Received: 11 May 2021 | Revised: 4 February 2022 | Accepted: 4 February 2022 | Published on: 26 February 2022

DOI: 10.1002/nav.22053

RESEARCH ARTICLE

WILEY

Truck and drone routing problem with synchronization on arcs

Hongqi Li[1] | Jun Chen[1] | Feilong Wang[1] | Yibin Zhao[2]

[1]School of Transportation Science and Engineering, Beihang University, Beijing, China
[2]Research Institute of Highway, Ministry of Transport, Beijing, China

Correspondence
Hongqi Li, School of Transportation Science and Engineering, Beihang University, No. 37 Xueyuan Road, Haidian District, Beijing 100191, China.
Email: lihongqi@buaa.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 71672005, 71972007.

Abstract

Truck–drone technology that enables drones to launch from or land on a moving truck without the need for the truck to stop is the focus of this study. We define the truck and drone routing problem with synchronization on arcs (TDRP-SA). The TDRP-SA is characterized by synchronization on arcs, time windows, classified customers, direct delivery, multiple trucks, and multiple drones carried by each truck. The TDRP-SA involves synchronization on arcs to permit trucks to dispatch and retrieve drones at suitable moving-LRLs (drone launch/retrieval locations) on arcs of truck routes. A moving-LRL is essentially a drone launch/retrieval location with a moving truck, and the drone launch/retrieval operations do not need special parking for the trucks. We develop a mixed integer nonlinear programming model to address the TDRP-SA. We propose a mathematical analysis method to locate moving-LRLs and to estimate the drone's arrival time at the customers. We develop boundary models through introducing linear piecewise functions to locate moving-LRLs. We provide an adaptive large neighborhood search (ALNS) heuristic. Through computational experiments, the effectiveness of the boundary models, the TDRP-SA model and ALNS-based heuristic are evaluated.

KEYWORDS

adaptive large neighborhood search, mixed integer programming, routing, synchronization on arcs, truck–drone combination

1 | INTRODUCTION

Since drones are less sensitive to road congestion, drone technologies provide a compelling alternative to the traditional last-mile parcel delivery mode, especially in time-sensitive situations (Boysen et al., 2021). Many companies are developing drone systems for delivery. Amazon, Google, JD.com, DHL, and UPS have invested in drone delivery modes (Chung et al., 2020; Macrina et al., 2020; Murray & Chu, 2015). Some strategies have been proposed to deal with the limited flight range of drones, one of which is a combined approach that permits drones to launch from trucks for deliveries.

The operations research community has paid attention to the operational challenges in adopting drones combined with trucks for parcel deliveries. Keeping in view the optimization of the cooperative routes for one truck working in collaboration with one drone, several studies introduce variants of the traveling salesman problem (TSP), including the flying sidekick TSP (FSTSP), TSP with drone (TSPD), and carrier-vehicle TSP (e.g., Agatz et al., 2018; Gambella et al., 2018; Murray & Chu, 2015). Considering the use of a fleet of truck–drone combinations for deliveries, in which several drones collaborate with each truck, variants of the vehicle routing problem (VRP), for example, VRP with drones (VRPD), are introduced (e.g., Sacramento et al., 2019; Wang et al., 2017). Variants involving drones in collaboration with trucks usually include a type of node that is called drone launch/retrieval location (LRL). In the literature, LRLs are located at customers or specific parking nodes. Almost all the literature assumes customers to be candidate LRLs. The literature that involves specific parking nodes generally assumes that only drones are used for deliveries, and trucks are used for drone launch/retrieval at specific parking nodes. A type of binary variable for deciding LRLs is usually involved in the developed mathematical models.

The research and development on drones have been evolving, and there has been an increased interest in making the drone operations more autonomous (Alotaibi et al., 2018).

The assumption of drones moving directly from one location to another and avoiding encountered obstacles is expected to be realized in the near future. Especially, Planck[1] technology enables fully autonomous drones for ground vehicles. A drone equipped with Planck's AVEM system can safely launch from a moving ground vehicle, without the need for the ground vehicle to stop. Meanwhile, the drone can return to the ground vehicle and executes a precision landing on the bed of the ground vehicle.

We cover relevant research and choose to focus on models and algorithms that are developed in papers published by mainstream transportation or operations research journals. Past mathematical models are not able to capture the characteristics involved in the precision landing by Planck's AVEM system, that is, a drone launching from and landing on a moving truck without the need for the truck to stop. Such feature implies a mathematically and methodologically interesting problem, in which the LRLs, which may be located at any nodes on arcs of truck routes, must be decided. In the situation that the drone launch/retrieval operation happens while trucks are running on arcs, the LRL decisions should be indicated by continuous variables.

In this paper, we formally define the truck and drone routing problem with synchronization on arcs (TDRP-SA) that focuses on the delivery applicability of a fleet of truck–drone combinations. Along truck routes, the locations where trucks can dispatch or retrieve drones, may be any nodes (including customers) on arcs. At a location for a moving truck dispatching or retrieving drones, the arrival time of the truck, and the departure or arrival time of the drones affect each other, which is addressed in this study through the synchronization on arcs. The synchronization on arcs involves flexible and mobile platforms for drone launch/retrieval. An LRL that can be located at any node on arcs traveled by trucks, which is called a moving-LRL in this paper, consists of a truck that keeps running. At a moving-LRL, one or several drones launch from their truck with parcels to serve customers, and the drones need to return to the paired truck at suitable moving-LRLs. The drone launch/retrieval operations do not need the trucks to stop, that is, they do not need special parking for the trucks. The synchronization on arcs does not consider the synchronization of truck and drone speed. It is assumed that at a moving-LRL the truck and drone speeds are synchronized instantaneously. The synchronization on arcs discussed in this study includes the satellite synchronization, introduced in the two-echelon VRP literature. If all the moving-LRLs are located at customers, and drone launch/retrieval operates when trucks park to serve customers, the TDRP-SA involves only satellite synchronization. If there is a moving-LRL located at a node (not a customer) on arcs, the TDRP-SA involves synchronization on arcs.

While the body of literature around truck–drone cooperative routing problems is growing quickly, we choose to tackle a particularly complex variant with synchronization on arcs. The contributions include the following. (i) We define the TDRP-SA to cater to the developing technology of drone launching from and landing on a moving truck without the need for the truck to stop. Especially, the synchronization on arcs is integrated. (ii) We introduce a mixed integer nonlinear programming model to exactly describe the TDRP-SA. By introducing linear piecewise functions to locate moving-LRLs, we develop two boundary models. (iii) We propose a mathematical analysis method to select moving-LRLs and to estimate the arrival time of the drones at customers, and the mathematical analysis method is included in an adaptive large neighborhood search (ALNS)-based heuristic. We evaluate experimentally the effectiveness of the TDRP-SA model and the applicability of the heuristic for large-scale instances, and the gains in delivery performance offered by synchronization on arcs, multiple drones carried by each truck, drone direct delivery from the depot, are evaluated.

The paper is organized as follows. Section 2 reviews the studies on the truck–drone routing variants. Section 3 defines the TDRP-SA and develops a mathematical model. Section 4 develops two boundary models. Section 5 presents an ALNS-based heuristic. Section 6 describes the test instances, computational results, and corresponding analyses. We summarize several extensions in Section 7, and conclude with Section 8.

2 | LITERATURE REVIEW

The FSTSP or TSPD are general forms of the TSP variants in truck–drone delivery modes. Murray and Chu (2015), de Freitas and Penna (2018, 2020), Dell'Amico et al. (2021), and Jeong et al. (2019) provided solution methods to solve the FSTSP and variants. Agatz et al. (2018), Yurek and Ozmutlu (2018), and Ha et al. (2018, 2020) proposed solution methods to solve the TSPD and variants. Marinelli et al. (2017) introduced a TSPD variant, in which the truck should stop at a point of an arc to launch and retrieve the drone and start again. The optimal position of the parking place was decided by finding the intersections between the best drone coverage circle with a radius around the drone customers (DRCs); the best radius corresponded to minimum waiting time of the vehicles. A greedy randomized adaptive search procedure algorithm was provided. To extend the FSTSP and the TSPD, Luo et al. (2018), Gonzalez-R et al. (2020), and Liu et al. (2020) relaxed the constraint of one drone route covering exactly one customer, that is, one drone route can cover more than one customer. Murray and Raj (2020), Moshref-Javadi, Hemmati, and Winkenbach (2020); Moshref-Javadi, Lee, and Winkenbach (2020), Salama and Srinivas (2020), Dell'Amico et al. (2020), Boysen, Schwerdfeger, and Weidinger (2018);

[1] https://www.planckaero.com/avem

Boysen, Briskorn, et al. (2018), Poikonen and Golden (2020), and Karak and Abdelghany (2019) relaxed the constraint of the truck operating in coordination with exactly one drone, that is, the truck can operate in coordination with several drones. Kitjacharoenchai et al. (2019) introduced the multiple TSPD model that permits drone interchange between trucks.

Wang et al. (2017) defined the VRPD. Schermer et al. (2019a), Sacramento et al. (2019), Kitjacharoenchai et al. (2020), and Euchi and Sadok (2021) proposed solution methods for the VRPD. Di Puglia Pugliese and Guerriero (2017), Di Puglia Pugliese et al. (2020), and Das et al. (2020) introduced the VRPD with time windows. Schermer et al. (2019b) introduced the VRPD with en route operations. Drones could be launched and retrieved both at vertices and on some discrete points on arcs. The service time required to serve a customer and the time required to launch/retrieve drones were ignored, and a truck could wait at certain location for drone arrival. A variable neighborhood search and a tabu search was proposed. Li et al. (2020) studied a variant with time windows from the two-echelon network perspective. Wang and Sheu (2019) investigated a variant in which each drone could be interchanged between trucks at service hubs.

In Table S1, we present a comparison of VRPD variants in the literature with that introduced in this study. The key point of distinction of the TDRP-SA is the involvement of the synchronization on arcs and time windows. The variants introduced by Di Puglia Pugliese and Guerriero (2017), Di Puglia Pugliese et al. (2020), and Das et al. (2020) involve time windows, while exactly one customer is covered by each drone route. Li et al. (2020) studied a variant with time windows, while the location at which a truck dispatches drones should be the same as that at which the truck retrieves drones. To the best of our knowledge, the assumption of drone launch/retrieval en route has been addressed in Marinelli et al. (2017) and Schermer et al. (2019b), while the constraint of one drone route covering exactly one customer was considered, and time windows were ignored. The mathematical models in the literature cannot capture the feature of a drone launching from or landing on a moving truck, without the need for the truck to stop.

3 | DEFINITION AND MATHEMATICAL FORMULATION OF TDRP-SA

3.1 | Some definitions related to the TDRP-SA

The TDRP-SA is characterized by synchronization on arcs, time windows, classified customers, direct delivery, multiple trucks, and multiple drones carried by each truck. Figure S1 illustrates an example of the routes in a TDRP-SA instance.

3.1.1 | Network

On the TDRP-SA network, there is one depot and a number of customers. The depot, where a homogeneous fleet of truck–drone combinations is available, is the only source of parcels. The network is denoted by an undirected complete graph, where the nodes represent the depot and customers, and the arcs represent the possible paths of the trucks and drones (i.e., each arc can be traveled on by both trucks and drones). It is assumed that drones fly on straight lines, that is, drones fly straight toward customers or trucks. The drones can fly along the arcs or fly from any node (or location) on the arcs of truck routes to customers, or fly from customers to nodes (or locations) on the arcs of truck routes.

