# Branch-Price-and-Cut for Trucks and Drones Cooperative Delivery

Lu Zhen [1], Jiajing Gao [1], Zheyi Tan [1], Shuaian Wang [2*], Roberto Baldacci [3]

[1] *School of Management, Shanghai University, Shanghai, China*
[2] *Faculty of Business, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
[3] *College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar*

* Corresponding author. Email: hans.wang@polyu.edu.hk

***Abstract***: The truck and drone based cooperative model of delivery can improve the efficiency of last mile delivery, and has thus increasingly attracted attention in academia and from practitioners. In this study, we examine a vehicle routing problem and apply a cooperative form of delivery involving trucks and drones. We propose a mixed-integer programming model and a branch-price-and-cut based exact algorithm to address this problem. To reduce the computation time, we design several acceleration strategies, including a combination of dynamic programming and calculus-based approximation for the pricing problem, and various effective inequalities for the restricted master problem. Numerical experiments are conducted to validate the effectiveness and efficiency of the proposed solution.

***Keywords***: collaborative delivery; route optimization; drones; branch-price-and-cut.

## 1. Introduction

Drone-based delivery has recently become a promising option in the last mile delivery industry (Macrina et al., 2020). Amazon first implemented drone deliveries in 2013, and many logistics companies have since developed new modes of drone delivery. In the current COVID-19 pandemic, drones have been increasingly useful because they avoid human contact during the delivery process. They can also circumvent the road restrictions encountered in the traditional truck-based delivery mode. The flight route of a drone is generally straight and is much shorter than the land route of a truck. Drones are also less affected by the terrain and thus particularly suitable for fulfilling delivery tasks in remote mountainous areas. They can contribute to alleviating road congestion and improve delivery efficiency. However, drones have some disadvantages. The maximum flying time and distance is restricted by the battery capacity, which limits their service range and makes it impossible to fulfil long-range delivery tasks, and their load capacity in terms of weight and volume is relatively limited compared with that of trucks. Weather conditions and flying altitude restrictions also limit drone-based delivery services.

The AMP company and the University of Cincinnati in the U.S. jointly developed a novel cooperative delivery system involving both trucks and drones in 2014, aimed at compensating for the shortcomings of solely drone delivery. Parcels are delivered by truck groups in this system. The concept “truck group” does not mean a fleet of trucks; a truck group is a pair of a truck and a drone carried by the truck. A drone in a truck group is launched from and returns to the truck. Figure 1 demonstrates this novel mode and gives a comparison with the traditional truck-based delivery mode and the drone-based mode. We make some estimation for the example in Figure 1 under the realistic settings. The truck mode's route length (two trucks), travel cost, completion time is 110km, 76 CNY (Chinese Yuan), 3.67 hour, respectively; the drone mode' route length (two drones), cost, completion time is 174km, 47 CNY, 3.63 hour, respectively. While for the cooperative mode, the above three metrics are: 75km (two trucks) & 45km (two drones), 64 CNY, 3.44 hour. The cooperative mode is the fastest. Although its cost is higher than the drone mode's cost, it should be noted that the drone mode only serves 11 customers due to the limited flying distance and other modes serve all the 21 customers. If we compare their average travel cost per customer, the cooperative mode is also the most competitive with respect to the economic metric. Therefore, the cooperative system inherits the relative merits of both the trucks and the drones, which can potentially improve its total performance in the last mile delivery.

