting time for time window opening, the service time, and the running time on the arc. Constraints (8) and (9) ensure that the truck visits TUCs by respecting the TUC time windows. According to constraints (10) and (11), the UV arrival time at the first customer is determined by the UV departure time from the depot and the running time on the arc that connects the depot and the first customer. According to constraints (12) and (13), the UV arrival time at a specified vertex is determined by the UV arrival time at previous vertex, the possible waiting time for time window opening, the service time, and the UV running time on the arc. Constraints (14) and (15) ensure that the dispatched UV visits customers by respecting the customer time windows.

$$dt^{\mathrm{t}}_k + \frac{d_{0i}}{v^{\mathrm{t}}} + M\cdot\left(1 - x^{\mathrm{t}}_{k0i}\right) \ge at^{\mathrm{t}}_{ki},\ k\in K,\ i\in V^{\mathrm{tuc}}, \tag{4}$$

$$dt^{\mathrm{t}}_k + \frac{d_{0i}}{v^{\mathrm{t}}} - M\cdot\left(1 - x^{\mathrm{t}}_{k0i}\right) \le at^{\mathrm{t}}_{ki},\ k\in K,\ i\in V^{\mathrm{tuc}}, \tag{5}$$

$$\begin{aligned}at^{\mathrm{t}}_{ki} + wt^{\mathrm{t}}_{ki} + s_i + \frac{d_{ij}}{v^{\mathrm{t}}} \\ + M\cdot\left(1 - x^{\mathrm{t}}_{kij}\right) \ge at^{\mathrm{t}}_{kj},\ k\in K,\ i\in V^{\mathrm{tuc}},\ j\in V^{\mathrm{tu}},\end{aligned} \tag{6}$$

$$\begin{aligned}at^{\mathrm{t}}_{ki} + wt^{\mathrm{t}}_{ki} + s_i + \frac{d_{ij}}{v^{\mathrm{t}}} \\ - M\cdot\left(1 - x^{\mathrm{t}}_{kij}\right) \le at^{\mathrm{t}}_{kj},\ k\in K,\ i\in V^{\mathrm{tuc}},\ j\in V^{\mathrm{tu}},\end{aligned} \tag{7}$$

$$at^{\mathrm{t}}_{ki} + wt^{\mathrm{t}}_{ki} + M\cdot\left(1 - \sum_{j\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kij}\right) \ge e_i,\ k\in K,\ i\in V^{\mathrm{tuc}}, \tag{8}$$

$$at^{\mathrm{t}}_{ki} + wt^{\mathrm{t}}_{ki} - M\cdot\left(1 - \sum_{j\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kij}\right) \le l_i,\ k\in K,\ i\in V^{\mathrm{tuc}}, \tag{9}$$

$$dt^{\mathrm{u}}_k + \frac{d_{0i}}{v^{\mathrm{u}}} + M\cdot(1 - x^{\mathrm{u}}_{k000i}) \ge at^{\mathrm{u}}_{ki},\ k\in K,\ i\in V^{\mathrm{c}}, \tag{10}$$

$$dt_k^{\mathrm{u}} + \frac{d_{0i}}{v^{\mathrm{u}}} - M \cdot \left(1 - x_{k000i}^{\mathrm{u}}\right) \le at_{ki}^{\mathrm{u}}, \quad k \in K,\ i \in V^{\mathrm{c}}, \tag{11}$$

$$at_{ki}^{\mathrm{u}} + wt_{ki}^{\mathrm{u}} + s_i + \frac{d_{ij}}{v^{\mathrm{u}}} + M \cdot \left(1 - x_{krcij}^{\mathrm{u}}\right) \ge at_{kj}^{\mathrm{u}}, \quad k \in K,\ r \in V^{\mathrm{tu}},\ c \in V^{\mathrm{tu}},\ i \in V,\ i \ne r,\ j \in V, \tag{12}$$

$$at_{ki}^{\mathrm{u}} + wt_{ki}^{\mathrm{u}} + s_i + \frac{d_{ij}}{v^{\mathrm{u}}} - M \cdot \left(1 - x_{krcij}^{\mathrm{u}}\right) \le at_{kj}^{\mathrm{u}}, \quad k \in K,\ r \in V^{\mathrm{tu}},\ c \in V^{\mathrm{tu}},\ i \in V,\ i \ne r,\ j \in V, \tag{13}$$

$$at_{kj}^{\mathrm{u}} + wt_{kj}^{\mathrm{u}} + M \cdot \left(1 - \sum_{i \in V} x_{krcij}^{\mathrm{u}}\right) \ge e_j, \quad k \in K,\ r \in V^{\mathrm{tu}},\ c \in V^{\mathrm{tu}},\ j \in V^{\mathrm{c}},\ j \ne r,\ j \ne c, \tag{14}$$

$$at_{kj}^{\mathrm{u}} + wt_{kj}^{\mathrm{u}} - M \cdot \left(1 - \sum_{i \in V} x_{krcij}^{\mathrm{u}}\right) \le l_j, \quad k \in K, r \in V^{\mathrm{tu}},\ c \in V^{\mathrm{tu}},\ j \in V^{\mathrm{c}},\ j \ne r,\ j \ne c. \tag{15}$$

Constraints on the vehicle visiting vertices ensure the limit on the times of vehicle visiting a node and the vehicle retention. Constraints (16)–(18) ensure that a TUC is visited exactly once by a truck or a UV when the TUC is not being used as the satellite or a TUC is visited more than once if it is being used as the satellite. Constraint (19) guarantees that each UC is visited exactly once by one UV. Constraint (20) ensures that for a truck, the number of arrivals at vertex *j* equals the number of departures from vertex *j*. Constraint (21) guarantees that for a UV, the number of arrivals at a customer vertex equals the number of departures from the customer vertex. According to constraint (22), each dispatched UV leaves once from the origin of a subtour and returns once to the destination of the subtour. Constraint (23) requires that a UV departs by itself or with the paired truck no more than once from the depot.

$$\sum_{k \in K} \sum_{i \in V^{\mathrm{tu}}} x_{kij}^{\mathrm{t}} + \sum_{k \in K} \sum_{r \in V^{\mathrm{tu}}} \sum_{c \in V^{\mathrm{tu}}} \sum_{i \in V} x_{krcij}^{\mathrm{u}} \ge 1,\ j \in V^{\mathrm{tuc}}, \tag{16}$$

$$\sum_{k \in K} \sum_{i \in V^{\mathrm{tu}}} x_{tij}^{\mathrm{t}} \le 1,\ j \in V^{\mathrm{tuc}}, \tag{17}$$

$$\sum_{k \in K} \sum_{i \in V^{\mathrm{tu}}} \sum_{r \in V^{\mathrm{tu}}} \sum_{c \in V^{\mathrm{tu}}} x_{krcij}^{\mathrm{u}} \le 1,\ j \in V^{\mathrm{tuc}}, \tag{18}$$

$$\sum_{k \in K} \sum_{r \in V^{\mathrm{tu}}} \sum_{c \in V^{\mathrm{tu}}} \sum_{i \in V} x_{krcij}^{\mathrm{u}} = 1, j \in V^{\mathrm{uc}}, \tag{19}$$

$$\sum_{i \in V^{\mathrm{tu}}} x_{kij}^{\mathrm{t}} = \sum_{a \in V^{\mathrm{tu}}} x_{kja}^{\mathrm{t}},\ k \in K,\ j \in V^{\mathrm{tu}}, \tag{20}$$

$$\sum_{i \in V} x_{krcij}^{\mathrm{u}} = \sum_{i \in V} x_{krcji}^{\mathrm{u}},\ k \in K,\ r \in V^{\mathrm{tu}},\ c \in V^{\mathrm{tu}},\ j \in V,\ j \ne r,\ j \ne c, \tag{21}$$

$$\sum_{i \in V} x_{krcri}^{\mathrm{u}} = \sum_{j \in V} x_{krcjc}^{\mathrm{u}},\ k \in K,\ r \in V^{\mathrm{tu}},\ c \in V^{\mathrm{tu}},\ i,\ j \ne r,\ i,\ j \ne c, \tag{22}$$

$$\sum_{j \in V^{\mathrm{tuc}}} x_{k0j}^{\mathrm{t}} + \sum_{j \in V^{\mathrm{c}}} x_{k000j}^{\mathrm{u}} \le 1,\ k \in K. \tag{23}$$

Constraints on the intersatellite synchronization ensure the connection between the main tour and its subtour(s). The intersatellite synchronization permits the origin and the destination of a subtour to differ. It is assumed that the truck arrives once at and leaves once from the subtour origin, and the truck arrives once and leaves once from the subtour destination. The dispatched UV leaves once from the subtour origin and arrives once at the subtour destination. After dispatching the carried UV and finishing the service at a satellite, the truck continues to visit other TUCs to complete each intended service. Only if the arrival time of the dispatched UV at a satellite is between the arrival time of the paired truck at the satellite and the departure time of the paired truck UV from the satellite can the truck pick up its UV. According to constraint (24), a TUC should be included in a main tour if the TUC is used as the subtour origin. According to constraint (25), the subtour destination should be included in a main tour. From the temporal perspective, constraints (26)–(28) ensure that a UV and the paired truck meet at the appointed time. According to constraint (26), the arrival time of a UV at the subtour destination is not later than the departure time of the paired truck from the subtour destination. If the depot is used as the subtour destination, the appointed time for a UV meeting the paired truck is not restricted. According to constraints (27) and (28), the arrival time of a dispatched UV at the first customer in a subtour is determined by the arrival time of the truck-UV combination at the satellite and the UV running time on the arc.

$$\sum_{j \in V^{\mathrm{c}}} \sum_{c \in V^{\mathrm{tu}}} x_{krcrj}^{\mathrm{u}} \le \sum_{i \in V^{\mathrm{tu}}} x_{kir}^{\mathrm{t}},\ k \in K,\ r \in V^{\mathrm{tuc}}, \tag{24}$$

$$\sum_{i \in V^{\mathrm{c}}} \sum_{r \in V^{\mathrm{tuc}}} x_{krcic}^{\mathrm{u}} \le \sum_{i \in V^{\mathrm{tu}}} x_{kic}^{\mathrm{t}},\ k \in K,\ c \in V^{\mathrm{tu}}, \tag{25}$$

$$at_{kc}^{\mathrm{t}} + wt_{kc}^{\mathrm{t}} + s_c + M \cdot \left(1 - \sum_{r \in V^{\mathrm{tuc}}} \sum_{i \in V^{\mathrm{c}}} x_{krcic}^{\mathrm{u}}\right) \ge at_{kc}^{\mathrm{u}},\ k \in K,\ c \in V^{\mathrm{tuc}}, \tag{26}$$

$$at_{kr}^{\mathrm{t}} + \frac{d_{ri}}{v^{\mathrm{u}}} + M \cdot \left(1 - x_{krcri}^{\mathrm{u}}\right) \ge at_{ki}^{\mathrm{u}},\ k \in K,\ r \in V^{\mathrm{tuc}},\ c \in V^{\mathrm{tu}},\ i \in V^{\mathrm{c}}, \tag{27}$$

$$at_{kr}^{\mathrm{t}} + \frac{d_{ri}}{v^{\mathrm{u}}} - M \cdot \left(1 - x_{krcri}^{u}\right) \leq at_{ki}^{u},\ k \in K,\ r \in V^{\mathrm{tuc}},\ c \in V^{\mathrm{tu}},\ i \in V^{\mathrm{c}}. \tag{28}$$

The results of the computational experiments on small-scale instances demonstrate that the TUVRP-TW formulation can be solved directly by commercial IP solver software (CPLEX in this paper). The exact solutions of small-scale instances that are found by CPLEX, which can be referenced in calibrating the heuristic, support the validity of the TUVRP-TW formulation. Instances are seldom solved directly via CPLEX in practical applications because practical instances are of typically large scale and too much computer memory and computation time are required by computers. To find satisfactory solutions for large-scale instances, we propose a hybrid algorithm based on a GRASP and a VNS. On small-scale instances, the exact solutions that are found using CPLEX and the satisfactory solutions that are output by the hybrid algorithm are compared to evaluate the performance of the heuristic. Large-scale instances are solved using the proposed heuristic.

