https://dx.doi.org/10.1016/j.trc.2023.104407

Multi-trucks-and-drones cooperative pickup and delivery problem

Jiajing Gao [1], Lu Zhen[*1], Shuaian Wang [2]

[1] *School of Management, Shanghai University, Shanghai 200444, China*
[2] *Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Abstract: This study aims to propose a decision methodology on scheduling trucks and drones for truck-and-drone cooperative delivery and pickup system. A fleet contains multiple truck groups; each truck group is a truck with carrying multiple drones. The fleet serves a set of dispersed customers who have the requirements of pickup and delivery services as well as their due time for service. A mixed-integer linear programming (MILP) model is formulated in this study for routing the trucks and drones in the fleet so that each customer's pickup or delivery requirements could be served by either a truck or a drone before their required due time. For solving the MILP model efficiently, this study designs a novel hybrid algorithm by combining the column generation and the logic-based Benders decomposition. Based on the main frame of column generation algorithm, the hybrid algorithm uses logic-based Benders decomposition to solve the pricing problem, and dynamic programming to solve subproblems of logic-based Benders decomposition for the purpose of accelerating the whole algorithm's solving process. Numerical experiments are also conducted on the context of the Hangzhou city so as to validate the efficiency of the proposed hybrid algorithm. Some managerial implications are also derived on the basis of some sensitivity analysis. The proposed methodology, i.e., the MILP model and the novel hybrid algorithm, is potentially useful for platform operators who run the truck-and-drone based urban delivery systems.

Keywords: Truck-and-drone cooperative system; pickup and delivery; multiple trucks and drones; column generation; logic-based Benders decomposition.

1. Introduction

Drones have been emerging in more and more civil applications. For example, the drones can deliver emergency supplies and aid in areas where medical personnel are unable to safely reach. The drone delivery is also gradually recognized as a suitable mode for time-critical delivery in some congested urban districts. Although the drone based delivery contains merits, the dominating status of the traditional truck (vehicle) based logistic systems is hardly replaced by the drone based systems in the near future because of the government's strict civil aviation administration and the drones' inherent limitations in some logistic contexts. Therefore, some logistics giants such as Amazon were

* Corresponding author: lzhen@shu.edu.cn (L. Zhen). Tel: +86-21-66137925. Fax: +86-21-66134284.

experimenting with packing a bunch of ground and flying drones into delivery trucks (vans). The delivery trucks or vans carry parcels (cargos) almost to their destination, then send out and direct drones to do the final drop-off. Such as a novel mode, which is named in this study as the truck-and-drone cooperative delivery systems, combines the relative merits of the two delivery models (trucks and drones) and has also attracted more and more attention in academia for investigating the routing and scheduling algorithms of this cooperative system.

As this novel mode has not been widely applied in reality, majority of the researches focus on the scheduling algorithm for single truck group, which is a truck (or a van) with carrying one or several drones (or small unmanned vehicles). These drones or unmanned vehicles act as the role of flying or land sidekicks; customers' parcels could be delivered by either these sidekicks or the trucks (vans). In addition, one of the important advantages of the third-party logistics systems lies in their large-scale coverage of the customers. Thus, designing an algorithm of scheduling multiple truck groups is more necessary for supporting this cooperative delivery systems' operation in large-scale application. Moreover, the automation, aviation, battery, and artificial intelligence related technologies have been developing very fast and widely applied in the trucks (vans) and drones contained in this system. In recent years, the capacity of the trucks and drones have been increased, more drones could be carried by one truck; more flying trips could be executed by a drone during a relatively long time; more customers could be visited (served) by a drone during one flying trip. Last but not the least, the customers' requirements become more and more diverse; some customers may need be delivered with some parcels bought from some online shopping malls, while some parcels may need be pickup and returned to sellers. Thus, the trucks and drones in this cooperative system need be scheduled to execute both the pickup tasks and the delivery tasks for customers. All of the above further complicates the design of a comprehensive decision methodology on scheduling trucks and drones for truck-and-drone cooperative delivery and pickup system.

This study investigates a comprehensive scheduling problem for the truck-and-drone cooperative delivery and pickup system. The studied problem takes account of a fleet containing multiple truck groups, each of which is constituted by one truck (or van) and more than one drones (or sidekicks); both the trucks and drones can execute the tasks of the deliveries and pickups for customers dispersed in a network. Each customer has their required due time for service. During a drone's flying trip that starts when the drone takes off from its truck and ends when the drone lands on the truck, the drone can visit more than one customer to perform the tasks of the deliveries and pickups. Moreover, in the planning horizon of the problem, a drone can execute more than one trips. By taking account of the above comprehensive set of realistic factors, this study proposes a mixed-integer linear programming (MILP) model for routing the trucks and drones in the fleet so that each customer's pickup or delivery requirements could be served by either a truck or a drone before their required due time. For solving the proposed MILP model within a reasonable time, this study combines the column generation (CG) and the logic-based Benders decomposition for designing a novel hybrid algorithm, which is also incorporated with the dynamic programming to solve subproblems of logic-based Benders decomposition for the purpose of accelerating the whole algorithm's solving process. Numerical experiments are also conducted on the context of the Hangzhou city, i.e., the most representative city in China in the aspect of e-commerce, so as to validate the efficiency of the proposed algorithm. Sensitivity analysis is also performed to obtain some managerial insights that are potentially useful for platform operators who run the truck-and-drone based urban delivery systems.

The remainder of this paper is organized as follows. The related works are reviewed in the next section. A formal problem description is given in Section 3. For the proposed methodology, an MILP model and a hybrid algorithm, are elaborated in Section 4 and Section 5, respectively. Computational experiments are presented in Section 6. The conclusions are then outlined in the last section.

2. Related works

Most of the literature on the truck-and-drone cooperative systems focuses on the delivery problems; while one of the important features of this study lies in the consideration of the pickup & delivery problems. Therefore, this section mainly reviews the literature from two streams: one is about the delivery problems based on truck-and-drone cooperative systems, the other is about the pickup & delivery problems based on these cooperative systems. Several excellent reviews on the cooperative delivery problems of trucks and drones have been written (Chung et al., 2020; Khoufi et al., 2019; Liang and Luo, 2022; Macrina et al., 2020; Otto et al., 2018; Viloria et al., 2021). These studies focus on different aspects of modeling, optimization techniques, solution techniques, performance metrics and application areas. The authors also categorized the related literature based on objectives, solutions, applications, constraints, and use of complementary vehicles. In addition, they review mathematical models, algorithms, and categorize the constraints and characteristics of the problem. These reviews provide a comprehensive overview of existing research in the field and can serve as a valuable resource for future research.

2.1 Delivery problems based on truck-and-drone cooperative systems

(1) Single truck with one or multiple drones

Drones, robots, and other devices expand the scope of traditional truck delivery services and replace manual door-to-door service at delivery terminals. Its application history can be traced back to the truck and drone collaborative delivery scheme proposed by Amazon in 2013. Truck and drone collaborative delivery was initially carried out by a truck group consisting of a single truck and a single drone. It was first studied by Murray and Chu (2015). Murray and Chu (2015) studied a flying sidekick traveling salesman problem (TSP) for a delivery system of a truck and a drone, and established an MILP model as well as a heuristic. The experimental results showed that compared with only truck delivery, the delivery system of a truck and a drone can shorten the delivery time. Subsequently, the delivery route problem of single truck and single drone (Agatz et al., 2018; Boccia et al., 2021; Carlsson and Song, 2018; Ha et al., 2018; Mara et al., 2022; Poikonen et al., 2019; Reed et al., 2022a, b; Roberti and Ruthmair, 2021; Vasquez et al., 2021). Among them, some studies proposed algorithms based on the CG. For example, by combining the branch-and-cut and the CG, Boccia et al. (2021) designed a column-and-row generation approach, through which the optimal solution of up to 20 customers can be obtained. Roberti and Ruthmair (2021) proposed a compact MILP model and an exact branch-and-price algorithm, whch can obtain the optimal solution for instances with up to 39 customers. Vasquez et al. (2021) studied a TSP with drone, and designed a Benders decomposition algorithm. In our paper, based on the ideas of CG and the Benders decompositon, we design a novel hybrid algorithm by combining them.

Based on the single truck and single drone delivery route problem, some scholars have studied the delivery route problem of single truck and multiple drones (Bruni et al., 2022; Cavani et al., 2021; Kang and Lee, 2021; Murray and Raj, 2020; Poikonen and Golden, 2020b; Salama and Srinivas, 2022). The distribution efficiency can be improved by expanding single drone distribution to multiple drones distribution. However, due to the increase of the number of drones, the constraints about the time association between trucks and drones have been increased significantly, which brings difficulty to planning the delivery routes. To reduce the complexity of multiple drones scheduling, Kang and Lee (2021) and Bruni et al. (2022) required a truck to remain stationary until drones complete delivery tasks and return to the truck. Canvani et al. (2021) studied a TSP of multiple drones and investigated the modeling complexity imposed by the presence of multiple drones; they designed an exact decomposition method. Salama and Srinivas (2022) discussed the problem of coordinating a truck and multiple drones for last-mile package delivery and introduced a new variant that allows drones and truck can stop at non-customer locations; they developed a two-phase algorithm hybridizing simulated annealing and variable neighborhood search.

(2) Multiple trucks with drones

In recent years, the distribution system consisting of multiple trucks and drones has also attracted scholars' attention. With the increase in customer demands as well as customer delivery time requirements, it is necessary to increase truck groups. The simultaneous operation of multiple trucks can reduce the total operation time and improve distribution efficiency. With the increase of the number of truck groups, it is necessary to consider the constraints related to truck capacity when establishing the optimization models, which makes the mathematical models more complex. Gu et al. (2022), Kuo et al. (2022), Poikonen and Golden (2020a), Sacramento et al. (2019) and Zhen et al. (2023) studied the multiple trucks distribution system with one drone on one truck. And some scholars have studied the problem of multi-truck delivery with multiple drones on one truck (Chen et al., 2021; Li et al., 2022; Li et al., 2020; Masmoudi et al., 2022; Schermer et al., 2019; Tamke and Buscher, 2021; Wang and Sheu, 2019; Yan et al., 2022). Compared with the distribution system consisting of single truck and single drone, the distribution system consisting of multiple trucks and drones is much more complex, which requires consideration of the synchronization of multiple drones and trucks, as well as the distribution activities for the customers among multiple trucks. The established mathematical models have much more decision variables, larger solution space and more constraints. Schermer et al. (2019) studied the delivery problem of multiple trucks and multiple drones by considering multiple drones carried by one truck, and proposed a metaheuristic based on problem specific features. Chen et al. (2021) studied a vehicle routing problem (VRP) of delivery robots with time windows, and proposed an adaptive large neighborhood search (ALNS) heuristic to plan routes of multiple trucks and multiple robots.

(3) One trip of drone visiting multiple customers

In many researches on cooperative delivery system of trucks and drones, due to the limitation of the loading capacity and endurance of drones, it is assumed that each drone serves one customer in a single flight (e.g., Agatz et al., 2018; de Freitas and Penna, 2020; Dell'Amico et al., 2021, 2022; Ha et al., 2018, 2020; Kuo et al., 2022; Poikonen et al., 2019; Raj and Murray, 2020; Roberti and Ruthmair, 2021; Sacramento et al., 2019; Salama and Srinivas, 2022; Yurek and Ozmutlu, 2021; Zhen et al., 2023). However, with the progress of technology, more and more scholars believe that it is possible for each drone to serve multiple customers in a single flight (e.g., Gonzalez-R et al., 2020; Gu et al., 2022; Leon-Blanco et al., 2022; Li et al., 2020; Luo et al., 2021; Masmoudi et al., 2022; Poikonen and Golden, 2020a, b; Reed et al., 2022a, b; Wang and Sheu, 2019). The problem of truck-drone delivery is further complicated by the fact that a drone visits multiple customers in one trip. The route planning of a trip of each drone need to consider the impact of the capacity and endurance of the drones. In addition, in the route planning, each drone needs to decide the order (sequence) for serving customers in one trip; then a series of binary decision variables are needed, resulting in much larger solution space, more constraints, and difficulty in solving the mathematical models. Poikonen and Golden (2020b) relaxed some of the constraints often found in the previous literature and proposed the k-MVDRP (k-Multi-Visit Drone Routing Problem), in which multiple drones can carry multiple parcels on a single truck; they designed a heuristic to minimize the time. Luo et al. (2021) studied a multi-visit TSP with multiple drones considering that drone can delivery multiple parcels in one trip; they established a MILP model to minimize the time, and designed a heuristic for solving the model. Masmoudi et al. (2022) considered a fleet of drones equipped with multi-pack payload bays in order to serve more customers in one trip, and designed an adaptive multi-start simulated annealing metaheuristic for maximizing total profit. Based on the above analysis, it is currently feasible for drones to serve multiple customers in a single trip, and therefore, we consider drones serving multiple customers in a single trip. In contrast to the traditional vehicle routing problem, the truck-drone delivery problem involves the coordination and integration of two different types of vehicles, which poses additional coordination and synchronization challenges, and the routing decisions need to take into account the capacity, constraints, and interactions of these two types of vehicles. For trucks and drones, both have different speeds and service ranges, which creates complexity in determining when and where to deploy drones from trucks to optimize the overall delivery process. The delivery system studied in this paper involves multiple drones, and it is necessary to consider the number, capacity, and speed of the drones to rationally assign tasks and coordinate the flight paths of multiple drones to maximize the delivery efficiency. In addition, we consider that simultaneous pickup and delivery by trucks and drones makes our research problem more complex and challenging.

2.2 Pickup & delivery problems based on truck-and-drone cooperative systems

(1) Categories of the pickup and delivery problems

Pickup and delivery problems are important route planning problems in which goods or passengers must be transported from different starting points to different destinations. Min (1989) first raised the issue of pickup and delivery on the issue of public libraries. Based on the work of Berbeglia et al. (2007), Battarra et al. (2014) and Koc et al. (2020), the pickup and delivery problem can be divided into three categories, including one-to-one problem, one-to-many-to-one problem, and many-to-many problem. The one-to-one problem is that each item has a pickup location and a delivery location, and the item must be delivered between the pickup and delivery locations. The one-to-many-to-one problem is that some items are delivered from the depot to many customers, while others are collected at the customer and shipped back to the depot. The scenario in which empty cans and bottles are collected while drinks are distributed and forward and reverse logistics systems are applied to this problem. The many-to-many problem is that each item may have multiple pickup and delivery locations, and any location may be a pickup and delivery location for multiple items.

From perspective of the pickup and delivery problem’s categories, the problem studied in this paper belongs to the most studied and common variant of the one-to-many-to-one pickup and delivery problem. Each customer has pickup demand and delivery demand. The goal is to build the plan of routes with the lowest total cost, which also allows vehicles to meet each customer's pickup and delivery requirements and do not exceed each vehicle's capacity. Instead of conventional vehicles, we used both trucks and drones in our study.

(2) The pickup & delivery problems based on truck-and-drone cooperative systems

As far as we know, there are only six papers on the pickup & delivery problems based on truck-and-drone cooperative systems this problem. Table 1 lists these relevant literatures. Jeon et al. (2021) and Gacal et al. (2020) studied the pickup and delivery system of a truck and a drone. Jeon et al. proposed the routing problem considering backhaul demands, established a mixed integer linear programming model, and constructed a heuristic algorithm to solve large-scale problems. Gacal et al. (2020) studied the flying sidekick traveling salesman problem with pickup and delivery, considering drone energy optimization, and proposed a mixed integer linear programming model with the goal of minimizing the total cost of route operations. Karak and Abdelghany (2019) took the system consisting of a single truck and multiple drones as the research object, and studied a one-to-many-to-one problem considering the pickup and delivery vehicle problem. In this distribution system, drones can serve multiple customers in a single flight, and all customers need drones to serve, while the truck only as the mobile depot of drones and does not serve customers. They formulated a mixed integer model with the aim of minimizing operation cost, and designed a heuristic algorithm to solve the model. The other three papers all studied multi-truck and multi-drone systems. Lu et al. (2022) proposed a multi-objective humanitarian pickup and delivery VRP with drones, which is closest to many-to-many pickup and delivery problem. A hybrid multi-objective evolutionary algorithm and a hybrid multi-objective ant colony algorithm were developed to solve the problem. Yu et al. (2022) studied a van-based robot logistic distribution system, and introduced a mixed-integer program based on the one-to-one pickup and delivery mode. In addition, they proposed an ALNS algorithm. Luo et al. (2022) studied a one-to-one pickup and delivery problem involving multiple trucks.

Table 1: The related literature on multi-trucks-and-drones cooperative pickup and delivery problem

Authors and years	**Number of trucks and drones**	**Problem features**					**Type of pickup & delivery**	**Objective**	**Solving methods**
		a truck carrying multiple drones	*one trip for multiple customers*	*both trucks and drones serve customers*	*nodes other than depot and customers*	*time window of customer*			
Jeon et al. (2021)	1-truck, 1-drone			✓	✓		one-to-many-to-one	time	order-first split-second, clustering-based heuristic
Gacal et al. (2020)	1-truck, 1-drone		✓	✓			one-to-many-to-one	cost	CPLEX
Karak and Abdelghany	1-truck, m-drone	✓	✓		✓		one-to-many-to-	cost	hybrid Clarke and Wright heuristic

(2019)							one		
Lu et al. (2022)	m-truck, m-drone		✓	✓	✓		many-to-many	time, fulfillment rate of nodes	multi-objective evolutionary algorithm
Chen et al. (2021)	m-truck, m-robot		✓	✓	✓	✓	one-to-one	cost	adaptive large neighborhood search
Luo et al. (2022)	m-truck, m-drone		✓	✓			one-to-one	cost	iterated local search
This paper	m-truck, n-drone	✓	✓	✓	✓	✓	one-to-many-to-one	cost	algorithm combining CG and Benders decomposition

Notes: The delivery robot in Chen et al. (2021) is similar to the drone in other papers mentioned.

