Optimizing an On-Demand Delivery Mode based on Trucks and Drones

Lu Zhen [1], Jiajing Gao [1*], Shuaian Wang [2], Gilbert Laporte [3,4], Xiaohang Yue [5]

[1] *School of Management, Shanghai University, Shanghai, 200444, China*

[2] *Faculty of Business, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

[3] *Department of Decision Sciences, HEC Montréal, Montréal, H3T 2A7, Canada*

[4] *School of Management, University of Bath, Bath, United Kingdom*

[5] *Lubar School of Business, University of Wisconsin-Milwaukee, Milwaukee, Wisconsin, 53201, USA*

* Corresponding author: Jiajing Gao.

Emails: *lzhen@shu.edu.cn (L. Zhen), jiajing_gao@shu.edu.cn (J. Gao), wangshuaian@gmail.com (S. Wang), gilbert.laporte@cirrelt.net (G. Laporte), xyue@uwm.edu (X. Yue)*

Abstract: We explore a novel on-demand delivery mode based on cooperation between trucks and drones. A fleet of trucks, each of which carries several drones, travels along a closed-loop route, and the drones are launched from the trucks to pick up (or deliver) ordered parcels from their origin (or to their destination). The fulfillment of an order (i.e., delivering the parcel from its origin to its destination) includes three steps: pickup by a drone, transport by a truck, and delivery by a drone. We investigate how to fulfill all the orders in one batch in order to minimize the total operational cost. We build a mixed-integer programming (MIP) model for this new on-demand delivery system in a network of multiple routes with transshipment. For drones, the assignment decision regarding the fulfillment stages for the orders and the location decision regarding the launching from and landing onto trucks are optimized by the proposed MIP model. An exact branch-and-price algorithm is designed to efficiently solve the model on large-scale instances. We validate the advantages of our algorithm in terms of computing time and solution quality through experiments on both artificial and real data. We validate the benefits of both implementing this new delivery mode and of allowing transshipments among routes, using a drone to serve multiple orders in one flying trip, and consolidating orders. We also investigate the influences of the number of drones, speed, endurance time, unit penalty cost, and the geographic distribution of orders on the system's operational cost.

Keywords: Cooperative delivery; on demand; trucks and drones; sidekicks; branch-and-price.

1 Introduction

The cooperative delivery mode based on trucks and drones, developed by the AMP Company and the University of Cincinnati in 2014 (Robinette, 2014), has attracted extensive academic and practical attention. In this mode, a truck departs from a depot, travels along a route, and carries one or more drones. Parcels ordered by customers are transported by the truck. A drone is launched from the truck when it approaches a customer's location, delivers the parcel to the customer, and then returns to the truck at another suitable location along the route. During the drone's delivery process, the truck does not necessarily need to wait for the drone and can continue its route. Many studies propose mathematical programming models and algorithms for designing the routes of trucks, along with the locations for drone launching or landing, to ensure that all customer deliveries are fulfilled. In the above cooperative delivery mode, the drone is one of various types of sidekicks. Unmanned vehicles or robots can also act as sidekicks that cooperate with the trucks. In addition, trucks, as primary vehicles, can be large vehicles such as ships or airplanes capable of moving in the Euclidean space. For example, Poikonen and Golden (2020) studied the mothership and drone routing problem, where the mothership can be a truck, a large ship, or airplane, among others. Figure 1 gives examples of other implemented truck-and-sidekick cooperative delivery mode systems. The system in the left part of Figure 1 shows that when a truck travels to a specific node, sidekicks of the truck carry out the distribution tasks while the truck waits for the sidekicks to return. In the delivery system shown on the right, the truck continues to travel while the sidekicks deliver parcels, and the sidekicks and truck meet at another node. We use the truck-and-drone system as an example to introduce proposed methodology, and the drones are equivalent to the sidekicks (unmanned vehicles or robots) in our study.

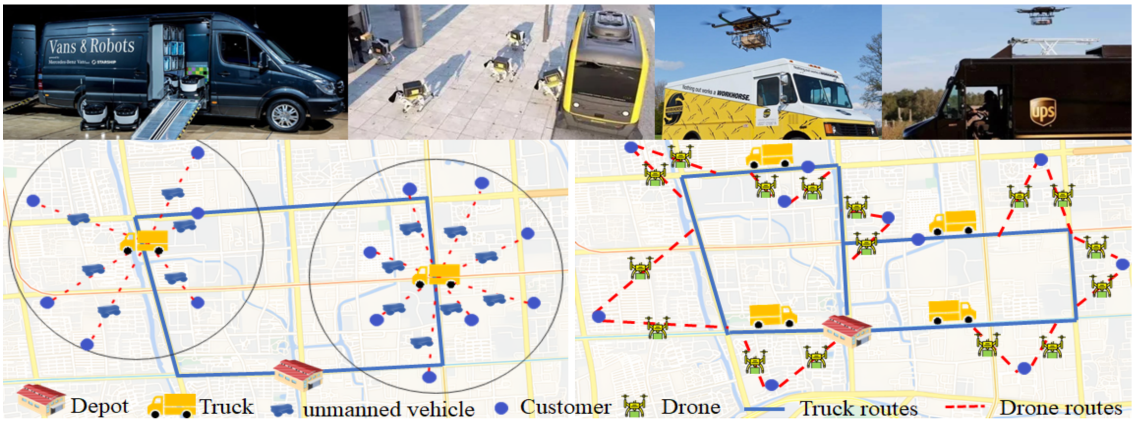
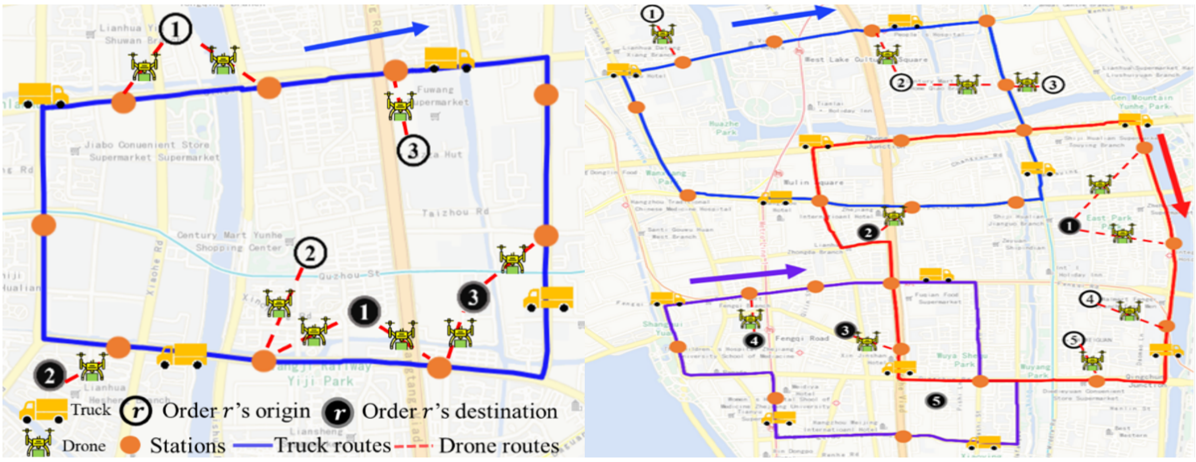


Figure 1: Truck-and-sidekick cooperative systems for last-mile delivery

The above cooperative delivery mode based on trucks and drones belongs to the last-mile delivery mode, which is to distribute the parcels from the depot to their customer locations. This last-mile delivery mode has been widely studied in recent years. However, the express delivery industry also contains another category, i.e., the on-demand delivery mode, which is to dispatch couriers (or vehicles, drones, etc.) to pick up a customer's parcel from its origin and then deliver it to its destination. The on-demand delivery mode can bring more profit for the delivery service providers than the last-mile delivery mode, and is becoming more and more popular in the express delivery industry. By using the similar infrastructure of the well-known truck-and-drone cooperative last-mile delivery systems, this study proposes a novel business mode for on-demand delivery, which is also based on the cooperation of trucks and drones, as shown in Figure 2. The infrastructure consists of a fleet of trucks, each of which carries one or more drones. However, a fleet of trucks travels along a specific closed-loop route like a bus tour route. A drone is launched from or lands on a traveling truck to either pick up a parcel from its origin to a truck or deliver a parcel from a truck to the order destination. The benefits of this novel mode include the following: (i) it combines the merit of the trucks that can undertake long-distance transits in a cost-efficient way with the merit of the drones that can respond to customer requests timely and overcome the restrictions of road traffic conditions; (ii) a fleet of trucks are operated in a bus tour-like way, which enables a drone on a truck can be timely dispatched to pick up an order's parcel from its origin, and then the drone that has picked up the parcel can transfer it to a truck so that it can be timely delivered to its destination. As the customers' demands for delivering their parcels are uncertain, the benefit of this bus tour-based on-demand delivery mode is similar to the benefit of the traditional bus mode in cities' public transit systems for satisfying passengers' uncertain travel demands.



Single route **Multiple routes with transshipment**

Figure 2: An on-demand delivery mode based on bus tour-like routes of trucks and drones

In this business mode, a platform is constructed to provide an on-demand delivery service to customers who require a parcel to be picked up from an origin and delivered to a destination. Customers release orders on the platform, and each order contains information on its origin, destination, and delivery due time; as shown in Figure 2, the orders' origins and destinations are denoted by the numbered circles with white color and black color, respectively, and the number in each circle denotes the index of the order related to the circle.

The order fulfillment process is executed as follows. A drone is chosen to fulfill an order released on the platform and is launched from a truck, flies to the origin, picks up the parcel, and returns to a truck, which is not necessarily the truck from which it was launched. The parcel is then transported by a truck along the closed-loop route. When the truck arrives at a suitable location near the order's destination, a drone is launched from the truck, flies to the destination with the parcel, delivers it, and then returns to the truck after fulfilling the delivery task. Thus, the fulfillment of an order on the platform consists of three steps: a flying trip by a drone for pickup, a land-transportation trip by a truck, and another flying trip by a drone for delivery. To efficiently operate this novel on-demand delivery mode, the platform must carefully schedule the drones so that the orders are fulfilled before their due time.

As is typical of on-demand service platforms, the delivery platform examined in this paper handles continuously arriving orders batch by batch. For each batch of orders accumulated during a decision period, the platform must decide: (i) how to assign orders to drones to fulfill pickup and delivery; (ii) how to assign orders to trucks to fulfill the land transportation process; and (iii) the locations (times) from which a drone is launched or where it lands when fulfilling its pickup and delivery trips. This study proposes a mixed integer linear programming (MILP) model to make these decisions; and the model's objective is to minimize the flying cost of drones and the penalty cost of delayed delivery for orders. Because the fleet of trucks is operated in the bus-tour mode and travels at a constant speed, the truck-related cost is a fixed one, which is not influenced by the decisions made by the model since the truck related cost is not included in the objective. In addition, our proposed model is an operational-level decision model for one batch of orders; in the rolling horizon with multiple batches, our model could be solved iteratively at a certain frequency to make plans for one batch at a time. As an operational-level decision model, its objective does not consider long-term fixed costs such as the deployment cost of drones and trucks. As aforementioned, our proposed model should support the platform to efficiently schedule the drones so that the orders are fulfilled before their due time; therefore, the model aims to minimize the sum of the drones' flying costs and the penalty cost due to the late delivery. The timely delivery before the due time is critical in the on-demand delivery service industry; thus the penalty cost of delayed delivery for orders is contained in our model's objective so as to improve the service level of the on-demand delivery processes fulfilled by our proposed new business mode.

To support the on-demand service platform and make timely decisions, this study also develops an exact branch-and-price algorithm to solve the model efficiently. We then extend the model, which is oriented to one route, to a more generic context in which there are multiple routes deployed with different numbers of trucks. Two routes may be connected by hub stations through which parcels can be transshipped between trucks along two routes, as shown in Figure 2. By considering the multiple routes for the transshipment hubs, we further propose a more advanced model to support on-demand delivery platforms that cover a larger customer service area. We conduct experiments to validate the effectiveness of the proposed models and the algorithm. Sensitivity analyses are also performed to derive insights for managers aiming to implement this mode of on-demand delivery.

The remainder of this paper is organized as follows. In Section 2, we review related studies, and in Section 3, we study the problem in the context of a single route. In Section 4, we propose a model for a single route. An exact algorithm is developed in Section 5 to solve the model. Experimental results are reported in Section 6. The model is extended to the multiple route context in Section 7. Section 8 concludes the paper.

2 Related works

The mode proposed in this study is essentially a type of truck-and-sidekick cooperative delivery, which has been widely studied in recent years (Chung et al., 2020; Li et al., 2021; Macrina et al., 2020). We examine two main streams of research. One considers the truck-and-drone (or truck-and-sidekick) cooperative delivery mode, and the other considers autonomous on-demand delivery modes based on drones or other unmanned vehicles. As our methodology involves mathematical programming, our literature review focuses on studies that propose mathematical programming models and algorithms.

2.1 Truck-and-drone (or truck-and-sidekick) cooperative delivery

First, we examine the widely studied truck-and-drone cooperative delivery mode, in which the drones can be considered to be sidekicks (robots). Since Murray and Chu (2015) conducted their initial research into the truck-and-drone cooperative delivery mode, many scholars have examined related issues. They mainly focus on the routing problem during last-mile delivery. In terms of the number of drones and trucks, these studies mainly assess vehicle routing with a single truck and single drone (Agatz et al., 2018; Boccia et al., 2021; Poikonen et al., 2019; Roberti and Ruthmair, 2021), with a single truck and multi-drones (Bruni et al., 2022； Carlsson and Song, 2018; Chen et al., 2021; Kang and Lee, 2021; Murray and Raj, 2020; Salama and Srinivas, 2022), and with multi-trucks and multi-drones (Chen et al., 2021; Masmoudi et al., 2022; Schermer et al., 2019; Tamke and Buscher 2021; Zhen et al., 2023). In the majority of the related research, heuristics are used and can solve their problems effectively. However, in recent years scholars have increasingly considered exact algorithms. Wang and Sheu (2019) proposed an MILP model and designed a branch-and-price algorithm a vehicle routing problem with drones. Tamke and Buscher (2021) developed an MILP model and designed a branch-and-cut algorithm for a vehicle routing problem with drones, in which there are two different time-oriented objective functions. Zhen et al. (2023) proposed an MILP and a branch-price-and-cut algorithm to solve the delivery routes of trucks and drones. Yin et al. (2023) built an MILP model that considers time window constraints to solve delivery routing problems for trucks and drones, proposed an enhanced branch-price-and-cut algorithm, and applied a bounded bidirectional labeling algorithm to solve challenging pricing problems. Xia et al. (2023) investigated the vehicle routing problem with load-dependent drones, and developed a branch-and-price-and-cut algorithm. Their proposed inequalities accelerate the convergence rate of the algorithm. Zhou et al. (2023) studied the two-echelon vehicle routing problem with drones, which is a new variant of the truck and drone delivery problem. To solve this problem, they proposed a branch-and-price algorithm and used a bidirectional labeling algorithm to solve the pricing problem. Li and Wang (2023) defined the truck-drone routing problem with time windows and proposed a branch-price-and-cut algorithm. In computational experiments, this algorithm optimally solves instances with up to 50 customers. Based on the above analysis, we design an exact algorithm for our model. In addition, Gao et al. (2023b) proposed a truck and drone scheduling decision-making method in a cooperative delivery and pickup system. It constructs a MILP model to plan the truck and drone routes in the fleet to meet the customers' pickup or delivery demands within a specified time, and also designs a hybrid algorithm to solve the model. However, the closed-loop related aspects of this study are not covered in Gao et al. (2023b). Hokama et al. (2024) revolve around the traveling salesman problem (TSP) with multiple flying sidekicks, which focuses on making deliveries to a group of customers using a truck and a drone, with the objectives of minimizing the time to service all customers and return the vehicle to the warehouse. Hokama et al. (2024) involve the truck and drone working in tandem and additionally solve the problem by fixing the initial tour sequence to insert the drone trips. However, our study considers the multiple closed-loop routes with transshipment between routes, while the TSP studied by Hokama et al. (2024) contains one route. Last but not least, the most essential difference between our study and the above related works lies in that our study is oriented to the on-demand delivery mode, while the related works studied the problems of last-mile delivery mode. The difference between the on-demand delivery mode (as shown in Figure 2) and the last-mile delivery mode (as shown in Figure 1) has been elaborated in the introduction section. Regarding the modeling of the problem, the former mode is much more complex than the latter mode; for example, the former one needs to consider the decision on the assignment of orders to the resources (i.e., trucks, and drones), while the concept of 'orders' does not exist in the latter mode.