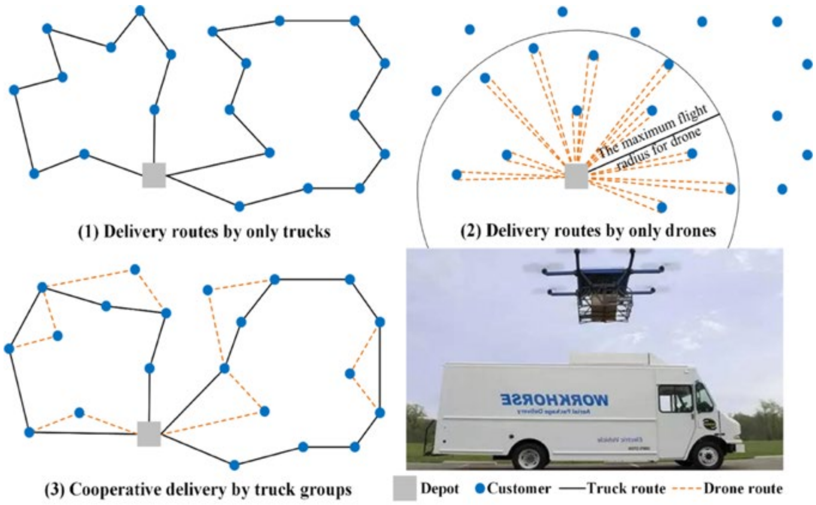


**Figure 1** Delivery route schemes of trucks and drones

An efficient schedule for the trucks and drones in terms of their routes and timing decisions is crucial for the effective performance of the cooperative system. Mathematical programming has been widely used in vehicle routing and has the potential to establish a decision model and an efficient algorithm for this cooperative delivery mode. Few studies have been conducted that focus on this specific system and the related scheduling problems. Thus, in this paper we propose a vehicle routing problem for truck and drone cooperative delivery, which decides the allocation of customers to truck groups and their routes (the groups' routes are the same as the trucks' routes) and the drones' routes; we assume that each truck is equipped with only one drone, and together they form a truck group.

We present a mixed-integer programming (MIP) model for the problem. The model considers specific factors, such as the influence of the weight of the carried cargo on the drone's maximum flying time. We designed a branch-price-and-cut (B&P&C) algorithm to solve the proposed MIP model, in which valid inequalities are used to strengthen the linear relaxation of the model and that takes an effective dynamic programming approach to solving the pricing problem. We conducted numerical experiments to validate the effectiveness and efficiency of the B&P&C algorithm.

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 describes the background of the problem. We describe our MIP model for the problem in Section 4. Section 5 presents an exact algorithm to solve the problem. Section 6 reports the results of our numerical experiments. Conclusions are outlined in the final section.

## 2. Related works

The problem investigated in this study represents a new variant of the well-known vehicle routing problem (VRP) (see for example, Toth and Vigo, 2014). The problem is more complex than other variants of VRPs, because trucks and drones can both serve customers independently and have a cooperative relationship. Research into the cooperative delivery of trucks and drones can be divided into two categories according to the number of trucks: first, a traveling salesman problem with drones (TSP-D); and second, the vehicle routing problem with drones (VRP-D). The routing problem for cooperative delivery using a single truck and a single drone is typically referred to as TSP-D with single drone, while that for cooperative delivery using a single truck and multiple drones is known as TSP-D with multiple drones. The problem studied in this paper belongs to the VRP-D class. The development of drone logistics has led to extensive research into the problem of cooperative delivery using trucks and drones. Table 1 lists the studies in this field. We examine the three research topics of single truck and single drone, single truck and multiple drones, and multiple trucks and multiple drones. There are also studies that focus on other aspects of drone scheduling. For example, Yi and Sutrisna (2021) examined the optimization of drone speed for construction management and designed an interesting dynamic programming algorithm for the problem. For brevity, these studies are not reviewed.

### 2.1 Single truck and single drone

The study of Murray and Chu (2015) may be the first to examine the TSP-D with single drone and defined the delivery of parcels by a truck and a drone as a flying sidekick traveling salesman problem (FSTSP). Yurek and Ozmutlu (2018) improved the FSTSP model and presented an iterative algorithm based on a decomposition approach, to minimize delivery completion time. Based on the problems raised by Murray and Chu (2015) and Agatz et al. (2018), de Freitas and Penna (2020) proposed a three-step method consisting of a hybrid variable neighborhood search, an exact method, and a heuristic to solve the TSP-D with single drone. Carlsson and Song (2018) defined the TSP-D with single drone as a horsefly routing problem (HRP), which they investigated by applying Euclidean plane theory and a real-time numerical simulation of a road network. Their results demonstrated that the efficiency improvement is proportional to the square root of the ratio of the speeds of the truck and the drone. Poikonen et al. (2019) proposed four algorithms based on branch and bound to solve the TSP-D with a single drone. They used dynamic programming to obtain an approximate lower bound for each node, aimed at further improving the solving efficiency. El-Adle et al. (2019) established a binary MIP model for the TSP-D with single drone to minimize the duration of the joint tour. To enhance the model, cut generation and other bound improvement strategies were applied. Jeong et al. (2019) considered more realistic factors and proposed the FSTSP with energy consumption and no-fly zone (FSTSP-ECNZ). They established an MILP model and a two-phase constructive and search heuristic to solve the problem. Roberti and Ruthmair (2021) proposed a compact MILP model for different TSP-D variants based on the timely synchronizing of truck and drone flows. Their proposed branch-and-price algorithm can solve to optimality instances with up to 39 customers.