As shown in Table 1, this paper is similar to more existing studies in that the paper considers drones can carry multiple packages and serve multiple customers. In addition, there are some differences between this paper and existing studies. First, this paper considers multiple trucks and multiple drones, with one truck having multiple drones., increasing the difficulty of solving the problem. Second, the one-to-many-to-one problem considering simultaneous pickup and delivery is studied. fourthly, time windows are often important and should be considered in the pickup and delivery problem, and this study considers the customers' time windows. Finally, as most existing studies are based on meta heuristics, this paper is the development of a hybrid algorithm that combines Benders decomposition and CG. The dynamic programming is also incorporated in the hybrid algorithm to solve subproblems of logic-based Benders decomposition for the purpose of accelerating the whole algorithm's solving process.

Compared with the most existing heuristics algorithms, the hybrid algorithm by combining the column generation and the logic-based Benders decomposition designed in this paper can gradually improve the quality of the solution through the decomposition and coordination of the problem, and can obtain an approximate solution close to the global optimal solution. In addition, the algorithm designed in this paper is usually suitable for more complex optimization problems, especially for the problems that need to generate a large number of decision variables or have large-scale cut planes. Their implementation is complex and requires problem decomposition and coordination, and then solving sub-problems to obtain results. Most importantly, the hybrid algorithm by combining the column generation and the logic-based Benders decomposition usually have higher solution accuracy and can find an approximate optimal solution or a global optimal solution. In contrast, heuristic algorithms usually have a faster solution speed, but the resulting solution may only be locally optimal, resulting in less precision than the hybrid algorithm. We experimentally verify the difference between the heuristic algorithm and the proposed hybrid algorithm in Section 6.3.

To sum up, there are some differences between our designed algorithm and other heuristic algorithms in solving ideas, application scope, complexity, and efficiency. Which algorithm to choose depends on the nature of the problem, its size, and the requirements for the solution. the truck-and-drone cooperative delivery and pickup system studied in this paper is a complex distribution system, which requires the routes of drones and trucks to be high, and requires time efficient connection. Based on this complex problem, we want to get high precision results and be able to handle this complex problem, so we design a hybrid algorithm by combining the column generation and the logic-based Benders decomposition.

3. Problem description

Suppose there is a region with a network of vertexes (nodes), each of which could denote a customer or a station; here a station could be regarded as a candidate location for drones taking off or landing at trucks; the depot is also a station in the network. A fleet of $|K|$ trucks and $(|K| \times |D|)$ drones performs the delivery and pickup services for the region, which contains $|N_V|$ stations and $|N_C|$ customers. A customer may have the demand for pickup and delivery; a customer could be served by a truck or a drone. Each truck carries $|D|$ drones; a truck and its carried $|D|$ drones is called as a truck group. At the beginning of the planning horizon, each truck group departs from a depot, then serves customers (by truck or drone), and returns to the depot before the end of the planning horizon. In the planning horizon, a drone may take off from (and land in) a truck for several times, which means a drone may perform several flying trips. During one flying trip of a drone, the drone could visit multiple customers to perform the delivery and pickup services. When drones are flying to perform delivery and pickup services, a truck need not stay somewhere to wait for its carried drones, but can continue travelling along its path.

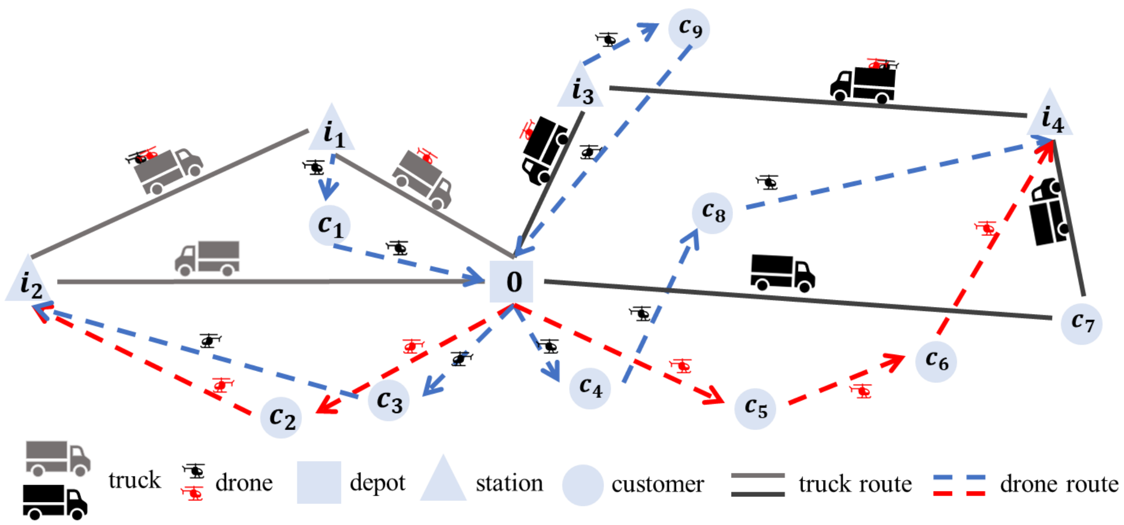

Figure 1: The delivery routing network on cooperative delivery of two truck groups

The delivery route network on cooperative delivery of truck groups is shown in Figure 1. There are two truck groups in Figure 1, and each truck group includes one truck and two drones. Truck 1 starts from the depot, serves customer c_7, visits the station i_3, i_4 and returns to the depot. Drone 1 on truck 1 takes off from the truck at the depot, then serves customer c_4, c_8, and lands on the truck at station i_4. Drone 2 takes off from the truck at the depot, then serves customer c_5 and c_6, and lands on the truck at station i_4. After that, the truck carries the two drones, and travels from station i_4 to i_3. Drone 1 takes off again from the truck at station i_3, then serves customer c_9, and lands on the truck at the depot. During the second flying trip of Drone 1, Drone 2 stays with the truck without flying to serve any customer. Truck 2 starts from the depot, visits the station i_2, i_1 and returns to the depot. Drone 1 on truck 2 takes off from the truck at the depot, then serves customer c_3, and lands on the truck at station i_2. Drone 2 takes off from the truck at the depot, then serves customer c_2, and lands on the truck at station i_2. After that, the truck carries the two drones, and travels from station i_2 to i_1. Drone 1 takes off again from the truck at station i_1, then serves customer c_1, and lands on the truck at the depot. During the second flying trip of Drone 1, Drone 2 stays with the truck without flying to serve any customer.

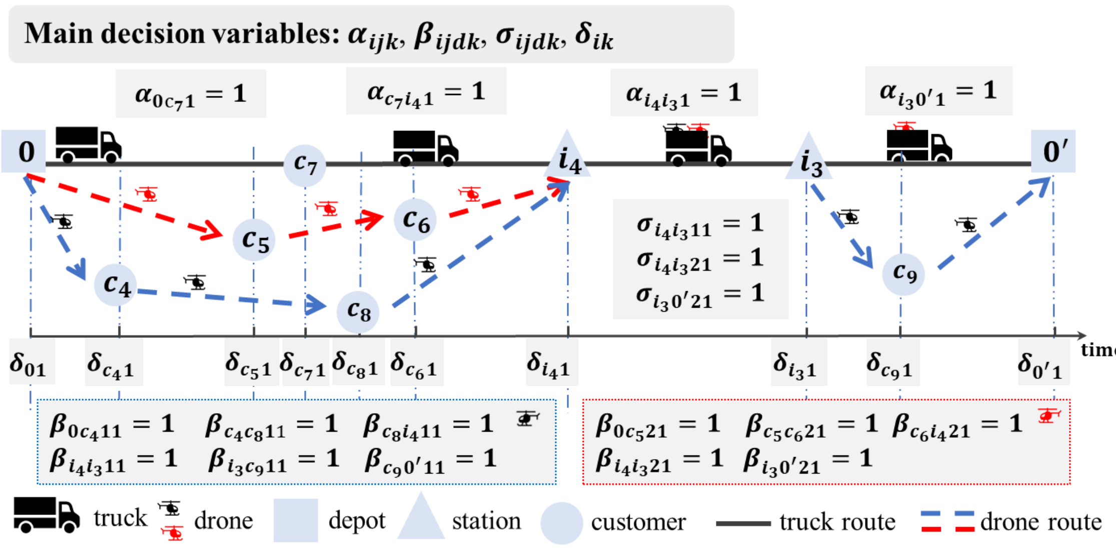

Figure 2: An example on cooperative delivery of a truck and two drones

More specifically, Figure 2 shows that the cooperative delivery of Truck 1 and two drones on Truck 1. As shown in Figure 2, the routes of truck and drones are denoted by the binary variables α_{ijk}, β_{ijdk} and σ_{ijdk}, respectively. More specifically, α_{ijk} equals one if truck k travels from node i to node j; β_{ijdk} equals one if drone d on truck k flies from node i to node j or truck k with carrying drone d travels from node $i \in N_-$ to node $j \in N_+$; σ_{ijdk} equals one if truck k with carrying drone d travels from node $i \in N_-$ to node $j \in N_+$. The time when truck group k departs from node $i \in N$ is represented by the decision variable δ_{ik}, which is affected by the travel time of a truck between node i and j, the flying time of a drone between node i and j, and the duration for a truck or a drone staying at customer i's site to perform delivery/pickup activity. The route of Truck 1 is depot $\rightarrow c_7 \rightarrow i_4 \rightarrow i_3 \rightarrow$ depot, which is represented by the decision variable α_{ijk} as $\alpha_{0c_71} = 1$, $\alpha_{c_7i_41} = 1$, $\alpha_{i_4i_31} = 1$, $\alpha_{i_30'1} = 1$. The route of Drone 1 on Truck 1 is depot $\rightarrow c_4 \rightarrow c_8 \rightarrow i_4 \rightarrow i_3 \rightarrow c_9 \rightarrow$ depot, which is represented by the decision variable β_{ijdk} as $\beta_{0c_411} = 1$, $\beta_{c_4c_811} = 1$, $\beta_{c_8i_411} = 1$, $\beta_{i_4i_311} = 1$, $\beta_{i_3c_911} = 1$, $\beta_{c_90'11} = 1$. The route of Drone 2 on Truck 1 is depot $\rightarrow c_5 \rightarrow c_6 \rightarrow i_4 \rightarrow i_3 \rightarrow$depot, which is represented by the decision variable β_{ijdk} as $\beta_{0c_521} = 1$, $\beta_{c_5c_621} = 1$, $\beta_{c_6i_421} = 1$, $\beta_{i_4i_321} = 1$, $\beta_{i_3c_921} = 1$, $\beta_{i_30'21} = 1$. In addition, Truck 1 with carrying Drone 1 travels from i_4 to i_3, and Truck 1 with carrying Drone 2 travels from i_4 to depot, so we have: $\sigma_{i_4i_311} = 1$, $\sigma_{i_4i_321} = 1$, $\sigma_{i_30'21} = 1$.

How to plan routes to optimize the total cost in the process of pickup and delivery is the research problem of this paper. To solve this problem, we establish a mathematical model to minimize the total cost, which includes the fixed cost of using truck groups, the travel cost of trucks and drones, and the penalty cost of late delivery or pickup serving for customers. First, the value of the fixed cost is determined by the number of truck groups in use. We define the binary variable ε_k to determine whether the truck group is used, and ε_k equals one if truck group k is used. In addition, s^G is defined the fixed cost of a truck group. Therefore, the fixed cost in this problem can be expressed as: $\sum_{k\in K} s^G \varepsilon_k$. In this problem, the loading capacity of each truck and drone with respect to carried cargos is considered. Suppose the load of cargos that need be delivered to and be picked up from customer i is denoted as q_i^D and q_i^P, respectively; the maximum loading capacity of a truck and a drone is denoted as m^K and m^D, respectively. The values of q_i^D, q_i^P, m^K, m^D, and some other variables' values influence the binary variable ε_k's value. Second, the value of the travel cost of trucks is mainly determined by α_{ijk}. We define the travel time of the truck between node i and j is t_{ij}^K and the travel costs of a truck per unit time is s^K. Therefore, the travel cost of trucks can be expressed as: $\sum_{k\in K} \sum_{i\in N_-} \sum_{j\in N_+} s^K t_{ij}^K \alpha_{ijk}$. Third, the value of the travel cost of drones mainly depends their routes which are determined by α_{ijk} and β_{ijdk}. We define the travel time of the drone between node i and j is t_{ij}^D and the travel cost of a drone per unit time is s^D. The drone routes include the flying routes of drones and the routes of the truck and drones travelling together. The travel cost of drones is the cost of drone flying routes, but does not include the route cost of the truck and the drone travelling together. We defined the binary variable σ_{ijdk}, which equals one if truck k with carrying drone d travels from node $i \in N_-$ to node $j \in N_+$. Therefore, the travel cost of drones can be expressed as: $\sum_{k\in K} \sum_{d\in D} \sum_{i\in N_-} \sum_{j\in N_+} s^D t_{ij}^D \left(\beta_{ijdk} - \sigma_{ijdk}\right)$. We consider that a drone can serve multiple customers in one flight without violating the endurance f

and the maximum loading capacity of the drone m^D. Forth, the value of the penalty cost of late delivery or pickup serving for customers is determined by δ_{ik}. In order to get the value of δ_{ik}, we define the auxiliary variables δ_{ik}^K and δ_{idk}^D. The duration for a truck or a drone staying at customer i's site to perform delivery/pickup activity is expressed by t_i^C, and $t_i^C = 0$ if node i is a station. The value of δ_{ck} should be within the time limit e_c of the truck and drone leaving customer c; the late from the limit e_c will be penalized; s^T is the penalty cost per unit time. Thus, the penalty cost of late delivery or pickup is: $\sum_{k\in K}\sum_{c\in N_C} s^T\left(\delta_{ck} - e_c\left(\sum_{i\in N_-}\sum_{d\in D}\beta_{icdk} + \sum_{i\in N_-}\alpha_{ick} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icdk}\right)\right)^+$. All the above four parts of cost should be minimized in the proposed model's objective.

Before addressing the model, the assumptions are clarified as follows:

(1) Each drone is dedicated to one truck.

(2) A drone can only take off or land on its dedicated truck when the truck is at a node (station or depot) in the network.

(3) The time for replacing batteries for drones is short and negligible.

4 Mathematical formulation

According to the previous problem description, this section proposes an MILP model for the multi-trucks-and-drones cooperative pickup and delivery problem.

4.1 Notations

Before formulating the MILP model, the indices, parameters, and decision variables are defined first. We use the Roman letters and Greek letters to denote the parameters and decision variables, respectively.

Indices and sets

K set of the truck groups (also trucks), indexed by k.

D set of the drones on each truck, indexed by d.

N_V set of the station nodes in the network, $N_V = \{0,1,2 \ldots\ldots |N_V|\}$, 0 and $|N_V|$ represent the starting and ending depots, respectively.

N_C set of the customers, $N_C = \{|N_V| + 1, |N_V| + 2 \ldots\ldots |N_V| + |N_C|\}$.

N set of the nodes, $N = N_V \cup N_C$, indexed by i, j, h, c; $N_- = N\backslash\{|N_V|\}, N_+ = N\backslash\{0\}, N_0 = N\backslash\{0, |N_V|\}$.

Parameters

t_{ij}^K travel time of a truck between node i and j.

t_{ij}^D flying time of a drone between node i and j.

q_i^D load of cargos that needs be delivered to customer i.

q_i^P load of cargos that needs be picked up from customer i.

m^K maximum loading capacity of a truck with respect to its carried cargos.

m^D maximum loading capacity of a drone with respect to its carried cargos.

e_c latest time when a truck (or a drone) should depart from customer c's site after serving it.

f endurance of a drone, i.e., a drone's maximum flying time of one trip.

t_i^C duration for a truck or a drone staying at customer i's site to perform delivery/pickup activity.

s^K travel cost of a truck per unit time.

s^D travel cost of a drone per unit time.

s^T penalty cost per unit time of late arrival at (departure from) a customer's site.

s^G fixed cost of activating a truck group.

M a sufficiently large positive number.

Decision variables

α_{ijk} binary, equals one if truck k travels from node $i \in N_-$ to node $j \in N_+$, and zero otherwise.

β_{ijdk} binary, equals one if drone d on truck k flies from node $i \in N_-$ to node $j \in N_+$ or truck k with carrying drone d travels from node $i \in N_-$ to node $j \in N_+$, and zero otherwise. This decision variable decides the drone's routing, both the flight routing of the drone and the routing of the drone on the truck.

σ_{ijdk} binary, equals one if truck k with carrying drone d travels from node $i \in N_-$ to node $j \in N_+$, and zero otherwise. This decision variable decides the routing of the drone on the truck.

γ_{ijhdk} binary, equals one if drone d, which takes off from truck k at node $i \in N_V$, flies from node $j \in N_-$ to node $h \in N_+$, and zero otherwise.

ζ_{ik} nonnegative, the remaining load of cargos when truck group k departs from node $i \in N$.

τ_{ik} nonnegative, the remaining load of cargos when truck k departs from node $i \in N$.

τ'_{idk} nonnegative, the remaining load of cargos when drone d on truck k departs from node $i \in N$.

ϱ_{ijdk} nonnegative, the remaining load of cargos when drone d departs from node $j \in N_+$ after taking off from truck k at node $i \in N_V$ in the drone d's flying trip.

φ_{ijhdk} nonnegative, the duration from the time when drone d takes off from truck k at node $i \in N_V$ to the time when the drone departs from node $h \in N_+$; the node $j \in N_-$ is immediately before the node h in the drone d's flying trip.

μ_{ik}^K integer, the order of node i in the sequence of nodes visited by truck k.

μ_{idk}^D integer, the order of node i in the sequence of nodes visited by drone d on truck k.

δ_{ik} nonnegative, the time when truck group k departs from node $i \in N$.

δ_{ik}^{K} nonnegative, the time when truck k departs from node $i \in N$, regardless of the effect of the time when drones depart from node $i \in N$.

δ_{idk}^{D} nonnegative, the time when drone d on truck k departs from node $i \in N$, regardless of the effect of the time when truck k departs from node $i \in N$.

ε_k binary, equals one if truck group k is used, and zero otherwise.

