European Journal of Operational Research 312 (2024) 125–137

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

A multi-visit flexible-docking vehicle routing problem with drones for simultaneous pickup and delivery services

Jie Jiang[a], Ying Dai[a], Fei Yang[a], Zujun Ma[b,*]

[a] School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China
[b] School of Business Administration, Zhejiang University of Finance & Economics, Hangzhou 310018, China

ARTICLE INFO

Article history:
Received 17 August 2022
Accepted 12 June 2023
Available online 25 June 2023

Keywords:
Logistics
Vehicle routing problem with drones
Pickup and delivery services
Mixed-integer linear programming
Adaptive large neighborhood search

ABSTRACT

This study investigates a multi-visit flexible-docking vehicle routing problem that uses a truck and drone fleet to fulfill pickup and delivery requests in rural areas. In this collaborative truck–drone system, each drone may serve multiple customers per trip (multi-visit services), dock to the same or different truck from where it launched (flexible docking), and perform simultaneous pickup and delivery. These characteristics complicate the temporal, spatial, and loading synchronization for trucks and drones, making the decisions of order allocation and vehicle routing highly interdependent and intractable. This problem is formulated as a mixed-integer linear programming model and solved by a tailored adaptive large neighborhood search metaheuristic. Numerical experiments are conducted on sparse rural networks to demonstrate the efficiency of the proposed method. We observe that the proposed truck–drone system shows an average cost saving of 34% compared to the truck-only case. Moreover, deep insights into the impacts of multi-visit services, flexible docking, and simultaneous pickup and delivery on the performance of the truck–drone system are discussed.

1. Introduction

Rural e-commerce has developed rapidly in recent years. In 2020, online retail sales in China's rural regions reached RMB 1.79 trillion (USD 277 billion), as reported in MOFCOM (2021). The surging demand has meant that rural logistics must provide faster and cheaper delivery and pickup services. However, satisfying the logistics needs of remote rural residents remains challenging because of sparse demand density, poor traffic conditions, and relatively high costs. To this end, logistics participants and the government are actively exploring ways to solve this dilemma.

Drones are an emerging technology with enormous promise for logistics practitioners to achieve cheaper, faster rural logistics. Drones are high-speed, low-cost, and not beholden to the available road network. However, because of their limited endurance and payload, drone delivery must be confined in a small region. To take full advantage of drone possibilities, Amazon proposed to combine trucks and drones to complete deliveries, exploiting their complementary advantages. Subsequently, other global enterprises, such as UPS, Google, Deutsche Post, SF Express, and JD Logistics, explored this delivery method and conducted a set of tentative drone delivery operations in rural areas. Furthermore, strict national drone regulations have been adjusted to encourage their application to commercial practices.

The use of drones, especially in last-mile delivery, has attracted scholars' attention in the operations management field. A collaborative truck–drone system achieves complementary advantages at the cost of requiring more precise scheduling of all vehicles. Two typical examples are the flying sidekick traveling salesman problem (FSTSP) and the vehicle routing problem with drones (VRPD), which were respectively proposed by Murray and Chu (2015) and Wang, Poikonen, and Golden (2017). They assumed that a drone is launched from a delivery truck, serves the designated customer, and then returns to the truck to replace its battery with a fresh one for subsequent flights. On this basis, Sacramento, Pisinger, and Ropke (2019), Dell'Amico, Montemanni, and Novellani (2021), and Euchi and Sadok (2021) devised more sophisticated algorithms. Other researchers also considered more practical situations, such as nonlinear energy consumption and no-fly zones (Jeong, Song, & Lee, 2019), time windows (Coindreau, Gallay, & Zufferey, 2021), and en route operations (Schermer, Moeini, & Wendt, 2019a). As drone technology evolves and application scenarios vary, the truck–drone system can incorporate more possibilities and more complex features. There are many issues worthy of further discussion.

Unlike most previous studies, we investigate a variant of VRPD in the rural context, featuring flexible and practical operations. This more complex collaborative truck–drone system reflects the real-

[*] Corresponding author.
E-mail address: zjma@zufe.edu.cn (Z. Ma).

https://doi.org/10.1016/j.ejor.2023.06.021
0377-2217/© 2023 Elsevier B.V. All rights reserved.

ities of rural logistics and boasts greater efficiency. It simultaneously incorporates three rich features, i.e., multi-visit services, flexible docking, and simultaneous pickup and delivery. First, logistics participants need to consider simultaneous pickup and delivery in routing decisions. Vehicles (both trucks and drones) must simultaneously meet delivery and pickup requests in bidirectional logistics. This feature implies a mixed load problem (Min, 1989), profoundly affecting the route decision. For example, vehicles' mixed loads (delivery and pickup loads) fluctuate over the service process, which could potentially exceed the vehicle's capacity, resulting in an infeasible routing scheme. Second, multi-visit services are a highly practical consideration. The evolution of energy storage technology allows drones to serve multiple customers on each flight. Unlike most studies that assumed a drone to have only one-unit capacity, we consider a more economical pattern in which the drone performs multi-visit services with limited endurance and payload. Third, the assignment of drones to trucks should not be specialized. In other words, after the drone has completed its assigned mission, it can return to any available truck nearby. This principle, which we call flexible docking, utilizes vehicle resources more effectively and unleashes the system's potential through the overall deployment and scheduling of all drones.

Consideration of the rich features listed above makes the problem more practical but also more complex and challenging to model and solve. We call such a VRPD a multi-visit flexible-docking vehicle routing problem with drones with simultaneous pickup and delivery (MVFDVRPDSPD, hereafter referred to as MFS-VRPD for simplicity). In this integrated truck–drone system, all vehicles are highly interdependent. For example, a change in one route may affect other routes; in the worst case, the change may make the solution infeasible. A similar interdependence is shown in the vehicle routing problem with multiple synchronization constraints (Drexl, 2012). According to Drexl's classification, the MFS-VRPD involves the following synchronization: (i) "Task synchronization": Each customer can only be serviced once by a truck or a drone; (ii) "Operation synchronization": Collaborative scheduling on the spatial and temporal side is required. Suppose a drone ends a flight and is ready for a new one; the start time of the next flight depends on the latest arrival time of the truck and the drone at the rendezvous node; and (iii) "Load synchronization": Mixed loads are transshipped between vehicles. The truck resupplies the drone before launching it and unloads the collected packages after recovering it. The transshipped volume is a variable to be determined.

The MFS-VRPD entails addressing the optimal scheduling and routing plan considering spatial, temporal, and loading synchronization requirements. Specifically, we need to determine the optimal order allocation, truck routes, and drone flights. To achieve accurate synchronization between any truck and any drone, we should decide which drone is launched or retrieved by which truck, when and where, and how many packages are transshipped between the two types of vehicles at rendezvous nodes. To answer these questions, we develop a mixed-integer linear programming (MILP) model that captures the high interdependence between vehicles. Because commercial solvers can only obtain the optimal solutions of small-scale instances within a reasonable time, we propose an adaptive large neighborhood search (ALNS) metaheuristic with tailored neighborhood structures to solve large-size cases. Next, a set of numerical experiments are conducted to demonstrate the effectiveness of the proposed algorithm. Lastly, we conduct extensive sensitivity analyses to obtain deep insights into the impacts of operational settings and model parameters on the performance of the proposed collaborative truck–drone system.

The main contributions of this study can be summarized as follows. First, motivated by pickup and delivery practice in rural areas, we propose a more effective and practical collaborative truck–drone operation, the MFS-VRPD, which allows multi-visit services, flexible docking, and simultaneous pickup and delivery. Second, we formulate the problem as a MILP model that adequately expresses the temporal, spatial, and loading synchronization between trucks and drones. Third, an ALNS metaheuristic that embeds a collection of elaborately tailored destroy and repair operators is well designed for the involving high interdependence and proves efficient in terms of solution quality and computational time. Finally, numerical experiments demonstrate that the proposed collaborative truck–drone operation can significantly reduce the total transportation cost compared to the truck-only case. Moreover, a truck–drone system that has fewer drones but allows multi-visit services and flexible docking performs better than a system that allows neither (even when equipped with more drones). Simultaneous pickup and delivery is economically beneficial as it reduces repeated visits to customers with both delivery and pickup demands. It effectively confirms the necessity to incorporate the above three characteristics into VRPD.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on the collaborative truck–drone operation. Section 3 describes and formulates the MFS-VRPD, and Section 4 discusses the ALNS metaheuristic. Detailed results of the numerical experiments and sensitivity analysis are presented in Section 5. Finally, Section 6 provides some concluding remarks.

2. Literature review

In recent years, there has been much productive research on truck–drone operation in last-mile delivery. It is assumed that trucks and drones can work in collaboration or in parallel. Here, we review only the literature on collaborative truck–drone systems, which can be grouped into the TSP-D or the VRPD; our work belongs to the latter category.

2.1. The TSP-D

The research on the truck–drone routing problem stems from a variant of the traveling salesman problem called FSTSP, which was first introduced by Murray and Chu (2015) to address the scenario in which a drone works in collaboration with a conventional truck to deliver parcels. A similar problem, TSP-D, which allows cyclic drone operation (the launching and retrieving nodes of a drone flight may be the same), was proposed by Agatz, Bouman, and Schmidt (2018).

Based on the FSTSP and TSP-D, which consider the case of a single truck and a single drone, many studies have been devoted to extended practice and more efficient algorithms. Algorithmically, variable neighborhood search (de Freitas and Penna, 2019) and iterative optimization algorithm (Es Yurek & Ozmutlu, 2018) have been proposed. Various exact methods such as branch-and-bound (Poikonen, Golden, & Wasil, 2019; Dell'Amico et al., 2021), column-and-row generation (Boccia, Masone, Sforza, & Sterle, 2021) and dynamic programming (Bouman, Agatz, & Schmidt, 2018) have also been developed. Practically, Ha, Deville, Pham, and Hà (2018) considered the cost objective, while Ha, Deville, Pham, and Hà (2020) compared the min-cost TSP-D with the min-time TSP-D. Wang, Biao, Mengting, and Lu (2020) addressed a bi-objective problem considering the operational cost and completion time. Carlsson and Song (2018) studied a variant of TSP-D that allows the drone to launch from and return to any point along the truck route rather than only at customer nodes. This feature is also addressed in a mothership and drone routing problem (Amorosi, Puerto, & Valverde, 2021; Poikonen & Golden, 2020a). Additionally, Jeong et al. (2019), Liu, Liu, Shi, Wu, and Pedrycz (2021), Luo, Poon, Zhang, Liu, and Lim (2021), and Poikonen and Golden (2020b) considered the nonlinear energy drain function influenced by several factors such as loaded weight, travel distance, specification, and flying speed.