3.3. A Variant without Intersatellite Synchronization. Considering that truck-UV combinations are used for "last-meter" delivery in the "mothership approach" of truck and UV delivery modes, we introduce the TUVRP-TW. The intersatellite synchronization in the TUVRP-TW allows a truck and its UV to meet at different satellites. Due to the involvement of the intersatellite synchronization, the TUVRP-TW differs from the 2E-VRP variants in the literature. In the 2E-VRP variant literature, satellite synchronization constraints require that first-echelon vehicles and second-echelon vehicles meet at the same satellite. The inner-satellite synchronization permits synchronization operations at the same satellite.

We also introduce and formulate a routing variant with inner-satellite synchronization, named the TUVRP-TW-B. In the TUVRP-TW-B, intersatellite synchronization is not permitted. The TUVRP-TW-B formulation is presented in the Appendix.

4. A GRASP and the VNS

The TUVRP-TW is complex because of the intersatellite synchronization and time continuity constraints. Since only small instances of the TUVRP-TW could be solved directly by using the mathematical formulation in Section 3, we provide a hybrid algorithm based on a GRASP and a VNS. The hybrid algorithm incorporates two powerful features, the effective construction and improving ability of the GRASP, and the flexibility of the VNS to explore different search spaces.

The GRASP consists of iterations made up from successive constructions of a greedy randomized solution and subsequent iterative improvements of it. The greedy randomized solutions are generated by adding elements to the solution set, and the list of elements is ranked by a greedy function according to the quality of the solution they will achieve. To obtain variability in the candidate set of greedy solutions, well-ranked candidate elements are placed in a restricted candidate list (RCL) and are randomly chosen when constructing the solution.

The VNS, proposed by Hansen and Mladenović [33], is an effective metaheuristic for solving combinatorial optimization problems. The VNS can exploit systematically the neighborhood change, both in descent to local minima and in escape from the valleys that contain them [34]. The basic idea is that a local optimum defined by one neighborhood structure is not necessarily the local optimum of another neighborhood structure, and thus the search can systematically traverse different search spaces which are defined by different neighborhood structures.

4.1. Constructing an Initial Solution by a GRASP. There are three types of routes in the TUVRP-TW solution: UV routes, truck routes, and combination routes. From the perspective of node insertion in solution construction operation, a TUC may be inserted into a truck route, a UV route, a main tour of a combination route, or a subtour of a combination route; a UC may be inserted into a UV route or a subtour of a combination route. In order to construct routes of an initial solution, the GRASP adopts different adaptive probabilistic mechanisms for TUCs and UCs. Considering that the types of TUC insertion operations are more than those of UC insertion operations, a reactive mechanism is adopted for TUC insertion operations, and a random plus greedy mechanism is adopted for UC insertion operations. The following ways to insert a TUC are adopted, i.e., constructing a new UV route, inserting into a constructed UV route, constructing a new truck route, inserting into a constructed truck route, inserting into a constructed subtour, and constructing a new subtour. The following ways to insert a UC are adopted, i.e., constructing a new UV route, inserting into a constructed UV route, constructing a new subtour, and inserting into a constructed subtour. The insertion cost of a customer is estimated as the travelling distance and the transformed distance of possible waiting times of vehicles at customers.

The capacity constraints are firstly used to decide the feasibility of the insertion operation. To handle the time windows, (i) when constructing a new truck route, constructing a new UV route or inserting into a constructed UV route is adopted, the departure time of the route is initialized as the opening time of the depot, and then the arrival time and vehicle waiting time at each customer in the route are estimated. The departure time of the route is postponed by the method provided in Li et al. [35], so as to decrease vehicle waiting time at each customer as much as possible, while customer time windows are guaranteed. (ii) When the way of constructing a new subtour or inserting into a constructed subtour is adopted, the opening time of the depot is used as the earliest departure time of the corresponding main tour, and the latest departure time of the corresponding main tour is estimated as the postponed time by the method provided in Li et al. [35]. The earliest and latest departure times of the main tour are referenced to estimate the departure time of

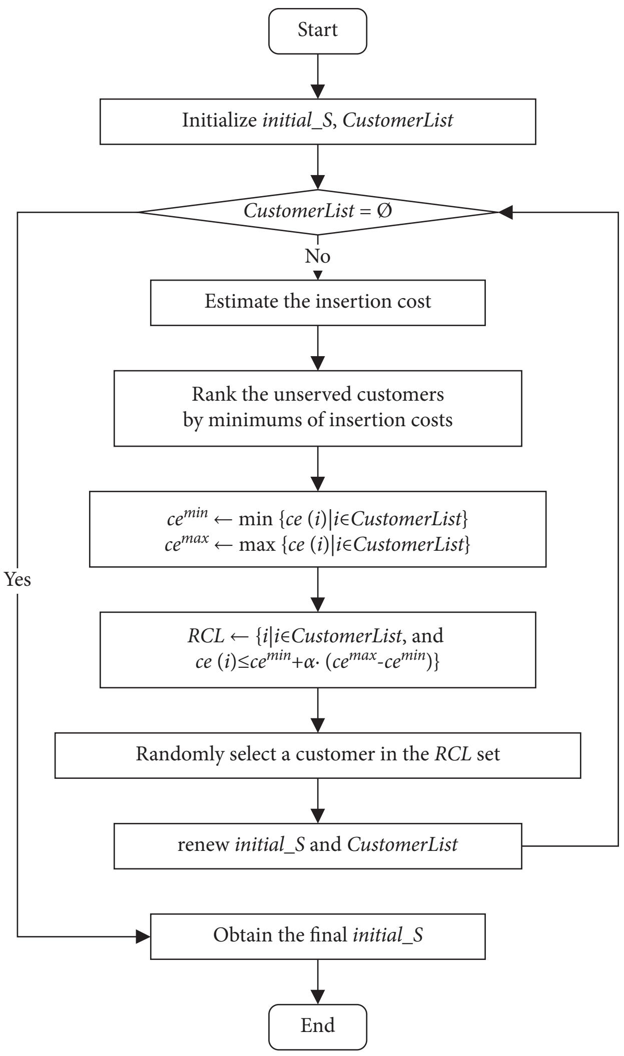
Start
Initialize *initial_S*, *CustomerList*
CustomerList = Ø
No
Yes
Estimate the insertion cost
Rank the unserved customers by minimums of insertion costs
$ce^{min} \leftarrow \min\{ce(i)|i \in CustomerList\}$
$ce^{max} \leftarrow \max\{ce(i)|i \in CustomerList\}$
$RCL \leftarrow \{i|i \in CustomerList$, and $ce(i) \le ce^{min} + \alpha \cdot (ce^{max} - ce^{min})\}$
Randomly select a customer in the *RCL* set
renew *initial_S* and *CustomerList*
Obtain the final *initial_S*
End

FIGURE 3: The GRASP process.

the subtour, and the result is not accepted if customer time windows are not ensured or if the UV covering the subtour cannot arrive at the subtour destination before its truck leaves. (iii) When the way of inserting into a constructed truck route is adopted, the handling method that is same as that used in the way of constructing a new truck route is used if there is no subtour involved. If there is one or several subtours corresponding to the constructed truck route, the handling method is the same as that used in the way of constructing a new subtour.

To handle the number of truck-UV pairs, when the way of constructing a new subtour or inserting into a constructed subtour is adopted, at a TUC, only if there is a UV carried by a truck, the TUC can be used as the origin of a subtour; at a TUC, only if there is no UV on a truck, the TUC can be used as the destination of a subtour.

We name the insertion of customer *i* by a way for customer insertion as *temp_in* and the insertion cost change of inserting customer *i* or not as *temp_obj*. The best insertion of customer *i* and its cost change are named *best_in* (*i*) and *best_obj* (*i*), respectively. For each customer insertion way, every location of each route is enumerated, for customer *i*. If a *temp_in* results in a solution that cannot ensure the TUVRP-TW constraints, the corresponding *temp_obj* is estimated as a large enough number. If a *temp_in* leads to "*temp_obj* < *best_obj* (*i*)," *best_in* (*i*) and *best_obj* (*i*) are updated. If *temp_in* leads to "*temp_obj* = *best_obj* (*i*)," *best_in* (*i*) and *best_obj* (*i*) are updated according to the probability of 50%.

The GRASP process is shown in Figure 3. The GRASP includes the following main steps.

Step 1. Initialize the solution (denoted as *initial_S*) and the unserved customer set (denoted as *CustomerList*).

Step 2. To insert customer *i* ($i \in CustomerList$), each feasible position is referred to estimate the insertion cost, and the minimum of insertion cost of customer *i* is identified and denoted as *ce* (*i*), and *ce* (*i*) = *best_obj* (*i*).

Step 3. Rank the unserved customers by minimum of insertion costs and introduce $ce^{\min} \leftarrow \min\{ce(i)|i \in CustomerList\}$, and $ce^{\max} \leftarrow \max\{ce(i)|i \in CustomerList\}$.

Step 4. Define the RCL set, $\text{RCL} \leftarrow \{i|i \in CustomerList$, and $\text{ce}(i) \le ce^{\min} + \alpha\cdot(ce^{\max} - ce^{\min})\}$, where $\alpha \in [0, 1]$.

Step 5. Randomly select a customer in the RCL set for customer insertion and renew *initial_S* and *CustomerList*.

Repeat Steps 2–5 until *CustomerList* is empty.

A feasible solution is constructed iteratively, one customer at a time. At each iteration, all unserved customers are ordered according to the insertion cost, and then the customers are selected to join the RCL with respect to an adaptive or greedy function. The customer to be inserted is selected randomly from the RCL set. The RCL set is updated at each iteration to reflect the changes brought on by the insertion of the previous customer, which indicates the adaptive mechanism. The probabilistic character is indicated by randomly selecting one of the customers in the RCL set.

For UC insertion, a random plus greedy mechanism is adopted. Instead of combining greediness and randomness at each iteration, the random plus greedy scheme applies randomness during the first *p* iterations to produce a random partial solution, and the parameter $\alpha = 1$. Then, one or more pure greedy iterations construct the feasible solution, and the parameter $\alpha = 0$. The balance between greediness and randomness can be controlled by changing the value of the parameter *p*. In the paper, the value of the parameter *p* is 50% of the number of UCs in the TUVRP-TW network.