4.2 Mathematical model

Based on the above definitions, an MILP model is established as follows.

$$\text{Minimize}\Big\{\sum_{k\in K} s^G \varepsilon_k + \sum_{k\in K}\sum_{i\in N_-}\sum_{j\in N_+} s^K t_{ij}^K \alpha_{ijk} + \sum_{k\in K}\sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D (\beta_{icdk} - \sigma_{icdk}) + \sum_{k\in K}\sum_{c\in N_C} s^T \left(\delta_{ck} - e_c\big(\sum_{i\in N_-}\sum_{d\in D}\beta_{icdk} + \sum_{i\in N_-}\alpha_{ick} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icdk}\big)\right)^+\Big\} \quad (1)$$

Subject to

$$\sum_{j\epsilon N_+} \alpha_{0jk} = 1 \qquad \forall k \in K \qquad (2)$$

$$\sum_{i\in N_-} \alpha_{i(|N_V|)k} = 1 \qquad \forall k \in K \qquad (3)$$

$$\sum_{i\in N_-} \alpha_{ijk} = \sum_{i\in N_+} \alpha_{jik} \le 1 \qquad \forall j \in N_0, k \in K \qquad (4)$$

$$\sum_{j\epsilon N_+} \beta_{0jdk} = 1 \qquad \forall k \in K, d \in D \qquad (5)$$

$$\sum_{i\in N_-} \beta_{i(|N_V|)dk} = 1 \qquad \forall k \in K, d \in D \qquad (6)$$

$$\sum_{i\in N_-} \beta_{ijdk} = \sum_{i\in N_+} \beta_{jidk} \le 1 \qquad \forall j \in N_0, k \in K, d \in D \qquad (7)$$

$$\sum_{k\in K}\sum_{i\in N_-}\sum_{d\in D}(\beta_{icdk} - \sigma_{icdk}) + \sum_{k\in K}\sum_{i\in N_-}\alpha_{ick} = 1 \quad \forall c \in N_C \qquad (8)$$

$$2\sigma_{ijdk} \le \beta_{ijdk} + \alpha_{ijk} \le 2\sigma_{ijdk} + 1 \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \qquad (9)$$

$$\sum_{i\in N_-} \sigma_{icdk} = \sum_{i\in N_+} \sigma_{cidk} \qquad \forall c \in N_C, k \in K, d \in D \qquad (10)$$

$$\sum_{i\in N_+} \gamma_{hjidk} = \sum_{i\in N_-} \gamma_{hijdk} \qquad \forall h \in N_V, j \in N_C, k \in K, d \in D \qquad (11)$$

$$\sum_{j\in N_+} \beta_{ijdk} \le \sum_{j\in N_+} \alpha_{ijk} \qquad \forall i \in N_V, k \in K, d \in D \qquad (12)$$

$$\beta_{ijdk} \le \alpha_{ijk} \qquad \forall j \in N_+ \backslash N_C, i \in N_V, k \in K, d \in D \qquad (13)$$

$$\beta_{ijdk} \ge \gamma_{hijdk} \qquad \forall j \in N_+, i \in N_-, h \in N_V, k \in K, d \in D \qquad (14)$$

$$\gamma_{hijdk} \le \sum_{c\in N_0} \alpha_{hck} \qquad \forall j \in N_+, i \in N_-, h \in N_V, k \in K, d \in D \qquad (15)$$

$$\gamma_{iijdk} \le \beta_{ijdk} - \sigma_{ijdk} \qquad \forall i \in N_- \backslash N_C, j \in N_C, k \in K, d \in D \qquad (16)$$

$$\delta_{jk}^K \ge \delta_{ik} + t_{ij}^K + t_j^C - M(1 - \alpha_{ijk}) \qquad \forall i \in N_-, j \in N_+, k \in K \qquad (17)$$

$$\delta_{jk}^K \le \delta_{ik} + t_{ij}^K + t_j^C + M(1 - \alpha_{ijk}) \qquad \forall i \in N_-, j \in N_+, k \in K \qquad (18)$$

$$\delta_{jdk}^D \ge \delta_{ik} + t_{ij}^D + t_j^C - M(1 - \beta_{ijdk}) \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \qquad (19)$$

$$\delta_{jdk}^D \le \delta_{ik} + t_{ij}^D + t_j^C + M\left(1-\beta_{ijdk}\right) \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \qquad (20)$$

$$\delta_{ik} \ge \delta_{idk}^D \qquad \forall i \in N_+, k \in K, d \in D \qquad (21)$$

$$\delta_{ik} \ge \delta_{ik}^K \qquad \forall i \in N_+, k \in K \qquad (22)$$

$$\delta_{0k} = 0 \qquad \forall k \in K \qquad (23)$$

$$\mu_{idk}^D - \mu_{jdk}^D \le N(1-\beta_{ijdk}) - 1 \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \qquad (24)$$

$$\mu_{ik}^K - \mu_{jk}^K \le N(1-\alpha_{ijk}) - 1 \qquad \forall i \in N_-, j \in N_+, k \in K \qquad (25)$$

$$\zeta_{0k} = \sum_{c \in N_C} q_c^D \left(\sum_{i \in N_-}\sum_{d \in D}\beta_{icdk} + \sum_{i \in N_-}\alpha_{ick} - \sum_{i \in N_-}\sum_{d \in D}\sigma_{icdk}\right) \qquad \forall k \in K \qquad (26)$$

$$\zeta_{0k} \le m^K \qquad \forall k \in K \qquad (27)$$

$$\tau_{jk} \ge \zeta_{ik} - q_j^D + q_j^P - M(1-\alpha_{ijk}) \qquad \forall i \in N_-, j \in N_+, k \in K \qquad (28)$$

$$\tau'_{jdk} \ge \zeta_{ik} - q_j^D + q_j^P - M\left(1-\beta_{ijdk}\right) \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \qquad (29)$$

$$\zeta_{ik} \le \tau'_{idk} \qquad \forall i \in N_+, k \in K, d \in D \qquad (30)$$

$$\zeta_{ik} \le \tau_{ik} \qquad \forall i \in N_+, k \in K \qquad (31)$$

$$\zeta_{ik} \le m^K \varepsilon_k \qquad \forall i \in N, k \in K \qquad (32)$$

$$\varrho_{iidk} = \sum_{j \in N_C} q_j^D \sum_{h \in N_-}\gamma_{ihjdk} \qquad \forall i \in N_-, k \in K, d \in D \qquad (33)$$

$$\varrho_{iidk} \le m^D \qquad \forall i \in N_-, k \in K, d \in D \qquad (34)$$

$$\varrho_{ijdk} \le m^D \sum_{h \in N_-}\gamma_{ihjdk} \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \qquad (35)$$

$$\varrho_{ihdk} \ge \varrho_{ijdk} - q_h^D + q_h^P - M(1-\gamma_{ijhdk}) \qquad \forall i \in N_-, j \in N_-, h \in N_+, d \in D, k \in K \qquad (36)$$

$$\varphi_{i_1jhdk} \ge \varphi_{i_1ijdk} + t_{jh}^D + t_h^C - M(2-\gamma_{i_1ijdk} - \gamma_{i_1jhdk})$$
$$\forall i \in N_-, j \in N_C, h \in N_+, i_1 \in N_\mathrm{V}, k \in K, d \in D \qquad (37)$$

$$\varphi_{i_1jhdk} \le \varphi_{i_1ijdk} + t_{jh}^D + t_h^C + M(2-\gamma_{i_1ijdk} - \gamma_{i_1jhdk})$$
$$\forall i \in N_-, j \in N_C, h \in N_+, i_1 \in N_\mathrm{V}, k \in K, d \in D \qquad (38)$$

$$\varphi_{iijdk} = (t_{ij}^D + t_j^C)\gamma_{iijdk} \qquad \forall i \in N_\mathrm{V}, j \in N_C, k \in K, d \in D \qquad (39)$$

$$\varphi_{hijdk} \le f\gamma_{hijdk} \qquad \forall i \in N_-, j \in N_+, h \in N_V, k \in K, d \in D \qquad (40)$$

$$\varepsilon_k \ge \beta_{ijdk} \qquad \forall i \in N_-, j \in N_C, k \in K, d \in D \qquad (41)$$

$$\varepsilon_k \ge \alpha_{ijk} \qquad \forall i \in N_-, j \in N_C, k \in K \qquad (42)$$

$$\alpha_{0(|N_V|)k} = 1 - \varepsilon_k \qquad \forall k \in K \qquad (43)$$

$$0 \le \mu_{ik}^K \le |N| \qquad \forall i \in N, k \in K \qquad (44)$$

$$0 \le \mu_{idk}^D \le |N| \qquad \forall i \in N, k \in K, d \in D \qquad (45)$$

$$\delta_{ik}, \delta_{ik}^K, \delta_{idk}^D, \varphi_{ijhdk}, \varrho_{ijdk}, \zeta_{ik}, \tau_{ik}, \tau'_{idk} \ge 0 \qquad \forall i, j, h \in N, k \in K, d \in D \qquad (46)$$

$$\alpha_{ijk}, \beta_{ijdk}, \sigma_{ijdk}, \gamma_{ijhdk}, \varepsilon_k \in \{0,1\} \qquad \forall i, j, h \in N, k \in K, d \in D. \qquad (47)$$

Objective (1) minimizes the total cost, which includes the fixed cost of using truck groups, the travel cost of trucks and drones, and the penalty cost of late delivery or pickup serving for customers. Constraints (2) and (3) ensure that truck k starts from the depot and eventually returns to the depot. Constraints (4) require flow conservation and each customer can only be served by a truck for at most once. Constraints (5)–(7) are drone routing constraints; the explanation for them is similar as Constraints (2)–(4). Constraints (8) require that each customer can only be served by either a truck or a drone for once. Constraints (9) ensure that if both β_{icdk} and α_{ick} are 1, σ_{icdk} is 0. Constraints (10) restrict that if a truck and a drone arrive at customer c from a node at the same time, the truck and drone need to depart from customer c at the same time. Constraints (11) state that for drone d, which takes off from truck k at node h, it needs to meet flow conservation constraints for each visited node such as customer j. Constraints (12) and (13) ensure that if a truck does not visit a station node ($i \in N_V$) or an arc (i, j) between two station nodes, the drones carried by the truck will not visit the node or the arc. Road network nodes are nodes on the road, and only when the truck passes through these road network nodes, the drone on this truck may pass through these road network nodes, which does not affect the service of the drone to customers. And constraints (12) and (13) do not defeat the original purpose of drones as a substitute for truck delivery. Constraints (14) indicate that if drone d on truck k travels from node $i \in N_-$ to $j \in N_+$ after its taking off from truck k at node h, i.e., $\gamma_{hijdk} = 1$, the drone flies the arc (i, j), i.e., $\beta_{ijdk} = 1$. Constraints (15) restrict that if drone d on truck k takes off from truck k at the node h, the node h is surely visited by truck k. Constraints (16) state that if drone d on truck k takes off from truck k at node i to serve customer j, i.e., $\gamma_{iijdk} = 1$, then the truck cannot travel the arc (i, j) with carrying the drone, i.e., $\sigma_{ijdk} = 0$, and the drone surely flies the arc (i, j), i.e., $\beta_{ijdk} = 1$. Constraints (17)–(22) connect the departure time of truck and drones from two consecutively visited nodes. Constraints (23) set the start time of the whole process is zero. Constraints (24) and (25) are established for subtour elimination. Constraints (26)–(32) are truck group capacity constraints. Constraints (26) and (27) ensure that the load of truck group k's carried cargos in the depot cannot exceed maximum loading capacity of truck k. Constraints (28) and (29) are the variation of cargo load in a truck route and a drone route, respectively. Constraints (30) and (31) are about the remaining load of cargos when truck group k departs from node $i \in N$. Constraints (32) ensure that the load of truck group k cannot exceed maximum loading capacity of truck k. Constraints (33) and (34) guarantee that the load of drone d's carried cargos at node $i \in N_V$ where drone d takes off from truck k cannot exceed maximum loading capacity of drone d. Constraints (35) ensure that the remaining load of cargos when drone d departs from node $j \in N_+$ after taking off from truck k

at node $i \in N_V$ in the drone d's flying trip cannot its maximum loading capacity; in addition, the constraints also ensure that if $\sum_{h \in N_-} \gamma_{ihjdk}$ equals zero, ϱ_{ijdk} should be zero. Constraints (36) are about the calculation on the variation of cargo load in drone d's flying trip. Constraints (37) and (38) update the accumulative flight time for each drone in single flight. Constraints (39) restrict the cumulative time of drone d when it departs from node j after taking off from truck k at node i equals to the drone's travel time from node i to node j, plus the operation time of truck or drone for delivering cargos to customer j and picking up cargos from customer j. And constraints (40) indicate that the cumulative time of drone d on truck k should be no greater than f. Constraints (41) and (42) are used to specify whether truck group k is used or not. Constraints (43) ensure that if a truck group is used, its truck should go through at least one node before returning to the depot. Constraints (44)–(47) define the domains of the decision variables.

5 Algorithm combining CG and logic-based Benders decomposition

In order to solve the above mathematical model efficiently, this study designs a hybrid algorithm by combining the CG and the logic-based Benders decomposition; and an acceleration strategy based on dynamic programming is also incorporated in the algorithm to further shorten the solution time. Figure 3 illustrate the algorithmic framework.

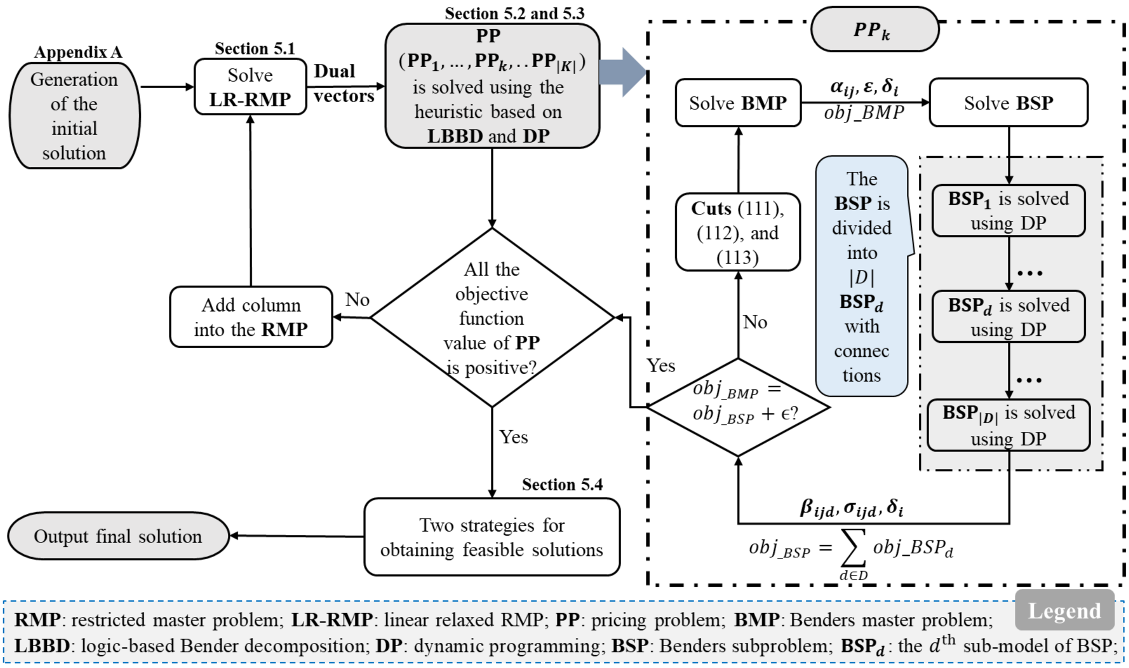

Figure 3: Flowchart of the proposed algorithm

As shown in the left part of Figure 3, the main flow of the algorithm is the CG, in which the pricing problem is divided into $|K|$ subproblems and each subproblem correspond to one truck group k. As shown in the right part of Figure 3, the logic-based Benders decomposition is designed to solve the

CG's pricing subproblem PP_k. The Benders decomposition's master problem (BMP) aims to obtain a plan for truck group k; while the Benders decomposition's subproblem (BSP) is also divided into $|D|$ submodels BSP_d, each of which aims to obtain a plan for drone d carried by the truck group k. A dynamic programming is also implemented in this study for solving the submodels BSP_d efficiently.

This study proposes a hybrid algorithm by combining the CG and the logic-based Benders decomposition to obtain plans for the $|K|$ truck groups and the $(|K| \times |D|)$ drones on the trucks when they performing delivery and pickup tasks for customers.

5.1 Restricted master model

According to the usual practice of CG, we firstly applied Dantzig Wolfe decomposition to reconstruct the original model established in Section 4 into a set partitioning model. Define R as the set of possible routes for truck groups; the routes are indexed by r, $r \in R$. Each route plan contains the route of a truck and the routes of all drones on this truck. Define parameters x_{ijr}, y_{ijdr}, z_{ijdr}, ρ_{ir}, λ, and c_r to represent truck routes, drone routes, common routes of truck and its carried drones, the latest time for a truck group leaving node i, truck group usage, and route cost, respectively. We define these parameters in detail. x_{ijr} is a binary parameter, if a truck travels from node $i \in N_-$ to node $j \in N_+$ in route r, $x_{ijr} = 1$. y_{ijdr} is a binary parameter, if drone d on the truck travels from node $i \in N_-$ to node $j \in N_+$ in route r or the truck with carrying drone d travels from node $i \in N_-$ to node $j \in N_+$ in route r, $y_{ijdr} = 1$. z_{ijdr} is a binary parameter, if drone d is staying on the truck and they travel from node $i \in N_-$ to node $j \in N_+$ together in route r, $z_{ijdr} = 1$. ρ_{ir} is the latest time for a truck group leaving node $i \in N$ in route r. λ is a binary parameter, if the truck group is used in route r, $\lambda = 1$. And c_r is the cost of route r. The values of parameters x_{ijr}, y_{ijdr}, z_{ijdr}, ρ_{ir}, λ, and c_r are equal to the values of decision variables α_{ijk}, β_{ijdk}, σ_{ijdk}, γ_{ijhdk} and ε_k obtained by solving PP_k.