Algorithm design is an important aspect in this research domain. This study proposes an exact algorithm, while some related works have also proposed exact algorithms. Table 1 provides a detailed comparison between our study and several of the above studies that have also developed some exact algorithms. It can be noticed that some literatures are delivery problems and do not consider the pickup demand. Our study considers on-demand distribution in which the process of pickup followed by delivery is more complex than the direct distribution process. In addition, the trucks in our study are parked by stations under fixed routes like buses, and the drones are launched and returned to the trucks at the stations.

Table 1: Overview of exact algorithms in recent academic literature

Literature	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Wang and Sheu (2019)	√	√		m/n	√			branch-and-price
Tamke and Buscher (2021)	√			m/m				branch-and-cut
Li and Wang (2023)		√		m/n				branch-price-and-cut
Xia et al. (2023)	√	√		m/n	√			branch-and-price-and-cut
Yin et al. (2023)				m/m				branch-price-and-cut
Zhen et al. (2023)				m/m				branch-price-and-cut
Zhou et al. (2023)				m/m				branch-and-price
This paper	√		√	m/n	√	√	√	branch-and-price

Notes: **(1)** indicates that drone-truck assignments are not fixed, i.e., a drone can be launched and returned in different trucks. **(2)** indicates that drones can carry multiple orders simultaneously; it is noted that this study is oriented to an on-demand instant delivery service, in which a drone is dedicated to picking up or delivering an order's parcel; therefore, this study does not consider this feature. **(3)** denotes that it considers both the pickup and the delivery. **(4)** denotes number of trucks/number of drones. **(5)** indicates the availability of other stations. **(6)** represents that truck routing decisions are given exogenously. **(7)** indicates that the transshipment of the delivered parcels is considered. **(8)** represents the solution method.

2.2 On-demand delivery modes based on drones, robots, or unmanned vehicles

The second related research stream concerns autonomous on-demand delivery modes based on drones, robots, or other types of unmanned vehicles. The application of unmanned devices in the field of on-demand instant delivery and logistics has attracted increasing attention from both academia and practitioners in recent years. Drones, as a type of unmanned device, have matured in terms of technology and become increasingly cost-effective, leading to retailers' logistics networks becoming further decentralized while they deliver products at a faster rate than before (Perera et al., 2020). Liu (2019) examined how to dispatch delivery drones in real time so that all orders can be delivered efficiently by studying a meal pickup and delivery routing problem for drones, and proposed a progressive algorithm for drone routing, scheduling, and order delivery in a dynamic real-time operation environment. Liu (2023) proposed a new business model for on-demand parcel delivery services using drones, and evaluated different modeling and solution approaches to drone routing problems that support service operation decisions. Gu et al. (2023) applied a Markov decision process methodology to study a dynamic truck-drone routing problem for executing delivery and pickup tasks in an on-demand logistics system, and proposed a heuristic solution framework. In addition, Dayarian et al. (2020) and Chen et al. (2022) considered same-day on-demand delivery based on vehicles and drones. Dayarian et al. (2020) introduced a system that uses drones and trucks to deliver goods to the doorstep on the same day. Unlike the previously studied traditional delivery modes, they considered a scenario in which the delivery trucks are regularly resupplied by drones. Chen et al. (2022) applied a deep learning methodology to study a vehicle and drone-based on-demand distribution problem, in which vehicles and drones are dynamically scheduled to deliver goods to customers before the deadline, while some customer requests can also be declined.

This study differs significantly from the above two research streams. In terms of the truck and drone (or sidekick) cooperative delivery mode, the trucks are operated in a bus-tour-based manner, while few studies in the related domain have considered the bus tour-like operations for trucks. In addition, in the literature, drones are usually dedicated to one truck or one route, on which a fleet of trucks travel; however, the drones are not necessarily dedicated to one truck in this study. Another difference is that most of the related works do not have the concepts of "order's origin" and "order's destination", which are the core concepts in our problem; however, in the literature, the customers' requests are satisfied by the deliveries from a depot (or a store, restaurant, etc.) to their locations.

This study contributes to the literature in the following aspects: (i) To our best knowledge, this paper is the first study to propose an on-demand delivery service mode in the context of cooperation between trucks and drones (or sidekicks); it is also the first study to combine the widely studied truck-and-drone mode and the well-known bus operation mode, so that their relative merits could be inherited in our proposed new on-demand delivery mode. (ii) This study proposes a decision model that considers a comprehensive set of factors such as there are multiple truck routes with transshipment; a drone can transfer among different routes, different trucks, and can execute both the pickup and the delivery tasks for customers; one truck can carry (be landed on by) multiple drones; each customer order has an origin and a destination. (iii) For such a complex decision problem, we design a novel exact algorithm by proposing some new tactics (as well as some rigorous proofs) for accelerating the solution process, which also contributes to the optimization-based studies in the domain of drones-based delivery in terms of algorithmic methodology.

3 Problem description

We study an on-demand delivery mode based on trucks and drones in the context of a delivery service platform. Customers release orders on the platform, and each order contains information about its origin's location, its destination's location, and the latest time for delivering the parcel for the order at the destination. A fleet of trucks and a fleet of drones are operated by the platform to fulfill customers' orders. The trucks travel along a closed-loop route, which is operated in a bus-tour mode (Zhen et al., 2020; Zeng et al. 2022; Chen et al., 2023). There is a certain distance between each pair of trucks traveling along the route, which is measured as the total length of the closed-loop route divided by the number of deployed trucks. Each truck may carry one or several drones. A drone is not dedicated to a single truck, which implies that a drone can be launched from a truck to execute a task and then land either on the same or any other truck after fulfilling its task.

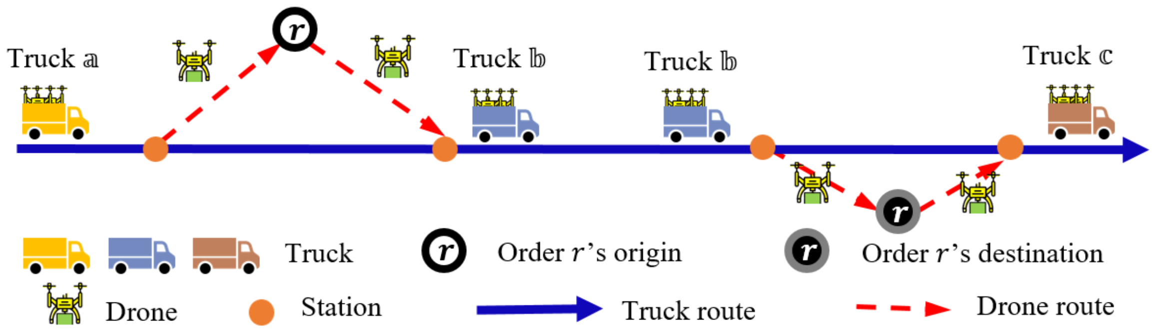


Figure 3: An on-demand delivery of Order r based on bus tour-like routes of trucks and drones

Figure 3 shows an example of this on-demand delivery process for Order r, which is to transport the order's parcel from its origin to its destination. The fulfillment for each order contains three steps: (i) Pickup step by drone: a drone should first be assigned to pick up the parcel of Order r at its origin. As shown in Figure 3, the drone is launched from truck 𝕒, flies to the order's origin, picks up the parcel, then flies back and lands on truck $\mathbb{b}$. Note that the two trucks $\mathbb{a}$ and $\mathbb{b}$ are not necessarily identical; (ii) Transport step by truck: the parcel is then transported by truck $\mathbb{b}$ to a location that is near the order's destination; (iii) Delivery step: a drone on truck $\mathbb{b}$ is then launched from the location carrying the parcel, and flies to the order's destination. After executing the delivery stage, the drone returns to truck $\mathbb{c}$, which is also not necessarily truck $\mathbb{b}$. Note that the same drone does not necessarily execute the abovementioned pickup and delivery steps.

To operate the bus tour-like routes of trucks efficiently, especially in the context of multiple drones' simultaneous operations at a truck, the take-off and landing of multiple drones should be able to take place simultaneously while a truck is at a station. To this end, multiple takeoff and landing areas are constructed on each truck. These areas are separated by physical isolation facilities (such as partitions, railings, etc.), and only one drone is allowed to park, take off, and land in each area. The above way can effectively prevent drones from colliding with each other during takeoff and landing. In addition, different areas are set up on a truck for storing parcels for different destinations. When a drone needs to pick up a parcel stacked under some other parcels, some automated device inside a truck can move the upper parcels away and place the target parcel near the truck's landing platform for the drone to pick up the parcel.

When the platform receives a batch of orders, the following decisions must be made for each order: (i) the assignment of a drone on a truck to fulfill an order's pickup or delivery task; (ii) the location where an assigned drone is launched from a truck to fulfill an order's pickup or delivery task, and (iii) the location where a drone lands on a truck after it fulfills an order's pickup or delivery task. The trucks travel along the route at a constant speed and do not stop to wait for drones, so decisions on the aforementioned locations implicitly contain time-related decisions on drone launching and landing events.

Before proposing a mathematical model for making these decisions, some assumptions underlying this study should be clarified:

(1) The route of the trucks is given. Determining drones' launching and landing locations in the route is conducted in a discrete decision manner. It means that a set of stations (points) exists along the route. A drone should thus be launched from a truck or land on a truck when the truck arrives at one of the stations in the set.

(2) Trucks travel at a constant speed in a bus-tour mode along the route without waiting for drones. A truck may pause for a while at a station for a drone launching or landing operation, but the pause time is negligible. If the truck has not yet arrived at the station, the drone can hover in the air and wait.

(3) Each drone is assumed to serve one order during a single flight, for the ease of introducing the model and algorithm. It should be noted that this assumption can be relaxed by extending our model, which can be referred to some recent works that consider multi-customer-visits in one flying trip (Gao et

al., 2023b; Jiang et al., 2024).

4 Mathematical model

This section proposes an MIP model on drone scheduling in this new on-demand delivery system based on truck-and-drone cooperative mode. As aforementioned, our investigated on-demand service platform handles continuously arriving orders batch by batch. The proposed MIP model actually belongs to a type of batch model, which makes the decisions for each batch of orders accumulated during a decision period.

4.1 Notations

Before formulating the model, the following notations are introduced. For ease of understanding, we use Roman letters and Greek letters to denote the parameters and the decision variables, respectively.

Sets and indices:

K set of all trucks, indexed by k.

R set of all orders, indexed by r.

I set of stations on trucks' route, indexed by i, j.

$\tilde{\imath}_k$ index of an initial station to denote the location of truck k at the decision epoch, $\tilde{\imath}_k \in I$.

Parameters:

e_r latest time (due time) for delivering parcel at order r's destination.

c^F unit flying cost of a drone with regard to its flying time.

c^P unit penalty cost of late delivery for orders with regard to the length of the late delivery per order; the unit of this parameter is $/delayed minute.

t_{ir}^{D+}, t_{ir}^{D-} drones' flying time from vertex i to order r's origin, or destination, respectively.

n_k number of available drones carried by truck k at the decision epoch.

$\overline{m}$ maximum number of drones that can be carried by a truck.

d_{ik} number of drones that should be on truck k at station $i, i \in I \backslash \{\tilde{\imath}_k\}$ but unavailable for being assigned to fulfill current batch's orders because they have been reserved for the previous batches' orders, which have been arranged before the decision epoch.

v_{ik} time when truck k arrives at station i. It is estimated according to the location of truck k at the decision epoch, station i's location, and truck speed.

T^{max} endurance of a drone, i.e., the maximum flying time of a drone.

M a sufficiently large positive number.

Decision variables:

α_{irk} binary, equals 1 if an empty drone is launched from truck k at station i for picking up order r from its origin; otherwise 0.

γ_{irk} binary, equals 1 if a laden drone lands on truck k at station i after picking up order r; otherwise 0.

φ_{irk} binary, equals 1 if a laden drone is launched from truck k at station i for delivering order r to its destination; otherwise 0.

δ_{irk} binary, equals 1 if an empty drone lands on truck k at station i after delivering order r; otherwise 0.

ϑ_r^+, ϑ_r^- continuous, a drone's arrival time at order r's origin, or destination, respectively.

Based on the above definitions, n_k is the number of available drones carried by truck k at the decision epoch, and $\bar{m}$ is the maximum number of drones that can be carried by a truck. Thus, $n_k \leq \bar{m}$. d_{ik} is the number of drones that are on truck k at station $i, i \in I\backslash\{\tilde{\imath}_k\}$ but are unavailable for being assigned to fulfill the current batch's orders because they have been reserved for the previous batches' orders, which have been arranged before the decision epoch, we have $d_{ik} + n_k \leq \bar{m},\ i \in I\backslash\{\tilde{\imath}_k\},\ k \in K$. Thus, $d_{ik} \leq \bar{m} - n_k,\ i \in I\backslash\{\tilde{\imath}_k\},\ k \in K$.

Regarding the definition of the parameter d_{ik}, we provide the following explanation. Our model applies to one batch of orders, and will be applied in the context of a rolling horizon with multiple batches, which implies that it will be solved iteratively at a certain frequency to make plans for one batch at a time, and the plan made for the previous batch will impact the plan made for the current batch. The abovementioned frequency usually corresponds to a short decision period, e.g., five minutes, which means one batch of orders is accumulated during five minutes. However, the fulfillment process for the orders in the batch usually needs a duration that is much longer than the decision period (five minutes). More specifically, when making the plan for the current batch (e.g., it is 10:00 now), a drone can be idle on a truck but cannot be used in the plan for the current batch because it has been involved in the previous batch's plan, which was made at 9:55 and may have a relatively long duration (e.g., one hour). In the process of executing the previous batch's plan (from 9:55 to 10:55), the drone may be launched from the truck at 10:30; thus the drone is idle now but unavailable for making plans for the current batch. This is the reason why we define a parameter d_{ik} as the number of drones that are on truck k at station i but are unavailable for being assigned to fulfill the current batch's orders.

4.2 Model formulation

The model is formulated to minimize the variable cost incurred by fulfillment of a batch of orders. The objective mainly includes the flying cost of drones and the penalty cost of delay delivery for orders. Because the fleet of trucks are operated in the bus-tour mode and travel in a constant speed, the truck-related cost is a fixed one, which is not influenced by the decisions in the model. The truck related cost is not included in the objective.