For TUC insertion, a reactive mechanism is adopted. The parameter α that is involved in defining the RCL set is not fixed but is selected at each iteration from a set of discrete values. The value of the parameter α is selected by referring to the solution quality found along the previous iterations. Let the set of discrete values of α be {0.0, 0.1, 0.2, 0.3, . . ., 0.8, 0.9, 1.0}. We denote the selection times of the *i*-th value for α as *usetimes* (*i*). We denote the incumbent solution at each iteration with the *i*-th value of α as *aobj (i)* and the average of objective values of all solutions found by using the *i*-th value for α as *aq* (*i*). Let *aq* (*i*) = *aobj* (*i*)/*usetimes* (*i*). The

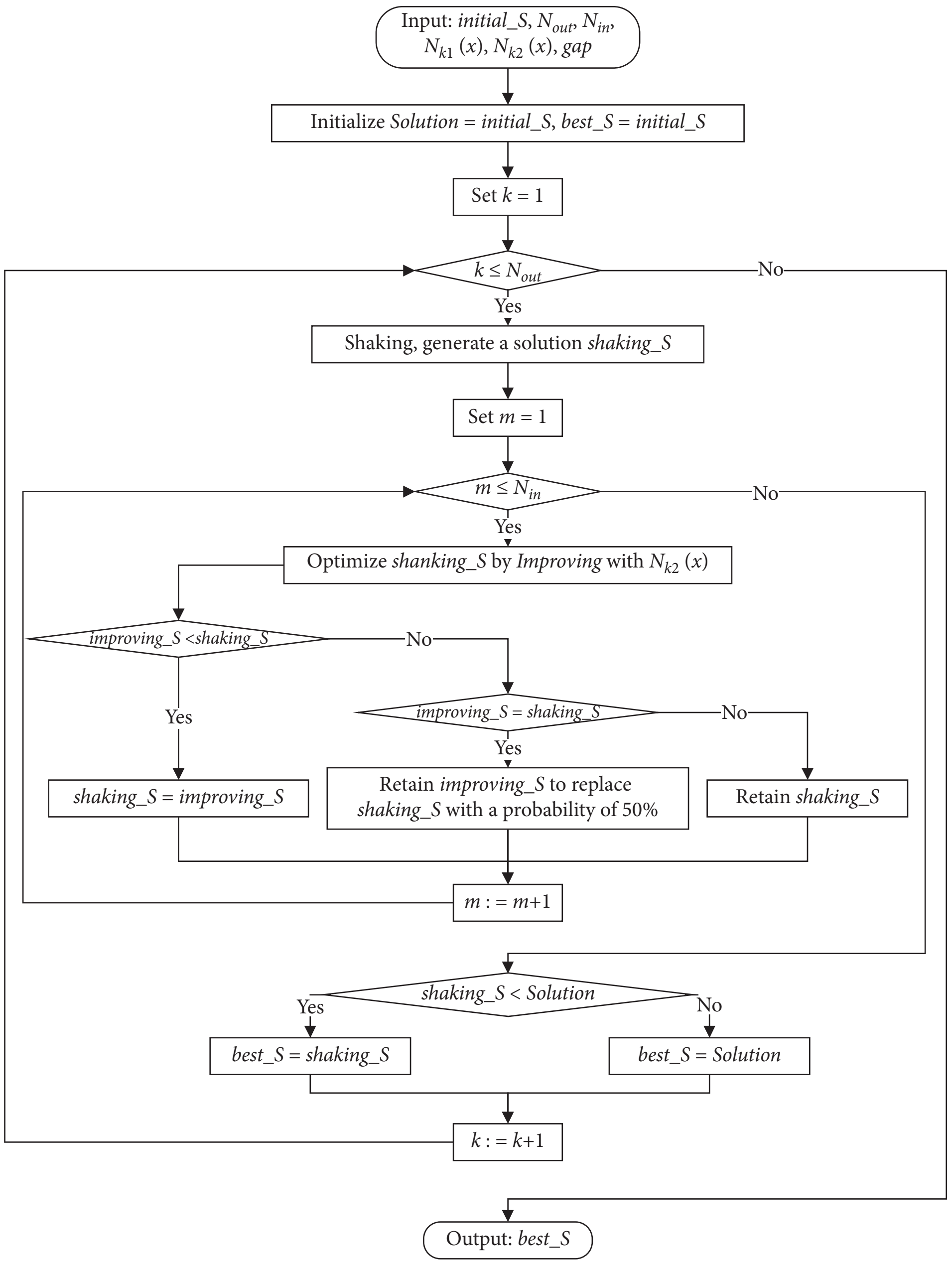
Input: initial_S, N_out, N_in, N_k1 (x), N_k2 (x), gap
Initialize Solution = initial_S, best_S = initial_S
Set k = 1
k ≤ N_out
No
Yes
Shaking, generate a solution shaking_S
Set m = 1
m ≤ N_in
No
Yes
Optimize shanking_S by Improving with N_k2 (x)
improving_S <shaking_S
No
Yes
improving_S = shaking_S
No
Yes
shaking_S = improving_S
Retain improving_S to replace shaking_S with a probability of 50%
Retain shaking_S
m : = m+1
shaking_S < Solution
Yes
No
best_S = shaking_S
best_S = Solution
k : = k+1
Output: best_S

Figure 4: The VNS process.

probabilities *ap* (i) associated with the selection of each value are initialized as *ap* $(i) = 1/11$. *usetimes* (i) and *aq* (i) are initialized as 0. When the i-th value of α is selected, *usetime* (i) is increased by 1, and *aq* (i) is increased by the insertion cost of the selected customer. The selection probability of the i-th value of α is periodically reevaluated by taking $ap\ (i) = aq\ (i)/\sum aq\ (i)$. The probabilities associated with the more appropriate values will increase when they are reevaluated.

4.2. VNS

4.2.1. The VNS Process. A VNS procedure is used to improve the initial solution (i.e., *initial_S*). We set the number of external iterations as N_{out} and the number of internal iterations as N_{in}. We define the optimization process as *Improving* and the disturbance process as *Shaking*. Within the *Shaking* process, a poor solution may be accepted, provided that its deviation does not exceed a given threshold (denoted as *gap*). N_{out} determines the number of shaking process, and N_{in} determines the number of internal iterations after the solution perturbation. The VNS combines deterministic and stochastic changes of neighborhood. The deterministic part is represented by a local search heuristic, and the stochastic phase is represented by the random selection of one point in one of the neighborhoods (Villegas et al., 2011).

At the beginning of the external iterations, *initial_S* obtained by the GRASP is regarded as the incumbent solution (denoted as *Solution*) and the best solution (denoted as *best_S*). At each external iteration, *Solution* is perturbed by *Shaking*.

Algorithm **VNS for the TUVRP-TW**

```
1: Input: initial_S, $N_{out}$, $N_{in}$, $N_{k1}(x)$, $N_{k2}(x)$, gap
2: Output: best_S
3: Initialize Solution = initial_S, best_S = initial_S
4: Set $k = 1$
5: While $k \le N_{out}$ do
6: Shaking: Based on Solution, generate a solution shaking_S from the k1-th
   neighborhood and shanking_S $\in N_{k1}(x)$, and accept a solution shanking_S whose
   divination does not exceed gap
7:  Set $m$=1
8:  While $m \le N_{in}$ do
9:    Optimize shanking_S by Improving with $N_{k2}(x)$ and find the best neighbor
      solution
      (denoted as improving_S) of shaking_S
10:   if improving_S < shaking_S
11:       shaking_S = improving_S
12:   else if improving_S = shaking_S
13:       Retain improving_S to replace shaking_S with a probability of 50%
14:   else
15:       Retain shaking_S
16:   end if
17:   $m := m+1$
18: end while
19: if shaking_S < Solution
20:     best_S = shaking_S
21: else
22:     best_S = Solution
23: $k := k+1$
24: end while
25: return best_S
```

FIGURE 5: The VNS algorithm.

Based on the selected operators of *Shaking*, *Solution* is changed, and a solution (denoted as *shaking_S*), which has the most disturbance within *gap*, is retained. After the *Shaking*, *Improving* is performed, which is iterated N_{in} times. If the attained result is worse than *Shaking_S*, *Shaking_S* is retained; if the attained result is better than *Shaking_S*, the better solution is retained. After each iteration, it will be judged whether the incumbent solution is optimized. If so, the best solution attained so far will be updated. We denote a finite set of preselected neighborhood structures with N_{k1} ($k1 = 1, \ldots, k_{\max1}$) in *shaking* and N_{k2} ($k2 = 1, \ldots, k_{\max2}$) in *Improving*. We introduce $N_{ki}(x)$ ($i = 1, 2$) to denote the set of solutions in the $k1$-th or $k2$-th neighborhood of x, where x is a feasible solution.

The process of the whole VNS for solving the TUVRP-TW is shown in Figure 4 and is described in pseudo-code in Figure 5.

In *Improving* and *Shaking*, various types of local search operators are used to search neighborhoods. The objective function of the TUVRP-TW is used as the estimation criterion for substituting the newly generated solution for the incumbent solution. Operators are designed based on relocation operators and interchange operators, which are typically used in the search phase for the VRP with time windows (VRPTW) in the literature. Operators for combination routes probably change the time continuity due to

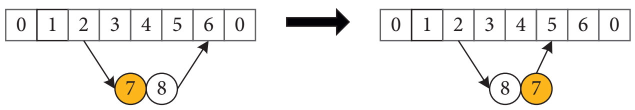

FIGURE 6: An example of the relocate operator for subtours.

the intersatellite synchronization and the varying subtour origins and destinations.

In *Shaking*, seven operators are adopted, that is, a relocation operator for truck routes or main tours of combination routes, a relocation operator for UV routes, a relocation operator for subtours, and four interchange operators for two randomly selected UV routes (denoted as $(1, 0)^{\text{PUR}}$, $(1, 1)^{\text{PUR}}$, $(1, 2)^{\text{PUR}}$, and $(2, 0)^{\text{PUR}}$).

Eighteen operators are adopted in *Improving*, that is, a relocation operator for truck routes and main tours of combination routes, a relocation operator for UV routes, a relocation operator for subtours, four interchange operators for two randomly-selected UV routes (denoted as $(1, 0)^{\text{PUR}}$, $(1, 1)^{\text{PUR}}$, $(1, 2)^{\text{PUR}}$ and $(2, 0)^{\text{PUR}}$), four interchange operators for two randomly-selected truck routes or main tours (denoted as $(1, 0)^{\text{PTR}}$, $(1, 1)^{\text{PTR}}$, $(1, 2)^{\text{PTR}}$ and $(2, 0)^{\text{PTR}}$), two interchange operators for two randomly-selected truck route (or main tour) and UV route (denoted as $(1,0)^{\text{PTR\_PUR}}$, $(0,1)^{\text{PTR\_PUR}}$, $(1,1)^{\text{PTR\_PUR}}$), two interchange operators for a randomly-selected subtour and a randomly-selected UV route (denoted as $(1, 0)^{\text{PUR\_SUB}}$ and $(1, 1)^{\text{PUR\_SUB}}$), and two interchange operators for a randomly-selected subtour and a randomly-selected truck route(or main tour) (denoted as $(1, 0)^{\text{PTR\_SUB}}$ and $(1, 1)^{\text{PTR\_SUB}}$).

4.2.2. Local Search Operators. The operators for truck routes or UV routes are similar to those usually adopted in the search phase for the routing variants involving time windows in the literature (e.g., [35]). The operators involving combination routes are relatively complex because of the intersatellite synchronization and time constraints. We take several routes shown in Figures 6–9 as examples to explain several special operators. In Figures 6–9, TUC is denoted as the box with the serial number of a customer. UC is denoted as the circle with the serial number of a customer.

Relocation operators aim to relocate a vertex of a randomly selected truck route, UV route, main tour, or subtour of a combination route. Figure 6 shows an illustrative example of the relocate operator for subtours. A combination route including the main tour "0-1-2-3-4-5-6-0" and the subtour "2-7-8-6" is randomly selected. A randomly selected vertex ("7") in the subtour is relocated, which results in a new feasible subtour "2-8-7-5" with the destination of the subtour being adjusted at vertex "5." It is worth noting that the relocate operator for subtours probably leads to the changing time continuity for the intersatellite synchronization and the subtour destination varying.