Since the capacity as well as other parameters is identical for all the truck groups, the above defined set of routes can be shared by all truck groups. The decision variable ξ_r is defined as an integer variable indicating the number of times, for which the route r has been selected. Due to the numerous feasible routes contained in the set R, it is difficult to directly solve the master problem. In the CG, a subset of all feasible routes is defined as $R' \subset R$, the restricted master problem (RMP) is established, the integer decision variable ξ_r in the RMP is relaxed as a continuous variable, and the linear relaxed RMP (LR-RMP) is established. The LR-MRP model is as follows.

[LR-RMP] Minimize $\sum_{r \in R'} c_r \xi_r$ (48)

Subject to

$$\sum_{r\in R'}\left(\sum_{i\in N_-}\sum_{d\in D} y_{ijdr} + \sum_{i\in N_-} x_{ijr} - \sum_{i\in N_-}\sum_{d\in D} z_{ijdr}\right)\xi_r = 1 \qquad \forall j \in N_C \tag{49}$$

$$\sum_{r\in R'} \xi_r \leq |K| \tag{50}$$

$$\xi_r \geq 0 \qquad \forall r \in R'. \tag{51}$$

$$c_r = s^G\lambda + s^K \sum_{i\in N_-}\sum_{j\in N_+} t_{ij}^K x_{ijr} + s^D \sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} t_{ic}^D (y_{icdr} - z_{icdr}) + \sum_{c\in N_C} s^T \left(\rho_{cr} - e_c\left(\sum_{i\in N_-}\sum_{d\in D} y_{icdr} + \sum_{i\in N_-} x_{icr} - \sum_{i\in N_-}\sum_{d\in D} z_{icdr}\right)\right)^+ \tag{52}$$

Constraints (49) ensure that each customer must be served. Constraints (50) ensure that at most $|K|$ routes are selected in the solution, that is, each truck group is assigned with at most one route. Constraint (51) define the domains of the decision variables. Formula (52) calculates the route cost. Define the dual variables of Constraints (49) and (50) as $\theta_j, j \in N_C$ and π, respectively. In each iteration of the algorithm, the dual variables are used to build the pricing problem (PP) to produce a new column.

At the beginning of the CG, a heuristic is used to generate a set of initial feasible route plans for the RMP. The heuristic is elaborated in Appendix A.

5.2 Pricing problem (PP)

The PP is used to produce a feasible column, which is then added to the LR-MRP. According to the characteristics of the problem in this study, the PP can be divided into $|K|$ pricing subproblems. Each subproblem is denoted by the model PP_k, which corresponds to truck group k and is formulated as follows.

[PP$_\text{k}$] Minimize$\left\{c_r - \sum_{j\in N_C}\left(\sum_{i\in N_-}\sum_{d\in D}\beta_{ijd} + \sum_{i\in N_-}\alpha_{ij} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{ijd}\right)\theta_j - \pi\right\}$ (53)

Subject to

$$\sum_{j\in N_+} \alpha_{0j} = 1 \tag{54}$$

$$\sum_{i\in N_-} \alpha_{i(|N_V|)} = 1 \tag{55}$$

$$\sum_{i\in N_-} \alpha_{ij} = \sum_{i\in N_+} \alpha_{ji} \leq 1 \qquad \forall j \in N_0 \tag{56}$$

$$\sum_{j\in N_+} \beta_{0jd} = 1 \qquad \forall d \in D \tag{57}$$

$$\sum_{i\in N_-} \beta_{i(|N_V|)d} = 1 \qquad \forall d \in D \tag{58}$$

$$\sum_{i\in N_-} \beta_{ijd} = \sum_{i\in N_+} \beta_{jid} \leq 1 \qquad \forall j \in N_0, d \in D \tag{59}$$

$$\sum_{i\in N_-}\sum_{d\in D}(\beta_{icd} - \sigma_{icd}) + \sum_{i\in N_-} \alpha_{ic} \leq 1 \qquad \forall c \in N_C \tag{60}$$

$$2\sigma_{ijd} \leq \beta_{ijd} + \alpha_{ij} \leq 2\sigma_{ijd} + 1 \qquad \forall i \in N_-, j \in N_+, d \in D \tag{61}$$

$$\sum_{i\in N_-} \sigma_{icd} = \sum_{i\in N_+} \sigma_{cid} \qquad \forall c \in N_C, d \in D \tag{62}$$

$$\sum_{i\in N_+} \gamma_{hjid} = \sum_{i\in N_-} \gamma_{hijd} \qquad \forall h \in N_V, j \in N_C, d \in D \tag{63}$$

$$\sum_{j\in N_+} \beta_{ijd} \leq \sum_{j\in N_+} \alpha_{ij} \qquad \forall i \in N_\text{V}, d \in D \tag{64}$$

$$\beta_{ijd} \leq \alpha_{ij} \qquad \forall j \in N_+ \backslash N_C, i \in N_V, d \in D \qquad (65)$$

$$\beta_{ijd} \geq \gamma_{hijd} \qquad \forall j \in N_+, i \in N_-, h \in N_V, d \in D \qquad (66)$$

$$\gamma_{hijd} \leq \sum_{c \in N_0} \alpha_{hc} \qquad \forall j \in N_+, i \in N_-, h \in N_V, d \in D \qquad (67)$$

$$\gamma_{iijd} \leq \beta_{ijd} - \sigma_{ijd} \qquad \forall i \in N_- \backslash N_C, j \in N_C, d \in D \qquad (68)$$

$$\delta_j^K \geq \delta_i + t_{ij}^K + t_j^C - M(1 - \alpha_{ij}) \qquad \forall i \in N_-, j \in N_+ \qquad (69)$$

$$\delta_j^K \leq \delta_i + t_{ij}^K + t_j^C + M(1 - \alpha_{ij}) \qquad \forall i \in N_-, j \in N_+ \qquad (70)$$

$$\delta_{jd}^D \geq \delta_i + t_{ij}^D + t_j^C - M(1 - \beta_{ijd} + \sigma_{ijd}) \qquad \forall i \in N_-, c \in N_+, d \in D \qquad (71)$$

$$\delta_{jd}^D \leq \delta_i + t_{ij}^D + t_j^C + M(1 - \beta_{ijd} + \sigma_{ijd}) \qquad \forall i \in N_-, c \in N_+, d \in D \qquad (72)$$

$$\delta_i \geq \delta_{id}^D \qquad \forall i \in N_+, d \in D \qquad (73)$$

$$\delta_i \geq \delta_i^K \qquad \forall i \in N_+ \qquad (74)$$

$$\delta_0 = 0 \qquad (75)$$

$$\mu_{id}^D - \mu_{jd}^D \leq N(1 - \beta_{ijd}) - 1 \qquad \forall i \in N_-, j \in N_+, d \in D \qquad (76)$$

$$\mu_i^K - \mu_j^K \leq N(1 - \alpha_{ij}) - 1 \qquad \forall i \in N_-, j \in N_+ \qquad (77)$$

$$\zeta_0 = \sum_{c \in N_C} q_c^D \left(\sum_{i \in N_-} \sum_{d \in D} \beta_{icd} + \sum_{i \in N_-} \alpha_{ic} - \sum_{i \in N_-} \sum_{d \in D} \sigma_{icd} \right) \qquad (78)$$

$$\zeta_0 \leq m^K \qquad (79)$$

$$\tau_j \geq \zeta_i - q_j^D + q_j^P - M(1 - \alpha_{ij}) \qquad \forall i \in N_-, j \in N_+ \qquad (80)$$

$$\tau'_{jd} \geq \zeta_i - q_j^D + q_j^P - M(1 - \beta_{ijd}) \qquad \forall i \in N_-, j \in N_+, d \in D \qquad (81)$$

$$\zeta_i \leq \tau'_{id} \qquad \forall i \in N_+, d \in D \qquad (82)$$

$$\zeta_i \leq \tau_i \qquad \forall i \in N_+ \qquad (83)$$

$$\zeta_i \leq m^K \varepsilon \qquad \forall i \in N \qquad (84)$$

$$\varrho_{iid} = \sum_{j \in N_C} q_j^D \sum_{h \in N_-} \gamma_{ihjd} \qquad \forall i \in N_-, d \in D \qquad (85)$$

$$\varrho_{iid} \leq m^D \qquad \forall i \in N_-, d \in D \qquad (86)$$

$$\varrho_{ijd} \leq m^D \sum_{h \in N_-} \gamma_{ihjd} \qquad \forall i \in N_-, j \in N_+, d \in D \qquad (87)$$

$$\varrho_{ihd} \geq \varrho_{ijd} - q_j^D + q_j^P - M(1 - \gamma_{ijhd}) \qquad \forall i \in N_-, j \in N_-, h \in N_+, d \in D \qquad (88)$$

$$\varphi_{i_1 jhd} \geq \varphi_{i_1 ijd} + t_{jh}^D + t_h^C - M(2 - \gamma_{i_1 ijd} - \gamma_{i_1 jhd}) \qquad \forall i \in N_-, j \in N_C, h \in N_+, i_1 \in N_{\mathrm{V}}, d \in D \qquad (89)$$

$$\varphi_{i_1 jhd} \leq \varphi_{i_1 ijd} + t_{jh}^D + t_h^C + M(2 - \gamma_{i_1 ijd} - \gamma_{i_1 jhd}) \qquad \forall i \in N_-, j \in N_C, h \in N_+, i_1 \in N_{\mathrm{V}}, d \in D \qquad (90)$$

$$\varphi_{iijd} = (t_{ij}^D + t_j^C)\gamma_{iijd} \qquad \forall i \in N_{\mathrm{V}}, j \in N_C, d \in D \qquad (91)$$

$$\varphi_{hijd} \leq f \gamma_{hijd} \qquad \forall i \in N_-, j \in N_+, h \in N_V, d \in D \qquad (92)$$

$$\varepsilon \geq \beta_{ijd} \qquad \forall d \in D, i \in N_-, j \in N_+ \qquad (93)$$

$$\varepsilon \geq \alpha_{ij} \qquad \forall i \in N_-, j \in N_+ \qquad (94)$$

$$\alpha_{0(|N_V|)} = 1 - \varepsilon \tag{95}$$

$$0 \le \mu_i^K \le |N| \qquad \forall i \in N, d \in D \tag{96}$$

$$0 \le \mu_{id}^D \le |N| \qquad \forall i \in N \tag{97}$$

$$\delta_i, \delta_i^K, \delta_{id}^D, \varphi_{ijhd}, \varrho_{ijd}, \zeta_i, \tau_i, \tau_{id}' \ge 0 \qquad \forall i, j, h \in N, d \in D \tag{98}$$

$$\alpha_{ij}, \beta_{ijd}, \sigma_{ijd}, \gamma_{ijhd}, \varepsilon \in \{0,1\} \qquad \forall i, j, h \in N, d \in D. \tag{99}$$

Objective (53) minimizes the reduced cost. The explanation for Constraints (54)–(59) and (61)–(99) is similar to the previous explanation for Constraints (2)–(7) and (9)–(47). Constraints (60) ensure that all customers should be served for at most once.

5.3 Logic-based Benders decomposition for solving the pricing problem

The above model PP_k is the shortest path problem with resource constraints. If the smallest reduced cost in all the PP_k ($k \in K$) is greater than or equal to zero, the CG is terminated, and the solution at this time is the optimal solution of the LP-RMP. The model PP_k in this paper involves more complex constraints, such as capacity constraints and time constraints. And the model PP_k takes into account the number of drones. In addition, the label algorithm usually requires traversal and maintenance of all nodes, resulting in high computational complexity. In this case, the traditional labeling algorithm can not meet the requirements of the problem, and other more complex algorithms or optimization methods need to be adopted to solve it. In order to solve the PP_k efficiently, this subsection proposes a logic-based Bender decomposition algorithm incorporated with a dynamic programming based acceleration tactic.

5.3.1 Logic-based Benders decomposition

Benders decomposition uses the concept of "divide and conquer" to decompose the original mixed integer programming model into smaller sub-models, i.e., the master problem and subproblems, which are solved independently (Piri et al., 2022). The subproblems of classical Benders decomposition are continuous linear programming problems; then the Benders cuts could be generated from the dual subproblem. While the logic-based Benders decomposition does not require the subproblem to be continuous linear programming problem because the Benders cuts are generated on the basis of some problem-specified logics. This study adopts the logic-based Benders decomposition to solve the PP_k and construct the Bender master problem (BMP) and Benders subproblem (BSP).

The BMP determines the subset of customers the truck serves, the subset of visited station nodes, and the order in which these nodes are visited. The BSP assigns the remaining customers to drones for service. First, the BMP is solved; the obtained solution is passed to the BSP; then the BSP is solved; and the tangent plane is returned to the BMP. The BMP and the BSP are solved iteratively. When the objective value of the BMP is equal to that of the BSP, the algorithm stops; and the obtained solution is the optimal solution of the original problem.

Benders subproblem (BSP)

Given a truck route r, the BSP aims to assign the remaining customers to drones for service and outputs a feasible solution for the PP_k. In model BSP, the subscript r in the parameters/variables is omitted. It is noted that the values of parameters x_{ij}, y, z_i equal to the solved values of variables $\alpha_{ij}, \varepsilon, \delta_i$ of model BMP, respectively. The model BMP is elaborated later. Each BMP corresponds to a truck group k; thus the subscript k in the BMP's parameters and variables such as $\alpha_{ij}, \varepsilon, \delta_i$ is omitted. The model formulation for the BSP is shown as follows.

[BSP] Minimize $\Big\{s^G y + \sum_{i\in N_-}\sum_{j\in N_+} s^K t_{ij}^K x_{ij} + \sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D(\beta_{icd} - \sigma_{icd}) +$

$\sum_{c\in N_C} s^T \left(\delta_c - e_c\left(\sum_{i\in N_-}\sum_{d\in D}\beta_{icd} + \sum_{i\in N_-} x_{ic} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icd}\right)\right)^+ - \sum_{j\in N_C}(\sum_{i\in N_-}\sum_{d\in D}\beta_{ijd} +$

$\sum_{i\in N_-} x_{ij} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{ijd})\,\theta_j - \pi\Big\}$ (100)

Subject to

Constraints (57)–(59), (62)–(63), (66), (68), (75)–(76), (79), (81)–(83) and (85)–(92)

$\sum_{i\in N_-}\sum_{d\in D}\beta_{icd} + \sum_{i\in N_-} x_{ic} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icd} \le 1 \quad \forall c \in N_C$ (101)

$2\sigma_{ijd} \le \beta_{ijd} + x_{ij} \le 2\sigma_{ijd} + 1 \quad \forall i \in N_-, j \in N_+, d \in D$ (102)

$\sum_{j\in N_+}\beta_{ijd} \le \sum_{j\in N_+} x_{ij} \quad \forall i \in N_V, d \in D$ (103)

$\beta_{ijd} \le x_{ij} \quad \forall d \in D, j \in N_+ \backslash N_C, i \in N_V$ (104)

$\gamma_{hijd} \le \sum_{c\in N_0} x_{hc} \quad \forall d \in D, j \in N_+, i \in N_-, h \in N_V$ (105)

$\delta_i \ge z_i \quad \forall i \in N$ (106)

$\delta_c \ge \delta_i + t_{ic}^D + t_c^C - M(1 - \beta_{icd} + \sigma_{icd} + x_{ic}) \quad \forall i \in N_-, c \in N_+, d \in D$ (107)

$\zeta_0 = \sum_{c\in N_C} q_c^D\left(\sum_{i\in N_-}\sum_{d\in D}\beta_{icd} + \sum_{i\in N_-} x_{ic} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icd}\right)$ (108)

$\tau_j \ge \zeta_i - q_j^D + q_j^P - M(1 - x_{ij}) \quad \forall i \in N_-, j \in N_+$ (109)

$\zeta_i \le m^K y \quad \forall i \in N.$ (110)

Constraints (100)–(105) and (108)–(110) are similar to Constraints (60)–(61), (64)–(65), (67), (78), (80) and (84), respectively. Constraints (106) ensure that the time when drones depart from node i is no earlier than the time when the truck departs from node i; and Constraints (107) connect the departure time of drones from two consecutively visited nodes.

Proposition 1. The BSP is feasible and bounded.

Proof: See Appendix B. ■

Benders master problem (BMP)

Proposition 1 indicates that it is sufficient to add only Benders optimality cuts in the BMP. The BMP determines the order of customers and stations the truck will visit. We define b_j as a binary variable, which equals one if customer j is served by the truck, and otherwise zero. Let ϖ be a non-positive continuous variable. b_j and ϖ are constrained by the results of BSP. We record the objective value of BSP in the l^{th} iteration, and calculate value of Φ_l as $\Phi_l = \sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D(\beta_{icdl} - \sigma_{icdl}) + \sum_{c\in N_C} s^T \left(\delta_{cl} - e_c\left(\sum_{i\in N_-}\sum_{d\in D}\beta_{icdl} + b_c - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icdl}\right)\right)^+ - \sum_{j\in N_C}(\sum_{i\in N_-}\sum_{d\in D}\beta_{ijdl} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{ijdl} + b_j)\,\theta_j$. Then, we have:

$$\varpi \geq \Phi_l \qquad \forall l \in \{1,2,\cdots,L\} \qquad (111)$$

Where L is the maximum number of iterations. Formulas (111) are the Benders optimality cuts. The proof is provided in Appendix C. ■

In addition, in the process of iteration, there will be cases where the objective of the BMP before and after the iteration is unchanged, and the objective of the BSP is unchanged, but the objective of the BMP is not equal to the objective of the BSP. Thus, it is necessary to change the route of the truck to affect the route of the drone to achieve the optimal effect. When the objective of the BMP does not change, we add some constraints for the BMP to constrain b_i. Let ϑ_{il} represents whether the truck group serves customer i in the l^{th} iteration. If the truck group serves customer i, then $\vartheta_{il} = 1$, otherwise $\vartheta_{il} = 0$. Constraints (112) can achieve the effect of changing the number of customers served by the truck. In each iteration, we add constraints (113) that limit the route of the truck, which leads to better results. Thus, we add constraints for the BMP as follows.

$$\sum_{i\in N_C} b_i \leq \max\{\sum_{i\in N_C}\vartheta_{il} - l, 0\} \qquad \forall l \in \{1,2,\cdots,L\} \qquad (112)$$

$$\sum_{i\in N_V} b_i \geq l + 1 \qquad \forall l \in \{1,2,\cdots,L\}. \qquad (113)$$

By summarizing the above, the BMP is formulated as follows.