[$\mathcal{M}\mathbf{0}$] Minimize $\sum_{r\in R}\left\{c^F\sum_{k\in K}\sum_{i\in I}\left[t_{ir}^{D+}(\alpha_{irk}+\gamma_{irk})+t_{ir}^{D-}(\varphi_{irk}+\delta_{irk})\right]+c^P(\vartheta_r^- - e_r)^+\right\}$(1-1)

subject to:

$\sum_{k\in K}\sum_{i\in I}\alpha_{irk}=1 \qquad r\in R \qquad (1\text{-}2)$

$\sum_{k\in K}\sum_{i\in I}\gamma_{irk}=1 \qquad r\in R \qquad (1\text{-}3)$

$\sum_{k\in K}\sum_{i\in I}\varphi_{irk}=1 \qquad r\in R \qquad (1\text{-}4)$

$\sum_{k\in K}\sum_{i\in I}\delta_{irk}=1 \qquad r\in R \qquad (1\text{-}5)$

$\sum_{i\in I}\gamma_{irk}=\sum_{i\in I}\varphi_{irk} \qquad r\in R,\ k\in K. \qquad (1\text{-}6)$

Constraints (1-2)–(1-5) state that for fulfilling order r, there should exist four related stations, i.e., the station where a drone is launched from a truck to pick up order r, the station where a drone lands on a truck after picking up order r, the station where a drone is launched from a truck to deliver order r, and the station where a drone lands on a truck after delivering order r. Constraints (1-6) connect the two drone trips (i.e., the pickup trip and the delivery trip) for fulfilling one order (i.e., order r), which may be executed by two different drones but are surely related to one truck (i.e., truck k); Constraints (1-6) state that the order r is transported by truck k.

$\sum_{r\in R}\alpha_{\tilde{\imath}_k rk}\le n_k \qquad k\in K \qquad (1\text{-}7)$

$\sum_{r\in R}\alpha_{irk}\le \bar{m}-d_{ik} \qquad i\in I\backslash\{\tilde{\imath}_k\},\ k\in K \qquad (1\text{-}8)$

$\sum_{r\in R}\varphi_{irk}\le \bar{m}-d_{ik} \qquad i\in I,\ k\in K \qquad (1\text{-}9)$

$\sum_{r\in R}\gamma_{irk}\le \bar{m} \qquad i\in I,\ k\in K \qquad (1\text{-}10)$

$\sum_{r\in R}\delta_{irk}\le \bar{m} \qquad i\in I,\ k\in K. \qquad (1\text{-}11)$

Constraints (1-7) ensure that the number of drones launched from truck k at the current station $\tilde{\imath}_k$ to fulfill this batch's orders should be no greater than the number of available drones on the truck. Similarly, Constraints (1-8) ensure that the number of drones launched from truck k at other stations is no greater than the number of available drones. Constraints (1-9) is similar to Constraints (1-8); both of them are about the drones' launching stage of the two types of trips (pickup trip and delivery trip). Constraints (1-10) and (1-11) are about the drones' landing stage of the two types of trips; they ensure that the number of drones on each truck does not exceed the maximum value.

$\vartheta_r^+\ge v_{ik}+t_{ir}^{D+}-M(1-\alpha_{irk}) \qquad i\in I, r\in R, k\in K \qquad (1\text{-}12)$

$\vartheta_r^+\le v_{ik}+t_{ir}^{D+}+M(1-\alpha_{irk}) \qquad i\in I, r\in R, k\in K \qquad (1\text{-}13)$

$\vartheta_r^+\le v_{ik}-t_{ir}^{D+}+M(1-\gamma_{irk}) \qquad i\in I, r\in R, k\in K \qquad (1\text{-}14)$

$\vartheta_r^-\ge v_{ik}+t_{ir}^{D-}-M(1-\varphi_{irk}) \qquad i\in I, r\in R, k\in K \qquad (1\text{-}15)$

$\vartheta_r^-\le v_{ik}+t_{ir}^{D-}+M(1-\varphi_{irk}) \qquad i\in I, r\in R, k\in K \qquad (1\text{-}16)$

$\vartheta_r^-\le v_{ik}-t_{ir}^{D-}+M(1-\delta_{irk}) \qquad i\in I, r\in R, k\in K \qquad (1\text{-}17)$

$\vartheta_r^+ \leq \vartheta_r^-$ $\quad r \in R \quad$ (1-18)

$\sum_{k\in K}\sum_{i\in I} t_{ir}^{D+}\alpha_{irk} + \sum_{k\in K}\sum_{i\in I} t_{ir}^{D+}\gamma_{irk} \leq T^{max}$ $\quad r \in R \quad$ (1-19)

$\sum_{k\in K}\sum_{i\in I} t_{ir}^{D-}\varphi_{irk} + \sum_{k\in K}\sum_{i\in I} t_{ir}^{D-}\delta_{irk} \leq T^{max}$ $\quad r \in R \quad$ (1-20)

$\sum_{i\in I} v_{ik}\gamma_{irk} \leq \sum_{i\in I} v_{ik}\varphi_{irk}$ $\quad r \in R,\ k \in K \quad$ (1-21)

Constraints (1-12) and (1-13) are used to calculate the order r's pickup time ϑ_r^+ as $v_{ik} + t_{ir}^{D+}$ if α_{irk} equals one. Constraints (1-14) connect the order r's pickup time and the return time of the drone, which executes the pickup trip for the order, if γ_{irk} equals one. It should be noted that the drone is allowed to wait for the truck on which the drone is planned to land; thus the time v_{ik} when the truck arrives at the station where the drone lands on the truck is no earlier than the time $\vartheta_r^+ + t_{ir}^{D+}$ when the drone arrives at the above station. Constraints (1-15)–(1-17) are similar to Constraints (1-12)–(1-14); the difference lies in that the former group of constraints are formulated for the delivery trips while the latter group of constraints are formulated for the pickup trips. Constraints (1-18) guarantee that the delivery time is no earlier than the pickup time for each order. Constraints (1-19) and (1-20) ensure that the flying time of the drone does not exceed the endurance time of the drone. Constraints (1-21) ensure that the time at which an order's parcel arrives at a truck after being picked up from its origin is no later than the time at which it departs from the truck for being delivered to its destination for the order.

$\alpha_{irk}, \gamma_{irk}, \varphi_{irk}, \delta_{irk} \in \{0,1\}$ $\quad i \in I, r \in R, k \in K \quad$ (1-22)

$\vartheta_r^+, \vartheta_r^- \geq 0$ $\quad r \in R \quad$ (1-23)

Constraints (1-22) and (1-23) define decision variables.

Objective (1-1) contains a nonlinearity, $(\cdot)^+$, and needs to be linearized. Define the decision variables ϕ_r^+. If the value of $\vartheta_r^- - e_r$ is positive, ϕ_r^+ equals $\vartheta_r^- - e_r$; otherwise, ϕ_r^+ equals zero. The mathematical model is formulated as follows.

[$\boldsymbol{\mathcal{M}1}$] Minimize $\sum_{r\in R}\{c^F \sum_{k\in K}\sum_{i\in I}[t_{ir}^{D+}(\alpha_{irk} + \gamma_{irk}) + t_{ir}^{D-}(\varphi_{irk} + \delta_{irk})] + c^P \phi_r^+\}$ (1-24)

subject to:

Constraints (1-2)–(1-23)

$\phi_r^+ \geq \vartheta_r^- - e_r$ $\quad r \in R \quad$ (1-25)

$\phi_r^+ \geq 0$ $\quad r \in R \quad$ (1-26)

In this delivery mode, there may be extreme situations where the drone flying time is short but the waiting time is long. This extreme example can be solved using the model $\mathcal{M}1$. Because each operation follows the logical relationship specified by the model, waiting for a long time does not directly violate these constraints. In addition, when calculating variables such as time, we need to handle them according to Constraints (1-12)–(1-21). However, if the waiting time for drones is too long, this will result in a long penalty time for the system, leading to a high total cost. Therefore, the decision made by $\mathcal{M}1$ can avoid similar extreme situations.

For the above batch model $\mathcal{M}1$, there exists a lower bound elaborated in the following proposition. This lower bound is used in our algorithm.

Proposition 1: A lower bound for model $\mathcal{M}1$ is calculated as: $LB_{\mathcal{M}1} = \sum_{r \in R} \left\{ c^F \cdot FT_r^{min} + c^P \cdot \left(DT_r^{min} - e_r\right)^+ \right\}$; here, $FT_r^{min} = 2 \cdot \left(\min_{i \in I} t_{ir}^{D+} + \min_{i \in I} t_{ir}^{D-}\right)$, $DT_r^{min} = \min_{i_1, i_2, i_3 \in I, k \in K} \left\{ \min_{k \in K} v_{i_1 k} + t_{i_1 r}^{D+} + t_{i_2 r}^{D+} + t_{i_3 r}^{D-} + t_{i_2 i_3}^{K} \right\}$ and $t_{i_2 i_3}^{K}$ is the trucks' travel time from vertex i_2 to i_3.

Proof: See Appendix A. ■

The remainder of this section simplifies the context of the problem and proposes additional bounds for the drones' flying cost in the model $\mathcal{M}1$. We simplify the context by ignoring the due time of orders; thus the penalty cost of delay delivery does not exist in the model's objective. Then the drones' flying cost is the objective of the model. The above mentioned bounds, which are introduced in the remainder of this section, could help the planners make some estimation of the drones' flying cost based on some preliminary information. In the section of computational experiments, these proposed bounds are also validated.

Since the fleet of trucks travel along a closed-loop route operated in a bus-tour mode, we assume a theoretically simplified context in which the truck route is a circle of radius r_1. Furthermore, the potential customers (i.e., their orders' origins and destinations) are also distributed on another circle of radius r_2. We assume the two circles have the same center and $r_2 > r_1$. Then we have the following proposition for estimating the drones' flying cost's range (i.e., lower and upper bounds) based on the above preliminary data of r_1 and r_2.

Proposition 2: For a theoretically simplified context with truck circle's radius as r_1 and order circle's radius as r_2, the calculation of the lower bound (denoted by $\widetilde{LB}_{drn}$) and of the upper bound (denoted by $\widetilde{UB}_{drn}$) for the flying cost of drones in model $\mathcal{M}1$ is summarized in two cases that consider or do not consider the drone's maximum flight time T^{max}:

(1) Without considering T^{max}, $\widetilde{LB}_{drn} = 4|R|c^F|r_2 - r_1|$, $\widetilde{UB}_{drn} = 4|R|c^F|r_2 + r_1|$.

(2) Considering T^{max}, $\widetilde{LB}_{drn} = 4|R|c^F|r_2 - r_1|$; for the upper bound, we have two cases: if $\frac{v^D T^{max}}{2\pi} \geq r_2 + r_1$, $\widetilde{UB}_{drn} = 4|R|c^F|r_2 + r_1|$; else, $\widetilde{UB}_{drn} = 2|R|c^F T^{max} v^D$, where, v^D is the speed of a drone.

Proof: See Appendix B. ■

As the model $\mathcal{M}1$ is a minimization model, some experiments are conducted in Section 6 to investigate whether the proposed lower bounds are tight by comparing them with the optimal results obtained by our proposed model and algorithm.

5 Exact algorithm

In order to solve model $\mathcal{M}1$ efficiently, we have designed an exact branch-and-price algorithm. Dynamic programming and lower bound formula based on calculus are used to accelerate the algorithm.

5.1 Set partitioning master model

Assume that $\mathcal{P}_r$ is a set of all possible service routes for order r. Each service route mainly includes four stations for fulfilling an order. A service route for order r can be described as: station 1 $\dashrightarrow$ the origin of order $r \dashrightarrow$ station 2 $\rightarrow$ station 3 $\dashrightarrow$ the destination of order r $\dashrightarrow$ station 4; here $\dashrightarrow$ denotes drone flying trips, while $\rightarrow$ denotes trucks' trips. Define P as a set of all service routes, $P = \bigcup_{r\in R}\mathcal{P}_r$. Let ξ_{p_r} be a binary decision variable, equal to one if route p_r is selected in solution, and zero otherwise. The parameter $\mathbb{c}_{p_r}$ is defined as the cost of route p_r. According to the original model's objective formula (1-1), $\mathbb{c}_{p_r}$ is calculated as follows:

$$\mathbb{c}_{p_r} = c^F \sum_{k\in K}\sum_{i\in I}\left[t_{ir}^{D+}\left(\alpha_{ip_rk} + \gamma_{ip_rk}\right) + t_{ir}^{D-}\left(\varphi_{ip_rk} + \delta_{ip_rk}\right)\right] + c^P \phi_{p_r}^{+} \qquad r \in R, p_r \in \mathcal{P}_r.$$

In the above formula, α_{ip_rk}, γ_{ip_rk}, φ_{ip_rk}, and δ_{ip_rk} are binary parameters for each route p_r. These parameters' definitions are similar to their previously defined meaning as variables. For example, if α_{ip_rk} equals one, this means that in route p_r, a drone is launched from truck k at station i to pick up order r from its origin. Due to limitation of space, the explanations on the binary parameters γ_{ip_rk}, φ_{ip_rk}, and δ_{ip_rk} are not elaborated here. Based on the above definition, the set partitioning master model is formulated as follows:

[**MP**] Minimize $\sum_{r\in R}\sum_{p_r\in\mathcal{P}_r}\mathbb{c}_{p_r}\xi_{p_r}$ (2-1)

subject to

$$\sum_{p_r\in\mathcal{P}_r}\xi_{p_r} = 1 \qquad r \in R \qquad (2\text{-}2)$$

$$\sum_{r\in R}\sum_{p_r\in\mathcal{P}_r}\alpha_{\tilde{\imath}_k p_r k}\xi_{p_r} \le n_k \qquad k \in K \qquad (2\text{-}3)$$

$$\sum_{r\in R}\sum_{p_r\in\mathcal{P}_r}\alpha_{ip_rk}\xi_{p_r} \le \bar{m} - d_{ik} \qquad i \in I\backslash\{\tilde{\imath}_k\},\ k \in K \qquad (2\text{-}4)$$

$$\sum_{r\in R}\sum_{p_r\in\mathcal{P}_r}\varphi_{ip_rk}\xi_{p_r} \le \bar{m} - d_{ik} \qquad i \in I,\ k \in K \qquad (2\text{-}5)$$

$$\sum_{r\in R}\sum_{p_r\in\mathcal{P}_r}\gamma_{ip_rk}\xi_{p_r} \le \bar{m} \qquad i \in I,\ k \in K \qquad (2\text{-}6)$$

$$\sum_{r\in R}\sum_{p_r\in\mathcal{P}_r}\delta_{ip_rk}\xi_{p_r} \le \bar{m} \qquad i \in I,\ k \in K \qquad (2\text{-}7)$$

$$\xi_{p_r} \in \{0,1\} \qquad r \in R, p_r \in \mathcal{P}_r. \qquad (2\text{-}8)$$

Objective (2-1) minimizes the cost of the routes used in the solution. Constraints (2-2) ensure that each order is served exactly once. Constraints (2-3)–(2-7) are similar to Constraints (1-7)–(1-11), and indicate that the number of drones used in the fulfillment of orders should not exceed the number of available drones. Constraints (2-8) define decision variables.

Due to existence of numerous feasible routes and the exponential growth of feasible routes, it is difficult to solve model MP directly. We define $\mathcal{P}_r'$ as a subset of feasible service routes for order r. Then a restricted master problem (RMP) model is constructed. The binary variable ξ_{p_r} in RMP is relaxed to a continuous variable, and the linear relaxation of RMP (LR-RMP) is constructed. Due to limitation of space, the model LR-RMP is not repeated here since it is similar to model MP. The LR-RMP differs from the MP in that "$p_r \in \mathcal{P}_r$" is replaced by "$p_r \in \mathcal{P}_r'$" and Constraints (2-8) is replaced with $\xi_{p_r} \geq 0,\ r \in R, p_r \in \mathcal{P}_r'$.