The collaboration of a single truck and multiple drones has been addressed in many studies. Based on the FSTSP, Murray and Raj (2020) and Raj and Murray (2020) researched the multiple flying sidekicks traveling salesman problem (mFSTSP). They considered the scheduling activities of multiple drones and nonlinear power consumption as a function of varying drone speeds and parcel weight. A variant of TSP-D that includes more than one drone and aims to minimize the total operational cost, called TSP-mD, was studied by Tu, Dat, and Dung (2018). In addition, Moshref-Javadi, Lee, and Winkenbach (2020) and Moshref-Javadi, Hemmati, and Winkenbach (2020) studied a similar problem called mTSP-D, but they focused on minimizing the sum of customer waiting time.

2.2. The VRPD

Compared with studies on the TSP-D or FSTSP, relatively few studies consider multiple drones working in collaboration with a fleet of trucks. This type of problem was first introduced by Wang et al. (2017) and is known as VRPD, which can be seen as a generalization of the FSTSP and TSP-D.

Several studies have explored VRPD with multiple trucks, each of which carries only one drone. For example, Schermer et al. (2019a) studied the vehicle routing problem with drones and en route operations (VRPDERO). They assumed that drones could launch from or return to customer nodes and discrete points on the arcs. Sacramento et al. (2019) formulated a MIP model for the VRPD that minimizes the total operational cost when considering vehicle capacity and working time limitations, and they proposed an ALNS algorithm to solve large-size problems. On this basis, Euchi and Sadok (2021) reformulated the VRPD and developed a hybrid genetic-sweep solution approach. Coindreau et al. (2021) proposed a MIP formulation and an ALNS algorithm for the VRPD with time windows.

In the case of multiple trucks equipped with multiple drones, Wang et al. (2017) and Poikonen, Wang, and Golden (2017) conducted a series of worst-case analyses and compared the maximum saving (time) between truck–drone delivery and truck-only delivery. Instead of showing the bounds on maximum saving, a mathematical model for the multi-truck multi-drone system with time windows was developed by Pugliese and Guerriero (2017). Schermer, Moeini, and Wendt (2019b) further studied the drone assignment and scheduling problem (DASP) that allows for cyclic drone operations. Similarly, Tamke and Buscher (2021) proposed a branch-and-cut algorithm for the VRPD with two time-oriented objectives: minimizing the maximum completion time (min-max) and minimizing the total completion time (min-sum). Additionally, Kitjacharoenchai, Min, and Lee (2020) addressed a two-echelon VRPD that determines trucks' routes at the first level while dealing with drones' delivery at the second level, and they designed two efficient heuristic algorithms.

In contrast with these studies, we consider the MFS-VRPD, which allows multi-visit services, flexible docking, and simultaneous delivery and pickup. Multi-visit services make the synchronization more complex in VRPD than in TSP-D with a single drone (Poikonen & Golden, 2020a; Gonzalez-R, Canca, Andrade-Pineda, Calle & Leon-Blanco, 2020; Liu et al., 2021) or multiple drones (Poikonen & Golden, 2020b; Luo et al., 2021). As a result, only Kitjacharoenchai et al. (2020), Wang and Sheu (2019), Gu, Poon, Luo, Liu, and Liu (2022), and Masmoudi, Mancini, Baldacci, and Kuo (2022) have examined this feature. In addition to multi-visit services, the MFS-VRPD proposed in this study processes more practical and complex resource constraints resulting from flexible docking and simultaneous pickup and delivery and services.

Flexible docking, differing from the commonly accepted assumption of a dedicated drone, only appeared in Kitjacharoenchai et al. (2019), Bakir and Tinic (2020), Wang and Sheu (2019), and Masmoudi et al. (2022). However, both Kitjacharoenchai et al. (2019) and Bakir and Tinic (2020) assumed a unit-capacity drone and aimed to minimize the overall makespan. In contrast, the MFS-VRPD in this study allows multi-visit and simultaneous pickup and delivery services, and its goal is to minimize the total transportation cost. So far, only Wang and Sheu (2019) and Masmoudi et al. (2022) have incorporated both multi-visit services and flexible docking into the VRPD. However, they only considered delivery services. In contrast, the MFS-VRPD further covers simultaneous pickup and delivery services. Furthermore, Wang and Sheu (2019) restricted the launch and landing of drones to a set of predefined docking hubs, whereas in the MFS-VRPD, all truck nodes have the potential to launch and retrieve drones. Masmoudi et al. (2022) realized multi-visit service by equipping the drones with different compartment configurations, significantly different from our work.

Simultaneous delivery and pickup is another significant feature considered in this study. To our knowledge, only Karak and Abdelghany (2019) have addressed this feature when studying the TSP-D. In their work, however, the single truck only visits a set of selected stations, supporting drones to fulfill all customers. They also assumed the infinite truck capacity and ignored its potential impact on routing decisions. In contrast, this study focuses on the MFS-VRPD, which fundamentally differs from the TSP-D. In our problem, both trucks and drones play an active role in serving customers, and any truck node can serve as a launch or return node. Thus, apart from routing decisions, order allocation should also be optimized. In addition, we respect capacity constraints and process the implied mixed load problem.

Finally, a comprehensive comparison between the closely related works and this study is given in online Appendix A. It shows that although the TSP-D and VRPD have been studied extensively, only a few works have considered the truck–drone system with multi-visit, flexible docking, or simultaneous pickup and delivery. Our work contributes to the literature by being the first to incorporate the three rich characteristics into the VRPD simultaneously. Moreover, the proposed MFS-VRPD allows all truck nodes to have the potential to launch and retrieve drones rather than restricting the launch and landing of drones to a set of predefined docking hubs. Thus, the truck-drone system can operate more flexibly and efficiently.

3. Problem formulation

This section specifies the MFS-VRPD, followed by a MILP model.

3.1. Problem description

The MFS-VRPD can be formally defined on a bilayer graph $G = \{N, A, M\}$, as shown in Fig. 1, where the elements denote the nodes, edges, and layers, respectively. The graph contains two layers $M = \{T, D\}$, corresponding to the truck road and drone flight layers. Each layer contains the nodes $N = \{0, 1, .., n+1\}$, including the physical depot 0 ($n+1$ is a copy of 0) and all customers $C = \{1, 2, .., n\}$. Each customer $i \in C$ is associated with a delivery demand d_i and a pickup demand p_i. For convenience of expression, we state the notions $N_0 = \{0, 1, .., n\}$ and $N_+ = \{1, 2, .., n+1\}$. Let the edge from $i \in N_0$ to $j \in N_+ (j \neq i)$ in each layer be A^M, presenting the existing road network A^T traveled by truck or the flight path network A^D by drone. Each edge is characterized by distance $d_{ij}^M (d_{ij}^T \geq d_{ij}^D)$ and travel time $t_{ij}^M (t_{ij}^T > t_{ij}^D)$.

A fleet of homogeneous trucks K and multirotor drones D are coordinated to provide delivery and pickup services to customers.

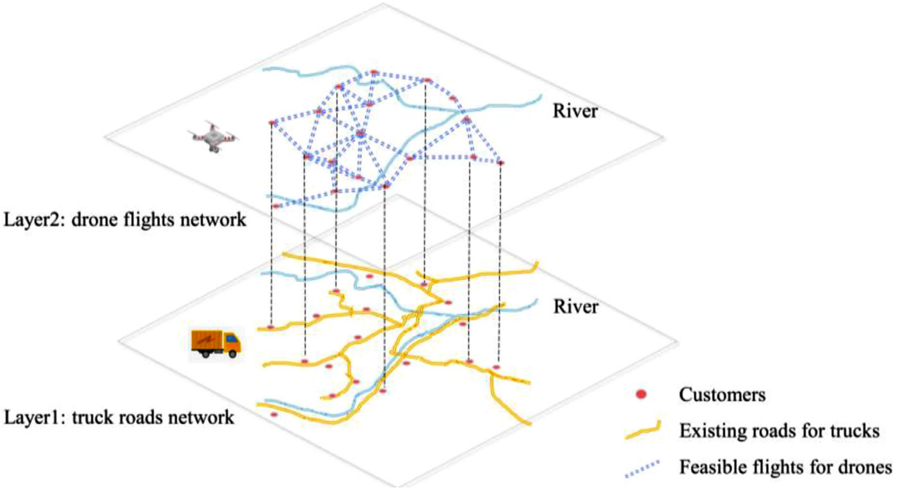

River
Layer2: drone flights network
River
Layer1: truck roads network
Customers
Existing roads for trucks
Feasible flights for drones

Fig. 1. The bilayer graph.

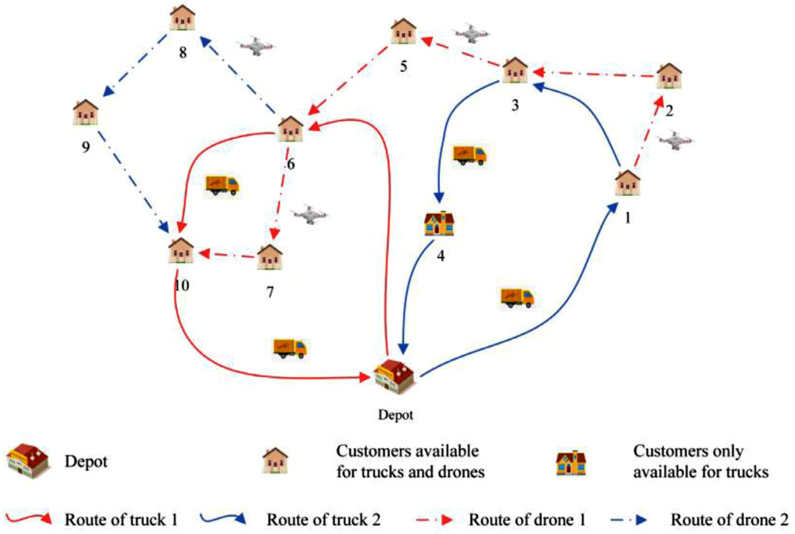


Fig. 2. An example of an MFS-VRPD solution.

Trucks and drones start from depot 0 and return to depot $n+1$. Each truck with a maximum capacity Q^T, carrying multiple drones, serves customers along its route. At the depot and any customer node visited by trucks, one or more drone(s) may be launched to carry out sub-routes independently. In each sub-route, the drone can serve one or multiple customers within its maximum payload Q^D and flying endurance time e. Once a sub-route ends, the drone returns to any nearby truck, which could be the one that launched the drone or any other truck, to unload the collected parcels and get a fully charged battery for the next dispatch. While drones are airborne, the truck can head out to serve the remaining customers instead of waiting in place for the drone to return. The maximum working hour of each vehicle must not exceed the predefined time L. Let c^T and c^D be the unit travel costs of trucks and drones, respectively. The MFS-VRPD aims to find the optimal routes for given trucks and drones to fulfill all customers collaboratively and minimize the total transportation cost. A visual example of an MFS-VRPD solution is depicted in Fig. 2.