Interchange operators for two randomly selected routes aim at selecting vertices in each of the two routes for exchange. Figure 7(a) gives an example of the $(1, 0)^{\text{PTR\_PUR}}$ operator. A TUC "2" in the main tour of the combination route is randomly selected and inserted into the UV route, on condition of respecting the TUVRP-TW constraints. Figure 7(b) gives an

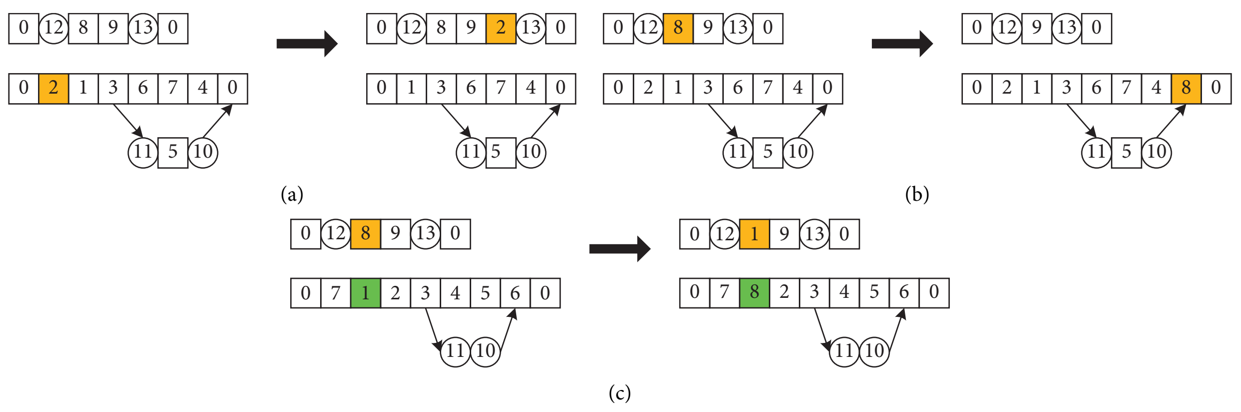

Figure 7: Examples of the interchange operators for one main tour (or truck route) and one UV route.

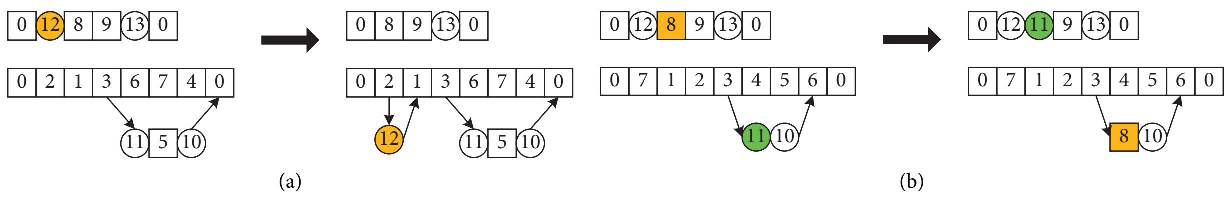

Figure 8: Examples of the interchange operators for one subtour and one UV route.

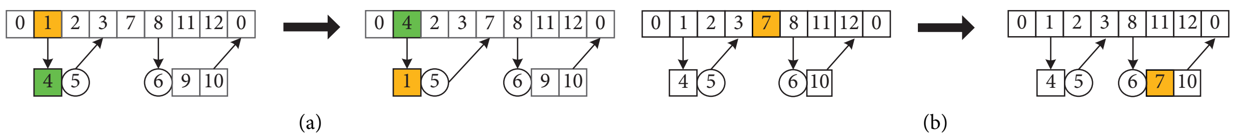

Figure 9: Examples of the interchange operators for one main tour (or truck route) and one subtour.

example of the $(0, 1)^{\mathrm{PTR\_PUR}}$ operator. A TUC "8" in the UV route is randomly selected and inserted into the main tour of the combination route, on condition of respecting the TUVRP-TW constraints. The main tour "0-2-1-3-6-7-4-0" is adjusted as "0-2-1-3-6-7-4-8-0," and the subtour "3-11-5-10-0" is transformed into "3-11-5-10-8." Figure 7(c) gives an example of the $(1, 1)^{\mathrm{PTR\_PUR}}$ operator. A TUC "1" in the main tour of the combination route is randomly selected, and a TUC "8" included in the UV route is randomly selected. The two selected TUCs are exchanged with each other.

Figure 8(a) gives an example of the $(1, 0)^{\mathrm{PUR\_SUB}}$ operator. A customer "12" in the UV route is randomly selected and assigned as the customer of TUC "2" included in the main tour of the combination route, on condition of respecting the TUVRP-TW constraints. A new subtour "2-12-1" is constructed. The UV route "0-12-8-9-13-0" is transformed into "0-8-9-13-0". Figure 8(b) gives an example of the $(1, 1)^{\mathrm{PUR\_SUB}}$ operator. A customer "11" included in a subtour of the combination route is randomly selected, and a customer "8" included in the UV route is randomly selected. The two selected customers are exchanged with each other.

Figure 9(a) gives an example of the $(1, 1)^{\mathrm{PTR\_SUB}}$ operator. In the main tour, a TUC "1" that is used as the origin of a subtour is randomly selected, and a TUC "4" included in the subtour is randomly selected. The TUCs "1" and "4" are exchanged with each other. Respecting the TUVRP-TW constraints, the main tour "0-1-2-3-7-8-11-12-0" is transformed into "0-4-2-3-7-8-11-12-0," and the subtour "1-4-5-3" is transformed into "4-1-5-7." The TUCs "4" and "7" are renewed as the origin and destination of the subtour, respectively. Figure 9(b) gives an example of the $(1, 0)^{\mathrm{PTR\_SUB}}$ operator. A TUC "7" in the main tour is randomly selected and inserted into a randomly selected subtour, on the condition that the TUVRP-TW constraints are ensured.

5. Computational Experiments

We conduct computational experiments to evaluate the performance of the intersatellite synchronization and to evaluate the effectiveness of the TUVRP-TW mathematical formulation and the applicability of the proposed heuristic for large-scale instances. The heuristic that is proposed in Section 4 is coded in C++. We use CPLEX 12.9 to solve directly the TUVRP-TW formulation and the TUVRP-TW-B formulation. The codes for the heuristic and the exact algorithm by CPLEX 12.9 are run on a computer with a Windows 10 operating system configured with an Intel(R) Core(TM) i5-10400F CPU @ 2.90-GHz processor with 16.00 GB memory.

This section is organized as follows. Section 5.1 describes the generated instances. Section 5.2 shows the intersatellite synchronization performance by referring to results of small-scale instances obtained by solving directly the TUVRP-TW formulation and the TUVRP-TW-B formulation. Section 5.3 shows heuristic results of small-scale instances obtained by the algorithm proposed in Section 4. Section 5.4 shows the performance of the TUVRP-TW heuristic for benchmark instances. Sections 5.5 and 5.6 show heuristic results of large-scale instances obtained by the proposed algorithm and the impact of customer time windows, respectively.

5.1. Test Instances. We select VRPTW test problems of Solomon [36] and convert them into two types of TUVRP-TW instances. Various scales of instances are generated for two types of computational experiments. In the first computational experiment, we use 21 small-scale instances to evaluate the intersatellite synchronization performance by solving directly the TUVRP-TW formulation and the TUVRP-TW-B formulation. In the second computational experiment, we use 21 small-scale instances and 27 large-scale instances to evaluate the performance of the TUVRP-TW formulation and the applicability of the proposed heuristic.

We derive the test instances from VRPTW benchmark problems of Solomon [36] with 100 customers. The Solomon [36] benchmark instances are classified into three categories: the R type, which consists of uniformly distributed customers; the C type, which consists of clustered customers; and the RC type, which is a mix of the R and C types. Each of the categories consists of two sets: sets R1, C1, and RC1 have narrow scheduling horizons, whereas sets R2, C2, and RC2 have wide scheduling horizons. We select the nine instances with 100 customers from set C1 for generating test instances for the two types of computational experiments.

Considering the computational experiment requirements for the TUVRP-TW and respecting the customer locations and the demand with time windows of the Solomon [36] benchmark instances, we modify the VRPTW benchmark instances by adjusting the number of customers and classifying the customers into TUCs and UCs. Each of the instances that has the same number of customers is converted into three types of test instances via a method that is similar to the method that Chao [37] used to generate the TTRP instances. In a VRPTW benchmark instance, the distance between each customer i and its nearest neighboring customer is calculated and denoted by A_i. In the first type of TUVRP-TW instances, 25% of the customers with the smallest Ai values are specified as UCs. This percentage is increased to 50% or 75% in the second type or the third type of TUVRP-TW instances, respectively.

The 21 small-scale instances that are used to evaluate the intersatellite synchronization performance are generated by selecting randomly a specified number of customers from the Solomon [36] instances with 100 customers from set C1. In an instance, the number of customers (denoted as *cn*) varies from 6 to 12 with an increment of 1. For instances that have the same number of customers, the UC percentage varies from 25% to 75% with an increment of 25%. The small-scale instances are classified according to *cn* and the UC percentage. Each small-scale instance is denoted by C1-*cn*-*cp*, where *cp* is the UC percentage. The 27 large-scale instances are generated by classifying 100 customers from set C1 according to the customer types of the Solomon [36] instances. Each instance with 100 customers from set C1 is converted into three types of large-scale instances via an approach that is similar to the method that was used to generate the small-scale instances. Each large-scale instance is denoted by "Solomon (1987) instance No."-*cp*. The parameters involved in the TUVRP-TW formulation and algorithm are evaluated in Table 2.

Table 2: Values of involved parameters.

Parameters	Value (unit)	
	Small-scale instance	Large-scale instance
Q^t	100 (kg)	100 (kg)
Q^u	100 (kg)	100 (kg)
v^t	50 (km/h)	50 (km/h)
v^u	50 (km/h)	50 (km/h)
τ	1	1
N_{out}	200	500
N_{in}	1000	2000
gap	30	70

Table 3: Exact solutions for small-scale instances.

Instance	TUVRP-TW formulation			TUVRP-TW-B formulation		
	Obj^E	m/s/U	Time (s)	Obj	m/s/U	Time (s)
C1-6-25	46.2	0/0/1	35	46.2	0/0/1	4
C1-6-50	46.2	0/0/1	5	46.2	0/0/1	<1
C1-6-75	46.2	0/0/1	2	46.2	0/0/1	<1
C1-7-25	70.6	1/1/0	673	82.2	0/0/2	214
C1-7-50	72.7	1/1/0	35	82.2	0/0/2	8
C1-7-75	72.7	1/1/0	11	82.2	0/0/2	3
C1-8-25	67.2	1/1/0	276	81.5	1/0/1	60
C1-8-50	67.2	1/1/0	33	81.5	0/0/2	5
C1-8-75	81.5	0/0/2	4	81.5	0/0/2	<1
C1-9-25	115.9	1/1/0	867	126.2	0/0/2	143
C1-9-50	126.2	0/0/2	138	126.2	0/0/2	7
C1-9-75	126.2	0/0/2	11	126.2	0/0/2	3
C1-10-25	106.4	1/1/0	14400*	121.9	1/0/1	3967
C1-10-50	121.9	0/0/2	14400*	121.9	0/0/2	654
C1-10-75	121.9	0/0/2	195	121.9	0/0/2	39
C1-11-25	207.3	0/0/3	14400*	207.3	0/0/3	2234
C1-11-50	207.3	0/0/3	10413	207.3	0/0/3	304
C1-11-75	207.3	0/0/3	670	207.3	0/0/3	57
C1-12-25	208.9	1/0/2	14400*	208.9	1/0/2	1656
C1-12-50	221.8	0/0/4	1036	221.8	0/0/4	93
C1-12-75	221.8	0/0/4	37	221.8	0/0/4	10

Note. ∗ denotes that the computation time is exhausted.

5.2. Intersatellite Synchronization Performance. Computational experiments on small-scale instances are conducted to evaluate the performance of intersatellite synchronization via a comparison of exact solutions for small-scale instances. For each small-scale instance, the total computation time for CPLEX 12.9 is set to 14400 s. CPLEX 12.9 runs with default settings until it has found an exact solution or the computation time has been exhausted. The computational results of the 21 small-scale instances are presented in Table 3. Column 1 lists the test instance. Columns 2–4 list the exact results of the TUVRP-TW

Table 4: Heuristic solutions for 21 small-scale instances.