To generate a set of initial feasible routes (columns) for the RMP, a heuristic is suggested and elaborated in Appendix C. For the LR-RMP's constraints that correspond to Constraints (2-2)–(2-7), their dual variables are defined as π_r, η_k, ι_{ik}, κ_{ik}, λ_{ik} and ϖ_{ik}, respectively. In each iteration of the algorithm, the dual variables are used to generate new columns for pricing problem.

5.2 Pricing problem

A pricing problem is constructed to generate a column with a negative reduced cost, which is then added to the LR-RMP. The pricing problem in this study can be decomposed into $|R|$ independent pricing subproblems, each of which corresponds to an order and generates a feasible service route for the order. The pricing subproblem of order r is represented by PP_r and is formulated as follows:

[$\mathbf{PP}_r$] Minimize $\sigma_r = \mathbb{c}_{p_r} - \pi_r - \sum_{k\in K} \alpha_{\tilde{\imath}_k k}\eta_k - \sum_{i\in I\setminus\{\tilde{\imath}_k\}} \sum_{k\in K} \alpha_{ik}\iota_{ik} - \sum_{i\in I}\sum_{k\in K} \varphi_{ik}\kappa_{ik} - \sum_{i\in I}\sum_{k\in K} \gamma_{ik}\lambda_{ik} - \sum_{i\in I}\sum_{k\in K} \delta_{ik}\varpi_{ik}$ (3-1)

subject to:

$$\sum_{k\in K}\sum_{i\in I} \alpha_{ik} = 1 \quad (3\text{-}2)$$

$$\sum_{k\in K}\sum_{i\in I} \gamma_{ik} = 1 \quad (3\text{-}3)$$

$$\sum_{k\in K}\sum_{i\in I} \varphi_{ik} = 1 \quad (3\text{-}4)$$

$$\sum_{k\in K}\sum_{i\in I} \delta_{ik} = 1 \quad (3\text{-}5)$$

$$\sum_{i\in I} \gamma_{ik} = \sum_{i\in I} \varphi_{ik} \qquad k \in K \quad (3\text{-}6)$$

$$\vartheta_r^+ \geq v_{ik} + t_{ir}^{D+} - M(1-\alpha_{ik}) \qquad i \in I, k \in K \quad (3\text{-}7)$$

$$\vartheta_r^+ \leq v_{ik} + t_{ir}^{D+} + M(1-\alpha_{ik}) \qquad i \in I, k \in K \quad (3\text{-}8)$$

$$\vartheta_r^+ \leq v_{ik} - t_{ir}^{D+} + M(1-\gamma_{ik}) \qquad i \in I, k \in K \quad (3\text{-}9)$$

$$\vartheta_r^- \geq v_{ik} + t_{ir}^{D-} - M(1-\varphi_{ik}) \qquad i \in I, k \in K \quad (3\text{-}10)$$

$$\vartheta_r^- \leq v_{ik} + t_{ir}^{D-} + M(1-\varphi_{ik}) \qquad i \in I, k \in K \quad (3\text{-}11)$$

$$\vartheta_r^- \leq v_{ik} - t_{ir}^{D-} + M(1-\delta_{ik}) \qquad i \in I, k \in K \quad (3\text{-}12)$$

$$\vartheta_r^+ \leq \vartheta_r^- \quad (3\text{-}13)$$

$$\sum_{k\in K}\sum_{i\in I} t_{ir}^{D+}\alpha_{ik} + \sum_{k\in K}\sum_{i\in I} t_{ir}^{D+}\gamma_{ik} \leq T^{max} \quad (3\text{-}14)$$

$$\sum_{k\in K}\sum_{i\in I} t_{ir}^{D-}\varphi_{ik} + \sum_{k\in K}\sum_{i\in I} t_{ir}^{D-}\delta_{ik} \leq T^{max} \quad (3\text{-}15)$$

$\sum_{i\in I} v_{ik}\gamma_{ik} \leq \sum_{i\in I} v_{ik}\varphi_{ik}$	$k \in K$	(3-16)
$\alpha_{ik}, \gamma_{ik}, \varphi_{ik}, \delta_{ik} \in \{0,1\}$	$i \in I, k \in K$	(3-17)
$\vartheta_r^+, \vartheta_r^- \geq 0.$		(3-18)
$\phi_r^+ \geq \vartheta_r^- - e_r$		(3-19)
$\phi_r^+ \geq 0$		(3-20)

Objective (3-1) minimizes the reduced cost. The explanations of Constraints (3-2)–(3-20) are similar to those of Constraints (1-2)–(1-6), Constraints (1-12)–(1-23), and Constraints (1-25) and (1-26). Similar to Constraints (1-2)–(1-5), Constraints (3-2)–(3-5) indicate that in order to fulfill order r, there must be four associated stations. Constraints (3-6) connect two drone trips for fulfilling order r, which may be executed by two different drones but are surely related to one truck (i.e., truck k). Constraints (3-7) and (3-8) are used to calculate the order r's pickup time ϑ_r^+ as $v_{ik} + t_{ir}^{D+}$ if α_{ik} equals one. Constraints (3-9) connect the order r's pickup time and the return time of the drone, which executes the pickup trip for the order. Constraints (3-10)–(3-12) are similar to Constraints (3-7)–(3-9). The difference lies in that the former group of constraints are formulated for the delivery trips while the latter group of constraints are formulated for the pickup trips. Constraints (3-13) guarantee that the delivery time is no earlier than the pickup time for each order. Constraints (3-14) and (3-15) ensure that the flying time of the drone does not exceed its endurance time. Constraints (3-16) ensure that the time at which an order's parcel arrives at a truck after being picked up from its origin is no later than the time at which it departs from the truck to be delivered to its destination for the order. Constraints (3-17) and (3-18) define variables. Constraints (3-19) and (3-20) need to linearize the model.

5.3 Tactics for algorithmic acceleration

5.3.1 Dynamic programming

An exact dynamic programming algorithm is proposed to solve the pricing problem PP_r in a fast way. The pricing problem is to generate a service route for an order; the route mainly includes four stations. A service route for order r can be described as: station 1 ⇢ the origin of order r ⇢ station 2 → station 3 ⇢ the destination of order r ⇢ station 4; here ⇢ denotes drone flying trips, while → denotes trucks' trips. It should be noted that the four stations (stations 1, 2, 3 and 4) could be the same station or different stations. The decisions on stations 1 and 2 are mainly affected by the origin of the order, while the decisions on stations 3 and 4 are mainly affected by the destination of the order. In addition, station 1 influences the decision on which truck's drone serves this order; station 2 influences which truck the drone returns to after it picks up the order's parcel; and station 4 influences which truck the drone returns to after it delivers the order's parcel. Since the truck route is fixed, we can decide on the four stations of the drone route, and the truck route from station 2 to station 3 does not need to be decided. Here we split the remaining route into two stages for decision-making. The first stage is: station 1 ⇢ origin of order r ⇢ station 2, and the second stage is: station 3⇢ destination of order r ⇢ station 4. Let S_n be the set of states in the n^{th} stage, and two stations need to be decided in each stage. Stations 1 and 2 need to be decided in the first stage, and stations 3 and 4 need to be decided in the second stage. For order r, $S_1 = S_2 = \{\{i_1, i_2, k_1, k_2\} | i_1, i_2 \in I, k_1, k_2 \in K\}$, and the total number of states in each stage is $(|I|)^2(|K|)^2$. Define Z_{ns} as the s^{th} $(s \le (|I|)^2(|K|)^2)$ state in the n^{th} stage, $Z_{ns} \in S_n$, and $u_n(Z_{ns})$ as the decision at the s^{th} state in the n^{th} stage. Define $d(u_n(Z_{ns}))$ as the value of the reduced cost when the decision $u_n(Z_{ns})$ is taken in the n^{th} stage. For example, there are two trucks (k_1, k_2) with three stations (i_1, i_2, i_3). Since the truck routes are fixed, the sequence of the stations and truck numbers is also fixed. Without considering the constraints of the pricing problem PP_r, all the 36 states (i.e., $(|I|)^2(|K|)^2 = 2^2 3^2 = 36$) in the first stage are as follows: $\{i_1, i_1, k_1, k_1\}$, $\{i_1, i_1, k_1, k_2\}$, $\{i_1, i_1, k_2, k_1\}$, …, $\{i_3, i_3, k_2, k_1\}$, $\{i_3, i_3, k_2, k_2\}$. The state of the second stage can be obtained similarly. According to the constraints of the pricing problem PP_r, we delete infeasible states.

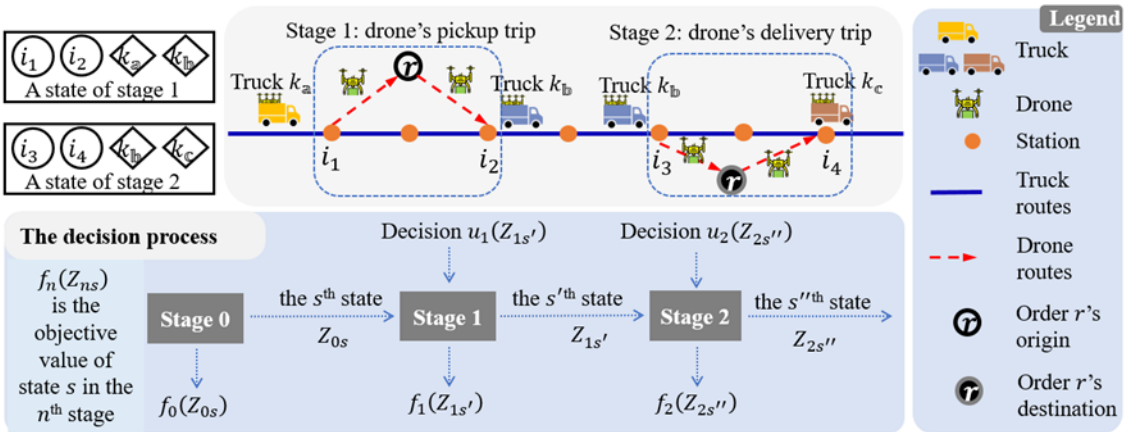


Figure 4: Dynamic programming algorithm for the pricing problem PP_r

Figure 4 depicts the dynamic programming algorithm for the pricing problem PP_r. Figure 4 illustrates a two-stage decision process for on-demand delivery of Order r based on bus tour-like routes of trucks and drones. The first stage involves a flying trip by a drone for pickup, and the second stage involves another flying trip by a drone for delivery. The trucks and drones in the figure together form a delivery system that involves multiple stages. The bottom decision process flowchart highlights the decision-making and state transfer process at each stage. The objective value $f_n(Z_{ns})$ of stage n is determined by the decision $u_n(Z_{ns})$, which starts from the initial state Z_{0s} in the virtual stage 0, and progresses to the final state $Z_{ns''}$ in Stage 2. This hierarchical structure shows how the dynamic programming approach solves the problem.

The calculation of the above $d(\cdot)$ follows Objective (3-1) and is explained as follows.

As the first stage is mainly related to the drone's pickup trip, the calculation of the above $d(\cdot)$ only includes the pickup trip related parts in Objective (3-1), i.e., the α and γ related parts. Then the $d(\cdot)$ is calculated as $d\big(u_1(Z_{1s})\big) = \begin{cases} c^F\big(t_{ir}^{D+} + t_{jr}^{D+}\big) - \eta_k - \lambda_{jk} & i,j,k \in Z_{1s}, i = \tilde{\iota}_k \\ c^F\big(t_{ir}^{D+} + t_{jr}^{D+}\big) - \iota_{ik} - \lambda_{jk} & i,j,k \in Z_{1s}, i \neq \tilde{\iota}_k \end{cases}$. Here $c^F\big(t_{ir}^{D+} + t_{jr}^{D+}\big)$ is the pickup trip related parts in the $\mathbb{c}_{p_r}$, i.e., the service route's cost for order r, which is included in Objective (3-1).

As the second stage is related to the drone's delivery trip, the calculation of the above $d(\cdot)$ includes the delivery trip related parts in Objective (3-1), i.e., the φ and δ related parts. The $d(\cdot)$ is calculated as $d\big(u_2(Z_{2s})\big) = c^F\big(t_{ir}^{D-} + t_{jr}^{D-}\big) + c^P\phi_r^+ - \kappa_{ik} - \varpi_{jk} - \pi_r$ where $i,j,k \in Z_{2s}$ and $\phi_r^+ = \big(v_{ik} + t_{ir}^{D-} - e_r\big)^+$. Here $c^F\big(t_{ir}^{D-} + t_{jr}^{D-}\big)$ is the delivery trip related parts in the $\mathbb{c}_{p_r}$, which is included in Objective (3-1). $c^P\phi_r^+$ is the penalty cost related parts in the $\mathbb{c}_{p_r}$. It should be noted that the "$-\pi_r$" in the Objective (3-1) is only included in the second stage for avoiding repeated counting.

Define $f_n(Z_{ns})$ represents the objective value under the state s in the n^{th} stage. The recursive equations for calculating $f_n(Z_{ns})$ are given as follows:

$$f_0(Z_{0s}) = 0 \qquad s = 1, \cdots, (|I|)^2(|K|)^2. \tag{4-1}$$

For the n^{th} stage ($n = 1, 2$), the objective under state s ($s = 1,2,\cdots,(|I|)^2(|K|)^2$) is calculated as:

$$f_n(Z_{ns}) = \min_{s'=1,\cdots,(|I|)^2(|K|)^2}\big\{f_{n-1}\big(Z_{n-1,s'}\big) + d\big(u_n(Z_{ns})\big)\big\} \qquad s = 1, \cdots, (|I|)^2(|K|)^2. \tag{4-2}$$

The dynamic programming process to solve pricing problem is provided in Appendix D.

5.3.2 Lower and upper bound formulas based on calculus approximation

We have designed a new method to accelerate the dynamic programming algorithm by using lower and upper bound formulas based on the calculus approximation principle (Carlsson and Song, 2018).