Throughout this paper, the following assumptions are made: (i) Each customer must be serviced once by a truck or a drone; (ii) Drones may fly from or return to the depot or any customer node visited by trucks; (iii) If a drone arrives at a rendezvous node earlier than a truck, it can land to save energy and wait for the truck's arrival to perform synchronization operations[1] and vice versa; (iv) The time needed to launch or retrieve a drone is not considered; and (v) The time required for trucks and drones to serve customers is negligible.

We use the following decision variables to model the MFS-VRPD. (i) Binary variables: x_{ijk} equals 1 if truck k travels through edge $(i,j) \in A^T$; y_{ijd} equals 1 if drone d flies through edge $(i,j) \in A^D$; z_{ijkd} equals 1 if truck k traverses edge $(i,j) \in A^T$ with drone d on board; w_{ikd} equals 1 if truck k launches/retrieves drone d at node i. (ii) Continuous variables: a^T_{ik} and a^D_{id} identify the arrival time of truck k and of drone d at node i, respectively; b^-_{id} and b^+_{id} indicate the cumulative flight time of drone d when it arrives at

[1] Emerging technologies for drone safety (Ganesh, S., & Menendez, 2016; Yariv & Regev, 2022; Zhou, Zhu, & Wu, 2021) can ensure that drones land safely while waiting for trucks. Moreover, some of the existing literature (Cheng, Adulyasak, & Rousseau, 2020; Gonzalez-R et al., 2020) had similar implicit assumptions. However, given that there is literature that posited that if the drone arrives at a rendezvous node earlier than the truck, it can hover there waiting for the truck to arrive (Ha et al., 2020; Tamke & Buscher, 2021), then only a minor adjustment to our model is needed to accommodate this situation. Specifically, simply add the item $\max(0, a^D_{jd} - a^T_{jk})$ to the right side of constraint (23), and it can be linearized by $b^-_{jd} \geq b^+_{id} + t^D_{ij} - (1 - y_{ijd})X$ and $b^-_{jd} \geq b^+_{id} + t^D_{ij} + a^D_{jd} - a^T_{jk} - (1 - y_{ijd})X$.

and leaves from node i since its last launch from a truck node or depot; F_{ik} and G_{ik} denote the delivery and pickup loads of truck k when it arrives at node i; f^-_{id} and g^-_{id} express the delivery and pickup loads of drone d when it arrives at node i; f^+_{id} and g^+_{id} specify the delivery and pickup loads of drone d after the load transshipment has been finished at node i.

3.2. Mathematical formulation

With the defined notations, we formulate the MFS-VRPD as a MILP model. The objective is to minimize the total transportation cost incurred by all vehicles, as follows:

$$\min c^T \sum_{k\in K}\sum_{(i,j)\in A^T} x_{ijk} d^T_{ij} + c^D \sum_{d\in D}\sum_{(i,j)\in A^D} y_{ijd} d^D_{ij} \tag{1}$$

The constraints fall into several categories, as follows.

- Routing constraints

$$\sum_{j\in N_+} x_{0jk} = \sum_{i\in N_0} x_{i,n+1,k} \cdot \forall k \in K \tag{2}$$

$$\sum_{j\in N_+} x_{0jk} \le 1 \cdot \forall k \in K \tag{3}$$

$$\sum_{i\in N_0} x_{ijk} = \sum_{h\in N_+} x_{jhk} \cdot \forall j \in C, k \in K \tag{4}$$

$$\sum_{j\in N_+} y_{0jd} + \sum_{k\in K}\sum_{j\in N_+} z_{0jkd} = \sum_{i\in N_0} y_{i,n+1,d} + \sum_{k\in K}\sum_{i\in N_0} z_{i,n+1,k,d} \cdot \forall d \in D \tag{5}$$

$$\sum_{i\in N_0} y_{ijd} + \sum_{k\in K}\sum_{i\in N_0} z_{ijkd} = \sum_{h\in N_+} y_{jhd} + \sum_{k\in K}\sum_{h\in N_+} z_{jhkd} \cdot \forall j \in C, d \in D \tag{6}$$

$$\sum_{i\in N_0} y_{ijd} + \sum_{k\in K}\sum_{i\in N_0} z_{ijkd} \le 1 \cdot \forall j \in N_+, d \in D \tag{7}$$

$$\sum_{k\in K}\sum_{i\in N_0} x_{ijk} \le 1 \cdot \forall j \in C \tag{8}$$

$$z_{ijkd} \le x_{ijk} \cdot \forall (i,j) \in A^T, k \in K, d \in D \tag{9}$$

Constraints (2) and (3) ensure that each truck starts and ends its route at the depot only once. Constraint (4) guarantees the truck flow conservation. Constraint (5) ensures that each drone departs from and returns to the depot, and the drone flow conservation is provided in constraint (6). Constraint (7) ensures that each drone visits a customer at most once, whether it flies independently or is carried by a truck. Constraint (8) guarantees that a truck can only access each customer once. Constraint (9) illustrates the relation of z_{ijkd} and x_{ijk}.

- Spatial constraints

$$\sum_{k\in K}\sum_{i\in N_0} x_{ijk} + \sum_{d\in D}\sum_{i\in N_0} y_{ijd} \ge 1 - \sum_{k\in K}\sum_{d\in D} w_{jkd} \cdot \forall j \in C \tag{10}$$

$$\sum_{k\in K}\sum_{i\in N_0} x_{ijk} + \sum_{d\in D}\sum_{i\in N_0} y_{ijd} \le 1 + \sum_{k\in K}\sum_{d\in D} w_{jkd} \cdot \forall j \in C \tag{11}$$

$$\sum_{i\in N_0} x_{ijk} + \sum_{i\in N_0} y_{ijd} + \sum_{h\in N_+} x_{jhk} + \sum_{h\in N_+} y_{jhd} \ge 3w_{jkd} \cdot \forall j \in C, k \in K, d \in D \tag{12}$$

$$\sum_{i\in N_0} x_{ijk} + \sum_{i\in N_0} y_{ijd} + \sum_{h\in N_+} x_{jhk} + \sum_{h\in N_+} y_{jhd} \le 2 + 2w_{jkd} \cdot \forall j \in C, k \in K, d \in D \tag{13}$$

Constraints (10) to (13) define the rules of arrival and departure for trucks and drones, as illustrated in Fig. 3. Constraints (10) and (11) restrict that each customer can be visited once by a truck or a drone independently, as illustrated in Figs. 3(a) and 3(b). While a truck synchronizes with a drone at a specific customer node, constraints (10) and (11) are nonbinding, and there are three possibilities illustrated in constraints (12) and (13). The drone may be retrieved (Fig. 3(c)) or launched (Fig. 3(d)) by the truck or first collected and then launched at a node (Fig. 3(e)), which is dependent on the value of y.

- Timing constraints

$$a^T_{jk} \ge a^T_{ik} + t^T_{ij} - (1 - x_{ijk})X \cdot \forall i \in N_0, j \in N_+, k \in K \tag{14}$$

$$a^T_{jk} \ge a^D_{id} + t^T_{ij} - (1 - x_{ijk})X \cdot \forall i \in N_0, j \in N_+, k \in K, d \in D \tag{15}$$

$$a^D_{jd} \ge a^D_{id} + t^D_{ij} - (1 - y_{ijd})X \cdot \forall i \in N_0, j \in N_+, d \in D \tag{16}$$

$$a^D_{jd} \ge a^T_{ik} + t^D_{ij} - (1 - y_{ijd})X \cdot \forall i \in N_0, j \in N_+, k \in K, d \in D \tag{17}$$

$$a^T_{jk} \le L \sum_{i\in N_0} x_{ijk} \cdot \forall j \in N_+, k \in K \tag{18}$$

$$a^D_{jd} \le L\left(\sum_{i\in N_0} y_{ijd} + \sum_{k\in K}\sum_{i\in N_0} z_{ijkd}\right) \cdot \forall j \in N_+, d \in D \tag{19}$$

Constraints (14) and (15) express the time relationship for truck k from node i to j, where X is a sufficiently large number (similarly hereinafter). Similarly, constraints (16) and (17) denote the time relationship for drone d. Constraints (18) and (19) guarantee that the maximum working hours for all vehicles are respected.

$$b^+_{id} \le \left(1 - \sum_{k\in K} w_{ikd}\right)X \cdot \forall i \in N, d \in D \tag{20}$$

$$b^+_{id} \ge b^-_{id} - X \sum_{k\in K} w_{ikd} \cdot \forall i \in C, d \in D \tag{21}$$

$$b^+_{id} \le b^-_{id} + X \sum_{k\in K} w_{ikd} \cdot \forall i \in C, d \in D \tag{22}$$

$$b^-_{jd} \ge b^+_{id} + t^D_{ij} - \left(1 - y_{ijd}\right)X \cdot \forall i \in N_0, j \in N_+, d \in D \tag{23}$$

$$b^-_{jd} \le e \sum_{i\in N_0} y_{ijd} \cdot \forall j \in N_+, d \in D \tag{24}$$

Constraints (20) to (24) guarantee the feasibility of the drones' flight endurance. Constraint (20) states that each drone is fully charged (full endurance time) when starting its new flight. During each flight, constraints (21) and (22) track the cumulative flight time of a drone when it leaves any customer node. Constraint (23) updates the cumulative flight time of a drone as it reaches to node j through edge (i, j). Constraint (24) ensures that the flight endurance of drones is respected.

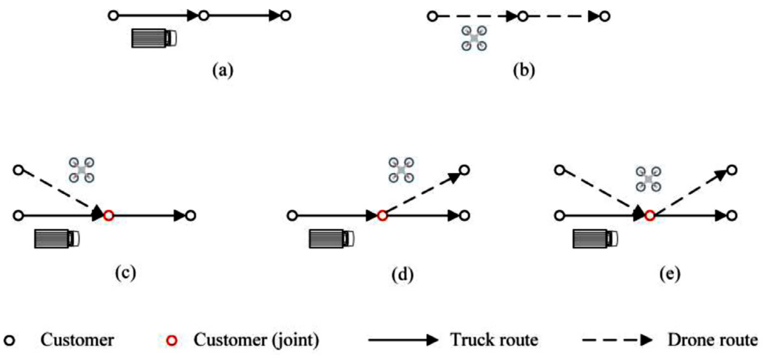


Fig. 3. Five potential cases of spatial synchronization.