3.1.2 | Customers

Each customer with time window should be served exactly once. The customers are classified into two types: truck–drone customer (TDC) and DRC. Each TDC can be served by one truck, and can also be served by one drone if the TDC demand does not exceed the drone capacity. Each DRC can only be served by one drone.

3.1.3 | Direct delivery

All trucks and drones in use should depart from and eventually return to the depot. A direct delivery from the depot is permitted. A truck or drone that is used for the direct delivery should depart from the depot, serve the assigned customers, and finally return to the depot.

3.1.4 | Vehicles

Each truck–drone combination, which works in a "paired" modality, includes one truck carrying a predetermined number of drones. This number is assumed to be the same. The "paired" modality implies that a drone launching from a truck must land on the same truck. It is assumed that the drones carried by each truck do not occupy the truck's full capacity. The total number of parcels to be delivered to the target customers in a route traveled by one truck–drone combination must not exceed the truck's capacity. A drone can carry several parcels to serve more than one customer.

Referring to a general assumption in the literature (e.g., Coutinho et al., 2018), each drone has a fixed maximum flying time for each takeoff and a constant velocity. In this paper, a "takeoff" is an entire flight sequence, which involves a drone launching from a moving-LRL or the depot, serving the assigned customer(s), and returning to a moving-LRL or the depot. Some papers in the literature use the term "operation," rather than "takeoff." Of each truck, the capacity, average velocity, variable cost, and maximum working time are the same. Of each drone, the capacity, average velocity, operating cost, and maximum flying time per takeoff are the same.

3.1.5 | Synchronization on arcs

On its route, a truck can dispatch and retrieve the carried drones at suitable moving-LRLs on the arcs. If a moving-LRL for trucks dispatching drones is located at a node except the TDCs, the truck arrival time and the drone departure time are the same; at a moving-LRL for trucks retrieving the drones, the truck arrival time and the drone arrival time are the same. If the moving-LRL is exactly located at a TDC, the drones can be dispatched or retrieved when their truck is waiting for the TDC's time window opening or is serving the TDC.

3.1.6 | Routes

Routes are classified into three types. The first and second types are called the drone route and truck route, which originate and terminate at the depot, and are traveled on by one drone or one truck for direct delivery. The third type is called the combination route, which is traveled on by one truck–drone combination. A combination route consists of one main tour and one or more subtours. In the main tour, one truck departs from the depot, visits TDCs, and returns finally to the depot. Subtours are traveled on by drones that launch from moving-LRLs.

3.1.7 | Objective

The objective of the TDRP-SA is to minimize the integrated cost, which includes vehicle operating costs and the penalty cost of truck or drone waiting time. The penalty cost of truck or drone waiting at customers is transformed in terms of monetary unit.

3.2 | Mathematical formulation

The TDRP-SA is presented on a graph $\Phi = (V, A)$. The node set $V = V^{\mathrm{tdc}} \cup V^{\mathrm{drc}} = \{0, 1, 2, \dots, n^{\mathrm{tdc}}, n^{\mathrm{tdc}}+1, \dots, n\}$, where 0 denotes the depot. The set of all the customers is $V^c = V \backslash \{0\}$. The node subset $V^{\mathrm{tdc}} = \{0, 1, 2, \dots, n^{\mathrm{tdc}}\}$ is the depot and the TDC set. The node subset $V^{\mathrm{drc}} = \{n^{\mathrm{tdc}}+1, \dots, n\}$ is the DRC set. d_{ij} denotes the traveling distance from locations i to j. where i or j can be any node or moving-LRL. It is assumed that the traveling distances satisfy triangular inequalities. The weight of the parcels required by customer $i \in V^c$ is known. $[e_i, l_i]$ denotes the time window of customer. s_i is the service time needed by customer i. The penalty for a truck waiting for a unit of time at a customer is denoted as τ. Let K denote the set of available trucks. The set of drones paired with each truck is denoted as U. Let G denote the set of drones that are used for direct delivery. L^U denotes the maximum flying time of one takeoff for each drone. v^t and v^u denote the average velocity of each truck and that of each drone, respectively. The variable cost of each truck is c^t and the operating cost per takeoff of each drone is c^u.

To set up the objective function and key constraints, we define some variables, as shown in Table 1.

3.2.1 | Objective function

The objective function is described as (1), which includes the following parts: the variable cost of trucks, the operating cost of drones, the transformed cost of waiting-time of trucks and drones at customers. Each part is estimated by the monetary unit.

$$
\begin{aligned}
\min z = & \sum_{k\in K}\sum_{i\in V^{\mathrm{tdc}}}\sum_{j\in V^{\mathrm{tdc}}} c^t d_{ij} x_{kij} + \sum_{k\in K}\sum_{i\in V^{\mathrm{tdc}}} c^t t^W_{ki} v^t \tau \\
& + c^u \left(\sum_{k\in K}\sum_{u\in U}\sum_{a\in V^{\mathrm{tdc}}}\sum_{b\in V^{\mathrm{tdc}}} r_{kuab} + \sum_{g\in G}\sum_{i\in V^c} z_{g0i} \right) \\
& + \sum_{k\in K}\sum_{u\in U}\sum_{i\in V} c^u t^W_{kui}/L^U + \sum_{g\in G}\sum_{i\in V} c^u t^W_{gi}/L^U. \qquad (1)
\end{aligned}
$$

3.2.2 | Constraints on truck routes and drone routes

On the condition that customers suitable for direct delivery are identified, optimizing truck routes and drone routes for direct delivery means solving the VRP with time windows (VRPTW). Constraints on truck routes and drone routes include: (i) Constraints on vehicle capacity and route duration. (ii) Constraints on time continuity, which indicate the time relationship between the arrival times of a vehicle at successive nodes, and respect the customer time windows. (iii) Constraints on vehicle visiting nodes, which ensure that all the vehicles in use depart from and return to the depot, and ensure that each customer is served exactly once.

3.2.3 | Constraints on synchronization on arcs in combination routes

Constraints on combination routes are listed in two categories: constraints on synchronization on arcs and usual constraints. The usual constraints include those on vehicle capacity, route duration, time continuity, and vehicle visiting nodes. For simplicity, these constraints are omitted. In combination routes, moving-LRLs, which may be located at any location on the arcs of the main tours, are used as subtour origins and destinations. The continuous variables t^r_{kuab} and t^c_{kuab} are used to address the localization of moving-LRLs, which indicates that the moving-LRLs are located in a continuous space on the arcs. Constraints on the synchronization on arcs should identify the locations of moving-LRLs and ensure time continuity. Some constraints include the parameter M, where M is a sufficiently large positive number.

$$
\begin{aligned}
& r_{kuaj} + M \cdot \left(2 - \sum_{b\in V^{\mathrm{tdc}}}\sum_{i\in V^c} y_{kuabai} - x_{kaj} \right) \geq 1, \quad k \in K, \\
& u \in U, a \in V^{\mathrm{tdc}}, j \in V^{\mathrm{tdc}}. \qquad (2)
\end{aligned}
$$

$$
\begin{aligned}
& r_{kuaj} - M \cdot \left(2 - \sum_{b\in V^{\mathrm{tdc}}}\sum_{i\in V^c} y_{kuabai} - x_{kaj} \right) \\
& \leq 1, k \in K, u \in U, a \in V^{\mathrm{tdc}}, j \in V^{\mathrm{tdc}}. \qquad (3)
\end{aligned}
$$

$$
\begin{aligned}
& c_{kubj} + M \cdot \left(2 - \sum_{a\in V^{\mathrm{tdc}}}\sum_{i\in V^c} y_{kuabib} - x_{kbj} \right) \geq 1, \\
& k \in K, u \in U, b \in V^{\mathrm{tdc}}, j \in V^{\mathrm{tdc}}. \qquad (4)
\end{aligned}
$$

TABLE 1 Variables in the truck and drone routing problem with synchronization on arcs model

Notation	Meaning
x_{kij}	A binary variable, which is 1 only if truck $k \in K$ travels on arc (i, j); $(i, j \in V^{\text{tdc}})$.
t_k^D	A continuous variable that denotes the departure time of truck k at the depot; $t_k^D \geq 0$·
t_{ki}^A	A continuous variable that denotes the arrival time of truck k at node $i \in V^{\text{tdc}}$; $t_{ki}^A \geq 0$·
t_{ki}^W	A continuous variable that denotes the waiting-time of truck k for customer time window opening at customer $i \in V^{\text{tdc}} \backslash \{0\}$; $t_{ki}^W \geq 0$.
y_{kuabij}	A binary variable, which is 1 only if drone $u \in U$, carried by truck $k \in K$ launches after truck k visits node $a \in V^{\text{tdc}}$ (while a is the last customer that is served by truck k before the drone launches), travels on arc (i, j) $(i, j \in V^c)$ and lands on truck k after truck k visits node $b \in V^{\text{tdc}}$ (while b is the last customer served by truck k before the drone is retrieved).
t_{kuai}^A	A continuous variable that denotes the arrival time of drone $u \in U$, carried by truck $k \in K$ at customer $i \in V^c$, while $a \in V^{\text{tdc}}$ is the last customer that is served by truck k before the drone launches; $t_{kuai}^A \geq 0$·
t_{kui}^W	A continuous variable that denotes the waiting-time of drone $u \in U$, carried by truck $k \in K$ for customer time window opening at node $i \in V^c$; $t_{kui}^W \geq 0$.
z_{gij}	A binary variable, which is 1 only if drone $g \in G$ travels on arc (i, j), $(i, j \in V)$.
t_{gi}^W	A continuous variable that denotes the waiting-time of drone $g \in G$ for customer time window opening at customer $i \in V^c$; $t_{gi}^W \geq 0$.
r_{kuab}	A binary variable, which is 1 only if drone $u \in U$, carried by truck $k \in K$ launches at a moving-LRL on arc (a, b), $(a, b \in V^{\text{tdc}})$.
c_{kuab}	A binary variable, which is 1 only if drone $u \in U$, carried by truck $k \in K$ lands on the paired truck at a moving-LRL on arc (a, b), $(a, b \in V^{\text{tdc}})$.
t_{kuab}^r	A continuous variable that denotes the difference between the launch time of drone $u \in U$, carried by truck $k \in K$ at the moving-LRL on arc (a, b) $(a, b \in V^{\text{tdc}})$ and the leaving time of truck k at node $a \in V^{\text{tdc}}$. $-s_a < t_{kuab}^r \leq d_{ab}/v^t$, and a negative value of t_{kuab}^r denotes that drone u carried by truck k launches at node a.
t_{kuab}^c	A continuous variable that denotes the difference between the arrival time of drone $u \in U$, carried by truck $k \in K$ at the moving-LRL on arc (a, b) $(a, b \in V^{\text{tdc}})$ and the leaving time of truck k at node $a \in V^{\text{tdc}}$. $-s_a < t_{kuab}^c \leq d_{ab}/v^t$, and a negative value of t_{kuab}^c denotes that drone u carried by truck k lands on the paired truck at node a.
d_{kuiab}^r	A continuous variable that denotes the traveling distance between the moving-LRL, where drone $u \in U$, carried by truck $k \in K$ launches on arc (a, b) $(a, b \in V^{\text{tdc}})$ and customer $i \in V^c$ who is the first customer in the subtour traveled by drone u.
d_{kuiab}^c	A continuous variable that denotes the traveling distance between the moving-LRL, where drone $u \in U$ carried by truck $k \in K$ lands on the paired truck on arc (a, b) $(a, b \in V^{\text{tdc}})$ and customer $i \in V^c$ who is the last customer in the subtour traveled by drone u.