[BMP] $\text{Minimize}\left\{s^G + \sum_{i\in N_-}\sum_{j\in N_+} s^K t_{ij}^K \alpha_{ij} - \sum_{j\in N_C} b_j\theta_j - \pi + \varpi\right\}$ (114)

Subject to

Constraints (54)–(56), (75), (77), (79), (96), (111), (112) and (113)

$$\sum_{i\in N_-}\alpha_{ic} \leq 1 \qquad \forall c \in N_C \qquad (115)$$

$$\delta_j \geq \delta_i + t_{ij}^K + t_j^C - M\left(1 - \alpha_{ij}\right) \qquad \forall i \in N_-, j \in N_+ \qquad (116)$$

$$\delta_i \leq e_i + M\left(1 - \alpha_{ij}\right) \qquad \forall i \in N_C \qquad (117)$$

$$\zeta_0 = \sum_{c\in N_C} q_c^D\left(\sum_{i\in N_-}\alpha_{ic}\right) \qquad (118)$$

$$\zeta_j \geq \zeta_i - q_j^D + q_j^P - M(1 - \alpha_{ij}) \qquad \forall i \in N_-, j \in N_+ \qquad (119)$$

$$\zeta_i \le m^K \qquad \forall i \in N \qquad (120)$$

$$\alpha_{0(|N_V|)} = 0 \qquad (121)$$

$$b_i \ge \sum_{j \in N_+} \alpha_{ij} \qquad \forall i \in N \qquad (122)$$

$$\delta_i \ge 0 \qquad \forall i \in N \qquad (123)$$

$$\alpha_{ij}, b_j \in \{0,1\} \qquad \forall i, j \in N. \qquad (124)$$

$$\varpi \le 0 \qquad (125)$$

Constraints (115) indicate that each customer is served by a truck at most once. Constraints (116) and (117) limit the departure time of trucks from two consecutively visited nodes. Constraints (118)–(120) are truck capacity constraints. Constraints (121) restrict the existence of a truck route. Constraints (122) link the variables b_i and α_{ij}. Constraints (123)–(125) define the domains of the decision variables.

The pseudocode of the logic-bases Benders decomposition for solving the PP_k is provided as follows.

Logic-based Benders decomposition for solving the model PP_k

Initialize $obj\_BSP = +\infty,\ obj\_BMP = -\infty,\ \epsilon = 0.001$

While $obj\_BSP > obj\_BMP + \epsilon$ **do**

  Solve the BMP by using CPLEX, let obj_BMP denote the optimal objective value of the BMP, and the solved values of $\alpha_{ij}, \varepsilon, \delta_i$ in the BMP are assigned to x_{ij}, y, z_i, respectively.

  Solve $|D|$ submodels of BSP (i.e., BSP_d, $d \in D$) by using the dynamic programming, let obj_BSP denote the optimal objective value of the BSP, and record the value of Φ_l.

  // The submodel BSP_d as well as the dynamic programming are elaborated in next subsection.

  Add cuts (112) and (113) to the BMP.

End while

Add the obtained route result to the RMP as a new column.

5.3.2 Dynamic programming for solving the BSP

Due to the complexity of BSP, it is still a bit time consuming to use the CPLEX to solve the BSP with large-scale instances during a short computation time. This study proposes an idea of further decomposing the BSP according to dimension of drones. More specifically, the BSP is divided into $|D|$ sub-models, each of which corresponds to one drone. The sub-model is denoted as BSP_d, which is established for drone d, $d \in D$. Define the RD as the set of customers that have already been served by the truck group, $RD \subseteq N_C$. Define m^{KD} as the truck group's remaining load of cargos, which can be picked up and delivered by drone d. The sub-model BSP_d is formulated as follows.

[BSP_d] Minimize$\left\{\frac{s^G}{|D|} y + \frac{s^K}{|D|} \sum_{i \in N_-} \sum_{j \in N_+} t_{ij}^K x_{ij} + \sum_{i \in N_-} \sum_{c \in N_+} s^D t_{ic}^D (\beta_{ic} - \sigma_{ic}) + \sum_{c \in N_C} s^T \left(\delta_c - \right.\right.$

$$e_c\left(\sum_{i\in N_-}\beta_{ic}+\sum_{i\in N_-}x_{ic}-\sum_{i\in N_-}\sigma_{ic}\right)\Big)^+-\sum_{j\in N_C}\theta_j\sum_{i\in N_-}\left(\beta_{ij}+x_{ij}-\sigma_{ij}\right)-\frac{\pi}{|D|}\Big\} \quad (126)$$

Subject to

Constraints (57)–(59), (62)–(63), (68), (75)–(76), (81)–(83), (85)–(92), (97) and (101)–(109) to remove d dimension

$$\sum_{i\in N_-}\beta_{ic}+\sum_{i\in N_-}x_{ic}-\sum_{i\in N_-}\sigma_{ic}=0 \qquad \forall c\in RD \quad (127)$$

$$\zeta_i\le m^{KD} \qquad \forall i\in N_+ \quad (128)$$

$$\zeta_0\le m^{KD}. \quad (129)$$

Objective (126) minimizes the reduced cost. Since BSP is decomposed into $|D|$ sub-models, the sum of the objective (126) of sub-model BSP_d should be greater than or equal to the objective (100) of BSP. Constraints (127) ensure that customers who have been served cannot be arranged in the model. Constraints (128) and (129) are truck group capacity constraints.

Dynamic programming algorithm is widely used to solve the shortest path problem with resource constraints. Because of the coupling relationship between multiple drones, PP_k is difficult to be solved by the widely used dynamic programing, although the PP in the CG is usually solved by the dynamic programing based algorithm (or labeling algorithm) in the CG related literature. In this study, the logic-based Benders decomposition is adopted to solve the PP_k; and the BSP in the Benders decomposition is decomposed again according to the dimension of drones. For each drone d, the submodel BSP_d is a combinatorial optimization problem, which can be solved by using dynamic programming algorithm.

Dynamic programming is a method to solve multi-stage decision problem. The multistage decision problem can be divided into several interrelated stages, each corresponding to this set of alternative decisions, and the selection of each decision depends on the current state. The submodel BSP_d of this paper solves the route problem of cooperative delivery between a truck and a drone, which is a multi-stage decision problem. The aim of the dynamic programing is to plan a drone's route given its truck group's route. The total number of nodes is the total number of stages in the dynamic programing. The sequence of stages is the increasing order of nodes' indices. For example, suppose there are five customers and three station nodes, and station nodes are indexed from numbers 1 to 3, and customers are indexed from numbers 4 to 8; then the problem can be divided into eight stages.

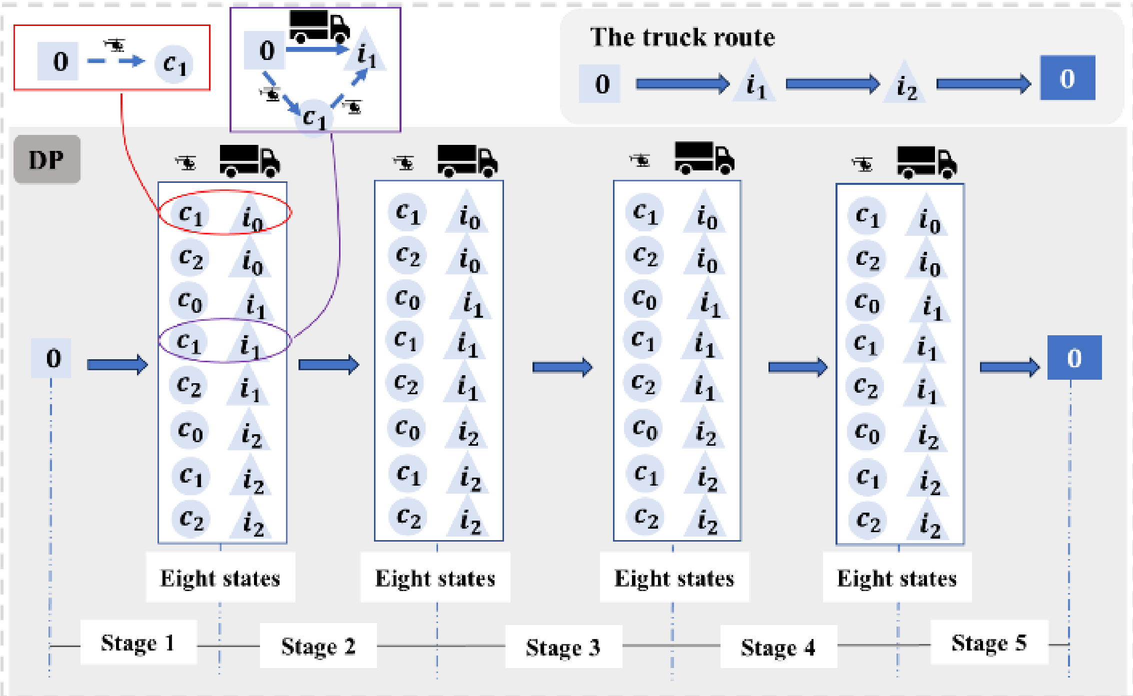

Figure 4: Stage and state partitioning of the dynamic programming algorithm

In each stage, we define the state representation as a combination of a customer node and a station node. We define S_m as the set of states in the m^{th} stage, define Z_{mn} as the n^{th} state in the m^{th} stage, $Z_{mn} \in S_m$. $Z_{mn} = \{i, j\}$, where $i \in N_c, j \in N_+$. When the customer index i corresponds to 0, it means that no customer is served in this stage; when the station index j is 0, it means that this stage does not include the station. Figure 4 shows that the stage and state partitioning for two customers and three stations. If the truck route {depot $\rightarrow i_1 \rightarrow i_2 \rightarrow$ depot} is obtained by CPLEX solving BMP. In each stage, there are eight states.

In addition, we define $u_m(Z_{mn})$ as the decision at the n^{th} state in the m^{th} stage, and $v\big(u_m(Z_{mn})\big)$ is defined as the part of the value of the reduced cost when the decision $u_m(Z_{mn})$ is taken in the m^{th} stage. The calculation of the above value v_m is $v\big(u_m(Z_{mn})\big) = \tilde{c}_{mn} - \sum_{i \in Z_{mn}} \theta_i$. $\tilde{c}_{mn}$ is the travel cost of the drone when the decision $u_m(Z_{mn})$ is taken in the m^{th} stage. The calculation of the above $\tilde{c}_{mnn'}$ is as follows. If truck and drone visit customer or station node together, $\tilde{c}_{mn} = 0$. If i_1 is the customer or station node in the $(m-1)^{\text{th}}$ stage, i is the customer node in n^{th} state in the m^{th} stage, and there is no station node in n^{th} state in the m^{th} stage, we have $\tilde{c}_{mn} = s^D(t^D_{i_1 i} + t^C_i)$. And if i_1 is the customer or station node in the $(m-1)^{\text{th}}$ stage, i is the customer node in n^{th} state in the m^{th} stage, and j is the station node in n^{th} state in the m^{th} stage, $\tilde{c}_{mn} = s^D\big(t^D_{i_1 i} + t^C_i + t^D_{ij} + t^C_j\big)$. Since the value of the dual variable π the fixed cost of the truck group and the truck traveling cost do not change in the process of dynamic programming, the value $v\big(u_m(Z_{mn})\big)$ takes into account the variable reduced cost. The formula for $v\big(u_m(Z_{mn})\big)$ is as follows.

$$v\big(u_m(Z_{mn})\big)=\begin{cases}0-\theta_i & case\ 1\\ s^D(t_{i_1 i}^D+t_i^C)-\theta_i & case\ 2\\ s^D\big(t_{i_1 i}^D+t_i^C+t_{ij}^D+t_j^C\big)-\theta_i-\theta_j & case\ 3.\end{cases} \tag{130}$$

Where, case 1 is the case that truck and drone visit customer or station node together; case 2 and case 3 are the cases that drone serves customer i and truck does not serve customer i. In case 2, i_1 is the customer or station node in the $(m-1)^{\text{th}}$ stage, i is the customer node in n^{th} state in the m^{th} stage, and there is no station node in n^{th} state in the m^{th} stage. In case 3, i_1 is the customer or station node in the $(m-1)^{\text{th}}$ stage, i is the customer node in n^{th} state in the m^{th} stage, and j is the station node in n^{th} state in the m^{th} stage.

The route of drones can be obtained by the following recursive equation. For the first stage, the calculation formula of the variable reduced cost under different states is as follows.

$$f_1(Z_{1n})=\begin{cases}0-\theta_i & case\ 1\\ s^D\big(t_{i_1 i}^D+t_i^C\big)-\theta_i & case\ 2\\ s^D\big(t_{i_1 i}^D+t_i^C+t_{ij}^D+t_j^C\big)-\theta_i-\theta_j & case\ 3.\end{cases} \tag{131}$$

Where, case 1,2 and 3 are the same as case 1,2 and 3 of formula (130).

For the following stages (e.g., the m^{th} stage, $m=2,\cdots,|N|$), the value of variable reduced cost under different states (e.g., the state n, $n=0,1,\cdots,|N*N|$) is calculated as:

$$f_m(Z_{mn})=\min\big\{v_m\big(u_m(Z_{mn})\big)+f_{m-1}\big(Z_{m-1,x}\big)\big\}. \tag{132}$$

The pseudocode of the dynamic programming to solve the BSP_d is provided in Appendix D.

5.4 Strategy for obtaining feasible solutions

As the usual practice of the CG, the solution for the LR-RMP may not be integer at the end of CG procedure. Therefore, we need design a useful and efficient strategy for output feasible integer solution for the RMP on the basis of the column pool obtained by the previous CG procedure.

From the set of columns obtained by solving the LR-RMP and PP (i.e., the CG procedure), the plan (column) with the most customers (i.e., the primal criteria) and the lowest cost (i.e., the secondary criteria) is selected, and is assigned to a truck group; then some sets of unserved customers, unused truck groups are updated. The above process is repeated until each truck group is assigned a plan or each customer is served. The detailed steps in the strategy is shown as follows.

Step 1: Define P is the set of customers who has not been served, $P\subset N_C$, R' is the set of plans (columns) generated by solving the LR-RMP and PP (i.e., the CG procedure), and PT is the set of truck groups that have not been used.

Step 2: Run the CG procedure to obtain set R'.

Step 3: From the set R', select the plan r that includes the most customers to serve; if there are several plans with the same number of customers, the plan with the lowest cost among them is selected. Remove the customers, which are contained in the above selected plan, from the set P. Assign the selected plan to a truck group in set PT, and then remove the truck group from the set PT.

Step 4: Repeat Steps 2 and 3 until $PT = \emptyset$ or $P = \emptyset$.

6 Numerical experiments

This section presents numerical experiments for evaluating the performance of the proposed hybrid algorithm that combines the CG and the logic-based Benders decomposition. In addition, some potentially useful managerial insights are derived from a series of sensitivity analysis experiments. All experiments are performed on a workstation with two Xeon E5-2680 V4 CPUs (12 cores) running at 2.4 GHz with 256 GB of memory under Windows 10. The proposed models and algorithms are implemented in C# (Visual Studio 2019). The time limit for all test instances is set to one hour.

6.1 Experimental settings

The experiments in this study are conducted in the context of an urban circle with a radius of 30km in Hangzhou, which is the most pioneering city in China to promote the mode of "urban drone delivery". Some representative companies (i.e., the urban drone delivery service providers) such as Antwork® has emerged in Hangzhou city and has been dedicated to constructing urban aerial drone operation network in cities. The truck-and-drone cooperative delivery and pickup system is a system that combines trucks and drones to achieve efficient delivery. The system first receives the cargo information and processes it. Based on customer requirements and vehicle availability, the system then assigns the cargo to the appropriate truck group for transport. This process can take into account factors such as geographical location, vehicle load and traffic conditions. During the transportation of the truck, the system will dispatch the drone on the truck to coordinate the delivery as needed, and the system can provide real-time monitoring and tracking to ensure the safety and timely delivery of the goods to the destination.

The experiments assume that each truck carries three drones. The speed of trucks is 30 km/h, the speed of drones is 48 km/h, the $m^K$and m^D, i.e., the maximum loading capacity of a truck and a drone, are set as 100 kg and 30kg, respectively. Customer demands to be delivered and picked up are randomly generated in the interval (0, 20] kg. We assume the gasoline price is 8.24 Chinese Yuan (CNY)/L, and the average fuel consumption of a truck is 0.10 L/km, so we set s^K as 24.72 CNY/h. In addition, we set s^D as 9.12 CNY/h, s^W as 1000 CNY/h, and s^G as 100 CNY per truck group; the setting of parameters is based on price of gasoline used by trucks and some related reference such as Wang and

Sheu (2019). The number of station nodes $|N_V|$ is 10. There are six instance groups (ISGs) with different scale; among the ISG 1, 2, …, and 6, the number of truck groups K ranges from 1 to 5, and the number of customers N_C ranges from 5 to 25. For each instance group, we set 5 instances.

Table 2: Experimental setting for six instance groups

Group ID	Number of customers ($|N_C|$)	Number of truck groups ($|K|$)
ISG1	5	1
ISG2	8	2
ISG3	10	2
ISG4	15	3
ISG5	20	4
ISG6	25	5

The settings for the six ISGs are shown in Table 2. It is noted that the ISG6 contains 25 customer nodes does not mean the city contains only 25 customer nodes; it just means there are 25 customers in this batch of order. Figure 5 illustrate the locations of customers distributed in the city and an example of the five truck groups' routes as well as the drones' routes for serving customers. In Figure 5, there are more than 25 customer nodes, and it can be seen from the route in the figure that 5 truck groups serve 25 customers, and the rest customers complete the delivery service in the next batch of order.