- Loading constraints

$$F_{jk} \geq F_{ik} - d_i - \sum_{d \in D} f^+_{id} - (1 - x_{ijk})X \cdot \forall i \in C, j \in N_+, k \in K \quad (25)$$

$$F_{jk} \leq F_{ik} - d_i - \sum_{d \in D} f^+_{id} + \left(1 - x_{ijk}\right)X \cdot \forall i \in C, j \in N_+, k \in K \quad (26)$$

$$G_{jk} \leq G_{ik} + p_i + \sum_{d \in D} g^-_{id} + \left(1 - x_{ijk}\right)X \cdot \forall i \in C, j \in N_+, k \in K \quad (27)$$

$$G_{jk} \geq G_{ik} + p_i + \sum_{d \in D} g^-_{id} - \left(1 - x_{ijk}\right)X \cdot \forall i \in C, j \in N_+, k \in K \quad (28)$$

$$F_{jk} \leq F_{0k} + \left(1 - x_{0jk}\right)X \cdot \forall j \in N_+, k \in K \quad (29)$$

$$G_{jk} \geq G_{0k} - \left(1 - x_{0jk}\right)X \cdot \forall j \in N_+, k \in K \quad (30)$$

$$F_{ik} + G_{ik} \leq Q^T \cdot \forall i \in N, k \in K \quad (31)$$

Constraints (25) to (31) ensure the feasibility of truck capacity. The delivery load of truck k is updated when it performs a delivery or replenishes the dispatched drone(s) at a node, as shown in constraints (25) and (26). Similarly, constraints (27) and (28) update the pickup load when truck k executes a pickup or transships the parcels collected by returning drone(s). Constraints (29) and (30) update the delivery and pickup load when truck k starts from the depot. Constraint (31) ensures that the capacity of each truck is respected.

$$f^-_{jd} \leq f^+_{id} - d_i\left(1 - \sum_{k \in K} w_{ikd}\right) + \left(1 - y_{ijd}\right)X \cdot \forall i \in C, j \in N_+, d \in D \quad (32)$$

$$g^-_{jd} \geq g^+_{id} + p_i\left(1 - \sum_{k \in K} w_{ikd}\right) - \left(1 - y_{ijd}\right)X \cdot \forall i \in C, j \in N_+, d \in D \quad (33)$$

$$f^-_{jd} \leq \left(1 - \sum_{k \in K} w_{jkd}\right)X \cdot \forall j \in N_+, d \in D \quad (34)$$

$$g^+_{jd} \leq \left(1 - \sum_{k \in K} w_{jkd}\right)X \cdot \forall j \in N_0, d \in D \quad (35)$$

$$f^-_{jd} \geq f^+_{jd} - X\sum_{k \in K} w_{jkd} \cdot \forall j \in C, d \in D \quad (36)$$

$$f^-_{jd} \leq f^+_{jd} + X\sum_{k \in K} w_{jkd} \cdot \forall j \in C, d \in D \quad (37)$$

$$g^-_{jd} \geq g^+_{jd} - X\sum_{k \in K} w_{jkd} \cdot \forall j \in C, d \in D \quad (38)$$

$$g^-_{jd} \leq g^+_{jd} + X\sum_{k \in K} w_{jkd} \cdot \forall j \in C, d \in D \quad (39)$$

$$f^-_{jd} + g^-_{jd} \leq Q^D \cdot \forall j \in N, d \in D \quad (40)$$

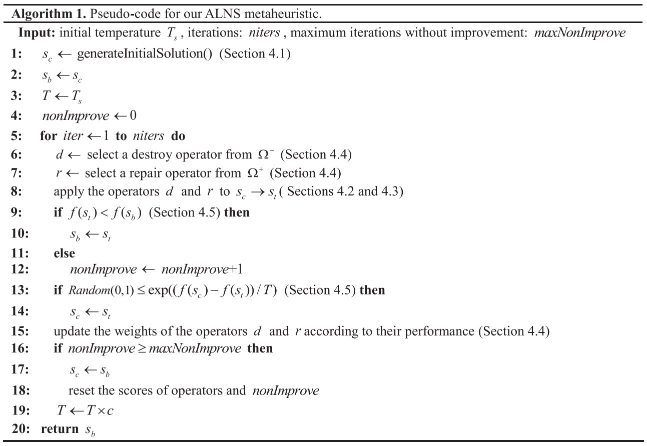
Constraints (32) to (40) ensure the feasibility of the drone payload. The loads (pickup and delivery loads) are updated when a drone arrives at node j after serving node i, as described in constraints (32) and (33). Constraint (34) requires that all assigned delivery tasks have been finished when each drone is retrieved at a rendezvous node. Constraint (35) requires that all collected packages have been unloaded before a drone starts a new sub-route. While the drone flies alone, no parcel transshipment occurs (constraints (36)–(39)); that is, $f^-_{jd} = f^+_{jd}$ and $g^-_{jd} = g^+_{jd}$. Constraint (40) ensures that the drone's payload is respected everywhere.

4. The ALNS metaheuristic

Because of the complexity of the MFS-VRPD, only small-scale instances can be solved optimally by commercial solvers within a reasonable running time. Therefore, it is necessary to develop an efficient solution technique for large-scale problems. The ALNS metaheuristic has proved its efficiency in VRP classes (Ropke & Pisinger, 2006; Chen, Demir, & Huang, 2021), which inspires us to design a tailored ALNS metaheuristic for solving the MFS-VRPD.

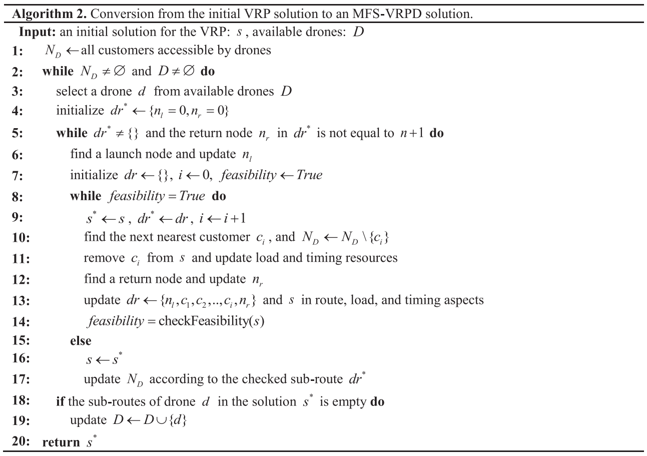
The general ALNS takes several destroy and repair operators as input. It begins with an initial solution and improves the objective value by probabilistically applying a combination of destroy and repair methods based on their past performance. The pseudo-code of our ALNS framework is given in Algorithm 1. First, a two-phase heuristic generates an initial solution (line 1). Then, it is gradually improved by repeatedly searching the neighborhood of the current solution until the predefined iterations *niters* are reached (lines 5–19). Each iteration first selects destroy and repair operators (lines 6–7) by an adaptive mechanism. Then, a new solution s_t is obtained by executing the chosen operators (line 8). Only when s_t improves on the global best solution s_b or meets the acceptance criterion of simulated annealing is it preserved and assigned to the current solution s_c before moving on to subsequent iterations (lines 9–19).

Algorithm 1
Pseudo-code for our ALNS metaheuristic.



```
Algorithm 1. Pseudo-code for our ALNS metaheuristic.
Input: initial temperature $T_s$, iterations: $niters$, maximum iterations without improvement: $maxNonImprove$
$s_c \leftarrow$ generateInitialSolution() (Section 4.1)
$s_b \leftarrow s_c$
$T \leftarrow T_s$
$nonImprove \leftarrow 0$
for $iter \leftarrow 1$ to $niters$ do
    $d \leftarrow$ select a destroy operator from $\Omega^-$ (Section 4.4)
    $r \leftarrow$ select a repair operator from $\Omega^+$ (Section 4.4)
    apply the operators $d$ and $r$ to $s_c \rightarrow s_t$ ( Sections 4.2 and 4.3)
    if $f(s_t) < f(s_b)$ (Section 4.5) then
        $s_b \leftarrow s_t$
    else
        $nonImprove \leftarrow nonImprove+1$
    if $Random(0,1) \le \exp((f(s_c)-f(s_t))/T)$ (Section 4.5) then
        $s_c \leftarrow s_t$
    update the weights of the operators $d$ and $r$ according to their performance (Section 4.4)
    if $nonImprove \ge maxNonImprove$ then
        $s_c \leftarrow s_b$
        reset the scores of operators and $nonImprove$
    $T \leftarrow T \times c$
return $s_b$
```

Algorithm 2
Conversion from the initial VRP solution to an MFS-VRPD solution.



```
Algorithm 2. Conversion from the initial VRP solution to an MFS-VRPD solution.
Input: an initial solution for the VRP: $s$, available drones: $D$
$N_D \leftarrow$ all customers accessible by drones
while $N_D \neq \varnothing$ and $D \neq \varnothing$ do
    select a drone $d$ from available drones $D$
    initialize $dr^* \leftarrow \{n_l = 0, n_r = 0\}$
    while $dr^* \neq \{\}$ and the return node $n_r$ in $dr^*$ is not equal to $n+1$ do
        find a launch node and update $n_l$
        initialize $dr \leftarrow \{\}$, $i \leftarrow 0$, $feasibility \leftarrow True$
        while $feasibility = True$ do
            $s^* \leftarrow s$, $dr^* \leftarrow dr$, $i \leftarrow i+1$
            find the next nearest customer $c_i$, and $N_D \leftarrow N_D \setminus \{c_i\}$
            remove $c_i$ from $s$ and update load and timing resources
            find a return node and update $n_r$
            update $dr \leftarrow \{n_l, c_1, c_2, .., c_i, n_r\}$ and $s$ in route, load, and timing aspects
            $feasibility =$ checkFeasibility($s$)
        else
            $s \leftarrow s^*$
            update $N_D$ according to the checked sub-route $dr^*$
    if the sub-routes of drone $d$ in the solution $s^*$ is empty do
        update $D \leftarrow D \cup \{d\}$
return $s^*$
```

4.1. Initial solution

This section shows a quick two-phase heuristic to construct the initial solution as a starting point of the ALNS algorithm. In the first phase, it generates a VRP solution with the nearest neighbor method. It requires that a truck starts from the depot and repeatedly visits the nearest customer, provided that capacity and duration limitations are respected. Once one of the constraints is violated, a new truck will be dispatched. This procedure keeps looping until all customers are served. In the second phase, the initial VRP solution is converted to an MFS-VRPD solution by assigning some truck nodes to drones, as outlined in Algorithm 2 and explained in online Appendix B.1.