$$c_{kubj} - M \cdot \left(2 - \sum_{a \in V^{\text{tdc}}} \sum_{i \in V^c} y_{kuabib} - x_{kbj}\right) \leq 1,$$
$$k \in K, u \in U, b \in V^{\text{tdc}}, j \in V^{\text{tdc}}. \quad (5)$$

$$\sum_{b \in V^{\text{tdc}}} \sum_{i \in V^c} y_{kuabai} \leq \sum_{j \in V^{\text{tdc}}} x_{kaj}, \;\; k \in K, u \in U, a \in V^{\text{tdc}}. \quad (6)$$

$$\sum_{a \in V^{\text{tdc}}} \sum_{i \in V^c} y_{kuabib} \leq \sum_{j \in V^{\text{tdc}}} x_{kjb}, \;\; k \in K, u \in U, b \in V^{\text{tdc}}. \quad (7)$$

$$r_{kuaj} \leq x_{kaj}, \;\; k \in K, u \in U, a \in V^{\text{tdc}}, j \in V^{\text{tdc}} \quad (8)$$

$$c_{kubj} \leq x_{kbj}, \;\; k \in K, u \in U, b \in V^{\text{tdc}}, j \in V^{\text{tdc}}. \quad (9)$$

$$t_{kuaj}^A + t_{kuj}^W + M \cdot \left(1 - \sum_{i \in V} y_{kuabij}\right) \geq e_j,$$
$$k \in K, u \in U, a \in V^{\text{tdc}}, b \in V^{\text{tdc}}, j \in V^c, j \neq a, j \neq b. \quad (10)$$

$$t_{kuaj}^A + t_{kuj}^W - M \cdot \left(1 - \sum_{i \in V} y_{kuabij}\right) \leq l_j,$$
$$k \in K, u \in U, a \in V^{\text{tdc}}, b \in V^{\text{tdc}}, j \in V^c, j \neq a, j \neq b. \quad (11)$$

$$t_k^D + t_{ku0j}^r + d_{kui0j}^r / v^u + M \cdot \left(2 - y_{ku0b0i} - x_{k0j}\right) \geq t_{ku0i}^A$$
$$k \in K, u \in U, b \in V^{\text{tdc}}, i \in V^c, j \in V^{\text{tdc}} \backslash \{0\}, j \neq i. \quad (12)$$

$$t_k^D + t_{ku0j}^r + d_{kui0j}^r / v^u - M \cdot \left(2 - y_{ku0b0i} - x_{k0j}\right) \leq t_{ku0i}^A$$
$$k \in K, u \in U, b \in V^{tdc}, i \in V^c, j \in V^{tdc} \backslash \{0\}, j \neq i. \quad (13)$$

$$t_{ka}^A + t_{ka}^W + s_a + t_{kuaj}^r + d_{kuiaj}^r / v^u$$
$$+ M \cdot \left(2 - y_{kuabai} - x_{kaj}\right) \geq t_{kuai}^A$$
$$k \in K, u \in U, a \in V^{\text{tdc}} \backslash \{0\}, b \in V^{\text{tdc}}, i \in V^c,$$
$$i \neq a, j \in V^{\text{tdc}}, j \neq i, j \neq a. \quad (14)$$

$$t_{ka}^A + t_{ka}^W + s_a + t_{kuaj}^r + d_{kuiaj}^r / v^u$$
$$- M \cdot \left(2 - y_{kuabai} - x_{kaj}\right) \leq t_{kuai}^A$$
$$k \in K, u \in U, a \in V^{\text{tdc}} \backslash \{0\}, b \in V^{\text{tdc}}, i \in V^c, i \neq a,$$
$$j \in V^{\text{tdc}}, j \neq i, j \neq a. \quad (15)$$

$$t_{kuai}^A + t_{kui}^W + s_i + d_{kui0j}^c / v^u$$
$$+ M \cdot \left(2 - y_{kua0i0} - x_{k0j}\right) \geq t_k^D + t_{ku0j}^c$$
$$k \in K, u \in U, a \in V^{\text{tdc}}, i \in V^c,$$
$$j \in V^{\text{tdc}} \backslash \{0\}, j \neq i. \quad (16)$$

$$\begin{aligned} & t^A_{kuai} + t^W_{kui} + s_i + d^c_{kui0j}/v^u \\ & \quad - M \cdot \left(2 - y_{kua0i0} - x_{k0j}\right) \leq t^D_k + t^c_{ku0j} \\ & \quad k \in K, u \in U, a \in V^{\text{tdc}}, i \in V^c, \\ & \quad j \in V^{\text{tdc}} \backslash \{0\}, j \neq i. \end{aligned} \tag{17}$$

$$\begin{aligned} & t^A_{kuai} + t^W_{kui} + s_i + d^c_{kuibj}/v^u + M \cdot \left(2 - y_{kuabib} - x_{kbj}\right) \\ & \quad \geq t^A_{kb} + t^W_{kb} + s_b + t^c_{kubj} \\ & k \in K, u \in U, a \in V^{\text{tdc}}, i \in V^c, b \in V^{\text{tdc}} \backslash \{0\}, \\ & i \neq b, j \in V^{\text{tdc}}, j \neq b, j \neq i. \end{aligned} \tag{18}$$

$$\begin{aligned} & t^A_{kuai} + t^W_{kui} + s_i + d^c_{kuibj}/v^u - M \cdot \left(2 - y_{kuabib} - x_{kbj}\right) \\ & \quad \leq t^A_{kb} + t^W_{kb} + s_b + t^c_{kubj} \\ & k \in K, u \in U, a \in V^{\text{tdc}}, i \in V^c, b \in V^{\text{tdc}} \backslash \{0\}, i \neq b, \\ & j \in V^{\text{tdc}}, j \neq b, j \neq i. \end{aligned} \tag{19}$$

Constraints (2) and (3) define the binary variable r_{kuaj}, which is equal to 1 only if drone u carried by truck k launches at a moving-LRL on arc (a, j), while arc (a, j) is traveled by truck k. Constraints (4) and (5) define the binary variable c_{kubj}, which equals to 1 only if drone u carried by truck k lands on the paired truck at a moving-LRL on arc (b, j), while arc (b, j) is traveled by truck k. Constraint (6) indicates that there is a subtour origin on the arc that takes TDC a as one vertex only if TDC a is included in a main-tour. Constraint (7) indicates that there is a subtour destination on the arc that takes TDC b as one vertex, only if TDC b is included in a main-tour. Constraints (8) and (9) indicate that there is no moving-LRL on an arc, if the arc is not traveled by a truck. Constraints (10) and (11) guarantee that the drone visits customers in a subtour, while simultaneously respecting the customer time windows. Constraints (12) and (13) indicate the time relationship between the drone launch time at a moving-LRL on arc (0, j) and the drone arrival time at the first customer in a subtour. Constraints (14) and (15) indicate the time relationship between the drone launch time at a moving-LRL on arc (a, j) and the drone arrival time at the first customer in a subtour. Constraints (16) and (17) indicate the time relationship between the time of the drone landing on its paired truck at a moving-LRL on arc (0, j,) and the drone arrival time at the last customer in a subtour. Constraints (18) and (19) indicate the time relationship between the time of the drone landing on its paired truck at a moving-LRL on arc (b, j) and the drone arrival time at the last customer in a subtour.

3.3 | Nonlinear constraints

To describe the synchronization on arcs, we have developed constraints (2)–(19). Constraints (2)–(11) are linear. However, constraints (12)–(19) are not linear because the continuous variables d^r_{kuiab} and d^c_{kuiab} should be estimated by the following way.

For a triangle with nodes i, a, and b ($a, b \in V^{\text{tdc}}$, $i \in V^c$) as the vertices, we introduce f_{iab} to denote the cosine of angle $\angle iab$. The continuous variables d^r_{kuiab} and d^c_{kuiab} are estimated as follows:

$$d^r_{kui0j} = \left[\left(t^r_{ku0j} v^t\right)^2 + (d_{i0})^2 - 2 f_{i0j} d_{i0} t^r_{ku0j} v^t\right]^{0.5},$$

$$d^r_{kuiaj} = \left[\left(t^r_{kuaj} v^t\right)^2 + (d_{ia})^2 - 2 f_{iaj} d_{ia} t^r_{kuaj} v^t\right]^{0.5},$$

$$d^c_{kui0j} = \left[\left(t^c_{ku0j} v^t\right)^2 + (d_{i0})^2 - 2 f_{i0j} d_{i0} t^c_{ku0j} v^t\right]^{0.5}, \text{ and}$$

$$d^c_{kuibj} = \left[\left(t^c_{kubj} v^t\right)^2 + (d_{ib})^2 - 2 f_{ibj} d_{ib} t^c_{kubj} v^t\right]^{0.5}.$$

4 | BOUNDARY MODELS

We develop boundary models that can be solved directly by the CPLEX solver. Some linear piecewise functions are introduced to estimate d^r_{kuiab} and d^c_{kuiab} approximately. Without loss of generality, we take the triangle ΔIAB (ΔIAB uses nodes I, A, and B as the vertices) as an example (Figures 1 and 2). Customers A and B should be visited by a truck. Customer I should be visited by a drone. The truck travels from A to B. On arc (A, B), O is the nearest node to I. It is assumed that a drone carried by the truck launches (or lands) at a moving-LRL E on arc (A, B).

4.1 | Boundary model-I

When angle $\angle IAB$ is an acute angle (Figure 1a), the traveling distance d_{EI} and d_{EA} satisfy Equations (20), (21), and (22).

$$\text{when } d_{EA} = 0, d_{EI} = d_{IA}. \tag{20}$$

$$\text{when } d_{EA} = d_{AB}, d_{EI} = d_{IB}. \tag{21}$$

$$\text{when } d_{EA} = d_{IA} \cdot \cos\theta, d_{EI} = d_{IA} \cdot \sin\theta. \tag{22}$$

Referring to Equations (20), (21), and (22), we introduce Equation (23) to estimate d_{EI}.

$$\hat{d}_{EI} = \begin{cases} d_{EA} \cdot (\sin\theta - 1)/\cos\theta + d_{IA}, & \\ \qquad \text{if } \; 0 \leq d_{EA} < d_{IA} \cdot \cos\theta & \\ (d_{IB} - d_{IA} \cdot \sin\theta) \cdot (d_{EA} - d_{IA} \cdot \cos\theta) / & \\ (d_{AB} - d_{IA} \cdot \cos\theta) + d_{IA} \cdot \sin\theta, & \\ \qquad \text{if } \; d_{IA} \cdot \cos\theta \leq d_{EA} \leq d_{AB} & \\ d_{IA}, \qquad \text{if } \; - s_A \cdot v^t \leq d_{EA} < 0 & \end{cases}. \tag{23}$$

When angle $\angle IAB$ is a right angle or an obtuse angle (Figure 1b), the traveling distances d_{EI} and d_{EA} satisfy Equations (20) and (21). Referring to Equations (20) and (21), we introduce Equation (24) to estimate d_{EI}.

$$\hat{d}_{EI} = \begin{cases} d_{EA} \cdot (d_{IB} - d_{IA}) / d_{AB} + d_{IA}, & \text{if } \; 0 \leq d_{EA} \leq d_{AB} \\ d_{IA}, & \text{if } - s_A \cdot v^t \leq d_{EA} < 0 \end{cases}. \tag{24}$$