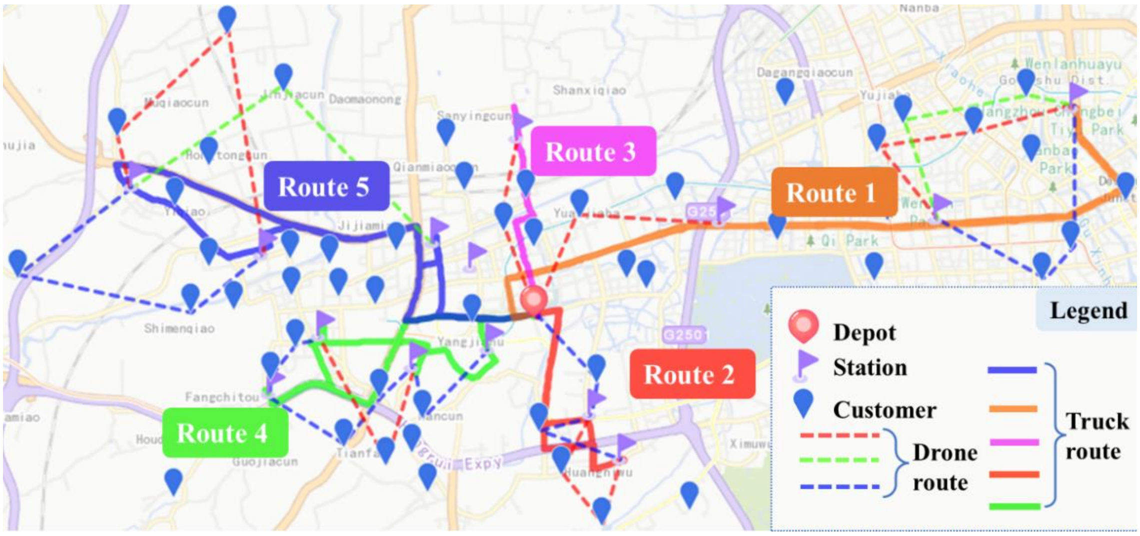

Figure 5: An example on planned routes of five truck groups and drones in Hangzhou city

6.2 Benefit of the logic-based Benders decomposition and the dynamic programming

The proposed hybrid algorithm combines several components such as the CG, the logic-based Benders decomposition, and the dynamic programming. First, experiments are conducted to investigate whether some algorithmic components are effectiveness in our proposed hybrid algorithm.

The main framework of our proposed algorithm is the CG. For solving the CG's PP, we have three different ways. The first way is to use the CPLEX to solve the CG's PP directly. The second way is to use the logic-based Benders decomposition (elaborated in Section 5.3.1) to solve the CG's PP; the subproblem contained in the Benders decomposition is solved by the CPLEX. The third way is to use the logic-based Benders decomposition to solve the CG's PP; the subproblem contained in the Benders decomposition is solved by the proposed dynamic programming (elaborated in Section 5.3.2). The objective values of the solutions obtained by the above three ways are denoted by F_1, F_2, F_3; and the solution time (in seconds) of the three ways are denoted by t_1, t_2, t_3 in Table 3. In the gap values of Table 3, Δ_{F_1} and Δ_{t_1} reflect the performance of the logic-based Benders decomposition (without dynamic programing) in the aspect of the solution quality (i.e., optimality gap in solving the PP) and the acceleration benefit in the solution time. Δ_{F_2} and Δ_{t_2} reflect the performance of the logic-based Benders decomposition incorporated with dynamic programming in the aspect of the solution quality and the acceleration benefit in the solution time. Δ_{F_3} and Δ_{t_3} reflect the benefit of incorporating the dynamic programming into the logic-based Benders decomposition in the aspect of the solution quality and the acceleration in the solution time.

Table 3: Comparison of three ways for solving the CG's PP (ISG1 & 2)

Instances		**Objective values and solution time**						**Gap**					
Scale	ID	F_1	t_1(s)	F_2	t_2(s)	F_3	t_3(s)	Δ_{F_1}	Δ_{t_1}	Δ_{F_2}	Δ_{t_2}	Δ_{F_3}	Δ_{t_3}
ISG1	1	104	301	104	88	108	6	0.00%	-70.76%	3.85%	-98.01%	3.85%	-93.18%
	2	104	405	104	77	108	6	0.00%	-80.99%	3.85%	-98.52%	3.85%	-92.21%
	3	104	278	104	93	106	5	0.00%	-66.55%	1.92%	-98.20%	1.92%	-94.62%
	4	104	227	104	149	107	7	0.00%	-34.36%	2.88%	-96.92%	2.88%	-95.30%
	5	104	487	104	103	108	7	0.00%	-78.85%	3.85%	-98.56%	3.85%	-93.20%
ISG2	6	--	3600	105	542	109	26	n/a	n/a	n/a	n/a	3.81%	-95.20%
	7	--	3600	104	406	109	20	n/a	n/a	n/a	n/a	4.81%	-95.07%
	8	--	3600	107	410	109	19	n/a	n/a	n/a	n/a	1.87%	-95.37%
	9	--	3600	205	427	209	19	n/a	n/a	n/a	n/a	1.95%	-95.55%
	10	--	3600	207	387	209	27	n/a	n/a	n/a	n/a	0.97%	-93.02%
Average								0.00%	-66.30%	3.27%	-98.04%	2.97%	-94.27%

Notes:(1) F_1, F_2, F_3 is the objective of solution obtained by the way for solving CG's PP by CPLEX, the normal logic-based Benders decomposition, the logic-based Benders decomposition in which its subproblem is solved by dynamic programming, respectively. And t_1, t_2, t_3 is the computation time of the above three ways. (2) $\Delta_{F_1}= (F_2 - F_1)/F_1$, $\Delta_{F_2}= (F_3 - F_1)/F_1$, $\Delta_{F_3}= (F_3 - F_2)/F_2$, $\Delta_{t_1}= (t_2 - t_1)/t_1$, $\Delta_{t_2}= (t_3 - t_1)/t_1$, $\Delta_{t_3}= (t_3 - t_2)/t_2$. (3) "--" means the proposed algorithm cannot obtain a feasible solution when the computation time reaches one hour. And "n/a" means no applicable.

According to the gap values of Δ_{F_1} and Δ_{t_1} in Table 3, it demonstrates that the logic-based Benders decomposition can solve the CG's PP optimally while much shorter computation time is used. The gap values of Δ_{F_3} show that incorporating the dynamic programming into the logic-based Benders decomposition will bring a bit loss of optimality, which is about 2.97% on average; however, the computation time of the whole algorithm is significantly reduced. As demonstrated by the average value of Δ_{t_3} gap in Table 3, the dynamic programming can help save the computation time by about 94.27% on average.

More experiments are conducted on the basis of ISG3 and ISG4 to further compare the above mentioned second way and the third way. Because the first way that use the CPLEX to solve the CG's PP is very time consuming, we only compare the second and the third ways; and the comparative results are shown in Table 4. The gap values of Δ_{F_3} and Δ_{t_3} in Table 4 further validate the benefit of incorporating the dynamic programming into the logic-based Benders decomposition. More specifically, the gap values of Δ_{F_3} show that incorporating the dynamic programming into the logic-based Benders decomposition brings a loss of optimality by about 1.27% on average; however, the computation time of the whole algorithm is significantly reduced; the dynamic programming can help save the time by about 96.79% on average. The about ninety percent saving of the computation time under the price of about one percentage loss of optimality validates the necessity of the proposed dynamic programming.

Table 4: Benefit of dynamic programing in solving Benders decomposition's subproblem (ISG3 & 4)

Instances		**Objective values and solution time**				**Gap**	
Scale	ID	F_2	t_2(s)	F_3	t_3(s)	Δ_{F_3}	Δ_{t_3}
ISG3	1	206	1261	214	26	3.88%	-97.94%
	2	207	1174	208	21	0.48%	-98.21%
	3	206	1531	210	29	1.94%	-98.11%
	4	205	1243	209	27	1.95%	-97.83%
	5	206	735	209	30	1.46%	-95.92%
ISG4	6	--	3600	318	68	n/a	n/a
	7	--	3600	314	57	n/a	n/a
	8	--	3600	213	80	n/a	n/a
	9	--	3600	315	58	n/a	n/a
	10	--	3600	315	103	n/a	n/a
Average						1.27%	-96.79%

Notes:(1) F_2, F_3 is the objective of solution obtained by the way for solving CG's PP by the logic-based Benders decomposition, the logic-based Benders decomposition with using dynamic programming to solve the Benders decomposition's subproblem, respectively. And t_2, t_3 is the computation time of the above two ways. (2) $\Delta_{F_3}= (F_3 - F_2)/F_2$, $\Delta_{t_3}= (t_3 - t_2)/t_2$. (3) "--" means the proposed algorithm cannot obtain a feasible solution when the computation time reaches one hour. And "n/a" means no applicable.

6.3 Solution quality of the proposed hybrid algorithm

The previous experiments validate the effectiveness of the components (the logic-based Benders decomposition and dynamic programming) contained in the proposed hybrid algorithm. Then further experiments are conducted to validate the solution quality of our proposed hybrid algorithm. More specifically, we solve the original model by the CPLEX directly, and obtain the solution's objective value as the optimal result; then we use the proposed hybrid algorithm to solve the same instance and calculate the gap between the result obtained by our algorithm and the optimal result. The optimality gap is reflected by the values Δ_{F_A} in Table 5, which indicates the optimality gap of the whole algorithm in small-scale instances is about 3.32%. It is noted that the solution time of our proposed algorithm is much shorter than the CPLEX, which is reflected by the values Δ_{t_A}. The results in Table 5 demonstrate that our proposed algorithm can save about 86.19% solution time than the CPLEX while the optimality gap is just 3.32% on average, which validates the efficiency of the proposed algorithm.

Because the CPLEX cannot solve the large-scale instances within a short time, the optimal results as well as the optimality gap cannot be obtained for evaluating our algorithm in the large-scale instances. Thus this study proposes a lower bound (LB) as an benchmark in the large-scale experiments. The details on the mathematical model for calculating the LB is elaborated in Appendix E. Before using the LB in the large-scale experiments, we first evaluate the optimality gap of the LB in the small-scale experiments, which is reflected by the Δ_{CPLEX} values in Table 5. It demonstrates that the optimality gap of the LB is about 0.96% on average, which validates the rationality of this proposed LB for the further usage as an benchmark in the large-scale experiments. Lastly, we also calculate the gap between our algorithm's result and the LB; the average of the Δ_{LB} gap values is about 3.96%, which is a bit larger than the above mentioned value (i.e., 3.32%) of the optimality gap of the solutions solved by our algorithm. The value of Δ_{LB} gap 3.96% will also be compared with the result in the large-scale experiment based solution quality evaluation.

Table 6 illustrates the algorithmic performance in the large-scale experiments. The Δ_{LB} values in Table 6 show that the objective value of the solution solved by our proposed algorithm deviates from the LB by about 3.93% on average. Recall that the gap Δ_{LB} in small-scale experiments is about 3.96%. Thus the results in Table 6 demonstrate that our proposed algorithm could obtain near-optimal results within a reasonable time; and the solution quality in the large-scale experiments is as stable as the quality in the small-scale experiments.

Table 5: Performance of the proposed algorithm in small-scale instances

Instances		**CPLEX**		**LB**	**Proposed algorithm**		**Gap**			
Scale	ID	F_{CPLEX}	t_{CPLEX}(s)	F_{LB}	F_A	t_A(s)	Δ_{F_A}	Δ_{t_A}	Δ_{CPLEX}	Δ_{LB}
	1	104	21	103	108	6	3.85%	-71.43%	0.96%	4.85%
	2	104	26	103	108	6	3.85%	-76.92%	0.96%	4.85%
ISG1	3	104	17	103	106	5	1.92%	-70.59%	0.96%	2.91%
	4	104	49	103	107	7	2.88%	-85.71%	0.96%	3.88%
	5	104	31	103	108	7	3.85%	-77.42%	0.96%	4.85%
ISG2	6	104	1327	103	109	26	4.81%	-98.04%	0.96%	5.83%

	7	226*	3600	103	109	20	n/a	n/a	n/a	5.83%
	8	104	863	103	109	19	4.81%	-97.80%	0.96%	5.83%
	9	205	1518	203	209	17	1.95%	-98.88%	0.98%	2.96%
	10	205	1672	203	209	18	1.95%	-98.92%	0.98%	2.96%
ISG3	11	206*	3600	204	214	28	n/a	n/a	n/a	4.90%
	12	361*	3600	204	208	21	n/a	n/a	n/a	1.96%
	13	206*	3600	204	210	29	n/a	n/a	n/a	2.94%
	14	207*	3600	204	209	27	n/a	n/a	n/a	2.94%
	15	206*	3600	204	209	30	n/a	n/a	n/a	2.45%
Average							3.32%	-86.19%	0.96%	3.96%

Notes: (1) F_{LB} is the objective value of the LB by the CPLEX, F_{CPLEX} and F_A denote the objective value of solving the original model by the CPLEX and our proposed algorithm, respectively. And t_{CPLEX} and t_A denote the computation time of CPLEX and the algorithm to solve the model, respectively. (2) $\Delta_{F_A}= (F_A - F_{CPLEX})/F_{CPLEX}$, $\Delta_{t_A}= (t_A - t_{CPLEX})/t_{CPLEX}$, $\Delta_{CPLEX}= (F_{CPLEX} - F_{LB})/F_{CPLEX}$, $\Delta_{LB}= (F_A - F_{LB})/F_{LB}$. (3) The value with the superscript "*" is the objective value of a solution obtained by the CPLEX when the computation time reaches one hour (3600 seconds). And "n/a" means no applicable.

Table 6: Performance of the proposed algorithm in large-scale instances

Instances		LB	Algorithm		Gap
Scale	ID	F_{LB}	F_A	t_A(s)	Δ_{LB}
ISG4	1	305	318	68	3.28%
	2	305	314	57	3.28%
	3	204	213	80	4.41%
	4	204	315	58	3.92%
	5	204	315	103	4.90%
ISG5	6	306	321	131	3.92%
	7	306	317	128	4.25%
	8	306	319	142	4.58%
	9	306	316	119	4.25%
	10	306	320	105	3.92%
ISG6	11	407	424	286	4.18%
	12	407	423	241	3.69%
	13	407	423	246	4.18%
	14	407	424	272	3.69%
	15	408	424	255	3.43%
Average					3.93%

Notes: (1) F_{LB} is the objective value of the LB by the CPLEX, F_A is the objective value of solving the original model by our proposed algorithm, and t_A denote the computation time of the algorithm to solve the model. (2) $\Delta_{LB}= (F_A - F_{LB})/F_{LB}$.

The above experiments validate the performance of the proposed hybrid algorithm. Then, we design a common heuristics algorithm based on neighborhood search, and further verify the quality of the solution obtained by this heuristics algorithm and the proposed hybrid algorithm. The pseudocode of the heuristics algorithm based on neighborhood search is stated in Appendix G. Experiments are conducted on six sets of instances with different scales and the results are shown in Figure 6. The results in Figure 6 show that the proposed hybrid algorithm can be solved with lower cost compared to the heuristic algorithm. Thus the hybrid algorithm designed in this paper combining column generation and logic-based Benders decomposition outperforms the designed heuristic algorithm to obtain an approximate solution close to the global optimal solution.

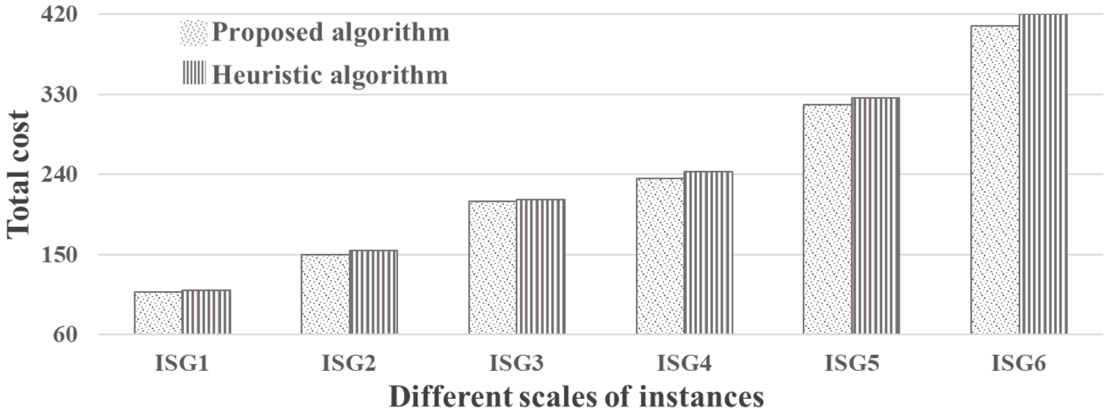

Figure 6: Comparison between the proposed algorithm and the heuristic algorithm

6.4 Managerial insights based on sensitivity analysis

This subsection performs some series of sensitivity analysis to investigate several important parameters' influence on the system's performance. These investigated parameters include the number of drones carried by each truck, the drone speed, drone's loading capacity and endurance, and the customers' time windows for service, all of which may have influence on this truck-and-drone cooperative delivery & pickup system's final performance (i.e., the total cost). Based on the results of the sensitivity analysis, some potentially useful managerial insights could be obtained.

6.4.1 Influence of the number of drones carried by each truck

The first series of sensitivity analysis is conducted by setting the number of drones carried by each truck as two, three, four, five, and six. The experiments are conducted on six groups of instances with different problem scales; and the results are shown in Figure 7. Although the shape of the six curves in Figure 7 is not identical, the trends of the curves are similar; the more drones are carried by each truck, a no greater cost could be realized by the system. It seems that carrying more drones on each truck does not surely decrease the delivery cost (or improves the system's performance).