4.2. Destroy methods

In each iteration, the ALNS metaheuristic removes customers and the associated sub-routes, then puts them into the removal-node set R_n and sub-route set R_s, respectively, until at least β customers are removed. The technical points of destroy methods are elaborated as follows.

4.2.1. Removal procedure

In the classical VRPD, the removal procedure only means removing the chosen customer. However, because of the high interdependency in the MFS-VRPD, our removal procedure implies the removal of both the selected node and the resulting infeasible parts. To illustrate the distinctive removal process, the following definitions are proposed.

Definition 1. Each sub-route $r^D = \{n_l, n_1, n_{2,...,}n_r\}$, which consists of a launch node n_l, one or multiple drone nodes, and a docking node n_r, is associated with a value $H(r^D)$ corresponding to whether r^D satisfies docking consistency (i.e., whether n_l and n_r are served by the same truck).

Based on the defined docking consistency, another definition is proposed.

Definition 2. Each selected customer j is associated with a state vector $(M(j), V^{in}(j), V^{out}(j), V(j), \bar{R}(j))$ corresponding to (i) the type of service vehicle; (ii) the set of inflow vehicles ($V^{in}(j) = \{k : \sum_{i\in N_0} x_{ijk} = 1\} \cup \{d : \sum_{i\in N_0} y_{ijd} = 1\}$); (iii) the set of outflow vehicles ($V^{out}(j) = \{k : \sum_{h\in N_+} x_{jhk} = 1\} \cup \{d : \sum_{h\in N_+} y_{jhd} = 1\}$); (iv) the total number of inflow and outflow vehicles ($V(j) = |V^{in}(j)| + |V^{out}(j)|$, $V(j) \ge 2$); (v) the sub-route(s) containing customer j, respectively. $\bar{R}(j)$ is meaningful only when customer j is visited by a drone ($V^{in}(j) \cup V^{out}(j) \cap D \neq \emptyset$); otherwise, it takes a value of *null*.

With the above definitions, the complex removal procedure can be explained as follows:

Definition 3. *General removal*, which requires the removal of the selected customer j, is encouraged if and only if $V(j) = 2$, indicating that customer j is visited and serviced by only one truck $k \in V^{in} = V^{out}$ or one drone $d \in V^{in} = V^{out}$.

Definition 4. *Simultaneous removal*, which involves the removal of the selected customer j and all associated sub-route(s) $r^D \in \bar{R}(j)$, is suggested if and only if $V(j) > 2$. It indicates that a truck $k \in (V^{in}(j) \cap V^{out}(j) \cap K)$ serves customer j, coupled with launching drone(s) $d \in (V^{in}(j) \cap D)$ or retrieving drone(s) $d \in (V^{out}(j) \cap D)$.

Definition 5. *Consistent removal* removes sub-route(s) ahead or behind because of the elimination of node j and associated sub-route r^D until the docking node of the sub-route closely prior to r^D and the launch node of the sub-route immediately behind r^D are consistent in the truck route. It is triggered only on two occasions: (i) $M(j) = D$, $|r^D \backslash \{j\}| = 2$, and $H(r^D) = 0$; or (ii) $M(j) = T$ and $H(r^D) = 0$.

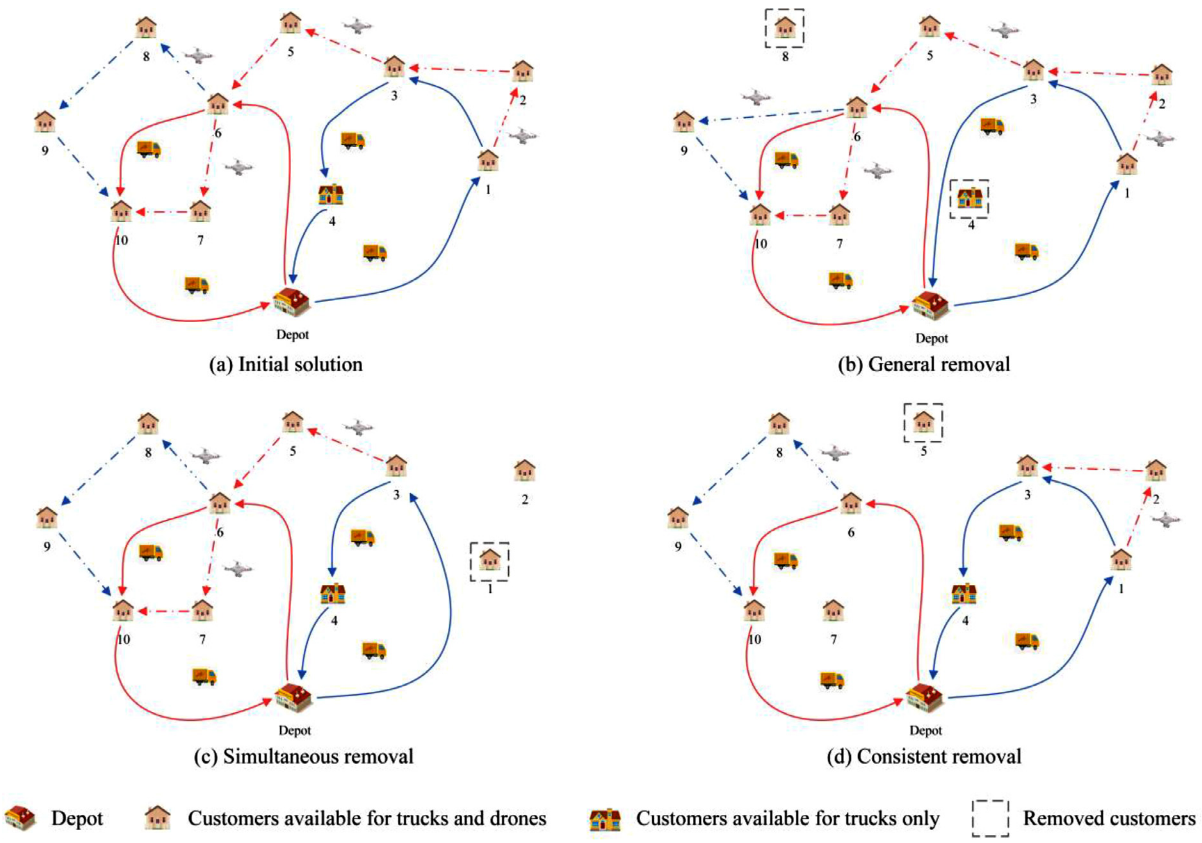
The visual depiction in Fig. 4 illustrates these types of removal more clearly. The current solution is given in Fig. 4(a). On this basis, Fig. 4(b) shows the general removal of a drone node (i.e., 8) and a truck node (i.e., 4), neither of which affects the other parts. Fig. 4(c) eliminates the sub-route $\{1, 2, 3\}$ because of the removal of launch node 1, as described in Definition 4. Removing drone node 5 and its associated sub-route $\{3, 5, 6\}$ conforms to Definition 5. Hence, the sequent sub-route $\{6, 7, 10\}$ becomes infeasible and should be removed, as depicted in Fig. 4(d).

4.2.2. Destroy operators

Three classical destroy operators are utilized in our ALNS metaheuristic. Each destroy operator gives a strategy for selecting a new removal customer, described as follows.

Shaw destroy. The main idea of this removal heuristic is to remove several customers with high similarity and then reinsert them in better positions. First, we randomly select a customer i. Second, we repeatedly select the next customer j with the highest similarity (the smallest $R(i, j)$), calculated as $R(i, j) = \omega_1 \min(\bar{d}^T_{ij}, \bar{d}^D_{ij}) + \omega_2(|\bar{d}_i - \bar{d}_j| + |\bar{p}_i - \bar{p}_j|)/2$, which is estimated from distance and load with the corresponding weights ω_1 and $\omega_2 = 1 - \omega_1$. $\bar{d}^T_{ij}, \bar{d}^D_{ij}, \bar{d}_i, \bar{p}_i$ are all normalized.

Worst destroy. It is logical to remove a customer with a high cost and insert it in a cheaper place to reduce the total cost. Based


8
5
3
2
9
6
1
4
10
7
Depot
Depot
Depot
Depot
Depot
Customers available for trucks and drones
Customers available for trucks only
Removed customers

(a) Initial solution

(b) General removal

(c) Simultaneous removal

(d) Consistent removal

Fig. 4. An illustration of the complex removal procedures.

on this idea, we repeatedly select a customer with the highest cost, where the cost of customer j between i and h is calculated as $\text{cost}(i, j, h) = c^M(d^M_{ih} - d^M_{ij} - d^M_{jh})$.

Random destroy. It removes a randomly chosen customer and the resulting infeasible parts.

4.3. Repair methods

The repair procedure repairs a destroyed solution to a new feasible one by inserting the removed sub-routes in R_s and customers in R_n into better positions using various problem-oriented and efficient methods.

To avoid losing high-quality components and accelerate the repair process, we first assign the removed sub-routes in R_s to a free drone (if possible). Otherwise, we add the drone nodes in the sub-routes to R_n. Second, each removed customer $j \in R_n$ is inserted into the partial solution with two alternatives: to drone sub-routes ($M(j) = D$) or to truck routes ($M(j) = T$). Detailed implementations of both alternatives are sketched in online Appendix B.2. The two options are differentiated in function and complexity. Specifically, the former directly reduces costs but is challenging to implement because of complex resource constraints. The latter is relatively easy to realize and indirectly offers potential cost savings by providing more potential rendezvous nodes. To reach a trade-off between efficiency and effectiveness, we elaborate three repair operators.

Equal-priority insertion. Each request is inserted in the least costly position with a cost of $\Delta f_i = \min(\Delta f^T_i, \Delta f^D_i)$, where Δf^T_i and Δf^D_i represent the minimum cost increment of customer i to be inserted as a feasible truck/drone node, respectively.

Drone-first truck-second insertion. To reduce costs rapidly, this method preferentially inserts the removed request into drone sub-routes at the cheapest position. If there is no feasible drone node insertion ($\Delta f^D_i = \infty$), the least costly and feasible truck node insertion is implemented.

Truck-first drone-second insertion. It first inserts all removed nodes into truck routes and then assigns some truck nodes to drone nodes with resource constraints and synchronization requirements being respected.