Instance	Initial solution		Optimized solution		Time (s)	Gap1 (%)	Gap2 (%)
	Obj^I	m/s/U	Obj^O	m/s/U			
C1-6-25	46.3	0/0/1	46.2	0/0/1	1.6	0.2	0.0
C1-6-50	46.3	0/0/1	46.2	0/0/1	1.5	0.2	0.0
C1-6-75	46.2	0/0/1	46.2	0/0/1	2.4	0.0	0.0
C1-7-25	85.5	0/0/2	70.6	1/1/0	8.3	17.4	0.0
C1-7-50	86.6	0/0/2	72.7	1/1/0	4.8	16.1	0.0
C1-7-75	77.6	1/1/0	72.7	1/1/0	5.3	6.3	0.0
C1-8-25	67.7	1/1/0	67.2	1/1/0	1.3	0.7	0.0
C1-8-50	86.0	0/0/2	67.2	1/1/0	7.6	21.9	0.0
C1-8-75	118.7	0/0/2	81.5	0/0/2	0.7	31.3	0.0
C1-9-25	214.3	1/1/2	115.9	1/1/0	6.1	45.9	0.0
C1-9-50	191.9	1/0/2	126.2	0/0/2	2.6	34.2	0.0
C1-9-75	165.7	0/0/2	126.2	0/0/2	2.4	23.8	0.0
C1-10-25	173.1	1/0/2	106.4	1/1/0	9.8	38.5	$0.0^{\#}$
C1-10-50	148.0	1/1/1	121.9	0/0/2	2.9	17.6	$0.0^{\#}$
C1-10-75	150.0	1/1/1	121.9	0/0/2	3.2	18.7	0.0
C1-11-25	276.1	1/0/3	207.3	0/0/3	5.5	24.9	$0.0^{\#}$
C1-11-50	307.3	1/0/3	207.3	0/0/3	3.2	32.5	0.0
C1-11-75	247.2	1/0/2	207.3	0/0/3	3.1	16.1	0.0
C1-12-25	265.1	0/0/4	208.9	1/0/2	8.2	21.2	$0.0^{\#}$
C1-12-50	316.2	0/0/4	221.8	0/0/4	3.6	29.9	0.0
C1-12-75	222.1	00/4	221.8	0/0/4	4.5	0.1	0.0

Note. # denotes that Gap2 is estimated by referring to a solution obtained by CPLEX 12.9 in the limited computation time.

formulation. Columns 2 and 3 list the results of the objective (Obj^E) and the numbers of main tours, subtours, and UV routes (denoted as "m/s/U"). Column 4 lists the computation time (seconds). Columns 5 and 7 list the exact results of the TUVRP-TW-B formulation. Columns 5 and 6 list the results of the objective (Obj) and the numbers of main tours, subtours, and UV routes (denoted as "m/s/U"). Column 7 lists the computation time (seconds).

The exact solutions for small-scale instances are affected by the instance data. We summarize several conclusions as follows.

(i) Intersatellite synchronization enables more flexible delivery operations of vehicles, which reduces the integrated distance in the TUVRP-TW model. Of the 17 instances with exact TUVRP-TW solutions, 6 instances have smaller objective values in the TUVRP-TW formulation than in the TUVRP-TW-B formulation. The other 11 instances have the same objective values in the TUVRP-TW formulation and in the TUVRP-TW-B formulation. Two instances, namely, C1-8-25 and C1-9-25, are selected as examples for demonstrating flexible delivery operations of vehicles. For instance C1-8-25, the optimal solution of the TUVRP-TW formulation includes one main tour, namely, "0-4-0," and one subtour, namely, "4-7-5-6-8-3-1-2-0," while the optimal solution of the TUVRP-TW-B formulation includes one truck route, namely, "0-2-0," and one UV route, namely, "0-4-7-5-6-8-3-1-0." For instance C1-9-25, the optimal solution of the TUVRP-TW formulation includes one main tour, namely, "0-7-6-3-0," and one subtour, namely, "3-8-4-5-9-2-1-0," while the optimal solution of the TUVRP-TW-B formulation includes two UV routes, namely, "0-7-6-8-3-0" and "0-4-5-9-2-1-0." Hence, the combination of the main tour and the subtour reduces the integrated distance, compared with pure UV routes or pure truck routes.

(ii) A lower UC percentage for an instance corresponds to more TUCs being included in the instance. The inclusion of more TUCs in an instance corresponds to the availability of more candidate satellites for constructing subtours, which leads to a higher computational workload in the solution process. For instances that have the same number of customers, the computation time varies substantially with the UC percentage of the instance.

(iii) The results of the computational experiments on small-scale instances demonstrate that the TUVRP-TW formulation can be directly solved by CPLEX 12.9. However, CPLEX 12.9 fails to solve directly some small-scale instances that include more customers. Various strategies (e.g., the decomposition method) for improving the performance of the exact method by CPLEX 12.9 may be useful. However, instances are seldom solved directly by CPLEX 12.9 for large-scale instances because too much computer memory and computation time are required. It is necessary to evaluate the performance of the heuristic.

Table 5: The TUVRP-TW heuristic results for TTRPTW benchmark instances.

Instance	BKS	Parragh and Cordeau [38]				TUVRP-TW algorithm in the paper			
		Best (of 5)	Dev (%)	Best seen	Dev (%)	Best	Dev (%)	Gap5 (%)	Gap6 (%)
C101-25	971.82	905.02	−6.87	903.84	−7.00	1015.21	4.5	12.2	12.3
C101-50	1106.90	1004.39	−9.26	1004.39	−9.26	1120.41	1.2	11.6	11.6
C101-75	1154.80	1068.92	−7.44	1058.85	−8.31	1179.84	2.2	10.4	11.4
C201-25	671.31	671.37	0.01	671.37	0.01	713.60	6.3	6.3	6.3
C201-50	713.06	709.07	−0.56	709.07	−0.56	711.46	−0.2	0.3	0.3
C201-75	711.45	711.46	0.00	711.46	0.00	711.46	0.0	0.0	0.0
R101-25	1651.16	1644.64	−0.39	1644.64	−0.39	1633.37	−1.1	−0.7	−0.7
R101-50	1644.64	1644.64	0.00	1644.64	0.00	1637.06	−0.5	−0.5	−0.5
R101-75	1644.64	1644.64	0.00	1644.64	0.00	1642.81	−0.1	−0.1	−0.1
R201-25	1166.64	1152.25	−1.23	1147.80	−1.61	1148.20	−1.6	−0.4	0.0
R201-50	1148.17	1149.79	0.14	1147.80	−0.03	1154.79	0.6	0.4	0.6
R201-75	1147.80	1147.88	0.01	1147.80	0.00	1156.31	0.7	0.7	0.7
RC101-25	1717.01	1654.39	−3.65	1654.39	−3.65	1708.13	−0.5	3.2	3.2
RC101-50	1770.40	1724.23	−2.61	1710.32	−3.39	1759.35	−0.6	2.0	2.9
RC101-75	1784.06	1779.61	−0.25	1779.45	−0.26	1794.56	0.6	0.8	0.8
RC201-25	1277.05	1265.56	−0.90	1265.56	−0.90	1284.39	0.6	1.5	1.5
RC201-50	1273.69	1266.11	−0.60	1265.56	−0.64	1274.42	0.1	0.7	0.7
RC201-75	1266.11	1266.11	0.00	1265.98	−0.01	1272.79	0.5	0.5	0.5
Average	1267.82	1245.00		1243.20		1273.23			

5.3. Heuristic Results of Small-Scale Instances. The performance of the proposed heuristic, which is measured in terms of the quality of the heuristic solutions, is assessed by comparing heuristic solutions with exact solutions for small-scale instances. Table 4 lists the heuristic results for the 21 small-scale instances, which include the results of the initial solution (columns 2 and 3) and the best solution that were obtained by the heuristic (columns 4 and 5). Column 1 lists the instance. Column 2 lists the objective value (denoted as $\mathrm{Obj}^{\mathrm{I}}$) of the initial solution. Column 3 lists the numbers of main tours, subtours, and UV routes (denoted as "m/s/U") in the initial solution. Column 4 lists the objective value (denoted as $\mathrm{Obj}^{\mathrm{O}}$) of the heuristic. Column 5 lists the numbers of main tours, subtours, and UV routes in the heuristic solution. Column 6 lists the computation time (seconds). Column 7 lists the percentage gap between the objective value of the initial solution ($\mathrm{Obj}^{\mathrm{I}}$) and the objective value of the heuristic solution ($\mathrm{Obj}^{\mathrm{O}}$), which is estimated as $\mathrm{Gap1} = 100\% \times (\mathrm{Obj}^{\mathrm{I}} - \mathrm{Obj}^{\mathrm{O}})/\mathrm{Obj}^{\mathrm{I}}$. Column 8 lists the percentage gap between the objective value of the heuristic solution and the objective value of the exact solution (denoted as $\mathrm{Obj}^{\mathrm{E}}$), which is estimated as $\mathrm{Gap2} = 100\% \times (\mathrm{Obj}^{\mathrm{O}} - \mathrm{Obj}^{\mathrm{E}})/\mathrm{Obj}^{\mathrm{O}}$.

According to the computational results, the proposed heuristic has a general and adaptable structure. For the 21 small-scale instances, the heuristic does not require instance-related iteration settings.

According to the computation time and the Gap2 values for the small-scale instances, the performance of the heuristic is satisfactory. Compared with the exact method by CPLEX 12.9, the heuristic can output the solution quickly, and the heuristic is relatively stable. Based on the heuristic running time of the heuristic solution, the heuristic can solve each of the 21 small-scale instances in less than 10 s, and the average computation time is 4.2 s. In terms of the solution quality, the heuristic performs well. Of the 17 small-scale instances with exact solutions, Gap2 is 0.0% in all the 17 cases; hence, these instances can be solved optimally via the proposed heuristic. The gap (Gap1) between the objective value of the initial solution and that of the heuristic solution is relatively large. Of all the 21 small-scale instances, Gap1 exceeds 20.0% in 10 cases. The largest value of Gap1 is 45.9% and the average value of Gap1 is 18.9%. Gap1 is less than 1.0% in 5 cases. Although the GRASP may obtain a satisfactory solution in some cases, the VNS is necessary and important for the proposed heuristic.

5.4. Performance of the TUVRP-TW Heuristic for Benchmark Instances. The TUVRP-TW is fundamentally a new variant. From the problem definition perspective, the TUVRP-TW situation is similar to the situation of the TTRP with time windows (TTRPTW). The combination vehicle used in the TUVRP-TW involves pairing one UV with one truck. Both trucks and UVs are autonomous and can serve assigned customers. The TTRPTW involves pairing one trailer with one truck, while a trailer is nonautonomous and cannot serve customers without its truck pulling.

We adopt the TUVRP-TW heuristic to solve the TTRPTW benchmark instances. Because vehicles (i.e., truck-trailer combination and truck) involved in the TTRPTW are obviously different from vehicles (i.e., truck with/without carried UV and UV) involved in the TUVRP-TW, we assume that truck and UV have the same capacities and velocities as those of truck-trailer combination and truck and adjust the objective function of the TUVRP-TW. The results attained by the proposed heuristic and the known results of the TTRPTW benchmark instances are shown in Table 5. Column 1 shows the instances. Column 2 shows the best known solution value (BKS), which is attained in Lin et al. [39]. Columns 3 and 4 show the best results reported by Parragh and Cordeau [38] and their deviations from the

Table 6: Heuristic solutions of 27 large-scale instances.