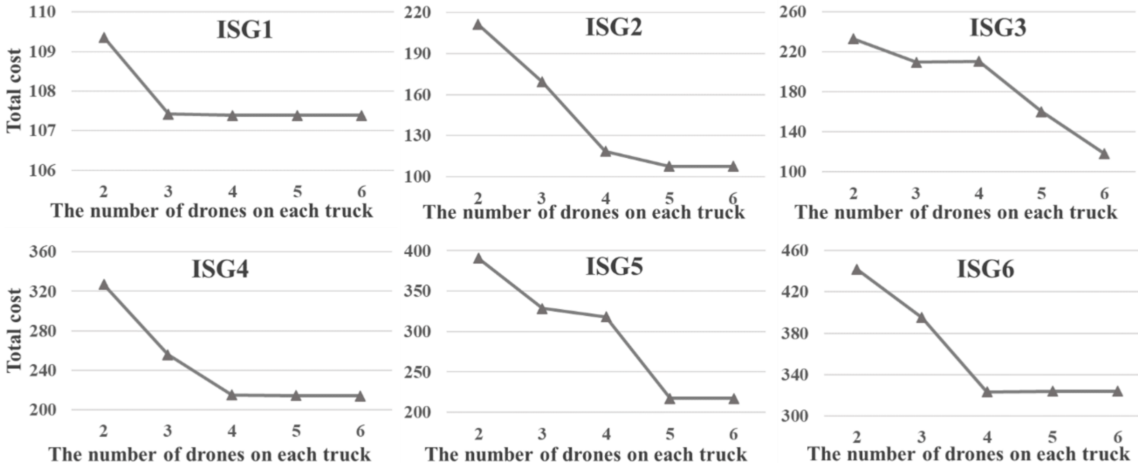

Figure 7: Influence of the number of drones carried by each truck

6.4.2 Influence of the drone speed

The drone speed is an important performance index for the drones. As mentioned in Section 6.1, the drone speed is set as 48 km/h for the previous experiments. In this series of sensitivity analysis, the drone speed is adjusted by increasing or decreasing; more specifically, the speed is set as 44, 46, 48, 50, and 52 km/h. The experiments are also conducted on six groups of instances with different problem scales; and the results are shown in Figure 8.

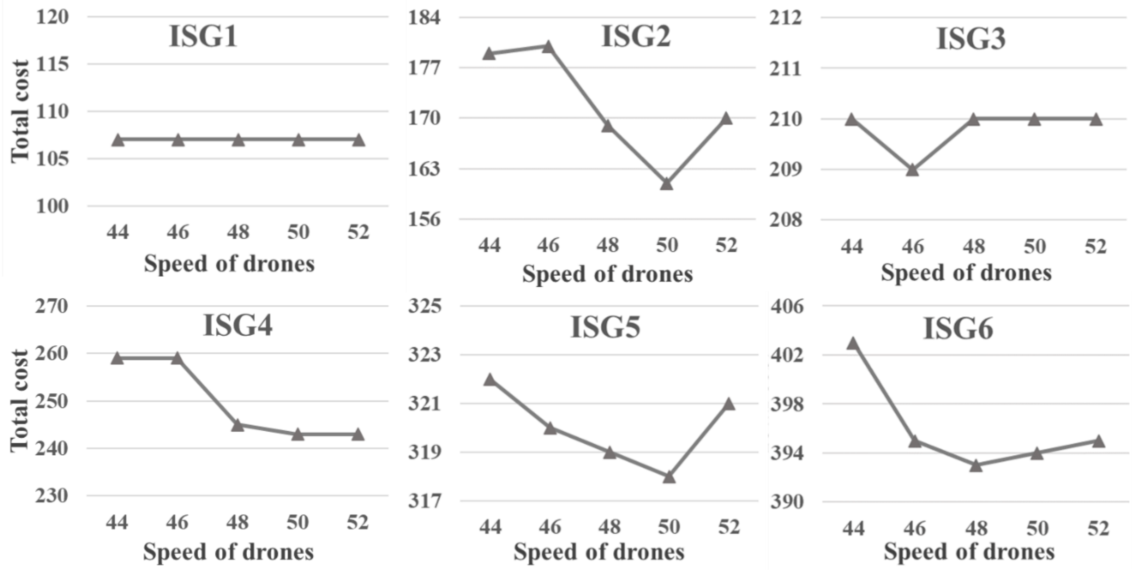

Figure 8: Influence of the drone speed

When the problem scale is very small in instances such as the ISG1, the drone speed seems not affect the system's total delivery cost. When the problem scale is growing, the trends of these curves have become more and more significant, and are demonstrated as the U shape. When the drone speed grows, the total cost is decreasing first and then becomes increasing. The results show that the high speed of drones does not always bring a better performance to the system; and there exists a certain value for the drone speed, at which the cooperative system's total delivery cost could be the lowest. The reason for the above phenomenon (i.e., faster drone may not realize better system performance) may lie in the existence of the customers' required time windows for service in this study's problem. When the speed of the drone increases, the drone is theoretically able to deliver the goods to the customer faster. However, if the drone speed is too high, it may increase the complexity of the handover, resulting in the inability to arrive on time at the customer location due to the customer's time window requirements, which will result in late arrival penalty costs and an increase in the total cost. As a result, the routes of the drones will change as the drone speed changes and the associated costs may increase or decrease.

6.4.3 Influence of the trade-off between loading capacity and endurance of drones

For drones, the loading capacity and the endurance (i.e., a drone's maximum flying time of one trip) are two types of importance performance index; and these two parameters are inverse proportional to each other if the drones are kept with the similar size and similar equipped battery power. It means given a certain size of drone or a certain battery power of drone, the higher is the drone's loading capacity, the shorter is its endurance, and vice versa. Thus the third series of sensitivity analysis is conducted to investigate the trade-off between the loading capacity and the endurance of drones. Five settings of the trade-off (i.e., the combinations of loading capacity and endurance) are used in this series of sensitivity analysis. More specifically, the five combinations, i.e., loading capacity (kg) / endurance (hour) are: 25/0.600, 30/0.500, 35/0.429, 40/0.375 and 45/0.333. For reflecting the above mentioned trade-off relationship between the two parameters, the product of the two parameters (e.g., 25×0.6, 30×0.5) in the five combinations is identical approximately. The experiments are also conducted on six groups of instances with different problem scales; and the results are shown in Figure 9.

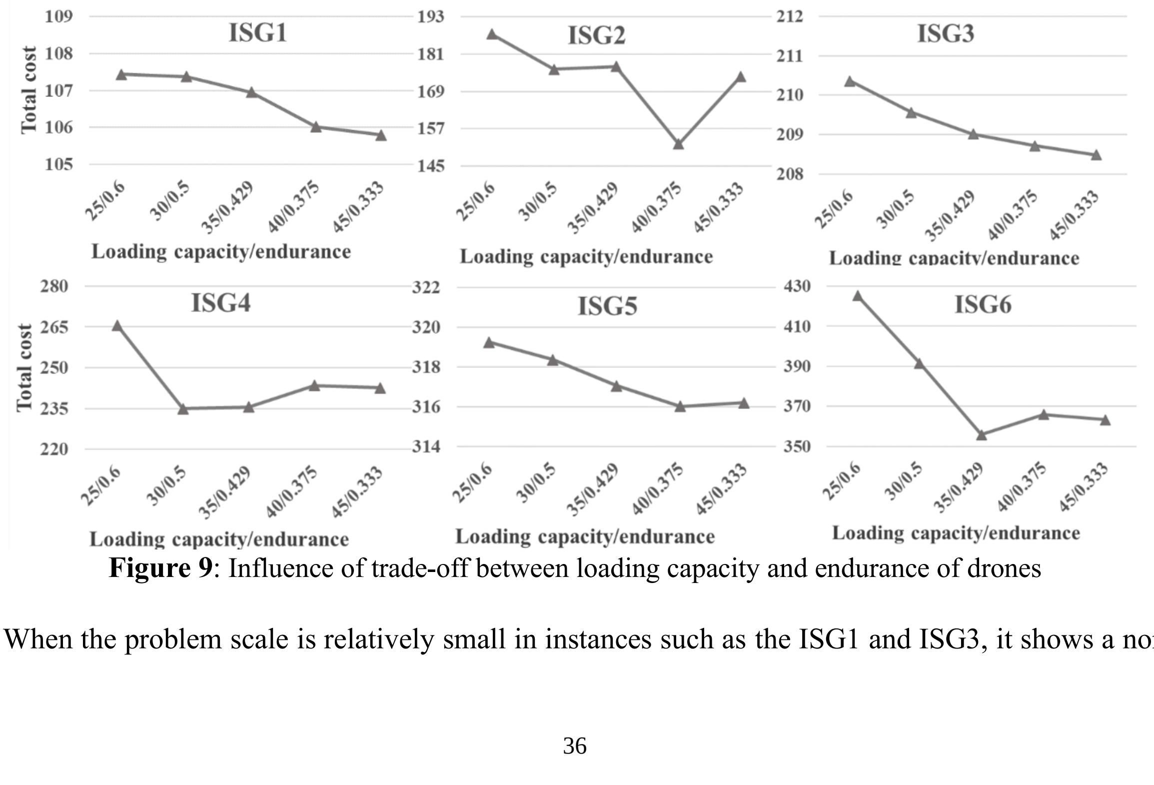

Figure 9: Influence of trade-off between loading capacity and endurance of drones

When the problem scale is relatively small in instances such as the ISG1 and ISG3, it shows a nonincreasing trend along with the drones' loading capacity increasing (or the drones' endurance decreasing). When the problem scale is relatively large, the trends of these curves have become more and more like the U shape. When the drones' loading capacity increases (or the drones' endurance decreases), the total cost is decreasing first and then becomes increasing. The results show that there exists a certain best combination of drones' loading capacity and endurance such that the cooperative system's total delivery cost could be the lowest. This phenomenon is also validated in another study by Karak and Abdelghany (2019).

6.4.4 Influence of customers' time windows

The customers' time windows are an important problem setting in this study's context. More specifically, the existence of this time window is reflected by the previously defined parameter e_c, i.e., the latest time when a truck (or a drone) should depart from customer c's site after serving it. The sensitivity analysis is conducted by setting the parameter e_c as a certain value (i.e., with time window) or setting the parameter $e_c = \infty$ (i.e., without time window). The experiments are also conducted on six groups of instances with different problem scales; and the results are shown in Figure 10, which demonstrates the customers' required service time window maybe increase the system's cost.

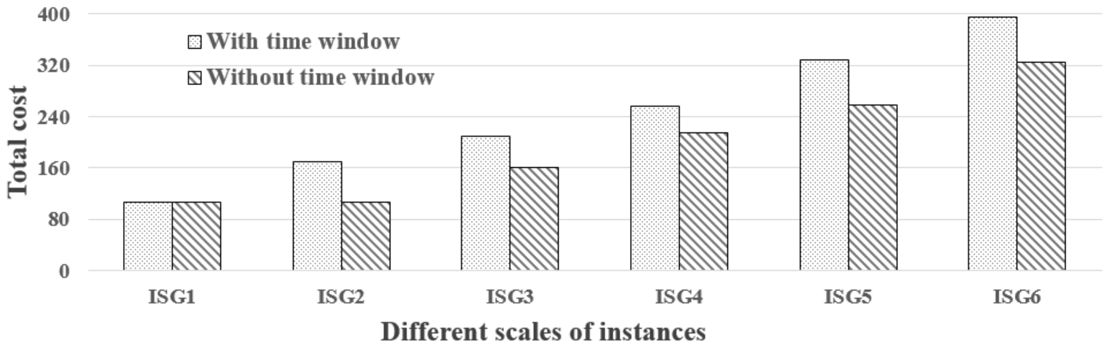

Figure 10: Influence of customers' time windows

7 Conclusions

This paper investigates a comprehensive routing and scheduling problem for the truck-and-drone cooperative pickup & delivery system. A decision model as well as an efficient algorithm are designed for this problem. Computational experiments are also conducted to validate the effectiveness of the proposed methodology (i.e., the model and the algorithm), which could act as an decision support tool for the operators of this cooperative pickup & delivery system. The main contributions of this paper are summarized from the following perspectives.

From the problem modeling perspective, a comprehensive set of problem contexts and factors are considered in the proposed model. This problem contains multiple truck groups; each truck group is a truck as well as multiple drones carried by the truck; each drone can visit multiple customers during one flying trip, which starts from the drone's taking-off from the truck to its landing onto the truck; multiple flying trips could be executed by each drone during the planning horizon of this problem; customers have the requirements of either picking up parcels from their sites or delivering parcels to their sites; a customer can be served by either a truck or a drone; the time-related requirements of customers are also considered. So far as we know, few studies have taken account of all the above issues simultaneously. This study formulates an MILP model for the above comprehensive decision problem.

From the algorithm design perspective, this study designs a novel hybrid algorithm to solve the above comprehensive and complex MILP model efficiently. The hybrid algorithm's outer framework is based on the CG; the logic-based Benders decomposition is embedded to solve the CG's PPs; and a dynamic programming is further embedded to solve the submodels of the Benders decomposition. Moreover, the decomposition is associated with the different layers in the above three-echelon framework of the hybrid algorithm. More specifically, the CG's PP is decomposed according to the dimension of truck groups, for each of which a PP subproblem is formulated and solved by the logic-based Benders decomposition; the Benders decomposition's subproblem is further decomposed into submodels according to the dimension of one truck group's drones, for each of which dynamic programming is designed to solve a submodel. So far as we know, few studies have proposed this hybrid algorithm, especially in the fields of the truck-and-drone cooperative delivery systems.

From the managerial implications perspective, numerical experiments are conducted to validate the effectiveness of the above embedded algorithmic components. Computational results show that the algorithmic optimality gap is just 3.32% on average, while the computation time is shorter than the CPLEX by about 86.19%. Moreover, some sensitivity analysis demonstrates that carrying more drones on each truck, and the higher speed of drones does not surely improve the system's performance; and there exists a certain best combination of drones' loading capacity and endurance such that the cooperative system's total delivery cost could be the lowest.

Limitations are also contained in the current study. Firstly, the proposed algorithm is a heuristic in nature although the algorithmic performance is good for solving the model. In the future, the proposed hybrid algorithm could be embedded in a branch and bound framework so as to design an exact solution method. Secondly, the problem scale of the solvable instances could be further increased by designing more efficient algorithmic acceleration tactics and incorporating them in the hybrid algorithm. In addition, some realistic factors such as the uncertain travel time between two locations, and a drone's battery power consumption as well as its dependance on the carried cargo's weight, could be taken account in the models of the future studies.

Appendices

Appendix A: Generation of the initial solution for RMP

At the beginning of the CG, we need generate a set of initial feasible route plans for the RMP; then we solve the LR-RMP so as to obtain the dual variables, which are used to construct the PP for generating the new route plans. This study uses a heuristic, which is elaborated as follows, for generating an initial solution.

Step 1: Sort the customers according to the increasing order of e_c, $c \in N_C$, and obtain a sequence of customers.

Step 2: Determine routes for drones. For each truck group k and for each drone d on one truck group, we determine a route for the drone as follows. In this initialization, the drone takes off from its truck for once, and the location for taking off is the depot. According to the above order obtained in Step 1, we judge whether or not each customer c, which is in the remainder of the sequence, can be added into the drone's route; the judgement is made under the limitation on the capacity of drone m^D, drone's maximum flying time of one trip f, and the latest time departing from the customer. After serving the added customers, the drone lands on its truck at a station node, which is the nearest to the last customer served by the drone d. According to the above route, the values of variables β_{ijdk} are determined for each drone.

Step 3: Determine routes for trucks. For each truck group k, we determine a route for the truck according to the station nodes, which are determined in Step 2 and are the locations where the $|D|$ drones land on the truck. For the above mentioned station nodes, there are at most $|D|$ nodes for each truck; and these nodes are surely on the truck's route. According to the order obtained in Step 1, we judge whether or not each customer c, which is in the remainder of the sequence, can be added into the truck's route; the judgement is made under the limitation on the capacity of drone m^K, and the latest time departing from the customer. According to the above route, the values of variables α_{ijk} are determined for each truck.

Step 4: According to the above obtained routes for trucks and drone, we determine the values of some decision variables for calculating the cost (objective value) of each truck group.

Step 5: Repeat the above steps to generate $|K|$ plans (routes of truck groups) as well as their costs.

Appendix B: Proof of Proposition 1

Proposition 1. The BSP is feasible and bounded.

Proof: Given any $\overline{\beta_{ijd}} \in \{0,1\}$, $\overline{\sigma_{ijd}} \in \{0,1\}$ and $\overline{\delta_i} \geq 0$, let $\sum_{d \in D} \sum_{i \in N_-} \sum_{c \in N_C} s^D t_{ic}^D (\overline{\beta_{icd}} -$

$\overline{\sigma_{icd}}) \geq 0$ and $\sum_{c\in N_C} s^T\left(\overline{\delta_c} - e_c(\sum_{i\in N_-}\sum_{d\in D}\overline{\beta_{icd}} + \sum_{i\in N_-} x_{ic} - \sum_{i\in N_-}\sum_{d\in D}\overline{\sigma_{icd}})\right)^+ = 0$. The BSP is feasible and bounded if and only if there exists a solution for BMP.

The BMP determines the order of customers and stations the truck will visit, and truck route in BMP satisfy constraints (49)–(51), constraints (64), constraints (66), constraints (68), constraints (86), constraints (101) and (102) and constraints (104) and (110). Hence, we can construct a feasible truck route. Then, based on the truck route, one can easily construct a feasible solution for the BSP. Therefore, the BSP is feasible and bounded. ■

Appendix C: Proof of Benders optimality cuts

In this paper, the Benders optimality cuts are obtained based on classical Benders decomposition.

Suppose that the set of poles of the polyhedron P is Ω_p, and when BSP has a bounded solution, the optimal solution is obtained at least one pole $\overline{\pi}_w$, where $w \in \Omega_p$. Suppose that the objective function value of BSP is η. Then we add the optimal cut to BMP as $\overline{\pi}_w^T(b - B\hat{y}) \leq \eta,\ w \in \Omega_p$, where b is the right vector matrix of the constraint of the original model, B is the vector matrix of the coefficients of $\hat{y}$, and $\hat{y}$ is the value of the decision variable obtained by BMP.