Because routing decisions are highly interdependent and extremely challenging, verifying feasibility in loading and timing aspects deserves attention. For the loading aspect, the feasibility can be determined by comparing the additional load incurred by the inserted request with the minimum remaining vehicle capacity. Specifically, inserting request i at the position p in the truck route $r^T = \{n_0 = 0, n_1, n_2, .., n_{|r^T|} = n + 1\}$ proves feasible in load if and only if both $d_i \le Q^T - \max_{l<p}(F_{n_l k} + G_{n_l k})$ and $p_i \le Q^T - \max_{p \le l \le |r^T|}(F_{n_l k} + G_{n_l k})$ are satisfied. The same applies to verifying the load feasibility of a drone node insertion. For the timing aspect, constraints (14) to (17) in Section 3 are formulated as an LP model to minimize the maximum makespan for all deployed vehicles, which is easily tackled with the Gurobi solver. Then, we take the given solution as an input, and the feasibility can be verified according to the output's objective value. If it is less than the maximum working time, it proves feasible; otherwise, infeasible.

4.4. Adaptive selection mechanism

Each operator i corresponds to weight w_{ij} at iteration j, which is updated based on its past performance. Each iteration selects one destroy operator and one repair operator individually with a probability $p_i = w_{ij}/\sum^n_i w_{ij}$. All weights are initially equal in value (i.e., 1) and updated as $w_{i,j+1} = (1 - \eta)w_{ij} + \eta\sigma$, where the reaction index $(0 < \eta < 1)$ determines how quickly the weights respond to the performance of a new solution s_t and σ measures the performance of s_t with four levels ($\sigma_1 > \sigma_2 > \sigma_3 > \sigma_4$), as detailed in online Appendix B.3. High scores σ_1 and σ_2 are rewarded to well-performing operators for pushing the search process forward. Lower scores σ_3 and σ_4 are given to poor-performing operators for bringing some diversity.

4.5. Simulated annealing-based acceptance criterion

Once a new solution is obtained, its performance is evaluated and a decision is made to accept it or not. We borrow the idea from simulated annealing of accepting a worse solution with a probability of $\exp(f(s_c) - f(s_t))/T$, which gradually decreases as the temperature T drops at the rate of $c(0 < c < 1)$.

5. Computational experiments

In this section, we report on a series of computational experiments conducted to verify the performance of the proposed model and algorithm. We first introduce the generation of test instances and parameter settings. Then, we validate the proposed ALNS metaheuristic on a set of instances with different sizes. Finally, we present some managerial insights and sensitivity analyses.

5.1. Generation of instances

Because the VRPD was recently introduced and widely accepted benchmark data that fit our problem are not available, we perform computational experiments for a variety of instances randomly generated with various problem sizes. Moreover, to simulate real-life road networks in rural areas, we refer to the method of creating sparse graphs outlined in Letchford, Nasiri, and Theis (2013). A diagram of the sparse road network is shown in online Appendix C.1, and the distance calculation is given in online Appendix C.2.

We generate a total of 75 MFS-VRPD instances partitioned into small (n=10, 25), medium (n=50, 75), and large sizes (n=100). For each size, 15 instances are generated with customers located in areas of 30×30, 40×40, and 50×50 (five instances per case). Experiments were conducted with $|D| = 2$ for all instances and $|K|$=2, 3, and 4 for small-, medium- and large-size instances, respectively. We assume all customers are eligible for trucks, while 90% are eligible for drones (Forbes, 2019). Both delivery and pickup demands (d_i and p_i) are randomly generated within the interval [0, 20] for each drone-eligible customer and [20,100] for those that can only be served by trucks. The truck fleet is assumed homogeneous with a capacity of $Q^T = 500kg$ and travels at a constant average speed of $v^T = 35$km/h. For drones, the maximum payload is assumed to be $Q^D = 30kg$ (Dayarian, Savelsbergh, & Clarke, 2020). According to the practice (Ehang, 2023; DHL, 2018; SF Technology, 2018), the average velocity of drones is between 54 km/h and 80 km/h; thus, we take a compromise value of $v^D = 70$km/h. The endurance time of the drone is 60 min. The standard working time takes the value of 8 h. The unit cost per kilometer for trucks is set to be 1, and the unit cost for drones is set to be 0.2. Further sensitivity analyses on several significant parameters are carried out in Section 5.6.

5.2. Parameter setting

All experiments were implemented on a computer with an Intel(R) Core (TM) i7–4790 CPU @ 3.6 GHz, 16.0 GB of RAM, and a 64-bit Windows 10 operating system. Gurobi optimizer 9.1.2 is utilized and the ALNS metaheuristic algorithm was coded in Python 3.8.

The parameters for the ALNS settings are determined in parameter tuning experiments by pursuing a trade-off between solution quality and solving time. We set $niters = 2000$ for small-size instances and $niters = 4000$ for large-size ones. The start temperature T_s is determined such that the probability of accepting a solution that is w% worse than the initial solution equals 0.5, where $w = 30$ and the cooling rate $c = 0.97$. The number of removal nodes β is a random integer between $\max(3, [r_L|C|])$ and $\min(30, [r_U|C|])$, where $r_L = 0.15$ and $r_U = 0.5$ for small-size instances and $r_U = 0.3$ for large-size ones. The parameters for the adaptive selection mechanism are set as follows: $\sigma_1 = 33, \sigma_2 = 13, \sigma_3 = 9, \sigma_4 = 1,$ andη= 0.6. The maximum number of consecutive iterations without improvement $maxNonImprove$ is 300.

5.3. Validation of performance of the ALNS metaheuristic

This section first uses a collection of small-size instances to verify the effectiveness of the proposed ALNS. The computational results for this experiment are reported in Table 1, which provides a comprehensive comparison of the results obtained respectively by our ALNS metaheuristic and the Gurobi solver in terms of solution quality and computational time. First, the MFS-VRPD is solved by the Gurobi optimizer in a run time limit of 7200 s. We report the lower bound LB, the obtained best objective value Z^G, and the running time t^G. Second, without losing generality, we run the ALNS 10 times for each instance and record the average objective $\overline{Z^A}$ and the average running time $\overline{t^A}$. Three indexes for scaling the performance of the ALNS are denoted as $Gap_G\% = 100 \times (Z^G - LB)/Z^G$, $\overline{Gap_A}\% = 100 \times (\overline{Z^A} - LB)/Z^G$, and $Imp\% = 100 \times (\overline{Z^A}/Z^G - 1)$.

For the instances with ten customers, Gurobi can only find 13/15 optimal solutions, while the ALNS offers equally good results. Moreover, our ALNS metaheuristic expends much less time (15.1 s on average) than Gurobi (1489s on average). When the number of customers increases to 25, all 15 instances cannot be solved to optimality within 7200 s, as the column Gap_G shows. It can be observed that our ALNS metaheuristic provides better solutions than Gurobi in almost all instances (except instance 8), with an average improvement (Imp) of 10.2%. Moreover, the ALNS has a significant advantage in computational time, averaging 46.5 s. To sum up, our ALNS metaheuristic is effective in terms of solution quality and computational time.

For medium- and large-scale instances, the experiments are conducted on 45 cases with 50, 75, and 100 customers. As the problem size increases, finding solutions to the MFS-VRPD becomes much more challenging due to its NP-hard nature. The average computational time of the instances with different numbers of customers is 255, 540 and 1056s, respectively, which are within the acceptable range. The best and the average objectives in ten repetitions, noted as Z^A and $\overline{Z^A}$, are reported in Table 2. The gap between the best and the average objectives, denoted as $Gap\% = 100 \times (\overline{Z^A} - Z^A)/\overline{Z^A}$, is between 0.7% and 4.3%. For the instances with different numbers of customers, the average gap is 2.2%, 2.8% and 2.9%, respectively. It implies that our ALNS metaheuristic is reliable for medium- and large-scale instances.

We also conduct extensive experiments to assess the performance of our ALNS further. First, we compare the results of the MILP formulation and the ALNS under different operational settings with varying instance parameters and delivery and pickup scenarios. Second, we test our algorithm with an adaptive multi-start simulated annealing algorithm (AMS-SA) proposed by Masmoudi et al. (2022) on a subproblem of the MFS-VRPD, which only incorporates flexible docking and multi-visit service into the VRPD. The findings show that our algorithm exerts excellent performance in solution quality and computation time under all contemplated conditions. See online Appendices E and F for details.

Table 1
Results of the ALNS algorithm and Gurobi on small-size instances ($K=2$, $D=2$).

Instances		LB	Z^G	$\overline{Z^A}$	Gap_G	$\overline{Gap_A}$	Imp	t^G	$\overline{t^A}$
$n=10$	1	53.6	53.6	53.6	0.0	0.0	0.0	59	13.8
	2	35.2	35.2	35.2	0.0	0.0	0.0	2680	16.3
	3	55.9	55.9	55.9	0.0	0.0	0.0	117	14.1
	4	41.3	41.3	41.3	0.0	0.0	0.0	865	15.7
	5	49.5	49.5	49.5	0.0	0.0	0.0	87	14.8
	6	61.9	61.9	61.9	0.0	0.0	0.0	182	15.2
	7	66.4	66.4	66.4	0.0	0.0	0.0	445	14.5
	8	32.5	46.8	46.8	30.6	30.6	0.0	7200	14.9
	9	41.5	41.5	41.5	0.0	0.0	0.0	178	15.5
	10	82.1	82.1	82.1	0.0	0.0	0.0	1095	15.0
	11	63.1	63.1	63.1	0.0	0.0	0.0	108	17.4
	12	94.1	94.1	94.1	0.0	0.0	0.0	133	15.6
	13	102.2	102.2	102.2	0.0	0.0	0.0	814	14.7
	14	73.5	90.3	90.3	18.8	18.8	0.0	7200	15.1
	15	110.3	110.3	110.3	0.0	0.0	0.0	1177	14.1
Avg.		–	**66.3**	**66.3**	**3.3**	**3.3**	**0.0**	**1489**	**15.1**
$n=25$	1	79.9	110.2	95.7	27.5	14.3	−13.2	7200	55.7
	2	66.2[a]	111.4	93.2	40.6	24.2	−16.3	7200	36.0
	3	55.5[a]	107.9	80.7	48.6	23.4	−25.2	7200	40.8
	4	38.6[a]	116.2	100.1	66.8	52.9	−13.8	7200	40.9
	5	67.0	79.4	78.4	15.6	14.3	−1.3	7200	47.1
	6	50.6[a]	137.4	130.3	63.2	58.0	−5.2	7200	50.5
	7	94.5	149.7	125.5	36.9	20.7	−16.2	7200	53.1
	8	82.0[a]	136.7	137.1	40.0	40.3	0.3	7200	59.7
	9	100.8	173.7	158.6	42.0	33.3	−8.7	7200	51.3
	10	54.5[a]	103.8	97.5	47.5	41.5	−6.0	7200	46.3
	11	86.6[a]	185.3	161.5	53.3	40.4	−12.8	7200	53.3
	12	64.8[a]	166.8	157.6	61.1	55.6	−5.5	7200	44.5
	13	41.0	114.6	106.0	64.2	56.7	−7.5	7200	39.5
	14	102.3	159.8	138.4	36.0	22.6	−13.4	7200	35.0
	15	93.0	164.1	149.5	43.0	34.4	−8.9	7200	43.9
Avg.		–	**134.5**	**120.7**	**45.7**	**35.5**	**−10.2**	**7200**	**46.5**

Note: superscript [a] denotes the lower bound the Lagrangian relaxation provides (see online Appendix D for details).