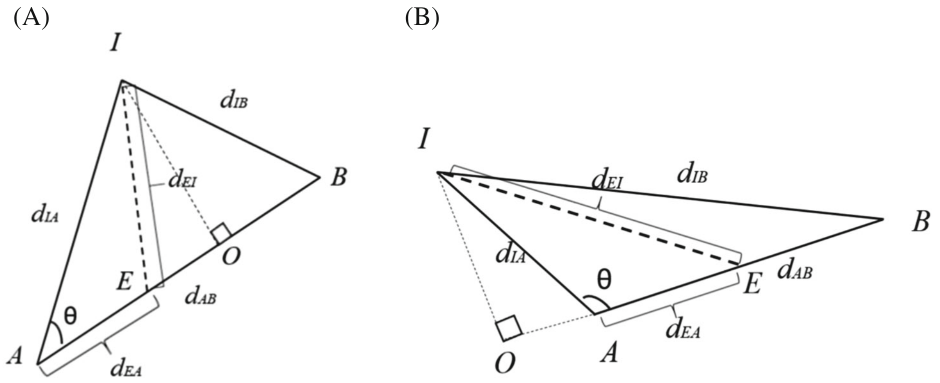


FIGURE 1 Triangle for illustrating the boundary model-I

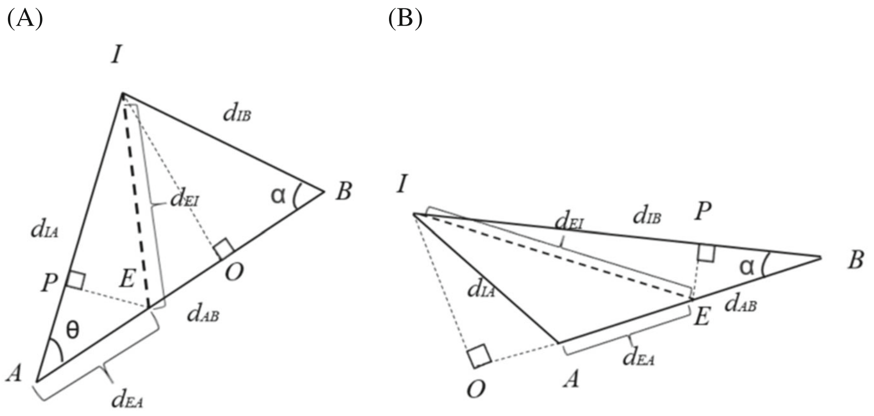


FIGURE 2 Triangle for illustrating the boundary model-II

When moving-LRLs are selected according to the linear piecewise functions (23) and (24), for the moving-LRL closer to nodes A, B, or O, the estimated traveling distance (i.e., $\widehat{d}_{EI}$) is closer to the real value. When the moving-LRL is located at nodes A, B, or O, the estimated traveling distance $\widehat{d}_{EI}$ is equal to the real value.

Lemma 1 $\widehat{d}_{EI} \geq d_{EI}$.

Proof Details of the proof are referred to Appendix S3. ∎

Based on Lemma 1, when linear piecewise functions (23) and (24) are referenced to develop a mixed-integer linear programming model, during the course of the solution, the drones take more time to fly along the subtours when the drone average velocity is constant. We name the mixed-integer linear programming model, which involves (23) and (24), as Boundary model-I (BM-I).

4.2 | Boundary model-II

When angle $\angle IAB$ is an acute angle (Figure 2a), we denote P on arc (I, A) as the nearest node to E if the selected moving-LRL is before O; we denote P on arc (I, B) as the nearest node to E if the selected moving-LRL is after O. We introduce Equation (25) to estimate d_{EI}.

When angle $\angle IAB$ is a right angle or an obtuse angle (Figure 2b), we denote P on arc (I, B) as the nearest node to E because the selected moving-LRL is always after O. Furthermore, we introduce Equation (26) to estimate d_{EI}.

$$\widehat{d}_{EI} = \begin{cases} d_{IA} - d_{EA} \cdot \cos\theta, & \text{if } 0 \leq d_{EA} < d_{IA} \cdot \cos\theta \\ d_{IB} - (d_{AB} - d_{EA}) \cdot \cos\alpha, & \text{if } d_{IA} \cdot \cos\theta \leq d_{EA} \leq d_{AB} \\ d_{IA}, & \text{if } -s_A \cdot v^t \leq d_{EA} < 0 \end{cases} . \tag{25}$$

$$\widehat{d}_{EI} = \begin{cases} d_{IB} - (d_{AB} - d_{EA}) \cdot \cos\alpha, & \text{if } 0 \leq d_{EA} \leq d_{AB} \\ d_{IA}, & \text{if } -s_A \cdot v^t \leq d_{EA} < 0 \end{cases} . \tag{26}$$

When moving-LRLs are selected according to the linear piecewise functions (25) and (26), for the moving-LRL nearer to nodes A or B, the estimated traveling distance (i.e., $\widehat{d}_{EI}$) is closer to the real value. When the moving-LRL is located at nodes A or B, the estimated traveling distance $\widehat{d}_{EI}$ is equal to the real value.

Lemma 2 $\widehat{d}_{EI} \leq d_{EI}$.

Proof Details of the proof are referred to Appendix S3. ∎

When linear piecewise functions (25) and (26) are referenced to develop a mixed-integer linear programming model, in the course of the solution, the drones take less time to fly

TABLE 2 Difference in d_{EI} and $\widehat{d}_{EI}$

		Difference in $\widehat{d}_{EI}$ and d_{EI}	
Conditions		**Boundary model-I**	**Boundary model-II**
$\theta \in \left[0, \frac{\pi}{2}\right]$	$d_{EA} = 0, d_{EA} = d_{AB}$, or $-s_A \cdot v^t \leq d_{EA} < 0$	$\widehat{d}_{EI} = d_{EI}$	$\widehat{d}_{EI} = d_{EI}$
	$d_{EA} = d_{IA} \cdot \cos\theta$	$\widehat{d}_{EI} = d_{EI}$	$\widehat{d}_{EI} < d_{EI}$
	$0 < d_{EA} < d_{IA}\cos\theta$, or $d_{IA}\cos\theta < d_{EA} < d_{AB}$	$\widehat{d}_{EI} > d_{EI}$	$\widehat{d}_{EI} < d_{EI}$
$\theta \in \left(\frac{\pi}{2}, \pi\right]$	$d_{EA} = d_{AB}$, or $-s_A \cdot v^t \leq d_{EA} < 0$	$\widehat{d}_{EI} = d_{EI}$	$\widehat{d}_{EI} = d_{EI}$
	$d_{EA} = 0$	$\widehat{d}_{EI} = d_{EI}$	$\widehat{d}_{EI} < d_{EI}$
	$0 < d_{EA} < d_{AB}$	$\widehat{d}_{EI} > d_{EI}$	$\widehat{d}_{EI} < d_{EI}$

along the subtours when the drone average velocity is constant. We name the mixed-integer linear programming model that involves (25) and (26) as Boundary model-II (BM-II).

4.3 | Evaluation of the boundary models

The only difference between the TDRP-SA model in Section 3 and the boundary models (i.e., BM-I and BM-II) is that in the boundary models, the drone flying distance on arcs of subtours is estimated by specific linear piecewise functions. Table 2 displays the difference in d_{EI} and $\widehat{d}_{EI}$.

The drone flying distance on arcs of subtours is not directly indicated in the objective function, which makes it difficult to directly judge the change in the objective function caused by the linear piecewise functions. Because the capacity, average velocity, operating cost, and maximum flying time per takeoff of each drone are considered in the TDRP-SA, the estimated drone-flying-distance on arcs of subtours directly affects the feasibility of the constructed subtours.

For each drone, the maximum flying distance per takeoff is defined as the product of the maximum flying time per takeoff and the average velocity. (i) On the condition that $\widehat{d}_{EI} \geq d_{EI}$, a drone flying on a subtour covers a longer distance because certain arcs are enlarged, which leads to extra drone occupations or maximum flying time per takeoff not being ensured. We take one subtour including more than one customer in the TDRP-SA solution as an example. As extra flying-time is needed to cover the distance $\widehat{d}_{EI} - d_{EI}$, to ensure the maximum flying time per takeoff, there are customers that cannot be served on time. As a result, the customers in one subtour in the TDRP-SA solution should be served by more than one subtour and/or drone route, which indicates an increase in the objective function. (ii) On the condition that $\widehat{d}_{EI} \leq d_{EI}$, a drone flying on a subtour covers a shorter distance because certain arcs are reduced, which leads to some free-time of the drone, and the maximum flying time per takeoff is certainly ensured. We take one subtour in the TDRP-SA solution as an example. To make full use of the drone free-time, the drone can visit new customers with a view to utilize the maximum flying time per takeoff. As a result, some customers are inserted into a subtour in the TDRP-SA solution, which indicates that the number of subtours or drone routes decreases and consequently, the objective function decreases.

BM-I and BM-II are formulated by vehicle flow formulations. An observation of the boundary models indicates that an optimal solution to the boundary models is actually an approximation of the optimal solution for the TDRP-SA model.

5 | SOLUTION METHOD

On the condition that customers served by the direct delivery mode are identified, constructing drone or truck routes for direct delivery means solving the VRPTW. Selecting moving-LRLs and ensuring time continuity are key to constructing combination routes. Generally, two moving-LRLs act as the origin and destination of a subtour because the drone launch/retrieval operations do not need the trucks to stop. We propose a mathematical analysis method to estimate a drone's arrival time at the first customer in a subtour and to select moving-LRLs.

5.1 | Seeking moving-LRLs

Considering that the main tours are traveled on by trucks departing from and returning to the depot, and considering that constructing truck routes for direct delivery means solving the VRPTW, it is assumed that there are several truck routes for constructing combination routes. Referring to data on the selected truck route traveled on by truck k, the arrival and leaving times of truck k at each served TDC are known. It is assumed that the first customer in a subtour is customer i while a moving-LRL (located on arcs of the route traveled on by truck k) is required by the subtour.

Lemma 3 *There is at least one moving-LRL r, a drone launching from which, can fly to arrive at customer i in time window TW^r, where*

$$TW^r = \left[t^A_{kr} + d_{ri}/v^u, t^A_{kr} + t^W_{kr} + s_r + d_{ri}/v^u\right]. \qquad (27)$$

Lemma 4 *There is at least one moving-LRL c, at which a drone landing on its truck, has completed serving customer i in time window TW^c, where*

$$TW^c = \left[t^A_{kc} - d_{ic}/v^u - s_i - t^W_{kui}, t^A_{kc} + t^W_{kc} + s_c - d_{ic}/v^u - s_i - t^W_{kui}\right]. \tag{28}$$

Proof Details of the proof are referred to Appendix S3. ■

We introduce time window $\left[t^{ue}_{krci}, t^{ul}_{krci}\right] = TW^r \cap TW^c \cap [e_i, l_i]$. If $TW^r \cap TW^c \cap [e_i, l_i] \neq \emptyset$, the drone's arrival time at customer i can be estimated as any value in the range $\left[t^{ue}_{krci}, t^{ul}_{krci}\right]$.

5.1.1 | Seeking a moving-LRL for drone launch

Without loss of generality, we consider the triangle with nodes I, A, and B as the vertices as an example (Figure 1a in Section 4) to illustrate the identification of a moving-LRL for drone launch. The lengths of the sides of the triangle are denoted as d_{AB}, d_{IA}, and d_{IB}, respectively. We aim to seek a moving-LRL on arc (A, B) for a drone launching to serve customer I.