All of the models considered in the above works aim to minimize time-related objectives, while other studies have taken the perspective of minimizing cost-related objectives. Agatz et al. (2018) and Bouman et al. (2018) proposed a local search-based method and a dynamic programming method for TSP-D with a single drone, respectively. Based on the work of Murray and Chu (2015), Ha et al. (2018) proposed a model to minimize the operating cost, and designed two heuristic methods by using the local search and greedy randomized adaptive search, respectively. The objectives of some studies include both time and cost related factors. Omagari and Higashino (2018) defined a constrained multi-objective optimization model for a drone delivery problem (DDP). Wang et al. (2020) presented a bi-objective TSP-D with single drone and proposed a non-dominated sorting genetic algorithm (INSGA-II) to solve the problem. Ha et al. (2020) proposed a hybrid algorithm combining a genetic algorithm (HGA) and local search to solve a bi-objective TSP-D with a single drone.

### 2.2 Single truck and multiple drones

Based on the above studies, TSP-D with multiple drones has also been examined. Murray and Chu (2015) proposed a parallel drone scheduling traveling salesman problem (PDSTSP). Ferrandez et al. (2016) used GA to solve the truck's route and the K-means algorithm to solve the drones' routes based on the truck's route. Murray and Raj (2020) proposed a multiple flying sidekick traveling salesman problem (mFSTSP). These studies established models to minimize the travel time for a truck and drones to return to the depot. Moshref-Javadi et al. (2020) proposed a single truck and multi-drone delivery problem and built an MILP model to minimize the waiting time of customers. Poikonen and Golden (2020) relaxed some constraints considered in the literature and proposed a k-multi-visit drone routing problem (k-MVDRP) in which a drone can carry multiple packages and visit multiple customers in one trip.