Table 2
The results on medium- and large-scale instances.

Instances	$n=50$				$n=75$				$n=100$			
	Z^A	$\overline{Z^A}$	Gap	$\overline{t^A}$	Z^A	$\overline{Z^A}$	Gap	$\overline{t^A}$	Z^A	$\overline{Z^A}$	Gap	$\overline{t^A}$
1	128.5	132.2	2.8	239	161.3	165.1	2.3	532	203.5	209.2	2.7	948
2	137.8	140.5	1.9	247	152.0	156.9	3.2	479	198.0	201.4	1.7	906
3	125.8	127.2	1.1	276	160.1	164.9	2.9	513	188.0	194.0	3.1	977
4	136.6	138.2	1.2	243	170.4	176.6	3.5	565	191.3	198.9	3.8	942
5	147.3	149.2	1.3	256	172.8	177.8	2.8	536	203.4	210.8	3.5	928
6	191.3	196.6	2.7	246	243.0	244.7	0.7	529	266.4	275.3	3.2	1061
7	177.4	181.8	2.4	249	195.5	200.1	2.3	551	253.3	258.0	1.8	1090
8	194.4	199.4	2.5	257	200.4	204.7	2.1	501	245.3	251.3	2.4	1038
9	190.5	197.2	3.4	271	214.0	223.4	4.2	558	253.8	259.3	2.2	1061
10	188.6	191.8	1.6	242	200.3	204.9	2.2	503	262.6	270.6	2.9	1179
11	244.1	248.8	1.9	264	278.6	290.9	4.3	619	310.0	321.8	3.7	1125
12	241.1	247.7	2.7	250	268.1	275.5	2.7	576	320.1	330.2	3.1	1143
13	252.0	257.4	2.1	271	254.5	261.6	2.7	562	335.2	344.5	2.7	1127
14	217.7	226.0	3.7	275	239.9	247.8	3.2	518	352.3	365.1	3.5	1169
15	229.9	234.0	1.8	245	261.9	268.3	2.4	559	339.7	351.6	3.4	1143
AVG.	**186.9**	**191.2**	**2.2**	**255**	**211.5**	**217.5**	**2.8**	**540**	**261.5**	**269.5**	**2.9**	**1056**

5.4. Performance of the proposed truck–drone system

In this subsection, we observe the behavior of our truck–drone system for a thoroughly understand. First, we measure the improvement of the MFS-VRPD compared to the truck-only case, calculated as $\Delta\% = 100 \times (Z_{|D|=0} - \overline{Z^A})/Z_{|D|=0}$, where $Z_{|D|=0}$ indicates the best-obtained solution for the truck-only case. Second, the featured information is noted: the number of truck customers (served by truck) n_t, the number of drone customers (served by drone) n_d, the number of sub-routes n_s, the cost per drone customer uc_d, and the cost per truck customer uc_t. The summary results are listed in Table 3, while the detailed outputs and an illustration of the MFS-VRPD solution are given in online Appendix G.

Table 3
The summary featured information on the solutions to MFS-VRPD instances.

n	$\overline{Z^A}$	n_t	n_d	n_s	uc_t	uc_d	$Z_{\lvert D\rvert=0}$	Δ
10	66.3	2.8	7.2	3.3	15.7	3.9	138.8	52.3
25	120.7	11.0	14.0	8.1	6.9	3.3	182.9	34.1
50	191.2	24.6	25.4	15.6	5.1	2.6	268.2	28.7
75	217.7	33.8	41.2	18.9	4.2	1.84	306.4	28.7
100	269.5	49.0	51.0	23.8	3.7	1.67	364.7	26.2

From Table 3, we can draw the following observations: (i) As column Δ shows, the proposed truck–drone system achieves substantial cost savings compared to the truck–only case, varying from 52.3% to 26.2% (i.e., 34.0% on average) for the instances of differ-

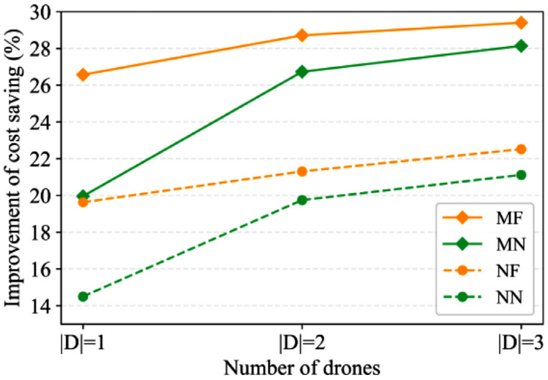


Fig. 5. The average improvement in cost saving when $|D|$ varies under different operational settings.

ent scales. It is noted that the cost savings decrease as the number of customers increases. The reason is that a small number of drones with finite spatial and temporal resources exert limited efficiency. More potential cost savings can be achieved by applying more drones (see SubSection 5.5.1). (ii) The cost per truck customer uc_t is significantly higher than the cost per drone customer uc_d, demonstrating the economic benefits of using drones in delivery and pickup. Moreover, uc_t and uc_d decrease as the number of customers increases, in line with the scale economies effect. (iii) Trucks, as mobile depots, effectively extend the service range of drones, which realize multiple sub-routes and serve more than half of customers, as column n_s and n_d shows. In summary, the proposed collaborative truck–drone system manifests substantial economic benefits by fully exerting the complementary advantages of trucks and drones.

5.5. Managerial insights for the MFS-VRPD

5.5.1. Impact of flexible docking and multi-visit services

This subsection systematically compares the results when the number of drones varies under four operational settings (Cases MF, MN, NF, and NN), corresponding to either, both, or neither of multi-visit services (M) and flexible docking (F). The experiment uses 15 instances with 50 customers considering different numbers of drones ($|D|$=1, 2, and 3) and four operational settings, yielding 15 × 3 × 4 = 180 experiments in total. The detailed outputs are reported in Tables H1 to H3 in online Appendix H.

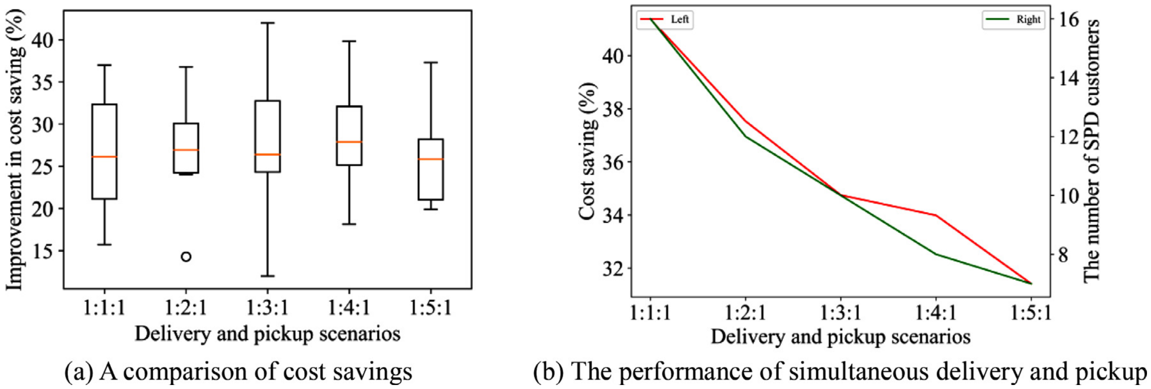
Fig. 5 provides the average improvement in cost saving compared with the truck-only case, as $|D|$ varies under different operational settings. We can make the following observations: (i) Both multi-visit services and flexible docking significantly improve solution quality. Case MF offers the best solution, while Case NN provides the worst. Specifically, in the case of $|D| = 2$, flexible docking brings an improvement with means of 2.0% and 1.3% when comparing Case MF with Case MN and comparing Case NF with Case NN, respectively. Multi-visit services achieve a more remarkable average improvement, and the average improvement of Case MF over Case NF and Case MN over Case NN is 7.4% and 6.7%, respectively. It is worth noting that this result is consistent regardless of the number of drones. (ii) It shows diminishing marginal improvement as drones increase. In other words, adding a drone for the first time brings a noticeable improvement. However, further increases in the number of drones would bring only relatively minor improvements. A similar trend is observed in the four operational scenarios. (iii) The marginal benefit of an additional drone when flexible docking is not allowed (i.e., Cases MN and NN) outweighs that when flexible docking is permitted (i.e., Cases MF and NF). The reason is that flexible docking improves overall deployment and scheduling, allowing drones to carry out more flights. (iv) Using fewer drones but allowing multi-visit services (i.e., Cases MF and MN) yields more substantial improvements than using more drones but not allowing multi-visit services (i.e., Cases NF and NN). This may be related to the improved utilization of drones' capacity and endurance. In conclusion, the collaborative truck–drone system with fewer drones allowing flexible docking and multi-visit service achieves more remarkable improvements than the general truck–drone system that prohibits multi-visit services and flexible docking (even with more drones). It demonstrates significantly enhanced resource utilization and higher efficiency.

More detailed analyses on the impact of flexible docking and multi-visit services on medium-size instances could refer to online Appendix I.1. A similar study on the impact of different operational settings on small-size instances can refer to online Appendix I.2.

5.5.2. Impact of different delivery and pickup scenarios

How different delivery and pickup scenarios affect the results interests us. To this end, an additional experiment is conducted on 50 instances with 50 customers. Given that customer demand for parcel delivery is usually higher, the proportion of customers with pickup, delivery, and both delivery and pickup demands is set to vary from 1:1:1, 1:2:1, 1:3:1, 1:4:1, and 1:5:1.

Fig. 6(a) shows that the average improvement in cost savings of the truck-drones system stabilized around 26.5% compared to the truck-only case, regardless of delivery and pickup scenarios. Furthermore, Fig. 6(b) indicates the performance of simultaneous pickup and delivery (SPD) compared to the scenarios (P/D) in which each vehicle can only fulfill pickup or delivery demands. It can be seen that as the proportion of customers with both delivery and pickup demands increases, so does the advantage of SPD in terms of cost savings. SPD achieves substantial cost savings



(a) A comparison of cost savings

(b) The performance of simultaneous delivery and pickup

Fig. 6. The impact of different delivery and pickup scenarios.

from 31% to 41% on average because of the reduction in repeated visits to customers with both delivery and pickup demands. Consequently, the truck-drone system benefits from the simultaneous delivery and pickup operation regardless of the delivery and pickup scenarios.