The lower bound formula is formulated as follows:

Proposition 3. At the first stage of dynamic programming, some required parameters are defined as follows. Based on Proposition 1, a_r is the proportion of the lower bound for order r to the lower bound for the model $\mathcal{M}1$, $a_r = \frac{c^F\cdot FT_r^{min} + c^P\cdot\left(DT_r^{min} - e_r\right)^+}{\sum_{r\in R}\left\{c^F\cdot FT_r^{min} + c^P\cdot\left(DT_r^{min} - e_r\right)^+\right\}}$. l_r is the route traveled by trucks in the PP$_r$ from stage zero to stage one and $length(l_r)$ is the length of the truck route l_r. Let σ_r be the reduced cost in the PP$_r$ from stage zero to stage one. A lower bound formula of reduced cost in the PP$_r$ from stage zero to stage one is formulated as follows:

$$LB_{\sigma_r} = \begin{cases} \frac{c^F}{v^D}\sqrt{2}length(l_r)a_r - \eta_k - \lambda_{jk} & i,j,k \in Z_{1s}, i = \tilde{\iota}_k \\ \frac{c^F}{v^D}\sqrt{2}length(l_r)a_r - \iota_{ik} - \lambda_{jk} & i,j,k \in Z_{1s}, i \neq \tilde{\iota}_k \end{cases}. \tag{4-3}$$

Proof: See Appendix E. ■

The upper bound formula is formulated as follows:

Proposition 4. At the first stage of dynamic programming, some required parameters are defined as follows. Based on Proposition 1, a_r is the proportion of the lower bound for order r to the lower bound for the model $\mathcal{M}1$, $a_r = \frac{c^F \cdot FT_r^{min} + c^P \cdot (DT_r^{min} - e_r)^+}{\sum_{r \in R}\left\{c^F \cdot FT_r^{min} + c^P \cdot (DT_r^{min} - e_r)^+\right\}}$. l_r is the route traveled by trucks in the PP_r from stage zero to stage one, and $length(l_r)$ is the length of the truck route l_r. Let σ_r be the reduced cost in the PP_r from stage zero to stage one. An upper bound formula of reduced cost in the PP_r from stage zero to stage one is formulated as follows:

$$UB_{\sigma_r} = \begin{cases} \frac{2c^P}{v^K} length(l_r) a_r - \eta_k - \lambda_{jk} & i,j,k \in Z_{1s}, i = \tilde{\imath}_k \\ \frac{2c^P}{v^K} length(l_r) a_r - \iota_{ik} - \lambda_{jk} & i,j,k \in Z_{1s}, i \neq \tilde{\imath}_k \end{cases}. \tag{4-4}$$

Proof: See Appendix F. ■

Combining dynamic programming with the principle of calculus approximation can effectively solve PP_r. Before performing dynamic programming, we calculate the lower bound (LB) and upper bound (UB) of the current state of stage m according to formulas (4-3) and (4-4). Define z as the reduced cost value from stage zero to stage one. If $LB < z$ and $UB > z$, the dynamic programming algorithm is executed at the current state; otherwise, the next state is selected. After traversing all the states, the optimal solution of PP_r is obtained. The dynamic programming's pseudocode is stated in Appendix G.

5.4 Branching and node selection strategy

The branching and node selection strategy used in this paper are designed as follows. We define $\xi_{p_r}^*$ as the optimal solution of LR-RMP. We use the nearly 0.5 rule commonly used in branching processes. Finding the value of $\xi_{p_r}^*$ closest to 0.5 among all $\xi_{p_r}^*$ is to find a special case that is in 'equilibrium' or 'in-between'. When $\xi_{p_r}^*$ is close to 0.5, this may indicate that the variable $\xi_{p_r}^*$ is at a critical point where the likelihood of identifying $\xi_{p_r}^*$ as either 1 or 0 is equal. By focusing on the value closest to 0.5, we are better able to understand and deal with these ambiguous or critical situations in branching strategies. Such branching strategies can be referenced in Forget and Parragh (2024) and Schrotenboer et al. (2019). If the value of $\xi_{p_r}^*$ is the closest to 0.5, we assign the service route p_r to the order r. Then, we divide the parent node into two child nodes. For the branch that generates the left child node, order r must select service route p_r. For the branch that generates the right child node, order r does not select route p_r.

For the left branch, which requires that order r must select service route p_r, we remove all plans of order r that are not service route p_r. We also require that PP_r does not have the newly generated service route for order r. For the pricing problems related to the orders other than r, no constraints are added. For the right branch, we remove service route p_r of order r. For PP_r, we require that the newly generated route cannot be service route p_r. For the pricing problems related to the orders other than r, no constraints are added. When all values of ξ_{p_r}, $\forall p_r \in \mathcal{P}_r$, $r \in R$ are integers, the optimal solutions of the RMP are integral and feasible.

For the node selection strategy, the depth-first-search rule and the best-lower-bound rule are combined. If the current node is not selected, the depth-first-search rule is used. And if the current node is pruned, the best-lower-bound rule is used.

5.5 Framework of the branch-and-price algorithm

Based on the above description on the core components in the proposed algorithm, this subsection summarizes the main framework of branch-and-price algorithm as follows:

Step 1: An initial solution is obtained by using a method described in Appendix C. A upper bound (UB) is set as the objective value of the initial solution and p^* is set as the initial feasible solution. UB is used to record the existing best objective value and p^* is used to record the existing best feasible solution.

Step 2: LP relaxation is obtained by solving the LR-RMP of the current parent node in Section 5.1–5.3. p' and $z(p')$ represent the LP solution and objective value of the current parent node, respectively. The lower bound (LB) represents the objective value of LR-RMP.

Step 2.1: If $z(p') < UB$ and p' is infeasible, go to Step 3.

Step 2.2: If $z(p') < UB$ and p' is feasible, update $UB = z(p')$, $p^* = p'$, record this node, and go to Step 4.

Step 2.3: If $z(p') \geq UB$ record this node, and go to Step 4.

Step 3: By branching the current parent into two child nodes as shown in Section 5.4. The left child ensures that order r selects service route p_r, while the right child ensures that order r does not select this service route p_r. If the left child node is chosen, the right child node is added to the set of nodes that currently have no record. and $LB = z(p')$. Then, go to Step 2.

Step 4: If all nodes are recorded, stop the algorithm; otherwise, select the node with the lowest LB among the currently unrecorded nodes and set it to the current node. After that, go to Step 2.

6 Computational experiments

We conduct computational experiments to evaluate the performance of the proposed model and algorithm, and we gained various management insights through sensitivity analyses. All experiments were performed on a workstation with two Xeon Gold 5218R CPUs running at 2.10 GHz with 32 GB of memory under Windows 10. The code was implemented in C#, which is in Visual Studio 2022. CPLEX version 22.1.0 was used to solve the models. The time limit for solving instances was one hour.

6.1 Experimental settings

The experimental context is Hangzhou, which is a representative city in terms of e-commerce in China.

The drone-related parameter setting is based on the operational data of Antwork, a technology company that develops urban aerial drones in Hangzhou and that obtained the first urban scene drone operation license issued by the Civil Aviation Administration of China in 2019. According to these data, the average speed of trucks is 30 km/h, the speed of drones is 45 km/h, and the endurance time of the drone is set to 0.44 hour. The parameter v_{ik} is determined by the truck route. According to the Antwork data, the full cost of a single five-km drone delivery trip is estimated to be about 7.6 CNY (Chinese Yuan), so we define the unit flying cost of a drone with regard to its flying time c^F as 68.3 CNY/h. Based on the research of Cheng and Lin (2024), The diesel truck fuel energy consumption rate is 7.1 miles per gallon. Assuming that the diesel price is 7 CNY/L and the speed of trucks is 30 km/h, the unit cost of a truck per unit of time is estimated as 83.59 CNY/h. Based on the above data, we can find that the use of drones can effectively reduce costs. Assuming c^P (i.e., unit penalty cost of late delivery) is set to 1000 CNY/h, $\bar{m}$ (i.e., maximum number of drones that can be carried by a truck) is set to 4, and the parameters e_r, n_k, and d_{ik} are randomly generated in instances. In the experiments, we generate six groups of problem instances with different sizes. We assume that the number of stations $|I|$ is 80, which are the road network nodes on the trucks' route. In addition, the number of trucks $|K|$ ranges from two to five, and the number of orders $|R|$ in one batch ranges from 15 to 50. The settings for $|K|$ and $|R|$ in the six different instance groups are listed in Table 1. Based on the Gongshu, Shangcheng and Xihu districts in Hangzhou city map (see Figure 5(a)), we form a bus tour-like route for trucks' travelling; the route is constituted by the Chaowang Road, Shaoxing Road, Zhaohui Road, East Ring Road, Qingchun Road, West Ring Road, Baisha Road and Baochu Road (see Figure 5(b)). In addition, 80 stations are evenly deployed on the truck route, and five trucks are evenly distributed in the truck route. Figure 5(c) and Figure 5(d) are the heat maps of the orders' origins and the orders' destinations, respectively. Two figures show the location distribution of the origins and destinations of orders accumulated over one hour. The colors in the heat map reflect the density of order origins and destinations. The darker the color, the more orders will be generated in the computational experiments. The order origin location areas are mainly restaurants and superstores; while, the order destination location areas are mainly residential neighborhoods.

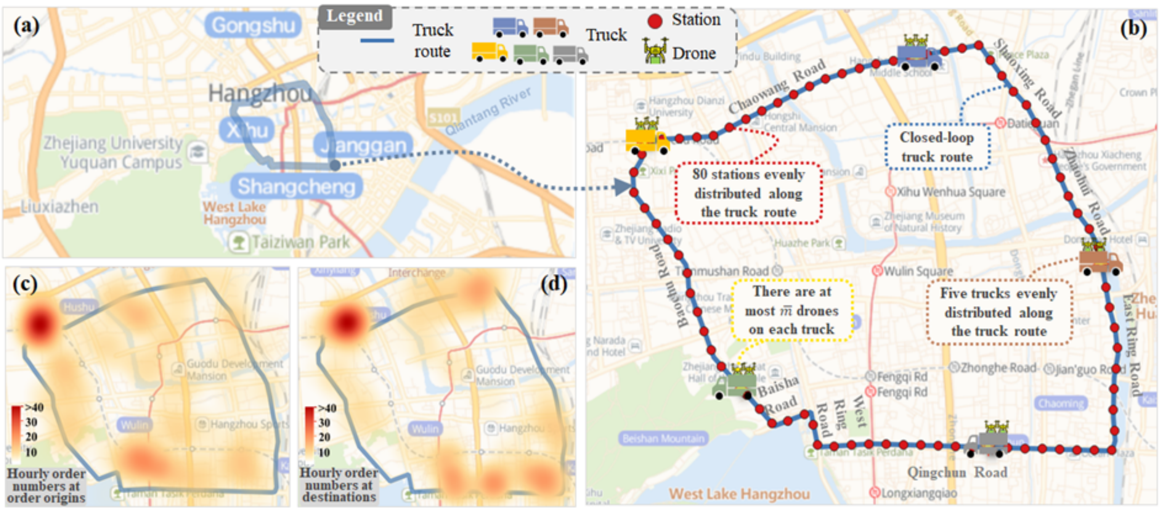


Figure 5: Demonstration of truck route, order origins and destinations in the case of Hangzhou city

Table 1: Parameters of problem scales for different instance groups

Instance groups	Number of orders ($\|R\|$) in one batch	Number of trucks ($\|K\|$)
ISG1	15	2
ISG2	20	2
ISG3	25	2
ISG4	30	3
ISG5	40	4
ISG6	50	5

6.2 Algorithmic performance

Before deriving managerial insights through computational experiments, we first investigate the performance of our algorithm. For the branch-and-price-based exact algorithm, the efficiency (computation time) of solving the pricing problem affects the overall performance of the whole algorithm. Therefore, we first investigate the performance of three strategies for solving the pricing problem. We then adopt the best strategy and investigate the overall performance of the whole algorithm (i.e., the branch-and-price-based exact algorithm) by comparing it to CPLEX.

6.2.1 Comparing three strategies for solving the pricing problem

In our proposed algorithm, we can use CPLEX to solve the pricing problem (PP) directly, and we can also use dynamic programming (elaborated in Section 5.3.1), or accelerated dynamic programming (elaborated in Section 5.3.2) to solve the PP. Thus, we compare these three strategies for solving the PP. In the algorithm solving, three different strategies are used to solve the PP, and the final objective value and algorithm solving time are obtained. The results of experiments based on small-scale and large-scale instances are shown in Tables 2 and 3, respectively. The objective values in Table 2 and Table 3 are not the objective values of the pricing problem, but the results obtained after the algorithm is solved.

Table 2: Comparing three strategies for solving PP (small-scale instances)

Instances	Objective value and computation time of different strategies for solving PP	Gap

Scale	ID	F_1	t_1(s)	F_2	t_2(s)	F_3	t_3(s)	$\Delta_{F_{21}}$	$\Delta_{t_{21}}$	$\Delta_{F_{31}}$	$\Delta_{t_{31}}$	$\Delta_{F_{32}}$	$\Delta_{t_{32}}$
ISG1	1	41.05	17	41.05	17	41.05	14	0.00%	0.00%	0.00%	-17.65%	0.00%	-17.65%
	2	42.15	20	42.15	20	42.15	15	0.00%	0.00%	0.00%	-25.00%	0.00%	-25.00%
	3	52.68	17	52.68	15	52.68	12	0.00%	-11.76%	0.00%	-29.41%	0.00%	-20.00%
	4	41.05	16	41.05	17	41.05	12	0.00%	6.25%	0.00%	-25.00%	0.00%	-29.41%
	5	52.68	20	52.68	17	52.68	12	0.00%	-15.00%	0.00%	-40.00%	0.00%	-29.41%
ISG2	6	59.43	41	59.43	37	59.43	24	0.00%	-9.76%	0.00%	-41.46%	0.00%	-35.14%
	7	59.77	31	59.77	31	59.77	25	0.00%	0.00%	0.00%	-19.35%	0.00%	-19.35%
	8	59.06	56	59.06	32	59.09	26	0.00%	-42.86%	0.00%	-53.57%	0.00%	-18.75%
	9	66.71	55	66.71	34	66.71	26	0.00%	-38.18%	0.00%	-52.73%	0.00%	-23.53%
	10	60.66	47	60.66	39	60.66	29	0.00%	-17.02%	0.00%	-38.30%	0.00%	-25.64%
ISG3	11	123.24	42	123.24	35	123.24	23	0.00%	-16.67%	0.00%	-45.24%	0.00%	-34.29%
	12	109.77	50	109.77	32	109.77	24	0.00%	-36.00%	0.00%	-52.00%	0.00%	-25.00%
	13	115.89	47	115.89	35	115.89	26	0.00%	-25.53%	0.00%	-44.68%	0.00%	-25.71%
	14	111.14	51	111.14	47	111.14	30	0.00%	-7.84%	0.00%	-41.18%	0.00%	-36.17%
	15	115.89	40	115.89	32	115.89	28	0.00%	-20.00%	0.00%	-30.00%	0.00%	-12.50%
Average								0.00%	-15.62%	0.00%	-37.04%	0.00%	-25.17%

Notes: (1) F_1, F_2, F_3 (or t_1, t_2, t_3) is the objective value of (or the computation time for obtaining) solutions solved by our proposed algorithm, in which the PP is solved by CPLEX, dynamic programming, and accelerated dynamic programming, respectively. (2) $\Delta_{F_{21}} = (F_2 - F_1)/F_1$, $\Delta_{F_{31}} = (F_3 - F_1)/F_1$, $\Delta_{F_{32}} = (F_3 - F_2)/F_2$, $\Delta_{t_{21}} = (t_2 - t_1)/t_1$, $\Delta_{t_{31}} = (t_3 - t_1)/t_1$, $\Delta_{t_{32}} = (t_3 - t_2)/t_2$.

These results show that all of the values of $\Delta_{F_\#}$ are zero, because both the dynamic programming and the accelerated dynamic programming are exact methods that can optimally solve the PP, like CPLEX. From the perspective of computation time, the values of $\Delta_{t_\#}$ demonstrate that the third strategy (i.e., accelerated dynamic programming) is the best, followed by the second strategy (i.e., dynamic programming). The computation times of accelerated dynamic programming and dynamic programming are shorter than that of CPLEX by about 37.04% and 15.62%, respectively.

To further validate the advantage of accelerated dynamic programming, the above experiments are also conducted on large-scale instances. The results are similar to those above. According to the values of $\Delta_{t_\#}$ in Table 3, the average computation time for dynamic programming is similar to that of CPLEX, while the time for accelerated dynamic programming is shorter than that of either the dynamic programming or CPLEX by about 32%–33% on average. Therefore, in the following experiments, we use accelerated dynamic programming in our branch-and-price algorithm.