**Table 1** Related literature on cooperative delivery of trucks and drones

| Authors and years | Key features of the problem | | | | | | | Problem abbreviation | Model | objective | | | | | Methodologies and solving methods |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | time | | cost | | | |
| | Num. of truck | Num. of drone | Correspondence | Drone multi-customer | Launch ≠ Return | Non-customer points | Variable flying duration | | | Makespan | Waiting /traveling | Traveling | Waiting | Fixed | |
| Murray and Chu (2015) | 1 | 1 | ✓ | | ✓ | | | FSTSP | MILP | ✓ | | | | | Heuristic |
| | 1 | n | | | ✓ | | | PDSTSP | MILP | ✓ | | | | | Heuristic |
| Yurek and Ozmutlu (2018) | 1 | 1 | ✓ | | ✓ | | | TSP-D | MILP | ✓ | | | | | Iterative algorithm |
| Carlsson and Song (2018) | 1 | 1 | ✓ | | ✓ | ✓ | | HRP | CAP | ✓ | | | | | Continuous approximation analysis |
| Poikonen et al. (2019) | 1 | 1 | ✓ | | | | | TSP-D | -- | ✓ | | | | | Heuristics based on the branch-and-bound |
| El-Adle et al. (2019) | 1 | 1 | ✓ | | ✓ | | | TSP-D | MIP | ✓ | | | | | Cut generation with bound improvement |
| Jeong et al. (2019) | 1 | 1 | ✓ | | ✓ | | ✓ | FSTSP-ECNZ | MILP | ✓ | | | | | TPCSA |
| Agatz et al. (2018) | 1 | 1 | ✓ | | | | | TSP-D | IP | | | ✓ | | | Fast route-first, cluster-second heuristics based on local search and DP |
| Bouman et al. (2018) | 1 | 1 | ✓ | | | | | TSP-D | DP | | | ✓ | | | DP and A* |
| de Freitas and Vaz Penna (2020) | 1 | 1 | ✓ | | ✓ | | | FSTSP, TSP-D | -- | ✓ | | | | | HGVNS |
| Ha et al. (2018) | 1 | 1 | ✓ | | ✓ | | | Min-cost TSP-D | MILP | | | ✓ | ✓ | | TSP-LS and GRASP |
| Omagari and Higashino (2018) | 1 | 1 | ✓ | | ✓ | | | DDP | -- | ✓ | | ✓ | | | Provisional-ideal-point |
| Wang et al. (2020) | 1 | 1 | ✓ | | ✓ | | | TSP-D | MILP | ✓ | | ✓ | ✓ | | INSGA-II |
| Ha et al. (2020) | 1 | 1 | ✓ | | ✓ | | | TSP-D | -- | ✓ | | ✓ | | | HGA |
| Roberti and Ruthmair (2021) | 1 | 1 | ✓ | | | | | TSP-D | MILP | ✓ | | | | | Branch-and-price |
| Ferrandez et al. (2016) | 1 | n | ✓ | | ✓ | | | TSP-D | -- | ✓ | | | | | GA and K-means |
| Murray and Raj (2020) | 1 | n | ✓ | | ✓ | | | mFSTSP | MILP | ✓ | | | | | Heuristic |
| Moshref-Javadi et al. (2020) | 1 | n | ✓ | | ✓ | | | STRPD | MILP | | ✓ | | | | TDRA |
| Poikonen and Golden (2020) | 1 | n | ✓ | ✓ | ✓ | ✓ | ✓ | k-MVDRP | ILP | ✓ | | | | | Heuristic |
| Kitjacharoenchai et al. (2019) | m | m | | | ✓ | ✓ | | mTSP-D | MIP | ✓ | | | | | ADI |
| Wang et al. (2017) | m | n | ✓ | | ✓ | | | VRP-D | -- | ✓ | | | | | Worst-case analysis |
| Poikonen et al. (2017) | m | n | ✓ | | | | | VRP-D | -- | ✓ | | | | | extend analysis |
| Ham (2018) | m | n | | ✓ | | | | PDSTSP+DP | CP | ✓ | | | | | VOH |
| Wang et al. (2019) | m | n | | ✓ | | ✓ | | HPDP | MILP | ✓ | | | | | HTDD |
| Sacramento et al. (2019) | m | m | ✓ | | ✓ | | | VRP-D | MIP | | | ✓ | | | ALNS |
| Wang and Sheu (2019) | m | n | | ✓ | ✓ | ✓ | | VRP-D | MIP | | | ✓ | | ✓ | Branch-and-price |
| This paper | m | m | ✓ | | ✓ | | ✓ | VRP-D | MIP | | | ✓ | ✓ | ✓ | B&P&C |

***Notes:*** **Correspondence**: A drone must land on the same truck from which it launched; **Drone multi-customer**: A drone can visit multiple customers in a single flight; **Launch≠Return**: Drone can't be launched from and return at the same node; **Non-customer points**: Non-customer nodes (for drones' launching, returning, charging etc.) are visited; **Variable flying duration**: The flying duration of the drone changes with the weight of the delivered goods; **FSTSP**: Flying Sidekick Traveling Salesman Problem; **MILP:** Mixed-Integer Linear Programming; **PDSTSP**: Parallel Drone Scheduling Traveling Salesman Problem; **HRP**: Horsefly Routing Problem; **CAP**: Continuous approximation paradigm; **FSTSP-ECNZ**: FSTSP considering Energy Consumption and No-fly Zone; **TPCSA**: Two-Phase Construction and Search Algorithm; **IP**: Integer Programming; **DP**: Dynamic Programming; **HGVNS**: Hybrid General Variable Neighborhood Search; **TSP-LS**: Traveling Salesman Problem Local Search; **GRASP**: Greedy Randomized Adaptive Search Procedure; **DDP**: Drone Delivery Problem; **INSGA-II**: Improved Non-dominated Sorting Genetic Algorithm; **HGA**: Hybrid genetic algorithm; **GA**: Genetic algorithm; **mFSTSP**: Multiple Flying Sidekicks Traveling Salesman Problem; **STRPD**: Simultaneous Traveling Repairman Problem with Drones; **TDRA:** Truck and Drone Routing Algorithm; **k-MVDRP**: k-Multi-Visit Drone Routing Problem; **ILP**: Integer linear program; **mTSPD**: Multiple Traveling Salesman Problem with Drones; **ADI**: Adaptive Insertion algorithm; **PDSTSP+DP**: PDSTSP Drop-Pickup; **CP**: Constraint programming; **VOH**: Variable ordering heuristics; **HPDP**: Hybrid truck-UAV cooperative Parcel Delivery Problem; **HTDD**: Hybrid Truck-Drone Delivery; **ALNS**: adaptive large neighborhood search.