5.6. Sensitivity analysis

As technology improves, drones will likely become able to carry more or heavier packages. Therefore, it is necessary to test the impact of drones' capacity on the computational results. Moreover, drone endurance is a crucial element of the collaborative truck–drone system that determines drones' service scope. We therefore ask: what will happen as the battery techniques improve? In addition, we investigate the truck–drone system from the perspective of cost minimization, which is inherently susceptible to the cost ratio of drones to trucks. How does the cost ratio affect the results? Due to space limitations, please refer to online Appendix J for detailed sensitivity analyses of these parameters.

6. Conclusion

This study has formally described a rich VRPD that uses a fleet of trucks collaborated with multiple drones to accomplish pickup and delivery services in rural areas. The process requires complex temporal, spatial, and loading synchronization between vehicles, resulting in a complex and challenging problem. We formulate this problem as a MILP model and propose an efficient ALNS algorithm with a collection of elaborately tailored destroy and repair operators to provide high-quality solutions for realistic large-scale problems. Computational experiments offer further insights. Our main conclusions are as follows.

- The proposed collaborative truck–drone operation significantly reduces the total transportation cost compared to the truck-only case.
- Both multi-visit services and flexible docking can substantially improve the economic performance of the collaborative truck–drone system, but in different ways. The former is achieved by the enhanced utilization of drones' payload and endurance resources per flight, while the latter is realized through the overall deployment and scheduling of drones.
- The collaborative truck–drone system with fewer drones under the relaxed operational settings achieves more remarkable improvements than the general truck–drone system that prohibits multi-visit services and flexible docking (even with more drones). It confirms the effectiveness and necessity of incorporating multi-visit services and flexible docking.
- Simultaneous pickup and delivery also benefits the proposed truck–drone system by reducing repeated visits to customers with both delivery and pickup demands.
- An increase in drones' loading capacity and endurance can improve the cost-effectiveness of the truck–drone system, but the results no longer improve once a certain threshold is reached.

Some extensions of this study might be investigated in the future. First, some assumptions might be further relaxed. For example, the locations where launching and retrieving operations occur can be discrete points on arcs. Second, it would be interesting to study the problem in more realistic settings, such as considering the nonlinear energy consumption of drones or the time window associated with each customer. Third, the formulation can be further analyzed based on other objectives, such as considering cost-, time-, and environment-related indicators. Finally, developing a more efficient algorithm to solve large-scale problems is still highly desirable and challenging.

Declaration of Competing Interest

The authors declare there is no conflict of interests.

Acknowledgments

We thank the Editor and two anonymous referees for their constructive comments that have greatly helped us to improve the paper. This work was supported by the National Key R&D Program of China [grant number 2018YFB1601401]; Natural Science Foundation of Sichuan, China (grant number 2023NSFSC0517); and the National Natural Science Foundation of China [grant number 72071164].

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2023.06.021. The instances are available at https://1drv.ms/f/s!AksCmHT7AVtNtnBeiy968OG12h8L.

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science, 52*(4), 965–981.
Amorosi, L., Puerto, J., & Valverde, C. (2021). Coordinating drones with mothership vehicles: The mothership and drone routing problem with graphs. *Computers & Operations Research, 136*, Article 105445.
Bakir, I., & Tiniç, G. Ö. (2020). *Optimizing drone-assisted last-mile deliveries: The vehicle routing problem with flexible drones* Published: 2020/04/09 https://optimization-online.org/wp-content/uploads/2020/04/7737.pdf .
Boccia, M., Masone, A., Sforza, A., & Sterle, C. (2021). A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transportation Research Part C: Emerging Technologies, 124*, Article 102913.
Bouman, P., Agatz, N., & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks, 72*(4), 528–542.
Carlsson, J. G., & Song, S. (2018). Coordinated logistics with a truck and a drone. *Management Science, 64*(9), 4052–4069.
Chen, C., Demir, E., & Huang, Y. (2021). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. *European Journal of Operational Research, 294*(3), 1164–1180.
Cheng, C., Adulyasak, Y., & Rousseau, L. M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological, 139*, 364–387.
Coindreau, M. A., Gallay, O., & Zufferey, N. (2021). Parcel delivery cost minimization with time window constraints using trucks and drones. *Networks, 78*(4), 400–420.
Dayarian, I., Savelsbergh, M., & Clarke, J. P. (2020). Same-day delivery with drone resupply. *Transportation Science, 54*(1), 229–249.
de Freitas, J. C., & Penna, P. H. V. (2019). A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research, 27*(1), 267–290.
Dell'Amico, M., Montemanni, R., & Novellani, S. (2021). Algorithms based on branch and bound for the flying sidekick traveling salesman problem. *Omega, 104*, Article 102493.
DHL. (2018). *DHL parcelcopter* https://www.dpdhl.com/content/dam/dpdhl/en/media-center/media-relations/infographics/2018/dhl-parcelcopter-infographic-2018.jpg (Accessed 20 March 2023).
Drexl, M. (2012). Synchronization in Vehicle Routing – A survey of VRPs with multiple synchronization constraints. *Transportation Science, 46*(3), 297–316.
Ehang. (2023). *Logictics* https://www.ehang.com/logistics (Accessed 20 March 2023).
Euchi, J., & Sadok, A. (2021). Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication, 44*, Article 101236.
Es Yurek, E., & Ozmutlu, H. C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies, 91*, 249–262.
Forbes. (2019). *Amazon's new delivery drone will start shipping packages 'in amatter of months'* https://www.forbes.com/sites/jilliandonfro/2019/06/05/amazon-new-delivery-drone-remars-warehouse-robots-alexa-prediction/#7c0c6f7d145f (Accessed 5 June 2019).
Ganesh, S., & Menendez, J. R. (2016). *Methods, systems and devices for delivery drone security*. Washington, DC: U.S. Patent and Trademark Office *U.S. Patent No. 9359074*.
Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., & Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C: Emerging Technologies, 114*, 657–680.

Gu, R., Poon, M., Luo, Z., Liu, Y., & Liu, Z. (2022). A hierarchical solution evaluation method and a hybrid algorithm for the vehicle routing problem with drones and multiple visits. *Transportation Research Part C: Emerging Technologies, 141*, Article 103733.

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies, 86*, 597–621.

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *Journal of Heuristics, 26*(2), 219–247.

Jeong, H. Y., Song, B. D., & Lee, S. (2019). Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. *International Journal of Production Economics, 214*, 220–233.

Karak, A., & Abdelghany, K. (2019). The hybrid vehicle-drone routing problem for pickup and delivery services. *Transportation Research Part C: Emerging Technologies, 102*, 427–449.

Kitjacharoenchai, P., Min, B.-C., & Lee, S. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics, 225*, Article 107598.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M. A., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering, 129*, 14–30.

Letchford, A. N., Nasiri, S. D., & Theis, D. O. (2013). Compact formulations of the Steiner traveling salesman problem and related problems. *European Journal of Operational Research, 228*(1), 83–92.

Liu, Y., Liu, Z., Shi, J., Wu, G., & Pedrycz, W. (2021). Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51*(12), 7450–7465.

Luo, Z., Poon, M., Zhang, Z., Liu, Z., & Lim, A. (2021). The multi-visit traveling salesman problem with multi-drones. *Transportation Research Part C: Emerging Technologies, 128*, Article 103172.

Masmoudi, M. A., Mancini, S., Baldacci, R., & Kuo, Y. H. (2022). Vehicle routing problems with drones equipped with multi-package payload compartments. *Transportation Research Part E: Logistics and Transportation Review, 164*, Article 102757.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pickup points. *Transportation Research Part A: General, 23*(5), 377–386.

MOFCOM. (2021). *E-commerce in China 2020* http://images.mofcom.gov.cn/dzsws/202110/20211022182630164.pdf (Accessed 15 September 2021).

Moshref-Javadi, M., Hemmati, A., & Winkenbach, M. (2020). A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Applied Mathematical Modelling, 80*, 290–318.

Moshref-Javadi, M., Lee, S., & Winkenbach, M. (2020). Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E: Logistics and Transportation Review, 136*, Article 101887.

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies, 54*, 86–109.

Murray, C. C., & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies, 110*, 368–398.

Poikonen, S., & Golden, B. (2020a). The mothership and drone routing problem. *INFORMS Journal on Computing, 32*(2), 249–262.

Poikonen, S., & Golden, B. (2020b). Multi-visit drone routing problem. *Computers & Operations Research, 113*, Article 104802.

Poikonen, S., Golden, B., & Wasil, E. A. (2019). A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing, 31*(2), 335–346.

Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks, 70*(1), 34–43.

Pugliese, L. D. P., & Guerriero, F. (2017). Last-mile deliveries by using drones and classical vehicles. In A. Antonio Sforza, & C. Sterle (Eds.), *Optimization and decision science: Methodologies and applications* (pp. 557–565). Springer International Publishing AG 2017.

Raj, R., & Murray, C. (2020). The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies, 120*, Article 102813.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455–472.

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies, 102*, 289–315.

Schermer, D., Moeini, M., & Wendt, O. (2019a). A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research, 109*, 134–158.

Schermer, D., Moeini, M., & Wendt, O. (2019b). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies, 106*, 166–204.

SF Technology. (2018). *Ark-octocoper-drone* https://www.sf-tech.com.cn/en/product/ark-octocoper-drone (Accessed 20 March 2023).

Tamke, F., & Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones. *Transportation Research Part B: Methodological, 144*, 174–203.

Tu, P. A., Dat, N. T., & Dung, P. Q. (2018). Traveling salesman problem with multiple drones. In *SoICT 2018: Proceedings of the ninth international symposium on information and communication technology* (pp. 46–53). December 6-7, 2018.

Wang, K., Biao, Y., Mengting, Z., & Lu, Y. (2020). Cooperative route planning for the drone and truck in delivery services: A bi-objective optimization approach. *Journal of the Operational Research Society, 71*(10), 1657–1674.

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: Several worst-case results. *Optimization Letters, 11*(4), 679–697.

Wang, Z., & Sheu, J.-B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological, 122*, 350–364.

Yariv, B. A. S. H., & Regev, A. (2022). *Dynamically arming a safety mechanism on a delivery drone*. Washington, DC: U.S. Patent and Trademark Office *U.S. Patent No. 11226619*.

Zhou, Y., Zhu, Z., & Wu, Z. (2021). *System and method for real-time location tracking of a drone*. Washington, DC: U.S. Patent and Trademark Office *U.S. Patent No. 11120560*.