Instance	Initial solution		Optimized solution		Time (s)	Gap1 (%)
	Obj^I	m/s/U	Obj^O	m/s/U		
C101-25	3642.4	6/0/14	2802.8	3/1/16	2533.4	23.0
C101-50	4020.4	5/2/18	2807.3	0/0/22	1211.8	30.2
C101-75	3857.4	5/3/19	2846.1	0/0/20	1139.5	26.2
C102-25	3110.0	6/1/12	2235.5	5/1/11	2309.1	28.1
C102-50	3438.3	4/2/16	2236.0	1/0/18	1385.1	35.0
C102-75	3472.0	6/2/19	2303.5	0/0/20	1342.3	33.7
C103-25	2913.3	6/3/11	1638.3	6/1/7	2463.1	43.8
C103-50	2807.0	4/0/15	1763.9	3/1/14	1715.6	37.2
C103-75	2975.7	4/0/17	1851.1	0/0/19	1217.3	37.8
C104-25	2576.5	5/2/11	1250.0	7/2/5	1557.2	51.5
C104-50	2705.2	3/5/14	1424.9	4/2/11	1787.2	47.3
C104-75	2721.6	2/1/16	1447.6	5/1/15	1433.0	46.8
C105-25	3048.5	8/4/9	2067.3	3/1/14	2965.7	32.2
C105-50	3438.4	5/2/12	2142.1	3/1/14	2740.5	37.7
C105-75	3518.2	3/3/15	2027.6	0/0/19	1193.9	42.4
C106-25	3268.6	3/0/15	1904.1	4/1/12	2070.4	41.7
C106-50	3329.7	5/3/10	1983.3	3/3/16	1770.1	40.4
C106-75	3352.6	3/1/17	2078.8	1/1/17	1775.8	38.0
C107-25	2890.7	4/1/11	1771.8	5/3/8	2674.2	38.7
C107-50	2895.6	4/1/11	1770.7	1/1/18	2280.6	38.8
C107-75	3075.0	5/3/14	1833.1	1/1/18	1344.2	40.4
C108-25	2646.8	6/1/8	1511.7	6/1/8	2664.5	42.9
C108-50	3227.9	3/0/14	1622.8	3/1/13	2023.3	49.7
C108-75	2951.6	3/2/16	1655.4	1/1/17	1399.5	43.9
C109-25	2454.6	6/2/7	1242.3	4/2/8	3179.6	49.4
C109-50	2726.9	3/0/13	1448.3	2/1/15	1790.1	46.9
C109-75	2797.0	3/2/14	1596.8	2/2/15	1856.2	42.9

BKS. Columns 5 and 6 show the best objective values encountered in Parragh and Cordeau [38] during all parameter tuning tests and their respective deviations from the best known results. Columns 7 and 8 show the best results reported by the TUVRP-TW heuristic and their deviations from the BKS. We introduce Gap5 to denote the percentage gap between columns 3 and 7. We introduce Gap6 to denote the percentage gap between columns 5 and 7.

To the best of our knowledge, the best results of TTRPTW benchmark instances are reported by Parragh and Cordeau [38]. Parragh and Cordeau [38] proposed a branch-and-price algorithm for the TTRPTW, using problem specific enhancements in the pricing scheme and alternative lower bound computations. They tailored an adaptive large neighborhood search algorithm to the TTRPTW in order to obtain good initial columns. We introduce Gap5 and Gap6 to compare results attained by the TUVRP-TW heuristic with the best results in the literature. For the 18 benchmark instances, the TUVRP-TW heuristic can attain new best solutions for four benchmark instances and the same solution for one benchmark instance. The smallest Gap5 is −0.7%, and the average Gap5 is 2.7%. The smallest Gap6 is −0.7%, and the average Gap6 is 2.9%. Generally speaking, the TUVRP-TW heuristic can obtain average gaps to best known solutions of less than 2.9%. We investigate the detailed results attained by the TUVRP-TW heuristic and find that instance data affect obviously the attained results. Some TTRPTW instances (e.g., C101-25, C101-50, and C101-75) may not be suitable for the TUVRP-TW situation.

5.5. Heuristic Applicability for Large-Scale Instances. The conclusions that are obtained by comparing heuristic solutions with exact solutions for small-scale instances and the TUVRP-TW heuristic results for TTRPTW benchmark instances support the satisfactory performance of the proposed heuristic. Several large-scale instances are solved by the heuristic in the computational experiments. Table 6 lists the heuristic results for the 27 large-scale instances. For each large-scale instance in the computational experiments, the heuristic is repeated ten times. The objective value, the number of tours or routes, and the heuristic running time of the best solution among the ten repetitions are reported. Column 1 lists the instance. Column 2 lists the objective value of the initial solution. Column 3 lists the numbers of main tours, subtours, and UV routes in the initial solution. Column 4 lists the objective value of the heuristic. Column 5 lists the numbers of main tours, subtours, and UV routes in the heuristic solution. Column 6 lists the computation time for finding the best solution. Column 7 lists the percentage gap between the objective value of the initial solution and the objective value of the heuristic solution, namely, Gap1.

The computation time of the heuristic for each of the 27 large-scale instance does not exceed 0.9 h, and the average computation time is 0.5 h. Hence, the proposed heuristic can solve one large-scale instance in an acceptable time for practices.

The initial solution that is constructed by the GRASP is feasible for enterprise operations; however, an initial solution that is constructed using the GRASP is likely to fall into

Table 7: Heuristic results of large-scale instances with changed time windows.

Instance	Obj^{O}	Time (s)	Gap3	Instance	Obj^{O}	Time (s)	Gap4
C_{br}101–25	2101.4	2814.0	−25.0	C_{na}101–25	3300.9	2357.0	17.8
C_{br}101–50	2144.5	1358.9	−23.6	C_{na}101–50	3380.7	1233.8	20.4
C_{br}101–75	2205.9	1215.8	−22.5	C_{na}101–75	3520.3	1295.4	23.7
C_{br}102–25	1750.8	2663.1	−21.7	C_{na}102–25	2663.5	2737.6	19.1
C_{br}102–50	1742.9	1808.0	−22.1	C_{na}102–50	2949.7	2406.7	31.9
C_{br}102–75	1796.2	1660.3	−22.0	C_{na}102–75	2743.8	1301.4	19.1
C_{br}103–25	1445.9	2940.9	−11.7	C_{na}103–25	2002.0	2138.3	22.2
C_{br}103–50	1575.3	2221.5	−10.7	C_{na}103–50	2069.5	2260.4	17.3
C_{br}103–75	1657.5	1246.5	−10.5	C_{na}103–75	2200.0	1264.6	18.9
C_{br}104–25	1193.9	2336.4	−4.5	C_{na}104–25	1398.7	2187.9	11.9
C_{br}104–50	1356.3	2380.4	−4.8	C_{na}104–50	1556.7	3175.7	9.2
C_{br}104–75	1403.7	1370.1	−3.0	C_{na}104–75	1707.0	1414.1	17.9
C_{br}105–25	1520.5	2763.8	−26.5	C_{na}105–25	2774.7	2420.7	34.2
C_{br}105–50	1570.9	2040.4	−26.7	C_{na}105–50	2901.9	2318.6	35.5
C_{br}105–75	1685.1	2322.3	−16.9	C_{na}105–75	2761.6	1591.4	36.2
C_{br}106–25	1428.8	2444.8	−25.0	C_{na}106–25	2462.3	1722.3	29.3
C_{br}106–50	1604.6	2260.6	−19.1	C_{na}106–50	2050.3	1222.6	3.4
C_{br}106–75	1610.9	1430.5	−22.5	C_{na}106–75	2587.1	1342.1	24.4
C_{br}107–25	1300.1	3063.9	−26.6	C_{na}107–25	2158.5	2688.6	21.8
C_{br}107–50	1474.5	2073.7	−16.7	C_{na}107–50	2291.4	1684.7	29.4
C_{br}107–75	1561.7	1294.7	−14.8	C_{na}107–75	2230.1	1277.7	21.7
C_{br}108–25	1160.3	3752.1	−23.2	C_{na}108–25	1923.2	2614.6	27.2
C_{br}108–50	1398.3	3265.9	−13.8	C_{na}108–50	2201.8	2195.1	35.7
C_{br}108–75	1426.4	1468.5	−13.8	C_{na}108–75	2033.0	1233.0	22.8
C_{br}109–25	1018.4	3337.0	−18.0	C_{na}109–25	1541.7	2825.2	24.1
C_{br}109–50	1191.3	2453.6	−17.7	C_{na}109–50	1645.3	1811.6	13.6
C_{br}109–75	1353.6	1807.3	−15.2	C_{na}109–75	1723.5	1297.0	7.9

a local optimum. The gap (Gap1) between the objective value of the initial solution and that of the heuristic solution is large. For the 27 large-scale instances, Gap1 is more than 40.0% in 14 cases. The largest value of Gap1 is 51.5%, and the average value of Gap1 is 39.5%. The VNS is highly important for finding a locally optimal solution. For the 27 large-scale instances, the average value of Gap1 is larger than the average value of Gap1 for the 21 small-scale instances. The scale of each large-scale instance is larger than that of any small-scale instance, and the initial solution of a large-scale instance may have more neighborhoods than a small-scale instance.

Comparing the initial solution with the final solution that is provided by the heuristic, various optimization strategies can be identified. For example, the number of main tours or subtours of the initial solution and the number of main tours or subtours of the final solution differ substantially. The number of main tours of the initial solution, the number subtours of the initial solution, the number of main tours of the final solution, and the number subtours of the final solution are denoted as NM^{I}, NS^{I}, NM^{O}, and NS^{O}, respectively. For 17 of the 27 large-scale instances, the percentage gap between NM^{I} and NM^{O}, namely, $\text{Gap3} = 100\% \times (\text{NM}^{\text{I}} - \text{NM}^{\text{O}})/\text{NM}^{\text{I}}$, exceeds 33.0%. The average value of Gap3 equals 33.9%. For 11 of the 27 large-scale instances, the percentage gap between NS^{I} and NS^{O}, namely, $\text{Gap4} = 100\% \times (\text{NS}^{\text{I}} - \text{NS}^{\text{O}})/\text{NS}^{\text{I}}$, exceeds 31.0%. The average value of Gap4 equals 31.8%. Hence, additional savings may be realized by optimizing the main tours and subtours. For enterprises, focusing on optimizing the main tours and subtours is more desirable.

The results of the computational trials on the 27 large-scale instances demonstrate that the instance data and the estimated values of various parameters (e.g., the vehicle velocity and the vehicle capacity) substantially affect the heuristic solutions. We summarize the management insights as follows, which are obtained by respecting the limitations of the instances and the computational experiments. (i) Catering to the "mothership approach" in the expected delivery practice, the TUVRP-TW includes intersatellite synchronization, which is expected to result in more efficient and more flexible delivery operations. Aiming at decreasing the integrated distance, it is not always suitable to adopt the "mothership approach." Of the 27 large-scale instances, six instances have no subtours in the heuristic solutions. The heuristic results of all 27 large-scale instances include 73 main tours and 29 subtours; hence, the approach of combining the main tour with one or several subtours is not always widely adopted in these instances. Considering the comparison of the exact solutions for small-scale instances that are obtained by directly solving the TUVRP-TW formulation and the TUVRP-TW-B formulation, we find that the "mothership approach" is suitable for pursuing savings in various practical scenarios. Both practitioners and researchers are suggested to identify suitable practical scenarios to employ truck-UV combinations for delivery. (ii) Using UVs for direct delivery from the depot seems necessary for reducing the integrated distance. For each instance, the number of UV routes exceeds the number of main tours or the number of subtours. Hence, the direct delivery strategy is necessary and important in implementing the "mothership approach."