Based on the above, the Benders optimality cuts in the BMP added in this paper should be:

$$s^G + \sum_{i\in N_-}\sum_{j\in N_+} s^K t_{ij}^K \alpha_{ij} - \sum_{j\in N_C} b_j\theta_j - \pi + \varpi \geq s^G + \sum_{i\in N_-}\sum_{j\in N_+} s^K t_{ij}^K x_{ijl} + \sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D(\beta_{icdl} - \sigma_{icdl}) + \sum_{c\in N_C} s^T\left(\delta_{cl} - e_c(\sum_{i\in N_-}\sum_{d\in D}\beta_{icdl} + \sum_{i\in N_-} x_{icl} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icdl})\right)^+ - \sum_{j\in N_C}(\sum_{i\in N_-}\sum_{d\in D}\beta_{ijdl} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{ijdl})\,\theta_j - \pi \qquad \forall l \in L \quad \text{(C-1)}$$

Since b_j is one if customer j is served by the truck group, $\sum_{j\in N_C} b_j = \sum_{j\in N_C}\sum_{i\in N_-}\alpha_{ij}$. After simplifying the above formula (C-1), we can obtain:

$$\varpi \geq \sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D(\beta_{icdl} - \sigma_{icdl}) + \sum_{c\in N_C} s^T\left(\delta_{cl} - e_c(\sum_{i\in N_-}\sum_{d\in D}\beta_{icdl} + b_c - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icdl})\right)^+ - \sum_{j\in N_C}\left(\sum_{i\in N_-}\sum_{d\in D}\beta_{ijdl} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{ijdl} + b_j\right)\theta_j \qquad \forall l \in L \quad \text{(C-2)}$$

Let $\Phi_l = \sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D(\beta_{icdl} - \sigma_{icdl}) + \sum_{c\in N_C} s^T\left(\delta_{cl} - e_c(\sum_{i\in N_-}\sum_{d\in D}\beta_{icdl} + b_c - \sum_{i\in N_-}\sum_{d\in D}\sigma_{icdl})\right)^+ - \sum_{j\in N_C}(\sum_{i\in N_-}\sum_{d\in D}\beta_{ijdl} - \sum_{i\in N_-}\sum_{d\in D}\sigma_{ijdl} + b_j)\,\theta_j$ in the l^{th} iteration. After simplification, Formula (C-1) turns to:

$$\varpi \geq \Phi_l \qquad \forall l \in \{1,2,\cdots,L\} \qquad \text{(C-3)}$$

Where L is the maximum number of iterations.

Appendix D: Pseudocode of the dynamic programing for solving the $\mathbf{BSP_d}$

For describing the detailed process of the dynamic programming of solving the pricing problem $\mathbf{BSP_d}$, the pseudocode of the procedure is elaborated as follows.

Dynamic programing for solving BSP_d

Input: $N,\ t_{ij}^{D},\ t_{ij}^{K},\ t_i^{C},\ s^{D},\ \theta_i$

Output: The route of drone d with the lowest variable reduced cost

For $n = 0, \dots, |N| * |N|$ **do**

$$f_1(Z_{1n}) = \begin{cases} 0 - \theta_i & case\ 1 \\ s^D\left(t_{i_1 i}^{D} + t_i^{C}\right) - \theta_i & case\ 2 \\ s^D\left(t_{i_1 i}^{D} + t_i^{C} + t_{ij}^{D} + t_j^{C}\right) - \theta_i - \theta_j & case\ 3. \end{cases}$$

, record the drone route, time, and capacity of n^{th} state in the first stage

End for

For $m = 2,3, \dots, |N|$ **do**

  For $n = 0, \dots, |N| * |N|$ **do**

    $f_m(Z_{mn}) \leftarrow \infty$

  End for

End for

For $m = 2,3, \dots, |N|$ **do**

  For $n = 0, \dots, |N| * |N|$ **do**

    For $i_1 = 0, \dots, |N|$ **do** // the customers and station nodes for the $(m-1)^{\text{th}}$ stage

$$v\left(u_m(Z_{mn})\right) = \begin{cases} 0 - \theta_i & case\ 1 \\ s^D\left(t_{i_1 i}^{D} + t_i^{C}\right) - \theta_i & case\ 2 \\ s^D\left(t_{i_1 i}^{D} + t_i^{C} + t_{ij}^{D} + t_j^{C}\right) - \theta_i - \theta_j & case\ 3. \end{cases}$$

    End for

    $r\_cost \leftarrow 0$ // r_cost is defined as the value of variable reduced cost in the m^{th} stage

    $r\_cost \leftarrow f_m(Z_{mn}) = \min\{v_m(u_m(Z_{mn})) + f_{m-1}(Z_{m-1,x})\}$

    If $r\_cost < f_m(Z_{mn})$

      $f_m(Z_{mn}) \leftarrow r\_cost$, record the drone route, time, and capacity of n^{th} state in the m^{th} stage

    End if

  End for

End for

Choose the route that drone serves the most customers and returns to the depot with the lowest variable reduced cost

Output the chosen route, the value of the lowest variable reduced cost

Appendix E: The complete model of the LB

In order to evaluate the performance and efficiency of the proposed algorithm in large-scale instances, the results obtained by the algorithm need to be compared with a lower bound (LB). Due to the complexity of the model in this paper, the LB model is obtained by relaxing some variables and constraints of the original model. In the LB model, we ignore the impact of the customer's penalty cost, and do not consider the flight duration and load capacity of the drone. Therefore, the flying routes of drones are not limited, and the related constraints and variables are removed. In addition, we relax the binary variables $\alpha_{ijk}, \beta_{ijdk}$ to continuous variables $\alpha_{ijk}, \beta_{ijdk} \geq 0$ and $\alpha_{ijk}, \beta_{ijdk} \leq 1$. The model of the LB is shown as follow.

$$\text{Minimize}\left\{\sum_{k\in K} s^G \varepsilon_k + \sum_{k\in K}\sum_{i\in N_-}\sum_{j\in N_+} s^K t_{ij}^K \alpha_{ijk} + \sum_{k\in K}\sum_{d\in D}\sum_{i\in N_-}\sum_{c\in N_+} s^D t_{ic}^D (\beta_{icdk} - \sigma_{icdk})\right\} \quad \text{(E-1)}$$

Subject to

$$\sum_{j\in N_+} \alpha_{0jk} = 1 \quad \forall k \in K \quad \text{(E-2)}$$

$$\sum_{i\in N_-} \alpha_{i(|N_V|)k} = 1 \quad \forall k \in K \quad \text{(E-3)}$$

$$\sum_{i\in N_-} \alpha_{ijk} = \sum_{i\in N_+} \alpha_{jik} \leq 1 \quad \forall j \in N_0, k \in K \quad \text{(E-4)}$$

$$\sum_{j\in N_+} \beta_{0jdk} = 1 \quad \forall k \in K, d \in D \quad \text{(E-5)}$$

$$\sum_{i\in N_-} \beta_{i(|N_V|)dk} = 1 \quad \forall k \in K, d \in D \quad \text{(E-6)}$$

$$\sum_{i\in N_-} \beta_{ijdk} = \sum_{i\in N_+} \beta_{jidk} \leq 1 \quad \forall j \in N_0, k \in K, d \in D \quad \text{(E-7)}$$

$$\sum_{k\in K}\sum_{i\in N_-}\sum_{d\in D} (\beta_{icdk} - \sigma_{icdk}) + \sum_{k\in K}\sum_{i\in N_-} \alpha_{ick} = 1 \quad \forall c \in N_C \quad \text{(E-8)}$$

$$2\sigma_{ijdk} \leq \beta_{ijdk} + \alpha_{ijk} \leq 2\sigma_{ijdk} + 1 \quad \forall i \in N_-, j \in N_+, k \in K, d \in D \quad \text{(E-9)}$$

$$\sum_{i\in N_-} \sigma_{icdk} = \sum_{i\in N_+} \sigma_{cidk} \quad \forall c \in N_C, k \in K, d \in D \quad \text{(E-10)}$$

$$\sum_{j\in N_+} \beta_{ijdk} \leq \sum_{j\in N_+} \alpha_{ijk} \quad \forall i \in N_V, k \in K, d \in D \quad \text{(E-11)}$$

$$\beta_{ijdk} \leq \alpha_{ijk} \quad \forall j \in N_+ \backslash N_C, i \in N_V, k \in K, d \in D \quad \text{(E-12)}$$

$$\delta_{jk}^K \geq \delta_{ik} + t_{ij}^K + t_j^C - M(1 - \alpha_{ijk}) \quad \forall i \in N_-, j \in N_+, k \in K \quad \text{(E-13)}$$

$$\delta_{jk}^K \leq \delta_{ik} + t_{ij}^K + t_j^C + M(1 - \alpha_{ijk}) \quad \forall i \in N_-, j \in N_+, k \in K \quad \text{(E-14)}$$

$$\delta_{jdk}^D \geq \delta_{ik} + t_{ij}^D + t_j^C - M(1 - \beta_{ijdk}) \quad \forall i \in N_-, j \in N_+, k \in K, d \in D \quad \text{(E-15)}$$

$$\delta_{jdk}^D \leq \delta_{ik} + t_{ij}^D + t_j^C + M(1 - \beta_{ijdk}) \quad \forall i \in N_-, j \in N_+, k \in K, d \in D \quad \text{(E-16)}$$

$$\delta_{ik} \geq \delta_{idk}^D \quad \forall i \in N_+, k \in K, d \in D \quad \text{(E-17)}$$

$$\delta_{ik} \geq \delta_{ik}^K \quad \forall i \in N_+, k \in K \quad \text{(E-18)}$$

$$\delta_{0k} = 0 \quad \forall k \in K \quad \text{(E-19)}$$

$$\mu_{idk}^D - \mu_{jdk}^D \leq N(1 - \beta_{ijdk}) - 1 \quad \forall i \in N_-, j \in N_+, k \in K, d \in D \quad \text{(E-20)}$$

$$\mu_{ik}^K - \mu_{jk}^K \leq N(1 - \alpha_{ijk}) - 1 \quad \forall i \in N_-, j \in N_+, k \in K \quad \text{(E-21)}$$

$$\zeta_{0k} = \sum_{c\in N_C} q_c^D \left(\sum_{i\in N_-}\sum_{d\in D} \beta_{icdk} + \sum_{i\in N_-} \alpha_{ick} - \sum_{i\in N_-}\sum_{d\in D} \sigma_{icdk}\right) \quad \forall k \in K \quad \text{(E-22)}$$

$$\zeta_{0k} \leq m^K \quad \forall k \in K \quad \text{(E-23)}$$

$$\tau_{jk} \geq \zeta_{ik} - q_j^D + q_j^P - M(1 - \alpha_{ijk}) \qquad \forall i \in N_-, j \in N_+, k \in K \tag{E-24}$$

$$\tau'_{jdk} \geq \zeta_{ik} - q_j^D + q_j^P - M\left(1 - \beta_{ijdk}\right) \qquad \forall i \in N_-, j \in N_+, k \in K, d \in D \tag{E-25}$$

$$\zeta_{ik} \leq \tau'_{idk} \qquad \forall i \in N_+, k \in K, d \in D \tag{E-26}$$

$$\zeta_{ik} \leq \tau_{ik} \qquad \forall i \in N_+, k \in K \tag{E-27}$$

$$\zeta_{ik} \leq m^K \varepsilon_k \qquad \forall i \in N, k \in K \tag{E-28}$$

$$\varepsilon_k \geq \beta_{ijdk} \qquad \forall i \in N_-, j \in N_C, k \in K, d \in D \tag{E-29}$$

$$\varepsilon_k \geq \alpha_{ijk} \qquad \forall i \in N_-, j \in N_C, k \in K \tag{E-30}$$

$$\alpha_{0(|N_V|)k} = 1 - \varepsilon_k \qquad \forall k \in K \tag{E-31}$$

$$0 \leq \mu_{ik}^K \leq |N| \qquad \forall i \in N, k \in K \tag{E-32}$$

$$0 \leq \mu_{idk}^D \leq |N| \qquad \forall i \in N, k \in K,, d \in D \tag{E-33}$$

$$\alpha_{ijk}, \beta_{ijdk}, \sigma_{ijdk}, \delta_{ik}, \delta_{ik}^K, \delta_{idk}^D, \zeta_{ik}, \tau_{ik}, \tau'_{idk} \geq 0 \qquad \forall i, j \in N, k \in K, d \in D \tag{E-34}$$

$$\varepsilon_k \in \{0,1\} \qquad \forall k \in K. \tag{E-35}$$

$$\alpha_{ijk}, \beta_{ijdk} \leq 1 \qquad \forall i, j \in N, k \in K, d \in D \tag{E-36}$$

Appendix F: Pseudocode of the heuristics algorithm based on neighborhood search

The pseudocode of the heuristics algorithm based on neighborhood search is elaborated as follows.

The heuristics algorithm based on neighborhood search

Input: The parameters of model (see Section 4.1), the maximum number of iterations is M^{max}.

Output: The optimal route solution and total cost.

Generate the initial solution (see Appendix A) and calculate the total initial cost F^*.

For $m = 0,1,2, \dots, M^{max}$ **do**

  For $k = 0,1,2, \dots, |K|$ **do** // Stations exchange

    For $i = 0,1,2, \dots, N_V$ **do**

      For $j = 0,1,2, \dots, N_V$ **do**

        If exchange-eligible station i has not been exchanged with station j in the truck k route

            Exchange i and j to obtain new truck and drone routes. Assign values to all decision variables and calculate the total cost F_2

          If $F_2 < F^*$

            Update route scheme and current optimal cost values, $F^* \leftarrow F_2$

          End if

End if

End for

End for

End for

For $k = 0,1,2,\dots,|K|$ **do** //Customers insert and delete operation

For $i = |N_V| + 1, |N_V| + 2, \dots, |N_V| + N_C$ **do**

If eligible station i has not been inserted into the truck group k route

Insert customer i into the truck k route, and delete customer i from other truck group routes. Assign values to all decision variables and calculate the total cost F_2

If $F_2 < F^*$

Update route scheme and current optimal cost values, $F^* \leftarrow F_2$

End if

End if

End for

End for

End for

Return the optimal route solution and total cost.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (Grant numbers 72025103, 72361137001, 71831008 and 72071173).

References

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

Battarra M., Cordeau J.F., Iori M. (2014) Pickup-and-delivery problems for goods transportation, in: Toth, P., Vigo, D. (Eds.), *Vehicle Routing: Problems, Methods, and Applications, 2nd Edition* Siam, Philadelphia, pp. 161–191.

Berbeglia G., Cordeau J.-F., Gribkovskaia I., Laporte G. (2007) Static pickup and delivery problems: a classification scheme and survey. *Top* 15: 1–31.

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

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

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

Cavani S., Iori M., Roberti R. (2021) Exact methods for the traveling salesman problem with multiple drones. *Transportation Research Part C-Emerging Technologies* 130: 103280.

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

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

de Freitas J.C., Penna P.H.V. (2020) A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research* 27(1): 267–290.

Dell'Amico M., Montemanni R., Novellani S. (2021) Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem. *Optimization Letters* 15(5): 1617–1648.

Dell'Amico M., Montemanni R., Novellani S. (2022) Exact models for the flying sidekick traveling salesman problem. *International Transactions in Operational Research* 29(3): 1360–1393.

Gacal J.B., Urera M.Q., Cruz D.E. (2020) Flying sidekick traveling salesman problem with pick-up and delivery and drone energy optimization. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* Electr Network, pp. 1167–1171.

Gonzalez-R P.L., Canca D., Andrade-Pineda J.L., Calle M., Leon-Blanco J.M. (2020) Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C-Emerging Technologies* 114: 657–680.

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

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

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

Jeon A., Kang J., Choi B., Kim N., Eun J., Cheong T. (2021) Unmanned aerial vehicle last-mile delivery considering backhauls. *IEEE Access* 9: 85017–85033.

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

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

Khoufi I., Laouiti A., Adjih C. (2019) A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones* 3(3): 66.

Koc C., Laporte G., Tukenmez I. (2020) A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research* 122: 104987.

Kuo R.J., Lu S.-H., Lai P.-Y., Mara S.T.W. (2022) Vehicle routing problem with drones considering time windows. *Expert Systems with Applications* 191: 116264.

Leon-Blanco J.M., Gonzalez P.L., Andrade-Pineda J.L., Canca D., Calle M. (2022) A multi-agent approach to the truck multi-drone routing problem. *Expert Systems with Applications* 195: 116604.

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

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

Liang Y., Luo Z. (2022) A survey of truck-drone routing problem: Literature review and research prospects. *Journal of the Operations Research Society of China* 10(2): 343–377.

Lu Y., Yang C., Yang J. (2022) A multi-objective humanitarian pickup and delivery vehicle routing problem with drones. *Annals of Operations Research* 319: 291–353.

Luo Z., Gu R., Poon M., Liu Z., Lim A. (2022) A last-mile drone-assisted one-to-one pickup and delivery problem with multi-visit drone trips. *Computers & Operations Research* 148: 106015.

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

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

Mara S.T.W., Rifai A.P., Sopha B.M. (2022) An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Systems with Applications* 205: 117647.

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

Min H. (1989) The multiple vehicle-routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A-Policy and Practice* 23(5): 377–386.

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

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

Otto A., Agatz N., Campbell J., Golden B., Pesch E. (2018) Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* 72(4): 411–458.

Piri L., Ghezavati V., Hafezalkotob A. (2022) Developing a new model for simultaneous scheduling of two grand projects based on game theory and solving the model with Benders decomposition. *Frontiers of Engineering Management* 9(1): 117–134.

Poikonen S., Golden B. (2020a) The Mothership and Drone Routing Problem. *INFORMS Journal on Computing* 32(2): 249–262.

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

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

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

Reed S., Campbell A.M., Thomas B.W. (2022a) Impact of Autonomous Vehicle Assisted Last-Mile Delivery in Urban to Rural Settings. *Transportation Science* 56(6): 1530–1548.

Reed S., Campbell A.M., Thomas B.W. (2022b) The Value of Autonomous Vehicles for Last-Mile Deliveries in Urban Environments. *Management Science* 68(1): 280–299.

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

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

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

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

Vasquez S.A., Angulo G., Klapp M.A. (2021) An exact solution method for the TSP with Drone based on decomposition. *Computers & Operations Research* 127: 105127.

Viloria D.R., Solano-Charris E.L., Munoz-Villamizar A., Montoya-Torres J.R. (2021) Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. *International Transactions in Operational Research* 28(4): 1626–1657.

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

Yan R., Zhu X., Zhu X., Peng R. (2022) Optimal routes and aborting strategies of trucks and drones under random attacks. *Reliability Engineering & System Safety* 222: 108457.

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

Yurek E.E., Ozmutlu H.C. (2021) Traveling salesman problem with drone under recharging policy. *Computer Communications* 179: 35–49.

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