Table 3: Comparing three strategies for solving PP (large-scale instances)

Instances		**Objective value and computation time of different strategies for solving PP**						**Gap**					
Scale	ID	F_1	t_1(s)	F_2	t_2(s)	F_3	t_3(s)	$\Delta_{F_{21}}$	$\Delta_{t_{21}}$	$\Delta_{F_{31}}$	$\Delta_{t_{31}}$	$\Delta_{F_{32}}$	$\Delta_{t_{32}}$
ISG4	1	163.81	74	163.81	68	163.81	45	0.00%	-8.11%	0.00%	-39.19%	0.00%	-33.82%
	2	177.96	88	177.96	82	177.96	52	0.00%	-6.82%	0.00%	-40.91%	0.00%	-36.59%
	3	172.11	87	172.11	67	172.11	52	0.00%	-22.99%	0.00%	-40.23%	0.00%	-22.39%
	4	162.71	78	162.71	55	162.71	42	0.00%	-29.49%	0.00%	-46.15%	0.00%	-23.64%
	5	177.35	73	177.35	60	177.35	42	0.00%	-17.81%	0.00%	-42.47%	0.00%	-30.00%

	6	296.23	194	296.23	190	296.23	128	0.00%	-2.06%	0.00%	-34.02%	0.00%	-32.63%
	7	289.29	413	289.29	130	289.29	80	0.00%	-68.52%	0.00%	-80.63%	0.00%	-38.46%
ISG5	8	286.70	121	286.70	162	286.70	100	0.00%	33.88%	0.00%	-17.36%	0.00%	-38.27%
	9	302.90	117	302.90	141	302.90	99	0.00%	20.51%	0.00%	-15.38%	0.00%	-29.79%
	10	302.63	99	302.63	162	302.63	98	0.00%	63.64%	0.00%	-1.01%	0.00%	-39.51%
	11	327.75	200	327.75	260	327.75	189	0.00%	30.00%	0.00%	-5.50%	0.00%	-27.31%
	12	335.85	493	335.85	372	335.85	231	0.00%	-24.54%	0.00%	-53.14%	0.00%	-37.90%
ISG6	13	334.96	243	334.96	288	334.96	187	0.00%	18.52%	0.00%	-23.05%	0.00%	-35.07%
	14	341.15	213	341.15	250	341.15	165	0.00%	17.37%	0.00%	-22.54%	0.00%	-34.00%
	15	343.00	267	343.00	299	343.00	202	0.00%	11.99%	0.00%	-24.34%	0.00%	-32.44%
Average								0.00%	1.04%	0.00%	-32.39%	0.00%	-32.79%

Notes: (1) F_1, F_2, F_3 (or t_1, t_2, t_3) is the objective value of (or the computation time for obtaining) solutions solved by our proposed algorithm, in which the PP is solved by CPLEX, dynamic programming, and accelerated dynamic programming, respectively. (2) $\Delta_{F_{21}} = (F_2 - F_1)/F_1$, $\Delta_{F_{31}} = (F_3 - F_1)/F_1$, $\Delta_{F_{32}} = (F_3 - F_2)/F_2$, $\Delta_{t_{21}} = (t_2 - t_1)/t_1$, $\Delta_{t_{31}} = (t_3 - t_1)/t_1$, $\Delta_{t_{32}} = (t_3 - t_2)/t_2$.

6.2.2 Comparing our algorithm with CPLEX

The solution quality of an algorithm is typically evaluated by comparing it with CPLEX. Table 4 illustrates the comparative results between our branch-and-price algorithm and CPLEX on small-scale instances, for which CPLEX can obtain an optimal solution within a reasonable time. The values of $\Delta_{F_{BP}}$ in Table 4 are zero, which validates the optimality of our proposed algorithm. By comparing the computation time t_{BP} and t_{CPLEX}, the time of our algorithm appears relatively stable, while the time of CPLEX varies significantly among instances.

Table 5 illustrates the comparative results on large-scale instances, which cannot all be solved by CPLEX within one hour, but can be solved by our branch-and-price algorithm within four minutes. The above experiments validate the efficiency of our branch-and-price-based exact algorithm, which can optimally solve the problem within a reasonable time.

Table 4: Algorithmic performance of the branch-and-price algorithm (small-scale instances)

Instances		**CPLEX**		**Branch-and-price**		
Scale	ID	F_{CPLEX}	t_{CPLEX}(s)	F_{BP}	t_{BP}(s)	$\Delta_{F_{BP}}$
	1	41.05	3	41.05	14	0.00%
	2	42.15	4	42.15	15	0.00%
ISG1	3	52.68	14	52.68	12	0.00%
	4	41.05	3	41.05	12	0.00%
	5	52.68	6	52.68	12	0.00%
	6	59.43	18	59.43	24	0.00%
	7	59.77	4	59.77	25	0.00%
ISG2	8	59.09	17	59.09	26	0.00%
	9	66.71	8	66.71	26	0.00%
	10	60.66	5	60.66	29	0.00%
	11	123.24	92	123.24	23	0.00%
	12	109.77	106	109.77	24	0.00%
ISG3	13	115.89	77	115.89	26	0.00%
	14	111.14	1583	111.14	30	0.00%
	15	115.89	79	115.89	28	0.00%
Average						0.00%

Notes: (1) F_{CPLEX} and F_{BP} denote the objective value of the solutions solved by CPLEX directly and the branch-and-price algorithm, respectively. And t_{CPLEX} and t_{BP} denote the computation time of CPLEX and the branch-and-price algorithm to solve the model, respectively. (2) $\Delta_{F_{BP}}= (F_{BP} - F_{CPLEX})/F_{CPLEX}$.

Table 5: Algorithmic performance of the branch-and-price algorithm (large-scale instances)

Instances		**CPLEX**		**Branch-and-price**		
Scale	ID	F_{CPLEX}	t_{CPLEX}(s)	F_{BP}	t_{BP}(s)	$\Delta_{F_{BP}}$
ISG4	1	163.81	>3600	163.81	45	0.00%
	2	177.96	>3600	177.96	52	0.00%
	3	172.11	>3600	172.11	52	0.00%
	4	162.71	>3600	162.71	42	0.00%
	5	177.35	>3600	177.35	42	0.00%
ISG5	6	296.23	>3600	296.23	128	0.00%
	7	289.29	>3600	289.29	80	0.00%
	8	286.70	>3600	286.70	100	0.00%
	9	302.90	>3600	302.90	99	0.00%
	10	302.63	>3600	302.63	98	0.00%
ISG6	11	327.75	>3600	327.75	189	0.00%
	12	335.85	>3600	335.85	231	0.00%
	13	334.96	>3600	334.96	187	0.00%
	14	341.15	>3600	341.15	165	0.00%
	15	343.00	>3600	343.00	202	0.00%
Average						0.00%

Notes: (1) F_{CPLEX} and F_{BP} denote the objective value of the solutions solved by CPLEX directly and the branch-and-price algorithm, respectively. And t_{CPLEX} and t_{BP} denote the computation time of CPLEX and the branch-and-price algorithm to solve the model, respectively. (2) $\Delta_{F_{BP}}= (F_{BP} - F_{CPLEX})/F_{CPLEX}$. (3) For the above instances, CPLEX cannot solve them within one hour. F_{CPLEX} denotes the objective value of a solution, which is obtained by CPLEX when the computation time reaches one hour.

6.2.3 Validating the lower bounds proposed in Proposition 2

As mentioned in Proposition 2, bounds are proposed based on a theoretically simplified context. These bounds could potentially be useful for practitioners to estimate the drones' fly cost, based on preliminary information. The quality of these bounds should be validated through experiments. As the model $\mathcal{M}1$ is a minimization problem, we conducted computational experiments to investigate whether the proposed lower bounds are tight by comparing them with the optimal results obtained by our algorithm. Table 6 illustrates the comparative results of the optimal solution's objective value as solved by our branch-and-price algorithm and the lower bound calculated according to Proposition 2. The average of the gap values $\Delta_{F_{BP}}$ in Table 6 is about 0.64%, which confirms that the lower bound of Proposition 2 is very close to the optimal solution value and can thus be used by practitioners to estimate the drones' flying cost, based only on preliminary information in a realistic environment.

Table 6: Comparing the optimal results obtained by our algorithm and the proposed lower bound

Instances		**LB**	**Branch-and-price**	
Radius ratio r_1/r_2	Num. of customers' orders $\lvert R\rvert$	F_{LB}	F_{BP}	$\Delta_{F_{BP}}$
1/2	30	181.98	182.01	0.02%
	40	242.64	242.66	0.01%
	50	303.3	303.32	0.01%
2/3	30	181.98	182.58	0.33%
	40	242.64	242.69	0.02%

	50	303.30	303.90	0.20%
3/4	30	181.98	182.63	0.36%
	40	242.64	242.75	0.05%
	50	303.30	305.07	0.58%
4/5	30	181.98	184.34	1.28%
	40	242.64	244.56	0.79%
	50	303.30	307.33	1.31%
5/6	30	181.98	184.34	1.28%
	40	242.63	246.00	1.37%
	50	303.30	309.46	1.99%
Average				0.64%

Notes: (1) Radius ratio r_1/r_2 is the ratio of truck route circle's radius to the customer circle's radius. (2) F_{LB} and F_{BP} denote the solution value of the lower bound of Proposition 2 and the objective value of the optimal solution solved by the branch-and-price algorithm, respectively. (2) $\Delta_{F_{BP}} = (F_{BP} - F_{LB})/F_{BP}$.

6.3 Sensitivity analyses

After validating the efficiency of our proposed algorithm, we perform sensitivity analyses to derive managerial implications. We investigate the influence of drone speed, the maximum number of drones that can be carried by a truck, the number of stations, the endurance time of drones, the unit penalty cost of late delivery for orders, the order distribution and drone unavailability on the model's objective value.

6.3.1 Influence of drone speed

The speed of drones was set as 45 km/h in the previous experiments. In the first series of sensitivity analyses, we increase or decrease the speed by five and 10 and assess the cost changes. The experimental results are shown in Figure 6 and indicate that the objective value decreases as the speed increases. As the speed increases further, the decreasing cost trend levels off. A detailed analysis of the influence of the drone speed on the objective value is elaborated in Appendix I. This implies that drones do not necessarily need to be deployed at a very high speed, as this may not significantly reduce the cost if the investment cost of high-speed drones is very high. Our results are consistent with the conclusions of Zhen et al. (2023) and Boysen et al. (2018).

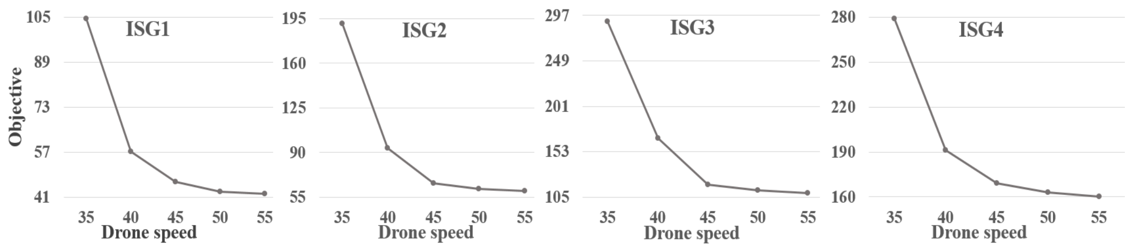


Figure 6: Influence of drone speed on the objective value

6.3.2 Influence of the maximum number of drones that can be carried by a truck

We conducted a second series of sensitivity analyses to investigate the influence on the objective value of the maximum number of drones that can be carried by a truck. In the previous experiments, this maximum number was set as four. We now set the number in a range between two and nine. According to the results shown in Figure 7, the more drones per truck, the lower the operational cost. This is consistent with the conclusions of Li et al. (2022). When the number of drones that can be carried by a truck increases, the overall delivery efficiency improves because the drones can share more tasks in the delivery process. As the number of drones a truck can carry increases, overall operating costs decrease and this decrease gradually levels off. This means that by continuing to increase the number of drones carried on trucks, there will be no further significant decrease in costs. Figure 7 also shows that the trend of the cost reduction levels off as the maximum number of drones per truck increases. This implies that we need not deploy trucks with too large a drone carrying capacity, as this may not reduce the operating cost significantly but will increase the cost of investing in trucks with a large-volume capacity.

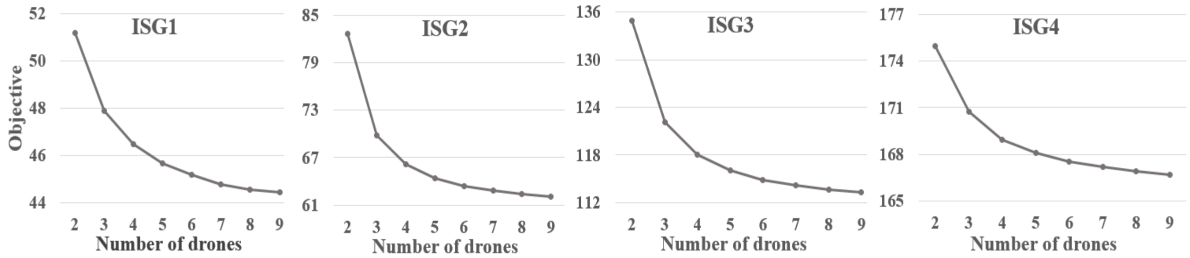


Figure 7: Influence of the maximum number of drones that can be carried by a truck

6.3.3 Influence of the number of stations

The number of stations deployed on the truck's bus-tour route has an effect on the performance of the delivery mode. More deployed stations provide more options for the drones in their takeoff and landing activities, which may be beneficial for reducing drones' waiting time and improving overall efficiency. As shown in Figure 8, the objective value follows a decreasing trend as the number of stations increases, which is similar to the conclusions obtained in related studies, such as Yu et al. (2022), Gao et al. (2023a), and Xia et al. (2023). However, solving a model with more decision variables and constraints, due to the increase in deployed stations, becomes more time-consuming. The dashed curves in Figure 8 show an increasing trend in the algorithm computation time as the number of stations increases; a shorter computation time usually means that a decision can be timelier in realistic applications. Thus, the balance between cost and decision time should be considered to determine the number of stations that can achieve good performance in real situations.

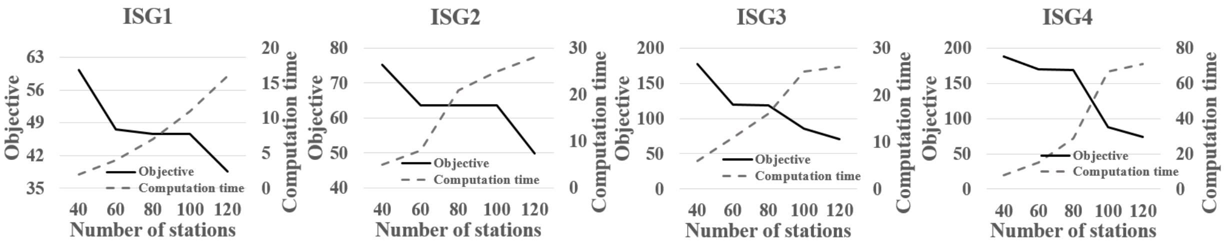


Figure 8: Influence of the number of stations on the objective value

The curves in Figure 8 decline then level off and then decline again; and the curves are analyzed as follows. Firstly, there is an initial decreasing phase, where the cost tends to decrease as the number of stations on the truck's bus-tour route increases; the reason lies in that more stations could provide more options for drones' take-offs and landings, which can potentially reduce their flying time as well as flying cost. Along with the number of stations growing, the initial decreasing phase is followed by a leveling phase. When the number of stations reaches a certain level, the contribution of additional stations to the reduction of drone flying time and cost decreases because the existing stations are already sufficient to meet most of the demand for drones' take-offs and landings. More specifically, although the deployment of more stations can provide additional takeoff and landing options for a drone, the contribution to reduction of the drone's flying time may become less significant because these new additions (of stations) only slightly shorten the distance that drones can fly, rather than significantly reducing their flying time. In this case, the addition of stations probably does not result in a proportional reduction of drones' flying time and cost. Finally, there is a subsequent decline phase. Because as the number of stations increases further, there may exist stations that could significantly reduce the drones' flying time. As a result, drones are able to choose better flying paths for reducing their flying time, which makes the cost curve drop again after the leveling phase.