5.6. Impact of Customer Time Windows. In the computational experiments on large-scale instances, sensitivity analysis is conducted to evaluate the impact of time windows on the large-scale instances. By changing the width of customer time windows, the large-scale instances designed in Section 5.1, in which the time window of customer i is $[e_i, l_i]$ and the width of the time window of customer i is denoted as w_i^{tw} ($w_i^{tw} = l_i - e_i$), are converted to another two types of instances. First, each large-scale instance is denoted by $C_{br}10J$-p. The only difference between instances $C_{br}10J$-p and $C10J$-p is the width of customer time windows. The time window of customer i in instances $C_{br}10J$-p is $[e_i - 0.5 \cdot w_i^{tw}, l_i + 0.5 \cdot w_i^{tw}]$. Second, each large-scale instance is denoted by $C_{na}10J$-p. The only difference between instances $C_{na}10J$-p and $C10J$-p is the width of customer time windows. The time window of customer i in instances $C_{na}10J$-p is $[e_i + 0.25 \cdot w_i^{tw}, l_i - 0.25 \cdot w_i^{tw}]$. The proposed heuristic is used to solve instances $C_{br}10J$-p and $C_{na}10J$-p. For each instance, the proposed heuristic is run 10 times, and the objective value and the computation time of the best solution of the 10 results are reported in Table 7.

To compare the objective values of solutions of instances $C_{br}10J$-p and $C10J$-p, we introduce a percentage gap, Gap3, where Gap3 = 100% × (the objective value of solution of instance "$C_{br}10J$-p" − the objective value of solution of instance "$C10J$-p")/the objective value of solution of instance "$C10J$-p." To compare the objective values of solutions of instances $C_{na}10J$-p and $C10J$-p, we introduce a percentage gap, Gap4, where Gap4 = 100% × (the objective value of solution of instance "$C_{na}10J$-p" − the objective value of solution of instance "$C10J$-p")/the objective value of solution of instance "$C10J$-p." The average, largest, and the smallest Gap3 values are −17.7%, −3.0%, and −26.7%, respectively. The average, largest, and the smallest Gap4 values are 22.1%, 36.2%, and 3.4%, respectively. It indicates that wide time windows may lead to the decrease of the integrated distance while narrow time windows may lead to the increase of the integrated distance. In terms of the heuristic computation time, the proposed heuristic is able to solve each of the 27 $C_{br}10J$-p instances in an average computational time of 2215 s and is able to solve each of the 27 $C_{na}10J$-p instances in an average computational time of 1927 s. Wide time windows may enlarge the neighborhood, so that more computation time is spent by the heuristic to find satisfactory solutions. Narrow time windows may cut the neighborhood, so that less computation time is spent by the heuristic to find satisfactory solutions.

6. Conclusions

The satellite synchronization that is considered in the 2E-VRP, the TTRP, and the 2E-LRP models in the literature, which is called inner-satellite synchronization in this paper, focuses on synchronization operations at a single satellite. We consider an important realistic feature in a practical scenario that logistics and supply-chain enterprises are presently facing in using UVs that are carried by trucks for "last-meter" delivery, and we propose the TUVRP-TW as a suitable methodology for overcoming the operational challenges in optimizing delivery routes for truck-UV combinations. We introduce intersatellite synchronization, which focuses on synchronization operations at multiple satellites. Using intersatellite synchronization, UVs need not return to the dispatched locations.

The TUVRP-TW involves a homogeneous fleet of truck-UV combinations for deliveries. UVs may be used for direct delivery from the depot. A UV that has been dispatched by its truck must be picked up by the same truck or must return by itself to the depot. Customers are classified into two types: TUCs and UCs. The TUCs where trucks drop off or pick up carried UVs are regarded as satellites. The TUVRP-TW involves truck-UV combinations or trucks delivering cargoes from the depot to TUCs, involves carried UVs being dispatched at satellites to visit TUCs or UCs, and involves UVs delivering cargoes directly from the depot. The TUVRP-TW objective is to minimize the integrated distance. To formulate the TUVRP-TW, both binary variables for identifying the appointed satellites and continuous variables for time continuity constraints are introduced to ensure the interaction between truck routes and UV routes.

We propose a mixed-integer linear programming model that can be solved directly using CPLEX. Compared with all types of models of the 2E-VRP, TTRP, and 2E-LRP, intersatellite synchronization constraints provide an innovative approach for formulating routing problems with satellite synchronization. A hybrid algorithm based on a GRASP and a VNS is proposed. We use computational experiments to evaluate the performances of the intersatellite synchronization and of the TUVRP-TW formulation and the applicability of the heuristic. Through a comparison of the exact solutions for small-scale instances, the satisfactory performance of the intersatellite synchronization is demonstrated. The results on small-scale and large-scale instances demonstrate that the TUVRP-TW formulation and the heuristic are effective. Besides, for 18 TTRPTW benchmark instances, the TUVRP-TW heuristic can attain new best solutions for four benchmark instances and the same solution for one benchmark instance.

For future research, extensions of the TUVRP-TW by considering, e.g., one truck carrying several UVs, are expected. In addition, more effective heuristics for the TUVRP-TW should be developed.

Appendix

Mathematical Formulation of the TUVRP-TW-B

The TUVRP-TW-B is presented on graph $G = (V, A)$ with the same customer demand and vehicle types as those for the TUVRP-TW. The parameters included in the TUVRP-TW formulation are used to describe the TUVRP-TW-B. To decision variables, a continuous variable wt_{ki}^{tu} ($k \in K$, $i \in V^{\mathrm{c}}$) is introduced to denote the possible waiting time of the truck k at the origin (that acts simultaneously as the destination) of subtours for waiting to pick up the dispatched UVs. Binary variables include the following kinds. x_{kij}^{t} is a binary variable which is 1 only if the truck k ($k \in K$) covers the arc (i, j) (i, $j \in V^{\mathrm{tu}}$). x_{krij}^{u} is a binary variable which is 1 only if the UV k ($k \in K$) carried by the truck k is dispatched at vertex r ($r \in V^{\mathrm{tu}}$) and covers the arc (i, j) (i, $j \in V$) in a subtour.

The objective function of the TUVRP-TW-B is

$$\begin{aligned}\min \sum_{k\in K}\sum_{i\in V^{\mathrm{tu}}}\sum_{j\in V^{\mathrm{tu}}} d_{ij}\cdot x_{kij}^{\mathrm{t}} &+ \sum_{k\in K}\sum_{i\in V^{\mathrm{c}}}\left(wt_{ki}^{\mathrm{t}} + wt_{ki}^{\mathrm{tu}}\right)\cdot v^{\mathrm{t}}\cdot\tau\\ &+ \sum_{k\in K}\sum_{r\in V^{\mathrm{tu}}}\sum_{i\in V}\sum_{j\in V} d_{ij}\cdot x_{krij}^{\mathrm{u}}\\ &+ \sum_{k\in K}\sum_{i\in V^{\mathrm{c}}} wt_{ki}^{\mathrm{u}}\cdot v^{\mathrm{u}}\cdot\tau.\end{aligned} \tag{A.1}$$

The objective is to minimize the integrated distance. The objective function (A.1) includes four parts: the travelling distance of trucks (with carried UVs), the transformed distance of possible waiting times of trucks at customers, the travelling distance of UVs, and the transformed distance of possible waiting times of UVs at customers.

$$\begin{aligned}&\sum_{i\in V^{\mathrm{c}},i\neq r}\sum_{j\in V} q_i x_{krij}^{\mathrm{u}} \leq Q^{\mathrm{u}},\ k\in K,\ r\in V^{\mathrm{tu}},\\ &\sum_{i\in V^{\mathrm{tuc}}}\sum_{j\in V^{\mathrm{tu}}} q_i x_{kij}^{\mathrm{t}} + \sum_{r\in V^{\mathrm{tuc}}}\sum_{i\in V^{\mathrm{c}},i\neq r}\sum_{j\in V^{\mathrm{c}}} q_i x_{krij}^{\mathrm{u}} \leq Q^{\mathrm{tu}},\ k\in K.\end{aligned} \tag{A.2}$$

The first inequation in Constraint (A.2) respect the UV capacity. The second inequation in Constraint (A.2) respect the truck-UV combination capacity.

$$\begin{aligned}&dt_k^{\mathrm{t}} + \frac{d_{0i}}{v^{\mathrm{t}}} + M\cdot\left(1 - x_{k0i}^{\mathrm{t}}\right) \geq \mathrm{at}_{ki}^{\mathrm{t}},\ k\in K,\ i\in V^{\mathrm{tuc}},\\ &dt_k^{\mathrm{t}} + \frac{d_{0i}}{v^{\mathrm{t}}} - M\cdot\left(1 - x_{k0i}^{\mathrm{t}}\right) \leq \mathrm{at}_{ki}^{\mathrm{t}},\ k\in K,\ i\in V^{\mathrm{tuc}},\\ &at_{ki}^{\mathrm{t}} + wt_{ki}^{\mathrm{t}} + s_i + \frac{d_{ij}}{v^{\mathrm{t}}} + M\cdot\left(1 - x_{kij}^{\mathrm{t}}\right) + wt_{ki}^{\mathrm{tu}} \geq \mathrm{at}_{kj}^{\mathrm{t}},\ k\in K,\\ &\qquad i\in V^{\mathrm{tuc}},\ j\in V^{\mathrm{tu}},\\ &at_{ki}^{\mathrm{t}} + wt_{ki}^{\mathrm{t}} + s_i + \frac{d_{ij}}{v^{\mathrm{t}}} - M\cdot\left(1 - x_{kij}^{\mathrm{t}}\right) + wt_{ki}^{\mathrm{tu}} \leq \mathrm{at}_{kj}^{\mathrm{t}},\ k\in K,\\ &\qquad i\in V^{\mathrm{tuc}},\ j\in V^{\mathrm{tu}},\\ &at_{ki}^{\mathrm{t}} + wt_{ki}^{\mathrm{t}} + M\cdot\left(1 - \sum_{j\in V^{\mathrm{tu}}} x_{kij}^{\mathrm{t}}\right) \geq e_i,\ k\in K,\ i\in V^{\mathrm{tuc}},\\ &at_{ki}^{\mathrm{t}} + wt_{ki}^{\mathrm{t}} - M\cdot\left(1 - \sum_{j\in V^{\mathrm{tu}}} x_{kij}^{\mathrm{t}}\right) \leq l_i,\ k\in K,\ i\in V^{\mathrm{tuc}}.\end{aligned} \tag{A.3}$$

The first and second inequations in Constraint (A.3) indicate that the truck arrival time at the first TUC is decided by the truck departure time from the depot and the running time on the arc. The third and fourth inequations in Constraint (A.3) indicate that the truck arrival time at a TUC or at the depot is decided by the truck arrival time at last vertex, the possible waiting time for time window opening, the serving time, the possible waiting time of truck k at the satellite for waiting to pick up the dispatched UV, and the running time on the arc. The fifth and sixth inequations in Constraint (A.3) ensure that the truck routes should respect TUC time windows.