We introduce t_A to denote the leaving time of the truck at customer A, t'_A to denote the arrival time of the truck at customer A, and t_I to denote the arrival time of the drone at customer I. It is assumed that node E is a moving-LRL on arc (A, B). The traveling distance between nodes E and A is denoted as d_{EA}, and the drone flying distance between nodes E and I is denoted as d_{EI}. From the time continuity perspective, Equation (29) is applicable. Let $h_1 = -v^u/v^t$ and $h_2 = (t_I - t_A)\,v^u$, and Equation (29) is transformed to Equation (30).

$$t_A + d_{EA}/v^t + d_{EI}/v^u = t_I. \tag{29}$$

$$d_{EI} = h_1 d_{EA} + h_2. \tag{30}$$

The cosine of angle $\angle IAB$ is estimated as (31).

$$\cos\theta = \left(d^2_{IA} + d^2_{AB} - d^2_{IB}\right) / \left(2 d_{IA} d_{AB}\right). \tag{31}$$

As the length of each side of the triangle (i.e., ΔIAB with nodes I, A, and B as the vertices) is known, the cosine of angle $\angle IAB$ is known, which is denoted as h_3, where $h_3 = \left(d^2_{IA} + d^2_{AB} - d^2_{IB}\right) / \left(2 d_{IA} d_{AB}\right)$.

In the triangle with nodes I, A, and E as the vertices, the cosine of angle $\angle IAB$ is estimated as

$$\cos\theta = \left(d^2_{IA} + d^2_{EA} - d^2_{EI}\right) / \left(2 d_{IA} d_{EA}\right). \tag{32}$$

Therefore,

$$h_3 = \left[d^2_{IA} + d^2_{EA} - \left(h_1 d_{EA} + h_2\right)^2\right] / \left(2 d_{IA} d_{EA}\right). \tag{33}$$

Equation (33) is solved to obtain d_{EA} as

$$d_{EA} = \left[h_1 h_2 + d_{IA} h_3 + \sqrt{d^2_{IA}\left(h^2_1 + h^2_3 - 1\right) + h^2_2 + 2 h_1 h_2 h_3 d_{IA}}\right] \Big/ \left(1 - h^2_1\right)$$

or

$$d_{EA} = \left[h_1 h_2 + d_{IA} h_3 - \sqrt{d^2_{IA}\left(h^2_1 + h^2_3 - 1\right) + h^2_2 + 2 h_1 h_2 h_3 d_{IA}}\right] \Big/ \left(1 - h^2_1\right). \tag{34}$$

If the estimated d_{EA} in (34) satisfies $0 < d_{EA} < d_{AB}$, there exists a moving-LRL on arc (A, B), which is located at node E for a drone launching to serve customer I, and the drone launch time at the moving-LRL is $t_A + d_{EA}/v^t$. Specifically, if $t'_A + d_{IA}/v^u \leq t_I \leq t_A + d_{IA}/v^u$, a drone can launch to serve customer I, when its truck is serving customer A.

5.1.2 | Seeking a moving-LRL for drone retrieval

We consider the triangle with nodes I, A, and B as the vertices as an example (Figure 1a in Section 4) to illustrate how to identify a moving-LRL for drone landing. It is supposed that node E is a moving-LRL on arc (A, B) for drone landing, and I is the last customer in the subtour. We introduce t_A to denote the arrival time of the truck at TDC A, t'_A to denote the leaving time of the truck at TDC A, and t'_I to denote the leaving time of the drone at customer I. From the time continuity perspective, Equation (35) exists. Let $h_4 = v^u/v^t$ and $h_5 = v^u\left(t'_A - t'_I\right)$, and Equation (35) is transformed to Equation (36).

$$t'_A + d_{EA}/v^t = t'_I + d_{EI}/v^u. \tag{35}$$

$$d_{EI} = h_4 d_{EA} + h_5. \tag{36}$$

The cosine of angle $\angle IAB$ is estimated as h_3. In the triangle that uses nodes I, A, and E as the vertices, the cosine of angle $\angle IAB$ is also estimated as (37).

$$h_3 = \left[d^2_{IA} + d^2_{EA} - \left(h_4 d_{EA} + h_5\right)^2\right] / \left(2 d_{IA} d_{EA}\right). \tag{37}$$

Equation (37) is solved to obtain d_{EA} as

$$d_{EA} = \left[h_4 h_5 + d_{IA} h_3 + \sqrt{d^2_{IA}\left(h^2_4 + h^2_3 - 1\right) + h^2_5 + 2 h_3 h_4 h_5 d_{IA}}\right] \Big/ \left(1 - h^2_4\right)$$

or

$$d_{EA} = \left[h_4 h_5 + d_{IA} h_3 - \sqrt{d^2_{IA}\left(h^2_4 + h^2_3 - 1\right) + h^2_5 + 2 h_3 h_4 h_5 d_{IA}}\right] \Big/ \left(1 - h^2_4\right). \tag{38}$$

If the estimated d_{EA} in (38) satisfies $0 < d_{EA} < d_{AB}$, there exists a moving-LRL on arc (A, B), which is located at node E for a drone landing on its truck, and the drone landing time at the moving-LRL is $t'_A + d_{EA}/v^t$. Specifically, if $t_A - d_{IA}/v^u \leq t'_I \leq t'_A - d_{IA}/v^u$, a drone can land on its truck when its truck is serving customer A.

```
Algorithm 1. Seeking a moving-LRL for drone launch from a truck route
Step 1: Given a truck route, initialize the first arc in the truck route as (a, b).
Step 2: If there is a drone available for launch, go to Step 3; otherwise, go to Step 4.
Step 3: If $t^{Am}_{kuai} - t_O < L^U - s_i$, go to Step 3.1; otherwise, go to Step 4.
    Step 3.1: If $e_i \le t^{Am}_{kuai} \le l_i$, $t^A_{kuai} = t^{Am}_{kuai}$, and node O on arc (a, b) acts as a moving-LRL. If $t^{Am}_{kuai} > l_i$, which indicates that the time window of customer i is not ensured, we set $t^A_{kuai} = l_i$, to respect the time window constraints. Similarly, if $t^{Am}_{kuai} < e_i$, we set $t^A_{kuai} = e_i$. The arrival time or the leaving time of the truck at TDC a is used as the launch time of the drone to serve customer i, and the drone arrival time at customer i is estimated as $TW^r$ (set $r = a$, i.e., $TW^r = TW^a$).
    Step 3.2: If $t^A_{kuai} \in TW^a$, node a acts as a moving-LRL, and the launch time of the drone to serve customer i is $t^A_{kuai} - d_{ia} / v^u$. If $t^A_{kuai} \notin TW^a$, the estimated $d_{Ea}$ from Equation (34) is used to locate a moving-LRL. There may exist a moving-LRL on arc (a, b) that is located at node E, and the drone launch time at the moving-LRL is $t_a + d_{Ea} / v^t$.
    Step 3.3: If the drone flying-time from the moving-LRL to customer i respects the maximum flying time per takeoff, the moving-LRL and the drone launch time are recorded. Go to Step 4.
Step 4: If arc (a, b) is the last arc in the truck route, stop; otherwise, arc (a, b) is substituted by the next arc in the same truck route, and go to Step 3.
```

FIGURE 3 Algorithm for seeking a moving-launch/retrieval locations for drone launch

5.1.3 | Algorithms for seeking moving-LRLs

Theoretically, in an arc there are innumerable nodes that can be used as moving-LRLs, and it is impossible to verify whether every node is suitable for a moving-LRL. With a view of saving the drone flying time as much as possible, we introduce an algorithm for seeking a moving-LRL for drone launch, and estimating the drone arrival time at the first customer i in a subtour, which is called Algorithm 1. Specifically, drone u, carried by truck k, which launches at a moving-LRL r on arc (a, b), should fly along arc (r, i) to arrive at customer i at time t^A_{kuai} by taking the minimum flying time.

Algorithm 1 is based on a triangle that comprises customer i and an arc in a truck route, and the arcs in all the truck routes are enumerated. We denote the arc included in a triangle as (a, b). On arc (a, b), O is the nearest node to customer i. It is assumed that a drone launches at node O to serve customer i; the drone launch time at node O is denoted as t_O and the drone arrival time at customer i is denoted as t^{Am}_{kuai}. Under the condition of $t^{Am}_{kuai} - t_O < L^U - s_i$, Algorithm 1 (Figure 3), is executed.

Using Algorithm 1, the moving-LRL for drone launch and the drone arrival time at the first customer i in a subtour is obtained. Using the time continuity constraints on subtours, the drone leaving time at the last customer i' in a subtour, which is denoted as t'_i, is determined. We introduce Algorithm 2 to seek a moving-LRL for drone retrieval. Algorithm 2 is based on the triangle that comprises customer i' and an arc in the truck route, including the moving-LRL for drone launch. We denote the arc included in the triangle as (a', b'). Algorithm 2 includes the steps shown in Figure 4.

The results of Algorithms 1 and 2 may include several pairs of moving-LRLs for drone launch/retrieval. The pair of moving-LRLs for drone launch/retrieval, which leads to the minimum flying time of the drone traveling the subtour, is finally selected.

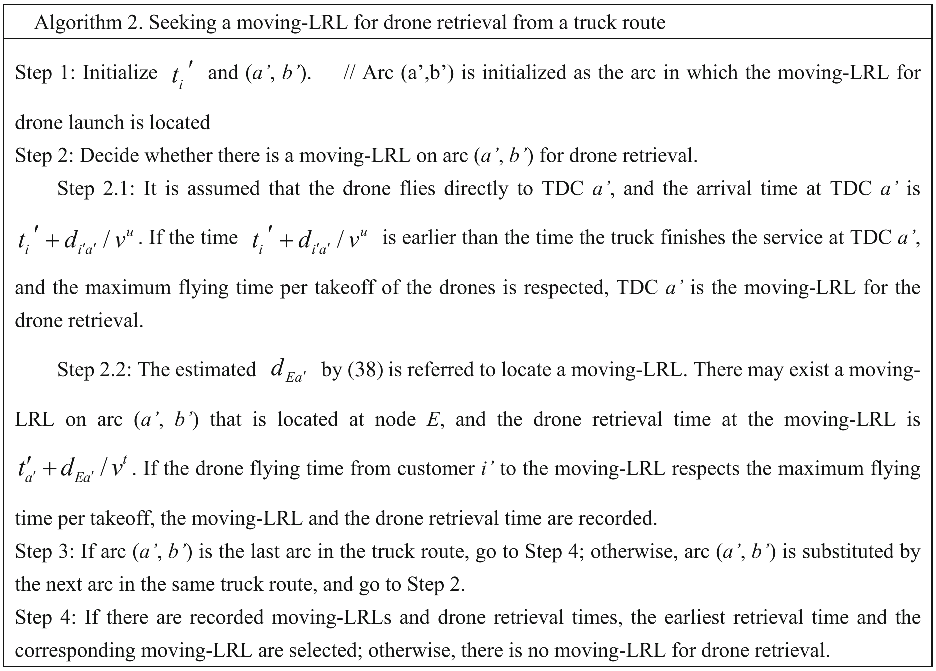
5.2 | An ALNS-based heuristic

5.2.1 | Initial solution

To construct an initial solution, three steps including classifying customers, construction of drone routes and truck routes, and construction of combination routes, are adopted.