6.3.4 Influence of the endurance time of drones

Other studies conclude that the endurance time of drones has a significant effect on the truck-drone cooperative delivery routing problem (Cheng et al., 2023; Gao and Zhen, 2024). Longer drone endurance allows a drone to cover a larger service region. The distance and time traveled by the trucks can then be reduced, which helps to improve delivery efficiency and reduce costs. However, in our study we did not find an influence of endurance on objective value, as Figure 9 indicates.

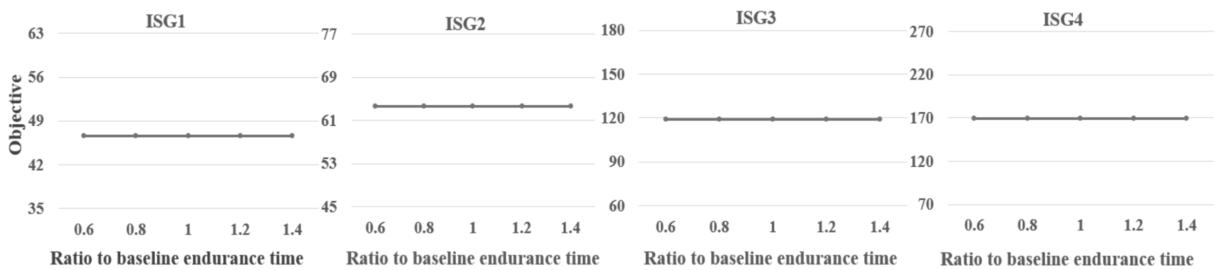
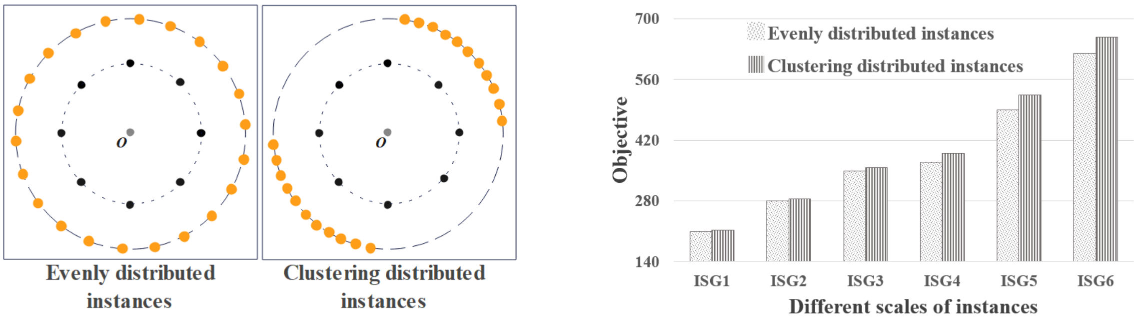


Figure 9: Influence of the endurance time of drones on the objective value

The endurance time of a drone mainly affects the set of orders (in a batch) that can be fulfilled by this drone. In our problem, all of the orders in the batch are fulfilled by a fleet of drones, so the revenue for fulfilling orders is not considered in the objective. Therefore, the endurance time of drones does not influence the model's objective. This suggests that the operator need not invest in maintaining a long endurance time for its drones. Therefore, drones do not need to be equipped with a larger and heavier battery, and thus they can have spare capacity for carrying more cargo.

6.3.5 Influence of the order distribution

In this subsection, we explore the impact of the geographic distribution of order locations on the objective value in six sizes of instance groups, i.e., ISG1, 2, 3, 4, 5, and 6, which contain 15, 20, 25, 30, 40, and 50 customers, respectively. According to Proposition 2, the experiments in this subsection are set in some concentric circles; the concentric setting can theoretically simplify the experimental environment so as to explore the impact of the geographical distribution of order locations on the objective in a clearer way. By comparing the realistic maps as experimental settings, the above theoretical way based on concentric circles can derive more conclusions that may be applicable for more generic contexts rather than only be applicable for a specific context. More specifically, in this theoretically simplified environment, the truck route is a circle of radius r_1. In addition, the origins and destinations of the orders are distributed on another circle of radius r_2, and both circles have the same center, $r_2 > r_1$.



(a) Different order distribution patterns (b) Results of order distribution

Figure 10: Influence of the order locations' geographic distribution on the objective value

As shown in Figure 10(a), two distribution patterns for order locations are considered in the experiments. In the first pattern, i.e., the random instances, order origins and destinations are uniformly distributed on the circle of radius r_2. In the second pattern, i.e., the clustering instances, order origins and destinations form two clusters on the circle of radius r_2. This experimental setting ensures that the sum of the distances of all order origins and destinations to the center of the circle O is identical for the two patterns, given an instance group.

Figure 10(b) illustrates the results obtained under the two different order distribution patterns. The randomized nature of the problem results in some differences, and we observe a clear tendency for the cost to increase as the number of customers increases. Similarly, the clustering distribution indicates higher costs, which suggests that this has a negative impact on cost reduction. The intrinsic reason for this counter-intuitive result lies in the mismatching between the "evenly distributed trucks in their bus-tour route" and the "clustering distributed orders' locations". In our problem setting (i.e., our proposed on-demand delivery mode), each drone is configured to serve only one order during a single flight. All orders' origins and destinations need to be visited by drones, which are launched from and land onto the evenly distributed trucks travelling along a bus-tour route. This means that each order requires a drone to fly back and forth; and the cost cannot be optimized by shared routes or bulk service. In the order clustering scenario, since trucks are uniformly distributed on fixed routes and orders are concentrated in a few areas, some orders may need to be served by trucks that are far away from these orders' locations, which results in that the drones need to fly long distances, and then the total cost is increased. The above "mismatching" context may lead to some relatively longer flying trips of drones when compared with the "matching" context, in which an order's location (either origin or destination) is visited by a drone that is the nearest to this location.

Besides the above sensitivity analyses, more experiments are also conducted on the Influence of the unit penalty cost of late order delivery and the Influence of drone unavailability; the results are elaborated in Appendix J.

7 Extensions

We now investigate various extensions so that the proposed methodology can be applied to more realistic contexts. For example, we consider multiple truck routes connected by transshipment hubs. A much larger service region can then be covered. Many more orders can also be handled using our proposed methodology if they can be handled in multiple batches. In addition, we consider the expansion of drones to serve multiple orders in a single flight, and order consolidation. The following four subsections address these four types of extensions.

7.1 Extension to multiple routes with transshipment

In this section, we extend the above proposed model to a more generic context with multiple truck routes. A network containing more routes, in which parcels could be transshipped by trucks between the designated stations in routes, could cover a much larger region for serving customers. The details of the model extension are given in Appendix H. In the extended model, each drone can freely take off from and land on multiple trucks on multiple routes. This implies that a drone can be allowed to fly from a truck on a given route to another truck on a different route. These key features (i.e., the transshipment of parcels between trucks on different routes and allowing drones to fly between different routes) further complicate the extended model. Computational experiments are then conducted on the extended model to derive managerial insights related to these two features.

The first series of comparative experiments is aimed at investigating the benefits of transshipments between routes. The experiments are designed as follows. Suppose a large service region is covered by either one long route (case 1) or three relatively short routes with transshipment (case 2), Figure 11 illustrates truck routes for both cases. In the same area, Figure 11(a) illustrates one long route covering the entire area and Figure 11(b) illustrates three relatively short routes with transshipment. For case 1, the previously proposed model M1 for a single route is applied; while for case 2 the extended model M2 for multiple routes with transshipment is applied, and the intersection vertexes between the routes serve as the transshipment locations. Given a set of orders, the above two models are solved.

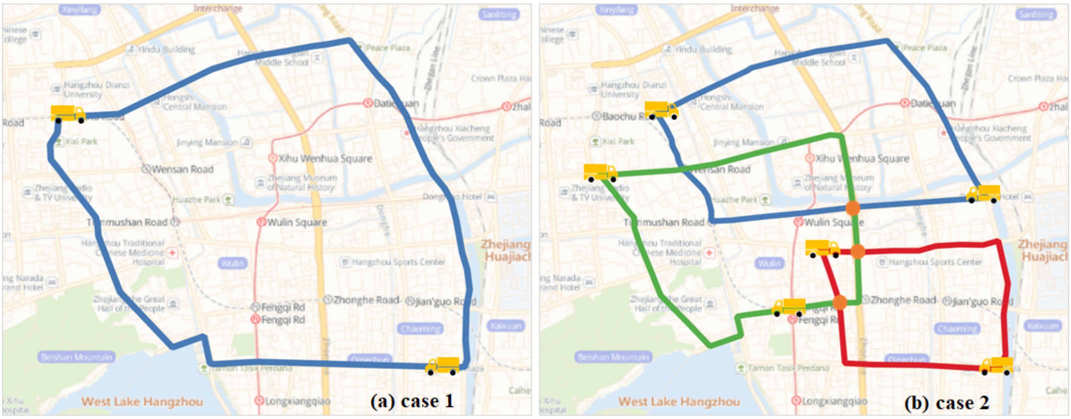
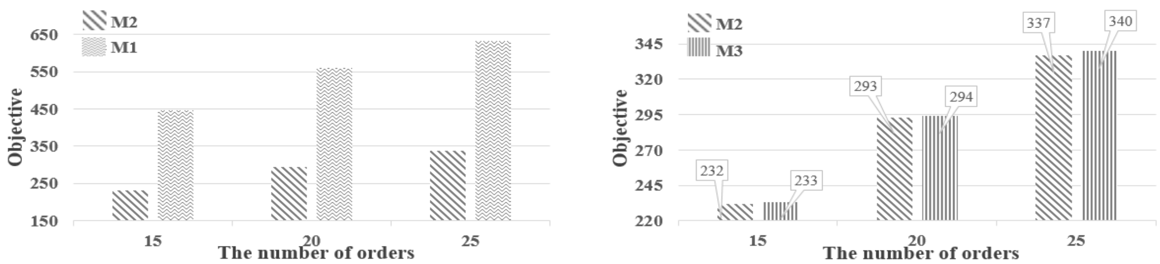


Figure 11: Two cases' routes for investigating the potential benefit of multiple routes with transhipment

For both cases, experiments were conducted using the same batch of orders and the results are shown in Figure 12. The objective values are given in Figure 12(a). The gap between the two objective values reflects the benefit of the transshipment between routes. As shown in Figure 12(a), three comparative experiments are conducted for instances with 15, 20, and 25 orders per batch. Figure 12(a) reflects the significant benefits of transshipment, which validates the extension of the model to a context containing multiple routes with transshipments.

As mentioned, the extended model M2 considers that a drone is allowed to fly between different routes. Thus, the second series of comparative experiments is aimed at investigating the benefits of allowing drones to fly between routes. The model M3 formulation is based on model M2. Each drone is limited to using one route and can only be launched from or landed on trucks that travel along the same route. A comparison of the two objective values obtained by solving models M2 and M3 then reflects the benefit of allowing drones to fly between routes, as shown in Figure 12(b). However, it is somewhat surprising that the investigated benefit does not appear to be significant.



(a) Benefit of transshipment between routes (b) Benefit of allowing drones to fly between routes

Figure 12: Benefits of the extended model that considers multiple routes with transshipment

7.2 The multi-batch mode

The multi-batch mode is used when there are many orders to be processed and all order information is known. As aforementioned, our model is oriented to one batch of orders, but it is applied in the context of a rolling horizon with multiple batches, which implies that our model will be solved iteratively at a certain frequency to make plans for one batch by one batch. The drones, which are idle at trucks and are not reserved in the plans made in the previous batches, could be reused in the plan made for the current batch or the later batches. Here, for the multi-batch mode, we conduct an experiment with 600 orders that are waiting to be delivered. We can complete the delivery of all orders by deciding the delivery path for 50 orders in one batch. Therefore, the above 600 orders can be divided into 12 batches of 50 orders each and solved in batches. According to the first-come-first-served principle, the 600 orders are divided into 12 batches. The first-come-first-served principle means that orders are included in batches according to the orders' arrival time in the system. We experimentally verify the ability of the algorithm to solve 600 orders. The number of orders in each batch is set to 50, 40, 30, and 25, corresponding to 12, 15, 20, and 24 batches, respectively. Here, the system performance is measured in terms of objective value and algorithmic computation time.

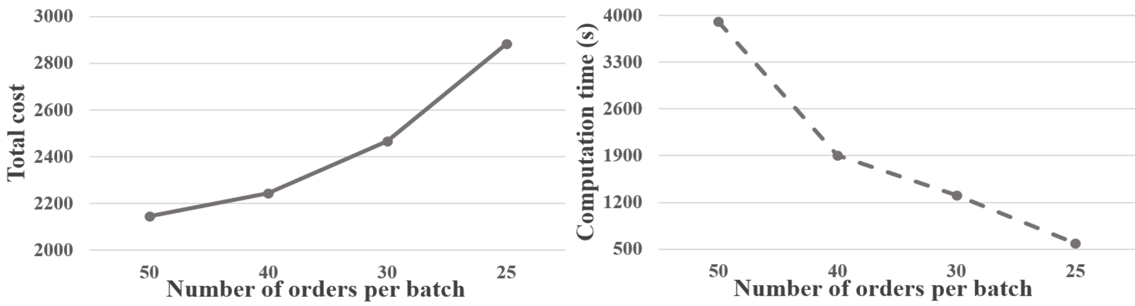


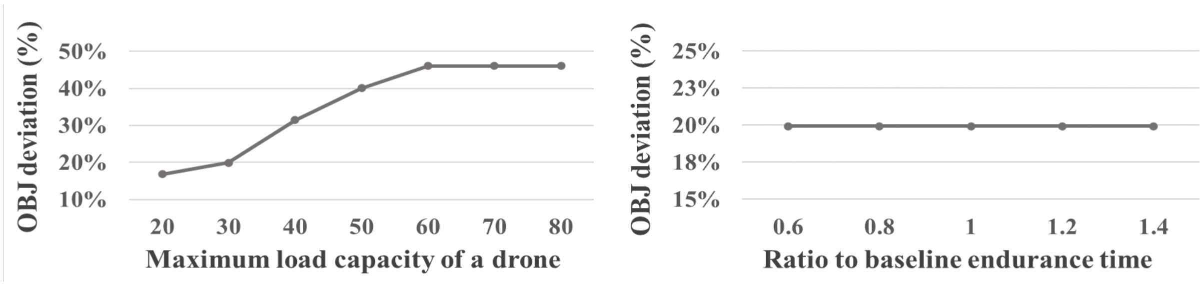
Figure 13: Performance when processing 600 orders in the multi-batch mode

The results in Figure 13(a) indicate that the objective value increases when the number of orders per batch decreases. This suggests that the batch size should be set to a larger value. However, Figure 13(b) shows that the computation time decreases as the number of orders per batch decreases. Therefore, the appropriate batch size should be determined by balancing the objective value against the computation time required to obtain the schedule.