$$\begin{aligned}&dt_k^{\mathrm{u}} + \frac{d_{0i}}{v^{\mathrm{u}}} + M\cdot\left(1 - x_{k00i}^{\mathrm{u}}\right) \geq at_{ki}^{\mathrm{u}},\ k\in K,\ i\in V^{\mathrm{c}},\\ &dt_k^{\mathrm{u}} + \frac{d_{0i}}{v^{\mathrm{u}}} - M\cdot\left(1 - x_{k00i}^{\mathrm{u}}\right) \leq at_{ki}^{\mathrm{u}},\ k\in K,\ i\in V^{\mathrm{c}},\\ &at_{kr}^{\mathrm{t}} + \frac{d_{ri}}{v^{\mathrm{u}}} + M\cdot\left(1 - x_{krri}^{\mathrm{u}}\right) \geq at_{ki}^{\mathrm{u}},\ k\in K,\ r\in V^{\mathrm{tuc}},\ i\in V^{\mathrm{c}},\\ &at_{kr}^{\mathrm{t}} + \frac{d_{ri}}{v^{\mathrm{u}}} - M\cdot\left(1 - x_{krri}^{\mathrm{u}}\right) \leq at_{ki}^{\mathrm{u}},\ k\in K,\ r\in V^{\mathrm{tuc}},\ i\in V^{\mathrm{c}},\\ &at_{ki}^{\mathrm{u}} + wt_{ki}^{\mathrm{u}} + s_i + d_{ij}/v^{\mathrm{u}} + M\cdot\left(1 - x_{krij}^{\mathrm{u}}\right) \geq at_{kj}^{\mathrm{u}},\ k\in K,\ r\in V^{\mathrm{tu}},\\ &\qquad i\in V^{\mathrm{c}},\ i\neq r,\ j\in V,\\ &at_{ki}^{\mathrm{u}} + wt_{ki}^{\mathrm{u}} + s_i + \frac{d_{ij}}{v^{\mathrm{u}}} - M\cdot\left(1 - x_{krij}^{\mathrm{u}}\right) \leq at_{kj}^{\mathrm{u}},\ k\in K,\ r\in V^{\mathrm{tu}},\\ &\qquad i\in V^{\mathrm{c}},\ i\neq r,\ j\in V,\\ &at_{kj}^{\mathrm{u}} + wt_{kj}^{\mathrm{u}} + M\cdot\left(1 - \sum_{i\in V} x_{krij}^{\mathrm{u}}\right) \geq e_j,\ k\in K,\ r\in V^{\mathrm{tu}},\\ &\qquad j\in V^{\mathrm{c}},\ j\neq r,\\ &at_{kj}^{\mathrm{u}} + wt_{kj}^{\mathrm{u}} - M\cdot\left(1 - \sum_{i\in V} x_{krij}^{\mathrm{u}}\right) \leq l_j,\ k\in K,\ r\in V^{\mathrm{tu}},\\ &\qquad j\in V^{\mathrm{c}},\ j\neq r.\end{aligned} \tag{A.4}$$

The first and second inequations in Constraint (A.4) indicate that the UV arrival time at the first customer is decided by the UV departure time from the depot and the running time on the arc. The third and fourth inequations in Constraint (A.4) indicate that in a subtour, the UV arrival time at the first customer is decided by the truck arrival time at the satellite and the UV running time on the arc. The fifth and sixth inequations in Constraint (A.4) indicate that the UV arrival time at a vertex is decided by the UV arrival time at last vertex, the possible waiting time for time window opening, the serving time, and the UV running time on the arc. The seventh and eighth inequations in Constraint (A.4) ensure that the UV should respect time windows of customers.

$$
\begin{aligned}
&\sum_{k\in K}\sum_{i\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kij} + \sum_{k\in K}\sum_{r\in V^{\mathrm{tu}}}\sum_{i\in V} x^{\mathrm{u}}_{krij} \geq 1,\ j \in V^{\mathrm{tuc}},\\
&\sum_{k\in K}\sum_{i\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kij} \leq 1,\ j \in V^{\mathrm{tuc}},\\
&\sum_{k\in K}\sum_{i\in V^{\mathrm{tu}}}\sum_{r\in V^{\mathrm{tu}}} x^{\mathrm{u}}_{krij} \leq 1,\ j \in V^{\mathrm{tuc}},\\
&\sum_{k\in K}\sum_{r\in V^{\mathrm{tu}}}\sum_{i\in V} x^{\mathrm{u}}_{krij} = 1,\ j \in V^{\mathrm{uc}},\\
&\sum_{i\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kij} = \sum_{a\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kja},\ k \in K,\ j \in V^{\mathrm{tu}},\\
&\sum_{i\in V} x^{\mathrm{u}}_{krij} = \sum_{a\in V} x^{\mathrm{u}}_{krja},\ k \in K,\ r \in V^{\mathrm{tu}},\ j \in V,\\
&\sum_{j\in V^{\mathrm{tuc}}} x^{\mathrm{t}}_{k0j} + \sum_{j\in V^{\mathrm{c}}} x^{\mathrm{u}}_{k00j} \leq 1,\ k \in K,\\
&\sum_{j\in V^{\mathrm{c}}} x^{\mathrm{u}}_{krrj} \leq \sum_{i\in V^{\mathrm{tu}}} x^{\mathrm{t}}_{kir},\ k \in K,\ r \in V^{\mathrm{tuc}},\\
&wt^{\mathrm{tu}}_{ki} = \max l\Big(at^{\mathrm{u}}_{ki} - \big(at^{\mathrm{t}}_{ki} + wt^{\mathrm{t}}_{ki} + s_i\big)\\
&\qquad - M\big(1 - {}^{ki}_{w}\big), 0\Big),\ k \in K,\ i \in V^{\mathrm{tuc}}.
\end{aligned}
\tag{A.5}
$$

The first, second, and third inequations in Constraint (A.5) ensure that a TUC is visited exactly once by a truck or a UV when the TUC is not being used as the satellite or a TUC is visited more than once if it is being used as the satellite. The fourth inequation in Constraint (A.5) guarantees that each UC is visited exactly once. The fifth inequation in Constraint (A.5) ensures that for each truck in use, the number of arrivals at vertex j is equal to the number of departures from vertex j. The sixth inequation in Constraint (A.5) guarantees that for each UV in use, the number of arrivals at a customer is equals to the number of departures from the customer. The seventh inequation in Constraint (A.5) requires that a UV departs by itself or departs with the paired truck no more than once from the depot. The eighth inequation in Constraint (A.5) indicates that vertex r is included in a main tour if the vertex r is used as the satellite. The ninth inequation in Constraint (A.5) defines the possible waiting time of truck k at the satellite for waiting to pick up the dispatched UV. "max$l$," which is defined by CPLEX solver, is a function to return the maximal value from a list of integers or floats.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the research grant from the National Natural Science Foundation of China (grant nos. 71672005 and 71972007).

References

[1] T. G. Crainic and A. Sgalambro, "Service network design models for two-tier city logistics," *Optimization Letters*, vol. 8, no. 4, pp. 1375–1387, 2014.

[2] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: a review of the state of the art and future directions," *Computers & Operations Research*, vol. 123, Article ID 105004, 2020.

[3] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, and G. Laporte, "Drone-aided routing: a literature review," *Transportation Research Part C: Emerging Technologies*, vol. 120, Article ID 102762, 2020.

[4] D. Rojas Viloria, E. L. Solano-Charris, A. Munoz-Villamizar, and J. R. Montoya-Torres, "Unmanned aerial vehicles/drones in vehicle routing problems: a literature review," *International Transactions in Operational Research*, vol. 28, 2020.

[5] H. Li, J. Chen, F. Wang, and M. Bai, "Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: a review," *European Journal of Operational Research*, vol. 294, no. 3, pp. 1078–1095, 2021.

[6] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[7] J. C. de Freitas and P. H. V. Penna, "A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem," *Electronic Notes in Discrete Mathematics*, vol. 66, pp. 95–102, 2018.

[8] J. C. de Freitas and P. H. V. Penna, "A variable neighborhood search for flying sidekick traveling salesman problem," *International Transactions in Operational Research*, vol. 27, no. 1, pp. 267–290, 2020.

[9] M. Dell'Amico, R. Montemanni, and S. Novellani, "Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem," *Optimization Letters*, vol. 15, pp. 1–32, 2019.

[10] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: payload-energy dependency and No-Fly zones," *International Journal of Production Economics*, vol. 214, pp. 220–233, 2019.

[11] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.

[12] E. Es Yurek and H. C. Ozmutlu, "A decomposition-based iterative optimization algorithm for traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 249–262, 2018.

[13] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 597–621, 2018.

[14] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "A hybrid genetic algorithm for the traveling salesman problem with drone," *Journal of Heuristics*, vol. 26, no. 2, pp. 219–247, 2020.

[15] M. Marinelli, L. Caggiani, M. Ottomanelli, and M. Dell'Orco, "En route truck–drone parcel delivery for optimal vehicle routing strategies," *IET Intelligent Transport Systems*, vol. 12, no. 4, pp. 253–261, 2017.

[16] S. Poikonen and B. Golden, "The mothership and drone routing problem," *INFORMS Journal on Computing*, vol. 32, 2019.

[17] P. L. Gonzalez-R, D. Canca, J. L. Andrade-Pineda, M. Calle, and J. M. Leon-Blanco, "Truck-drone team logistics: a

heuristic approach to multi-drop route planning," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 657–680, 2020.

[18] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 368–398, 2020.

[19] M. Moshref-Javadi, A. Hemmati, and M. Winkenbach, "A truck and drones model for last-mile delivery: a mathematical model and heuristic approach," *Applied Mathematical Modelling*, vol. 80, pp. 290–318, 2020a.

[20] M. Moshref-Javadi, S. Lee, and M. Winkenbach, "Design and evaluation of a multi-trip delivery model with truck and drones," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, Article ID 101887, 2020b.

[21] M. Salama and S. Srinivas, "Joint optimization of customer location clustering and drone-based routing for last-mile deliveries," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 620–642, 2020.

[22] M. Dell'Amico, R. Montemanni, and S. Novellani, "Matheuristic algorithms for the parallel drone scheduling traveling salesman problem," *Annals of Operations Research*, vol. 289, pp. 1–16, 2020.

[23] S. Poikonen and B. Golden, "Multi-visit drone routing problem," *Computers & Operations Research*, vol. 113, Article ID 104802, 2020.

[24] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: several worst-case results," *Optimization Letters*, vol. 11, no. 4, pp. 679–697, 2017.

[25] D. Schermer, M. Moeini, and O. Wendt, "A matheuristic for the vehicle routing problem with drones and its variants," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 166–204, 2019.

[26] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 289–315, 2019.

[27] J. Euchi and A. Sadok, "Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones," *Physical Communication*, vol. 44, Article ID 101236, 2021.

[28] L. Di Puglia Pugliese and F. Guerriero, "Last-mile deliveries by using drones and classical vehicles," in *Proceedings of the International Conference on Optimization and Decision Science*, pp. 557–565, Springer, Sorrento, Italy, September 2017.

[29] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5772–5782, 2021.

[30] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *International Journal of Production Economics*, vol. 225, Article ID 107598, 2020.

[31] H. Li, H. Wang, J. Chen, and M. Bai, "Two-echelon vehicle routing problem with time windows and mobile satellites," *Transportation Research Part B: Methodological*, vol. 138, pp. 179–201, 2020.

[32] N. Boysen, S. Schwerdfeger, and F. Weidinger, "Scheduling last-mile deliveries with truck-based autonomous robots," *European Journal of Operational Research*, vol. 271, no. 3, pp. 1085–1099, 2018.

[33] P. Hansen and N. Mladenović, "Variable neighborhood search for the p-median," *Location Science*, vol. 5, no. 4, pp. 207–226, 1997.

[34] P. Hansen, N. Mladenović, and D. Urošević, "Variable neighborhood search and local branching," *Computers & Operations Research*, vol. 33, no. 10, pp. 3034–3045, 2006.

[35] H. Li, L. Zhang, T. Lv, and X. Chang, "The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems," *Transportation Research Part B: Methodological*, vol. 94, pp. 169–188, 2016.

[36] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[37] I.-M. Chao, "A tabu search method for the truck and trailer routing problem," *Computers & Operations Research*, vol. 29, no. 1, pp. 33–51, 2002.

[38] S. N. Parragh and J.-F. Cordeau, "Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows," *Computers & Operations Research*, vol. 83, pp. 28–44, 2017.

[39] S.-W. Lin, V. F. Yu, and C.-C. Lu, "A simulated annealing heuristic for the truck and trailer routing problem with time windows," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15244–15252, 2011.