Firstly, customers in subtours of combination routes, drone routes, and truck routes are distinguished. If the demand of customer i is not more than the drone's capacity, and the drone flying distance between the depot and the customer exceeds $\left(L^U - s_i\right) / (2v^u)$, customer i is classified into a subset of customers in the subtours. If the demand of a customer is greater than the drone's capacity, the customer is classified into a subset of customers in the truck routes. The remaining customers are classified into a subset of customers in the drone routes.

Secondly, the drone routes are constructed by using a savings-based algorithm. Based on the subset of customers in the drone routes, the customer with the minimum latest service-starting time of time windows is inserted as the first



```
Algorithm 2. Seeking a moving-LRL for drone retrieval from a truck route

Step 1: Initialize $t_i'$ and ($a'$, $b'$).    // Arc (a',b') is initialized as the arc in which the moving-LRL for drone launch is located
Step 2: Decide whether there is a moving-LRL on arc ($a'$, $b'$) for drone retrieval.
    Step 2.1: It is assumed that the drone flies directly to TDC $a'$, and the arrival time at TDC $a'$ is $t_i' + d_{i'a'} / v^u$. If the time $t_i' + d_{i'a'} / v^u$ is earlier than the time the truck finishes the service at TDC $a'$, and the maximum flying time per takeoff of the drones is respected, TDC $a'$ is the moving-LRL for the drone retrieval.
    Step 2.2: The estimated $d_{Ea'}$ by (38) is referred to locate a moving-LRL. There may exist a moving-LRL on arc ($a'$, $b'$) that is located at node $E$, and the drone retrieval time at the moving-LRL is $t'_{a'} + d_{Ea'} / v^t$. If the drone flying time from customer $i'$ to the moving-LRL respects the maximum flying time per takeoff, the moving-LRL and the drone retrieval time are recorded.
Step 3: If arc ($a'$, $b'$) is the last arc in the truck route, go to Step 4; otherwise, arc ($a'$, $b'$) is substituted by the next arc in the same truck route, and go to Step 2.
Step 4: If there are recorded moving-LRLs and drone retrieval times, the earliest retrieval time and the corresponding moving-LRL are selected; otherwise, there is no moving-LRL for drone retrieval.
```

FIGURE 4 Algorithm for seeking a moving-launch/retrieval locations for drone retrieval

customer. Then, the customers with the maximum savings are added. Based on the subset of customers in the truck routes, the truck routes are constructed using a savings-based algorithm similar to that used in constructing the drone routes.

Thirdly, subtours are constructed through an algorithm based on Algorithms 1 and 2. As shown in Algorithm 2, there may be no suitable moving-LRL. To ensure that all the customers are served in the initial solution, the constraint on the maximum flying time of one takeoff for each drone is ignored, and some direct services, whose integrated cost is estimated as a large enough integer, are constructed. In a direct service, a drone departs from the depot, visits a customer in the subset of customers in subtours, and directly returns to the depot.

5.2.2 | Adaptive probabilistic mechanism

The ALNS in this paper adopts an adaptive probabilistic mechanism to select a destroy-and-repair operator through a particle swarm optimization. It is assumed that there are a predetermined number of particles. The initial position and velocity of the mth particle are denoted as x_m^0 and v_m^0, respectively. All the destroy-and-repair operators are labeled by natural numbers. The position of a particle corresponds to one destroy-and-repair operator. The initialized position of each particle is a randomly generated integer in the range of [1, N_{op}] (N_{op} denotes the number of destroy-and-repair operators), that is, $x_m^0 \in \left[1, N_{op}\right]$. The initial velocity is $v_m^0 = 0$. The best positions of all the particles are denoted as x_m^{best}. The global iteration of the ALNS includes two types of internal iterations.

Based on the initial solution (denoted as sol_0), the first internal iteration generates new solutions by destroy-and-repair operators paired with particle positions, and updates the positions of the particles. The predetermined number of particles indicates the number of iterations of the first internal iteration. The position of the mth particle indicates that at an internal iteration, the corresponding destroy-and-repair operator is adopted to generate a new solution (denoted as sol′). The particle position and velocity, referring to which, the currently best solution (denoted as sol*) is attained, are referenced to update the position and velocity of every particle by Equations (39) and (40). v_m^I and x_m^I denote the velocity and position of the mth particle at the Ith internal iteration, respectively. At the beginning, $x_{\text{global}}^{\text{best}}$ denotes the best position of all the particles in the first internal iteration; then, if sol* is better than sol′, $x_{\text{global}}^{\text{best}}$ is updated as x_m^I. ω, ε_1, and ε_2 are predetermined parameters, and rand_1^I or rand_2^I is a randomly generated number in range [0, 1]. The position and velocity of each particle are initialized every five internal iterations. The second internal iteration estimates and updates the position and velocity of each particle, by referring to the position and velocity of the particle with the best performance.

$$v_m^{I+1} = \omega v_m^I + \varepsilon_1 \cdot \text{rand}_1^I \cdot \left(x_m^{\text{best}} - x_m^I\right) + \varepsilon_2 \cdot \text{rand}_2^I \cdot \left(x_{\text{global}}^{\text{best}} - x_m^I\right). \tag{39}$$

$$x_m^{I+1} = \left\lfloor x_m^I + v_m^{I+1} \right\rfloor. \tag{40}$$

5.2.3 | Destroy operators and repair operators

The destroy operators include random removal, worst removal, and Shaw removal. For the introduction of these destroy operators, we refer to Ropke and Pisinger (2006). Each of the destroy operators is adjusted to remove one customer or to remove a set of customers at an iteration. A set of removed customers are identified by the following rules: on the condition that one customer has been selected for removal by certain criterion of a destroy operator, if the removed customer is included in a subtour, all other customers in the same subtour are removed; if the removed customer is included in a main tour, all customers included in the subtours that are connected with the main tour are removed. To adopt the Shaw removal operator, the degree of similarity between two customers is estimated by the distance. The two customers referred to which the degree of similarity is estimated are identified by the following rules. If there is only one removed customer, the customer that is the nearest to the removed customer is removed. If there is a set of removed customers, the customer in subtours that is visited by a drone at the latest time is referred to find the nearest customer for removal.

We provide two types of repair operators: deep-greedy repair operator and regret-2 repair operator. For the introduction of these repair operators, we refer to Ribeiro and Laporte (2012). Routes in the solution brought by the destroy operators are classified into three types, that is, drone routes (also including direct service), truck routes, and combination routes. When a removed customer is inserted into a drone route or a truck route, constraints on vehicle capacity, route duration and time windows should be ensured. When a removed customer is inserted into a main tour or a subtour, constraints on vehicle capacity, route duration, time windows and synchronization on arcs should be ensured, and Algorithms 1 and 2 are used to relocate moving-LRLs, if necessary.

The following ways are used one by one, so as to seek all feasible positions for a removed customer: (i) Constructing a new drone route for direct delivery. If constraints on drone capacity and duration are respected, the integrated cost is recorded; otherwise, a penalty cost denoted by a large enough integer is recorded. (ii) Inserting into a drone route. The integrated cost change corresponding to each feasible position is recorded. (iii) Constructing a new truck route if the removed customer is a TDC, and the integrated cost is recorded. (iv) Inserting into a truck route if the removed customer is a TDC, and the integrated cost change corresponding to each feasible position is recorded. (v) Constructing a new subtour, Algorithms 1 and 2 are used to seek moving-LRL, the integrated cost of the newly constructed subtour is recorded. (vi) Inserting into a subtour, and the integrated cost change corresponding to each feasible position is recorded. (vii) If the inserted customer is a TDC, constructing a new truck route when in the current solution there is drone direct-service (that violates the constraint on the maximum flying time of one takeoff for each drone). Based on the newly constructed truck route, Algorithms 1 and 2 are used to seek moving-LRLs for the direct service to construct a new combination route.

The deep-greedy repair operator involves comparing the integrated cost changes. When the removed customer is inserted into the position that increases the least integrated-cost, the corresponding position is accepted. The regret-2 repair operator locates where to insert a removed customer with the largest integrated-cost difference that is between the best position and the second-best position. When there are more than one feasible position, the integrated-cost difference between its best and second-best insertion is estimated as the regret value. When there is only one feasible position, the integrated-cost for a new drone- or truck-route is estimated, and the regret value is estimated as the difference between its insertion cost and the cost of the new route. When it is not feasible for a removed customer to be inserted into any route, the regret value is estimated as a small enough number.

6 | COMPUTATIONAL EXPERIMENTS

In some small-scale instances, we directly use the CPLEX 12.9 solver to solve the boundary models. The method provided in Section 5 is coded in C++. The codes are run on a computer with a Windows 8 operating system configured with an Intel(R) Core(TM) 3.2-GHz processor with 8-GB memory.

6.1 | Test instances

We choose the instances from Solomon (1987) VRPTW benchmark problems and convert them to TDRP-SA instances. We select instance C101 to generate small-scale instances, and instances C101–C109, R101–R112, RC101–RC108 to generate large-scale benchmark instances. All instances are available in Mendeley Data (https://data.mendeley.com/datasets/cxc6p5x4ts/1).

6.1.1 | Small-scale instances

The depot in the VRPTW instance C101 is used as the depot of the TDRP-SA small-scale instance. Let *nc* denote the number of customers. The *nc* customers are classified into TDCs and DRCs. Let A_i denote the distance between customer i and the neighboring customer. In the first type of TDRP-SA small-scale instances, 25% of customers with the smallest A_i are specified as DRCs. This percentage increases to 50% and 75% for the second and third types of TDRP-SA small-scale instances, respectively. The TDC or DRC demand is a random number in the range (0, truck capacity) or (0, drone capacity), respectively. The customer time windows with the width ranged from 0.5 to 1.5 h are randomly generated. The service times at a TDC and a DRC are taken as 15 and 5 min, respectively. *nc* customers included in the VRPTW instance C101 are randomly selected, on the condition that the generated instance has feasible solutions.

Each truck carries two drones. The drone and truck capacity are 5 and 1000 kg, respectively. The maximum working time of each truck is 10 h, and the maximum flying time of one takeoff for each drone is 0.67 h. The average velocity of each truck and that of each drone are 60 and 65 km/h, respectively. The variable cost of each truck is 0.8 Chinese Yuan/km, and the operating cost per takeoff of each drone is 15 Chinese Yuan. The penalty of a truck waiting for a unit of time at customers is estimated as 1. Small-scale instances are named by C“nc”-1, C“nc”-2, or C“nc”-3, where C“nc”-1, C“nc”-2, or C“nc”-3 has the percentage of DRCs 25%, 50%, or 75%, respectively.

6.1.2 | Large-scale benchmark instances

We modify the benchmark instances C101–C109, R101–R112, and RC101–RC108 by classifying customers into TDCs and DRCs. Each of the VRPTW benchmark instances is converted into three types of TDRP-SA test instances. For customer i in the VRPTW benchmark instance, the Euclidean distance between the depot and customer i is denoted by A_i. All customers are sorted from 1 to 100 in ascending order of A_i ($i \in V^c$), so that each customer has a sequence number. Firstly, customer i with $A_i \leq \left(L^U - s_i\right) \cdot v^u/2$ is preferentially regarded as a candidate DRC, and such customers are included in set CS1. Secondly, if the weight of the parcels required by customer i is not more than the drone capacity, and L^U is respected when a drone flies from the depot to visit customer i, and arrives finally at the nearest TDC location j, customer i is regarded as candidate DRC. Such customers are included in set CS2. Thirdly, if the weight of the parcels required by customer i is not more than the drone capacity, and the Euclidean distance between customer i and the nearest TDC j satisfies $d_{ij} \leq \left(L^U - s_i\right) \cdot v^u/2$, then customer i is regarded as a candidate DRC. Such customers are included in set CS3. For customers in sets CS2 and CS3, e_i is adjusted to be the same as e_j, and l_i is adjusted to be the sum of l_j and d_{ij}/v^u. Customers in CS1, CS2, or CS3 are sorted in ascending order of their sequence numbers. Customers in CS1 are preferentially specified as DRCs. If customers in CS1 are not enough, customers in CS2 and then in CS3 are selected as DRCs.