7.3 Benefits of multi-orders per flying trip

This study focuses on the on-demand delivery mode, which is very sensitive to delays. Thus, when an order enters an on-demand delivery system, it is immediately dispatched to a vehicle (or a courier or drone), which is typically dedicated to serving one order in one trip. Our model also assumes that a drone serves one order per flying trip, as in other related studies such as that of Chen and Hu (2023). However, if several orders in nearby neighborhoods are released, a drone can be dispatched to serve all of these orders in one trip. Therefore, we conducted experiments to relax the above assumption and allow a drone to serve two orders in one trip. In this extended context, the load capacity of a drone should be further considered, as the weight of the parcels carried by a drone in one trip should not exceed the load capacity.

Experiments are conducted by setting the maximum load capacity of a drone to 20kg, 30kg, 40kg, 50kg, 60kg, 70kg, and 80kg. Given an instance with a batch of orders, allowing multiple orders to be served by a drone in one trip will further decrease the objective value of the instance. The objective value reduction percentage is denoted by "objective deviation (%)" in the vertical axis of Figure 14(a) with the abovementioned "load capacity of a drone" on the horizontal axis. With the increase in load capacity, the benefits of multi-orders per flying trip, as reflected by the objective deviation, becomes increasingly significant. However, when the load capacity exceeds a specific threshold, the benefit does not appear to increase further. This confirms the benefit of allowing a drone's trip to serve multiple orders. The drones' load capacity should also be increased to achieve this, but it is not necessary to equip the drones with as high a load capacity as possible.



(a) (b)

Figure 14: Benefits of multi-orders per flying trip

In addition to load capacity, the endurance time of drones also affects their ability to serve multiple orders in one flying trip. Figure 14(b) illustrates the benefits of serving multiple orders per flying trip under different endurance time settings. Given an instance as a baseline, we increase or decrease the endurance time in the instance by 20% and 40% and then calculate the aforementioned OBJ deviation, which reflects the benefit of serving multiple orders per trip. The result shown in Figure 14(b) implies that the drones' endurance time has little effect on the benefit of serving multiple orders.

7.4 Benefits of order consolidation

In reality, one customer may release orders in relatively close time instants. The strategy of order consolidation adopted in some on-demand delivery systems reduces the operational cost by consolidating the orders released by the same customer (Chen et al., 2024; Wagner et al., 2023). Identify orders generated by the same customer or in the same area within a short period of time and consolidate these orders into a single order. The weight and volume of the consolidated order must not exceed the load capacity of the drone and must meet the time requirements of all orders. For example, if a customer places three orders within five minutes and all three orders have similar delivery locations, they can be merged into one order. For the consolidated order, delivery is carried out in the same way as for a single order, i.e., a drone is responsible for picking up the order and delivering it to the appropriate location. The order consolidation strategy reduces operational costs mainly by reducing the number of orders. When multiple orders are consolidated into a single order, the number of take-offs and landings of the drones is reduced, which in turn reduces the flight costs. To investigate the benefits of order consolidation, we designed comparative experiments similar to those previously devised. If too many orders are consolidated, it may lead to overloading of the drone and the need to choose a drone with a larger load capacity. In the next experiments, the load capacity of a drone should be further considered, as the weight of the parcels carried by a drone in one trip should not exceed the load capacity.

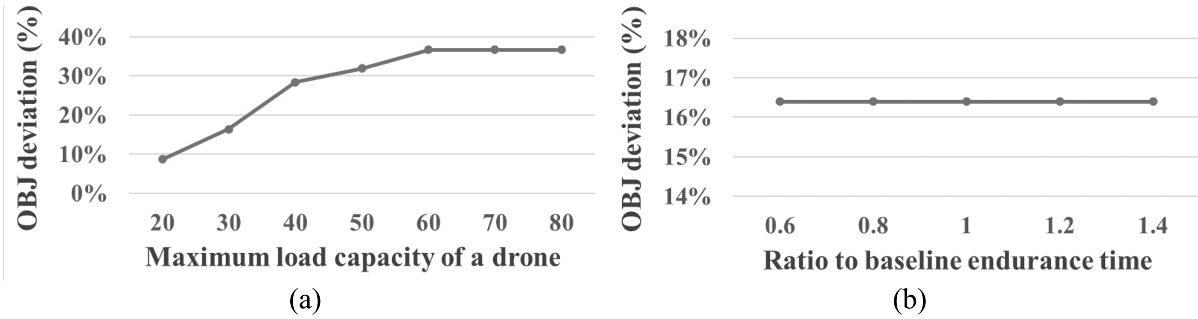


Figure 15: Benefits of order consolidation

Experiments are conducted by setting the maximum load capacity of a drone to 20kg, 30kg, 40kg, 50kg, 60kg, 70kg, and 80kg. Given an instance containing a batch of orders, allowing the drone to multiple orders to be consolidated during a single trip. The order consolidation strategy will further reduce the objective value of the instance. The objective value reduction percentage is denoted by "objective deviation (%)" in the vertical axis of Figure 15(a) with the abovementioned "load capacity of a drone" on the horizontal axis. As the load capacity increases, the benefits of order consolidation become more and more significant. However, the benefit does not seem to increase further when the load capacity exceeds a specific threshold. The results shown in Figure 15 validate the benefit of implementing order consolidation. According to Figures 14 and 17, the benefits of order consolidation are somewhat smaller than those of multi-orders per flying trip, and the influences of load capacity (and endurance time) on these two strategies are also the same.

8 Conclusions

We have proposed a novel on-demand delivery mode based on the cooperation of trucks and drones, which is different from the widely studied last-mile delivery approach based on a fleet of trucks carrying drones. Our proposed on-demand delivery mode can achieve a swift pickup of parcels from orders' origins and the timely delivery of them to the orders' destinations. This mode can also be applied in more generic contexts with trucks and other types of sidekicks such as robots, unmanned vehicles, and human couriers, in addition to our context with trucks and drones. The study makes the following three main contributions.

From the perspective of problem modeling, we formulated MIP models that optimize a set of comprehensive decisions, including the assignment of a drone to pick up an order's parcel from its origin and deliver it to its destination, as well as the location from where a drone is launched from a truck. The proposed model can be applied to a network of multiple truck routes with the transshipment of parcels between routes, where a drone is not dedicated to a single route, nor to a given truck traveling along a route. The objective of the proposed model is to minimize the total operational cost for executing a batch of orders by considering all of these decisions and problem features. The proposed methodology can be used in a rolling horizon manner, and the model is solved batch by batch and thus can handle a large volume of orders continuously entering the system.

From the perspective of algorithmic design, we have designed an exact branch-and-price algorithm to solve the MIP model. Various novel tactics and propositions for algorithmic acceleration were put forward. For example, we combined dynamic programing with the principle of calculus approximation to efficiently solve the pricing problem embedded in the algorithm. Based on small-scale instances, comparisons with CPLEX were conducted to validate the optimality of the solutions obtained by our exact algorithm. Our algorithm is significantly superior to CPLEX in terms of computation time and of the size of the instances it can solve. In addition, comparative experiments with a tight lower bound based on large-scale instances reveal that the average gap to the lower bound for our solution value is only about 0.64%.

From the perspective of managerial insights, we derived important conclusions by conducting sensitivity analyses. Operators of on-demand delivery systems or those interested in adopting this new business mode can benefit from our findings. For example, we found that the endurance time of drones does not appear to affect the operational cost, although drone speed and number do. However, higher values do not necessarily lead to lower operational costs. The clustering distribution of orders has a negative impact on cost reduction. In addition, operators should not set too high a unit penalty cost to increase the service level. When the unit penalty cost value exceeds a certain threshold, increasing the value further does not reduce delivery delays. In a network of routes, managers should also recognize the benefits of transshipment among routes, multi-orders per flying trip, and order consolidation.

In this study, we have considered an operational level decision problem for this new delivery mode. Future research can be conducted into strategic-level decision problems. For example, some models can be designed to optimize truck routes, the number of trucks deployed on each route, and the subdividing decision of the demand into batches. The current model can also be extended to consider the following variants: (1) the battery charging or swapping for drones, (2) the influence of parcel weight or number of parcels carried on a drone's endurance, (3) parcels could be picked up or delivered by the trucks directly if their origins or destinations are not far away from the bus-tour of the trucks. In addition, in future research, we can introduce the elasticity relationship of price and service time to demand, and design a new model for maximizing the platform's profit.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (Grant numbers 72025103, 72394360, 72394362, 72361137001, and 72371221), China Postdoctoral Science Foundation (Grant number 2024M761921), the Project of Science and Technology Commission of Shanghai Municipality China (Grant number 23JC1402200), and the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant number HKSAR RGC TRS T32-707/22-N).

References

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

Boccia, M., A. Masone, A. Sforza, C. Sterle (2021) A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transportation Research Part C: Emerging Technologies* 124: 102913.

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

Bruni, M.E., S. Khodaparasti, M. Moshref-Javadi (2022) A logic-based Benders decomposition method for the multi-trip traveling repairman problem with drones. *Computers & Operations Research* 145: 105845.

Carlsson, J.G., S. Song (2018) Coordinated logistics with a truck and a drone. *Management Science* 64(9): 4052–4069.

Chen, C., E. Demir, Y. Huang (2021) An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. *European Journal of Operational Research* 294(3): 1164–1180.

Chen, M., M. Hu (2023) Courier dispatch in on-demand delivery. *Management Science*. In Press.

Chen, S., Z. Yan, Y.F. Lim (2024) Managing the personalized order-holding problem in online retailing. *Manufacturing & Service Operations Management* 26(1): 47–65.

Chen, X., Z. Cheng, J. Jin, M. Trépanier, L. Sun (2023) Probabilistic forecasting of bus travel time with a bayesian Gaussian mixture model. *Transportation Science* 57(6): 1403–1719.

Chen, X., M.W. Ulmer, B.W. Thomas (2022) Deep Q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research* 298(3): 939–952.

Cheng, R., Y. Jiang, O.A. Nielsen, D. Pisinger (2023) An adaptive large neighborhood search metaheuristic for a passenger and parcel share-a-ride problem with drones. *Transportation Research Part C: Emerging Technologies* 153: 104203.

Cheng, X., J. Lin (2024) Is electric truck a viable alternative to diesel truck in long-haul operation? *Transportation Research Part D: Transport and Environment* 129: 104119.

Chung, S.H., B. Sah, J. Lee (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.

Dayarian, I., M.W.P. Savelsbergh, J.-P. Clarke (2020) Same-day delivery with drone resupply. *Transportation Science* 54(1): 229–249.

Dukkanci, O., A. Koberstein, B.Y. Kara (2023) Drones for relief logistics under uncertainty after an earthquake.

European Journal of Operational Research 310(1): 117–132.

Forget, N, S.N. Parragh (2024) Enhancing branch-and-bound for multiobjective 0-1 programming. *INFORMS Journal on Computing* 36(1): 285–304.

Gao, J., L. Zhen (2024) Intelligent distribution transportation route planning considering traffic congestion. *IEEE Intelligent Transportation Systems Magazine* 16(3): 96–110.

Gao, J., L. Zhen, G. Laporte, X. He (2023a) Scheduling trucks and drones for cooperative deliveries. *Transportation Research Part E: Logistics and Transportation Review* 178: 103267.

Gao, J., L. Zhen, S. Wang (2023b) Multi-trucks-and-drones cooperative pickup and delivery problem. *Transportation Research Part C: Emerging Technologies* 157: 104407.

Gu, R., Y. Liu, M. Poon (2023) Dynamic truck-drone routing problem for scheduled deliveries and on-demand pickups with time-related constraints. *Transportation Research Part C: Emerging Technologies* 151: 104139.

Hokama, P.H.D.B., C.N. Lintzmayer, M.C. San Felice (2024) A faster heuristic for the traveling salesman problem with drone. *Optimization Letters*. In press.

Jiang, J., Y. Dai, F. Yang, Z. Ma (2024) A multi-visit flexible-docking vehicle routing problem with drones for simultaneous pickup and delivery services. *European Journal of Operational Research* 312(1): 125–137.

Kang, M., C. Lee (2021) An exact algorithm for heterogeneous drone-truck problem. *Transportation Science* 55(5): 1088–1112.

Li, H., J. Chen, F. Wang, M. Bai (2021) Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. *European Journal of Operational Research* 294(3): 1078–1095.

Li, H., J.Chen, F. Wang, Y. Zhao (2022) Truck and drone routing problem with synchronization on arcs. *Naval Research Logistics* 69(6): 884–901.

Li, H., F. Wang (2023) Branch-price-and-cut for the truck-drone routing problem with time windows. *Naval Research Logistics* 70(2): 184–204.

Liu, Y. (2019) An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research* 111: 1–20.

Liu, Y. (2023) Routing battery-constrained delivery drones in a depot network: A business model and its optimization-simulation assessment. *Transportation Research Part C: Emerging Technologies* 152: 104147.

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

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

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

Perera, S., M. Dawande, G. Janakiraman, V. Mookerjee (2020) Retail deliveries by drones: How will logistics networks change? *Production and Operations Management* 29(9): 2019–2034.

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

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

Rave, A., P. Fontaine, H. Kuhn (2023) Drone location and vehicle fleet planning with trucks and aerial drones. *European Journal of Operational Research* 308(1): 113–130.

Roberti, R., M. Ruthmair (2021) Exact methods for the traveling salesman problem with drone. *Transportation Science* 55(2): 315–335.

Robinette, T. (2014) HorseFly 'Octocopter' primed to fly the future to your front door. https://www.uc.edu/news/articles/legacy/enews/2014/06/horsefly-octocopter-primed-to-fly-the-future-to-your-front-door.html.

Salama, M.R., S. Srinivas (2022) Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites. *Transportation Research Part E: Logistics and Transportation Review* 164: 102788.

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

Schrotenboer, A., E. Ursavas, I. Vis (2019) A branch-and-price-and-cut algorithm for resource-constrained pickup and delivery problems. *Transportation Science* 53(4): 1001–1022.

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

Xia, Y., W. Zeng, C. Zhang, H. Yang (2023) A branch-and-price-and-cut algorithm for the vehicle routing problem with load-dependent drones. *Transportation Research Part B: Methodological* 171, 80–110.

Yin, Y., D. Li, D. Wang, J. Ignatius, T.C.E. Cheng, S. Wang (2023) A branch-and-price-and-cut algorithm for the truck-based drone delivery routing problem with time windows. *European Journal of Operational Research* 309(3): 1125–1144.

Yu, S., J. Puchinger, S. Sun (2022) Van-based robot hybrid pickup and delivery routing problem. *European Journal of Operational Research* 298(3): 894–914.

Wagner, L., E. Calvo, P. Amorim (2023) Better together! The consumer implications of delivery consolidation. *Manufacturing & Service Operations Management* 25(3): 903–920.

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

Zeng, L., S. Chopra, K. Smilowitz (2022) A bounded formulation for the school bus scheduling problem. *Transportation Science* 56(5): 1111–1408.

Zhen, L., J. Gao, Z. Tan, S. Wang, R. Baldacci (2023) Branch-price-and-cut for trucks and drones cooperative delivery. *IISE Transactions* 55(3): 271–287.

Zhen, L., J. Xia, L. Huang, Y. Wu (2020) Bus tour-based routing and truck deployment for small-package shipping companies. *Transportation Research Part E: Logistics and Transportation Review* 136: 101889.

Zhou, H., H. Qin, C. Cheng, L.-M. Rousseau (2023) An exact algorithm for the two-echelon vehicle routing problem with drones. *Transportation Research Part B: Methodological* 168: 124–150.