We modify time windows of the benchmark instances C101–C109 with classified customers, so that the generated instance has feasible solutions. The service time of each TDC or DRC is 10 and 3, respectively. The drone and truck capacity are 60 and 1000, respectively. The maximum working time of each truck is 1000, and the maximum flying time of one takeoff for each drone is 40. The average velocity of each truck and that of each drone are 1 and 1.1, respectively. The variable cost of each truck is 1, and the operating cost per takeoff of each drone is 5.

Large-scale benchmark instances are named by “name of VRPTW benchmark instance”-1, “name of VRPTW benchmark instance”-2, or “name of VRPTW benchmark instance”-3, which has the number of DRCs 10, 20, or 30, respectively.

6.2 | Effectiveness of the boundary models and the heuristic

We use CPLEX 12.9 to solve the boundary models. CPLEX 12.9 runs with default settings until it finds an exact solution or until it stops because the computer memory is exhausted. For instances with exhausted computer memory, we set the computation time of CPLEX 12.9 to 14 400.00 s. Implementing the conditions specified in BM-I and BM-II, the exact results of the small-scale instances, and the results using the heuristic are presented in Table S2. We introduce Gap1 to show the percentage gap between the objective value of the exact solution obtained by boundary models and that of the heuristic solution of each instance.

When there are 10–11 customers, CPLEX 12.9 is unable to generate optimal solutions for five out of six cases owing to computer memory overflow or computation time limitation. The solution obtained by CPLEX 12.9 is used for comparison. Firstly, for 16 out of the 21 small-scale instances, CPLEX 12.9 can obtain the optimal boundary solutions. Of the 16 small-scale instances with optimal boundary solutions, the objective value of BM-I is equal to that of BM-II. The only difference between the two boundary models is the linear piecewise function used to estimate the drone flying distance on arcs of subtours. Considering that in the objective function, the operating cost of drones is calculated by referring to the “per takeoff” unit, the same objective value indicates that the number of subtours in the optimal solutions does not change, on the condition of different boundary models. Referring to the optimal solutions, although the objective value for an instance can be the same on the condition of different boundary models, the moving-LRLs can be located in different arcs, or the moving-LRLs are located in different nodes on the same arc. Secondly, for two instances with up to 10 or 11 customers, CPLEX 12.9 requires the full computation time limit to solve BM-I; for three instances with up to 10 or 11 customers, CPLEX 12.9 runs out of computer memory. For five instances with up to 10 or 11 customers, CPLEX 12.9 requires the full computation time limit to solve BM-II. For five instances with up to 10 or 11 customers, none of the solutions to BM-I or BM-II formulations are provably optimal.

The heuristic can solve each of the 21 small-scale instances in approximately 6.0 s. The longest and average computational times are 6.3 and 3.6 s, respectively. The heuristic solution and the solution found by the CPLEX solver solving the boundary models are very close. Of the 16 small-scale instances having exact solutions for the condition of the boundary models, Gap1 is 0.00% in all the 16 cases. Of the five instances, in which CPLEX 12.9 cannot find the exact solutions of the boundary models in the limited computation time, Gap1 is 0.00% in two cases and Gap1 is less than 0.00% in three cases. For C10-1, C11-1, and C11-3, the heuristic can find a better solution than the CPLEX solver.

6.3 | Application of heuristic for large-scale benchmark instances

6.3.1 | Performance of the ALNS-based heuristic

Among all the routing variants introduced in the literature, the truck and trailer routing problem with time windows (TTRPTW) is a variant that is most similar to the TDRP-SA.

TABLE 3 Results of large-scale benchmark instances on the condition of the truck and drone routing problem with synchronization on arcs

Instance	Avg.Obj^H	Avg.N^t	Avg.N^u	Avg.N^{mr}	Avg.N^{sr}	Avg.T^H (s)
C	873.18	1.89	7.89	6.37	12.78	715.46
R	930.11	0.42	7.78	7.31	23.11	2272.44
RC	995.32	0.25	8.13	7.58	20.71	1563.67

We employ the TDRP-SA heuristic to solve the TTRPTW benchmark instances. As the vehicles (i.e., truck-trailer combination or truck) involved in the TTRPTW are different from vehicles involved in the TDRP-SA, we assume that the trucks or drones have the same capacities and velocities as those of a truck-trailer combination or a truck, and adjust the objective function of the TDRP-SA. The results of TTRPTW-benchmark-instances attained by the TDRP-SA heuristic and the known results of the TTRPTW benchmark instances in the literature are shown in Table S3. To the best of our knowledge, the best results of the TTRPTW benchmark instances are reported by Parragh and Cordeau (2017). We introduce Gap2 to denote the percentage gap between the best objective values reported by Parragh and Cordeau (2017) and the best results reported by the TDRP-SA heuristic.

For the 18 TTRPTW-benchmark-instances, the TDRP-SA heuristic can achieve new best solutions or similar solutions for five cases. The smallest Gap2 is −5.80%. The TDRP-SA heuristic can obtain average gaps to best-known solutions of less than 1.67%. The computational results of the 18 TTRPTW-benchmark-instances indicate the good performance of the TDRP-SA heuristic. Moreover, we can estimate the values of parameters of the ALNS-based heuristic by referring to the computational experiments on the TDRP-SA heuristic solving the TTRPTW-benchmark-instances.

6.3.2 | Heuristic results of large-scale benchmark instances

For each large-scale benchmark instance, the ALNS-based heuristic is repeated 10 times, and the objective value and computation times of the best solution of the 10 results are reported. Table 3 shows the average of the heuristic solutions of the large-scale benchmark instances, and details of the heuristic solutions are referred to Tables S4, S5, and S6. Obj^H shows the objective value of the heuristic solution. T^H (s) shows the computation time (s). The ALNS-based heuristic can solve any one of the 87 large-scale benchmark instances in less than 4069 s. The shortest and the average computation times are approximately 209 and 1594 s, respectively.

All routes traveled by trucks can be classified into two types: those visited by trucks for direct delivery and those involving moving-LRLs. The number of the former is denoted by N^t, and the number of latter, by N^{mr}. All routes traveled by drones can be classified into two types: those visited by drones for direct delivery and those using moving-LRLs as the origins or destinations. The number of the former is denoted by N^u, while the number of the latter is denoted by N^{sr}. In the heuristic results of every large-scale benchmark instance, N^t is obviously less than N^{mr}, and N^u is evidently less than N^{sr}, which indicates that the moving-LRL approach is more common than the direct delivery from the depot.

TABLE 4 Results of large-scale benchmark instances on the condition of the truck and drone routing problem with synchronization on arcs variant with satellite synchronization

Instance type	Avg.Obj$^{H\text{-}V}$	Avg.T^H (s)	Avg.$N^{sr/mr}$	# inc.obj	# dec.N^{sr}
C	965.9	1088.4	1.5	27/27	23/27
R	1039.1	1434.3	2.3	36/36	27/36
RC	1081.1	1857.5	2.1	24/24	20/24

Abbreviations: "# inc.obj.", the number of instances with objective value being larger than that of the benchmark condition; "# dec.N^{sr}", the number of instances with N^{sr} being smaller than that of the benchmark condition.

Based on the computational experiments on large-scale benchmark instances, we evaluate the gains in delivery performance offered by synchronization on arcs, multiple drones carried by each truck, drone direct delivery from the depot, etc. Tables 4–7 show the average of the heuristic solutions, and details of the solutions are referred to Tables S7–S21.

Comparison between synchronization on arcs and satellite synchronization

When all moving-LRLs are required to be located at customers, the TDRP-SA is transformed to a variant that has the main characteristic of satellite synchronization at customers. The proposed ALNS-based heuristic is adjusted by simplifying the algorithm to search for moving-LRLs, so as to solve the TDRP-SA variant with satellite synchronization. The heuristic can solve any one of the 87 large-scale benchmark instances in less than 2451 s, and the shortest and the average computation times are 928 and 1444 s, respectively. For every large-scale benchmark instance, the integrated cost of the TDRP-SA is always less than that of the TDRP-SA variant with satellite synchronization. We introduce Gap3, and Gap3 = 100% × (Obj^H − Obj$^{H\text{–}V}$) / Obj$^{H\text{–}V}$, where Obj$^{H\text{-}V}$ denotes the objective value of the TDRP-SA variant with satellite synchronization. The average, largest, and smallest values of Gap3 are −8.92%, −1.31%, and −19.07%, respectively. The synchronization on arcs is always advantageous for decreasing the integrated costs.

For 70 out of the 87 large-scale benchmark instances, $N^{sr} > N^{sr-V}$ (where N^{sr-V} denotes the number of sub-tours in the solution of the TDRP-SA variant with satellite synchronization). The number of subtours per combination route is estimated by $N^{sr/mr}$ or $N^{sr-V/mr-V}$ (where N^{mr-V} denotes the number of main tours in the solution of the TDRP-SA variant with satellite synchronization). The smallest, largest, and average values of $N^{sr/mr}$ are 1.3, 4.8, and 2.7, respectively; the smallest, largest, and average values of $N^{sr-V/mr-V}$ are 1.0, 3.0, and 2.0, respectively. The number of subtours per combination route indicates the synchronization on arcs and the usage frequency of the carried drones.

TABLE 5 Results of large-scale benchmark instances on the condition of different values of the penalty for a truck waiting for a unit of time at customers

	$\tau = 0$				$\tau = 2$			
Instance type	**Avg.Obj^N**	**Avg.R^t**	**Avg.R^u**	**# dec.obj.**	**Avg.Obj^E**	**Avg.R^t**	**Avg.R^u**	**# inc.obj.**
C	810.20	7.4	1.8	27/27	896.90	8.2	1.6	19/27
R	857.72	6.4	1.4	34/36	963.53	7.3	1.4	31/36
RC	959.89	6.8	1.5	22/24	1018.94	7.3	1.5	18/24

Abbreviations: "# dec.obj.", the number of instances with objective value being smaller than that of the benchmark condition. R^t or R^u, the average number of truck–drone customers included in one truck route or that included in one drone route.

TABLE 6 Results of large-scale benchmark instances on the condition of various numbers of drones carried by each truck

	$\lvert U \rvert = 4$		$\lvert U \rvert = 6$		$\lvert U \rvert = 8$		$\lvert U \rvert = 10$	
Instance type	**Avg.Obj^H**	**# dec.obj.**	**Avg.Obj^H**	**# dec.obj.**	**Avg.Obj^H**	**# dec.obj.**	**Avg.Obj^H**	**# dec.obj.**
C	867.21	22/27	865.55	20/27	863.22	22/27	862.11	21/27
R	887.90	30/36	886.88	31/36	882.16	32/36	889.08	29/36
RC	969.74	18/24	957.82	21/24	946.67	24/24	951.22	23/24

TABLE 7 Results of large-scale benchmark instances on the condition of various values of truck velocity

	$v^t = 0.67$					$v^t = 1.33$				
Instance type	**Avg.Obj^H**	**Avg.$N^{sr/mr}$**	**Avg.N^{sr}**	**# inc.$N^{sr/mr}$**	**# inc.N^{sr}**	**Avg.Obj^H**	**Avg.$N^{sr/mr}$**	**Avg.N^{sr}**	**# dec.$N^{sr/mr}$**	**# dec.N^{sr}**
C	1033.90	18.1	2.1	15/27	27/27	890.94	9.7	1.9	17/27	20/27
R	970.69	28.8	3.3	21/36	36/36	959.35	19.3	2.9	22/36	30/36
RC	1021.65	26.1	3.1	17/24	22/24	1010.72	17.8	2.5	19/24	17/24

Abbreviations: "# inc.$N^{sr/mr}$", the number of instances with $N^{sr/mr}$ being larger than that in the condition "$v^t = 1$". "# dec.$N^{sr/mr}$", the number of instances with $N^{sr/mr}$ being smaller than that in the condition "$v^t = 1$".

Impact of waiting time at customers

To experimentally explore the impact of the waiting cost, we set several values to the penalty for a truck waiting for a unit of time at customers. Let τ have different values, that is, $\tau = 0$, $\tau = 1$, or $\tau = 2$. For 83 out of the 87 large-scale benchmark instances, the integrated cost on the condition of "$\tau = 0$" is less than that on the condition of "$\tau = 1$". For 68 out of the 87 large-scale benchmark instances, the integrated cost on the condition of "$\tau = 2$" is more than that on the condition of "$\tau = 1$". We introduce Gap4 and Gap5, Gap4 = $100\% \times (\text{Obj}^N - \text{Obj}^H) / \text{Obj}^H$, Gap5 = $100\% \times (\text{Obj}^E - \text{Obj}^H)/\text{Obj}^H$, where Obj^N or Obj^E indicates the objective function value on the condition of $\tau = 0$ or $\tau = 2$. The average, largest and smallest values of Gap4 are −5.85%, 1.79%, and −21.06%, respectively. The average, largest, and smallest values of Gap5 are 2.71%, 15.34%, and −5.82%, respectively. From the perspective of the average number of customers included in each truck route, the computational results on the condition of " $\tau = 1$" indicate a better capacity utilization of trucks.

Impact of the number of drones carried by each truck

In the TDRP-SA model, each truck is paired with multiple drones. Multiple drones can be launched on a same arc in truck routes. Computational results on setting different numbers of drones carried by each truck are obtained. Referring to the computational results, the integrated cost usually decreases along with the number of drones carried by each truck increases. For practice, companies should balance some factors to decide the number of drones carried by each truck.

Impact of the truck velocity

The average velocity of each truck obviously decides the moving-LRLs. To make a comparison between different values of the average velocity of each truck, the value of the average velocity of each truck is assumed to be 1, 0.67, and 1.33, respectively. Computational results indicate that the average velocity of each truck affects the use of synchronization on arcs. We take the condition of "$v^t = 1$" as the benchmark. When the average velocity of each truck decreases, of 85 out of the 87 large-scale benchmark instances, the number of subtours increases; of 53 out of the 87 large-scale benchmark instances, the average number of subtours in each combination route increases; which indicates synchronization on arcs is used more frequently with a lower velocity of trucks. When the average velocity of each truck increases, of 67 out of the 87 large-scale benchmark instances, the number of subtours decreases; of 58 out of the 87 large-scale benchmark instances, the average number of subtours in each combination route decreases; which indicates synchronization on arcs is used less frequently with a higher velocity of trucks.

Impact of the drone direct delivery from the depot

When the drone direct delivery is not involved, the TDRP-SA is transformed to a variant. The ALNS-based heuristic is adjusted by simplifying the algorithm not to construct and optimize the drone route for direct delivery. Details of the heuristic solutions under the condition of the TDRP-SA variant without drone direct delivery are referred to Tables S19–S21. We introduce Gap6 to denote the percentage gap between the objective value of the TDRP-SA solution and that of the solution of the TDRP-SA variant without drone direct delivery. Not permitting drone direct delivery leads to six instances without feasible solutions. The heuristic can solve any one of the 81 large-scale benchmark instances in less than 5450 s, and the shortest and the average computation times are 467 and 2200 s, respectively. For 71 out of the 81 large-scale benchmark instance, the computation time for obtaining the solution of the TDRP-SA without drone direct delivery is more than the computation time for obtaining the solution of the TDRP-SA. For the 81 large-scale benchmark instance, the average, largest and smallest values of Gap6 are 8.76%, 25.98%, and 0.27%, respectively.

7 | LIMITATIONS AND EXTENSIONS OF THE TDRP-SA

Several assumptions can be improved, so that the TDRP-SA model is extended.

(i) Extending customer types. In the TDRP-SA model, customers are classified into TDCs and DRCs, and customers that cannot be accessible by drones are neglected. In practice, some customers, which are called truck customers in this section, can only be served by trucks. To ensure truck customers being modeled within the TDRP-SA framework, certain extension of the TDRP-SA is expected.

(ii) Relaxing synchronization on arcs. We introduce the synchronization on arcs based on the pair modality. The synchronization on arcs is rather strict. Except waiting for customers' time window opening, trucks or drones should not wait for each other at moving-LRLs. Drones are allowed to return to the depot with their paired trucks, or drones can return alone to the depot if their paired trucks return to the depot at the same time. Allowing a drone to return alone to the depot may relieve its truck from waiting for drone retrieval. Involving such a route that a drone launching from its truck at moving-LRLs and returning to the depot leads to an extension of the synchronization on arcs.

(iii) Adding launch/retrieval positions. We assume in the TDRP-SA that trucks not only serve as a moving hub for drones, but also make deliveries. Adding some nodes for trucks to make detour can derive a generic problem setting. Adding launch/retrieval positions into the TDRP-SA increases the number of nodes that can be visited by trucks. Meanwhile, in a truck route the selection of launch/retrieval positions is another important decision.

(iv) Relaxing one drone launch per arc. The assumption of "a single launch per drone and per arc" is included in the TDRP-SA model. However, drone multiple launches along one arc will lead to much better solutions and a much more complicated model. The TDRP-SA with drone multiple launches along one arc is an important variant.

8 | CONCLUSIONS

Considering the significant advancements in drone technologies for delivery systems, especially the developing technology that enables drones to be launched from or land on a moving truck without the need for the truck to stop, we formally define the TDRP-SA. The TDRP-SA has the characteristics of synchronization on arcs, time windows, classified customers, direct delivery, multiple trucks, and multiple drones carried by each truck. The synchronization on arcs permits trucks to dispatch and retrieve drones at moving-LRLs on arcs of truck routes. We develop a mixed integer nonlinear programming model to exactly describe the TDRP-SA. We propose a mathematical analysis method to select moving-LRLs and to estimate the arrival time of the drones at customers. To overcome the computational difficulties, we introduce linear piecewise functions to locate moving-LRLs, and we develop boundary models. To solve the TDRP-SA, we propose an ALNS-based heuristic. On the basis of the computational experiments, the effectiveness of the boundary models, the TDRP-SA model and ALNS-based heuristic are evaluated. For practitioners and researchers, this study provides a reference to demonstrate the route feasibility of delivery by truck–drone combinations and synchronization on arc modes. Owing to the increased interest in this field, more effective algorithms for the TDRP-SA are expected. In addition, the proposed model can be extended in several directions.

ACKNOWLEDGMENTS

This research was supported by the Research Grant from the National Natural Science Foundation of China (grant numbers 71672005, 71972007).

CONFLICT OF INTEREST

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

DATA AVAILABILITY STATEMENT

All instances are available in Mendeley Data (https://data.mendeley.com/datasets/cxc6p5x4ts/1).

REFERENCES

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, *52*(4), 965–981.

Alotaibi, K. A., Rosenberger, J. M., Mattingly, S. P., Punugu, R. K., & Visoldilokpun, S. (2018). Unmanned aerial vehicle routing in the presence of threats. *Computers & Industrial Engineering*, *115*, 190–205.

Boysen, N., Briskorn, D., Fedtke, S., & Schwerdfeger, S. (2018). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, *72*(4), 506–527.

Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: A survey from an operational research perspective. *OR Spectrum*, *43*, 1–58. https://doi.org/10.1007/s00291-020-00607-8

Boysen, N., Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, *271*(3), 1085–1099.

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, *123*, 105004.

Coutinho, W. P., Battarra, M., & Fliege, J. (2018). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*, *120*, 116–128.

Das, D. N., Sewani, R., Wang, J., & Tiwari, M. K. (2020). Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, *22*(9), 5772–5782. https://doi.org/10.1109/TITS.2020.2992549

de Freitas, J. C., & Penna, P. H. V. (2018). A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. *Electronic Notes in Discrete Mathematics*, *66*, 95–102.

de Freitas, J. C., & Penna, P. H. V. (2020). 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). Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. *Optimization Letters*, *15*(5), 1617–1648.

Dell'Amico, M., Montemanni, R., & Novellani, S. (2020). Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. *Annals of Operations Research*, *289*(2), 211–226.

Di Puglia Pugliese, L., & Guerriero, F. (2017). *Last-mile deliveries by using drones and classical vehicles*. In A. Sforza & C. Sterle (Eds.), Optimization and decision science: Methodologies and applications. ODS 2017 *Springer Proceedings in Mathematics & Statistics* (Vol. *217*). Springer.

Di Puglia Pugliese, L., Macrina, G., & Guerriero, F. (2020). Trucks and drones cooperation in the last-mile delivery process. *Networks*, *78*(4), 371–399.

Euchi, J., & Sadok, A. (2021). Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication*, *44*, 101236.

Gambella, C., Lodi, A., & Vigo, D. (2018). Exact solutions for the carrier–vehicle traveling salesman problem. *Transportation Science*, *52*(2), 320–330.

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.

Ha, Q. M., Deville, Y., Pham, Q. D., & Ha, 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., & Ha, 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 pick-up 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*, 107598.

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

Li, H., Wang, H., Chen, J., & Bai, M. (2020). Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodological*, *138*, 179–201.

Liu, Y., Liu, Z., Shi, J., Wu, G., & Pedrycz, W. (2020). Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(12), 7450–7465. https://doi.org/10.1109/TSMC.2020.2968839

Luo, Z., Liu, Z., Shi, J., Wang, Q., Zhou, T., & Liu, Y. (2018, June). The mathematical modeling of the two-echelon ground vehicle and its mounted unmanned aerial vehicle cooperated routing problem. In 2018 IEEE intelligent vehicles symposium (IV), Changshu, China, (pp. 1163–1170). IEEE.

Macrina, G., Pugliese, D. P., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, *120*, 102762.

Marinelli, M., Caggiani, L., Ottomanelli, M., & Dell'Orco, M. (2017). En route truck–drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems*, *12*(4), 253–261.

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*, 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.

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

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

Ribeiro, G. M., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, *39*(3), 728–735.

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.

Salama, M., & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, *114*, 620–642.

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

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

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265.

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.

Yurek, E. 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.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Li, H., Chen, J., Wang, F., & Zhao, Y. (2022). Truck and drone routing problem with synchronization on arcs. *Naval Research Logistics (NRL)*, *69*(6), 884–901. https://doi.org/10.1002/nav.22053