### 2.3 Multiple truck and multiple drones

The previous TSP-D is extended to a multiple traveling salesman problem with drones (mTSP-D) to consider multiple trucks and drones. Kitjacharoenchai et al. (2019) proposed the mTSP-D, which allows the drone to fly from the truck, deliver a parcel, and fly back to any available truck nearby, regardless of capacity limitations. An MIP model and an adaptive insertion heuristic algorithm were designed to solve various large-scale instances with up to 100 customers. Wang et al. (2017) introduced the VRP with drones (VRP-D), and analyzed the worst case, which was dependent on the number of drones per truck and the speed of the drones. Poikonen et al. (2017) further extended this work by combining it with another practical variant of the VRP, based on Amdahl's Law. Although Wang et al. (2017) and Poikonen et al. (2017) considered VRP-D, they did not propose models and algorithms and mainly conducted theoretical analyses.

Ham (2018) examined a cooperative delivery problem with trucks and drones in a multi-depot context and proposed a constrained programming (CP) model. Wang et al. (2019) proposed a more efficient truck-drone parcel delivery system that uses trucks, truck-carried drones, and independent drones. They designed a hybrid truck-drone delivery (HTDD) algorithm to solve the problem. Sacramento et al. (2019) also formulated a mathematical model for the VRP-D and proposed an adaptive large neighborhood search (ALNS) metaheuristic. Wang and Sheu (2019) proposed an MIP model and developed a branch-and-price algorithm, which to the best of our knowledge represents the first exact algorithm for the cooperative delivery problem with multiple trucks and multiple drones.

In Table 1, we report the features considered in the problem addressed in this paper (see the "This paper" row) and compare them to the problem examined by Wang and Sheu (2019). The main difference is that we consider the different cost terms associated with a solution, as formally stated in the next section.

## 3. Problem description

In this problem, a truck group consists of a truck carrying a drone. Each drone is dedicated to a truck. Customers' packages are delivered by a set of truck groups. Both the trucks and the drones can deliver packages to the customers. Each customer is served (or visited) by either a truck or a drone. A drone can only be launched from and return to its dedicated truck when the truck stops at a customer node or depot. Figure 2 shows an example of the problem, which considers 2 truck groups and 23 customer points. In Figure 2, the solid lines represent the routes of trucks, and the dotted lines the routes of drones.

The problem can be defined on a graph $G = (N, E)$, where set $N = \{0,1,2 \ldots, n, n+1\}$ represents all of the nodes in the network, including customer nodes $i, j \in N_c = \{1,2 \ldots, n\}$, and starting and ending depots $\{0, n+1\}$ represent the same location. The weight or demand of the goods required by customer $i$ is $q_i$. The set of arcs $E$ represents possible freight routes. The depot houses $|K|$ truck groups where $K$ is the set of trucks or drones indexed by $k$.

We assume that the maximum flying duration of a drone is inversely proportional to its weight plus the demand the drone carries. If the maximum flying duration of drone $k$ is $e_k^D$ when it flies empty, when it delivers goods for customer $i$ the duration is denoted by $e_{ki}^D$, which equals $e_k^D f/(f+q_i)$, where $f$ is the net weight of an empty drone.

A drone's delivery trip is defined as a 3-tuple with the form $\langle i,j,h\rangle$; here $i$ is a node where a drone departs from a truck, $j$ is a customer node to which the drone delivers the cargo, and $h$ is a node where the drone returns to the truck. $F$ is the set of all possible tuples, and is generated as follows. For each customer node $j$ that a drone can serve, we determine the possible launching node $i$ and the returning node $h$ according to a distance constraint, i.e., the sum of the distance from $i$ to $j$ and the distance from $j$ to $h$ is no greater than the drone's maximum flying distance when the drone carries the customer's cargo.

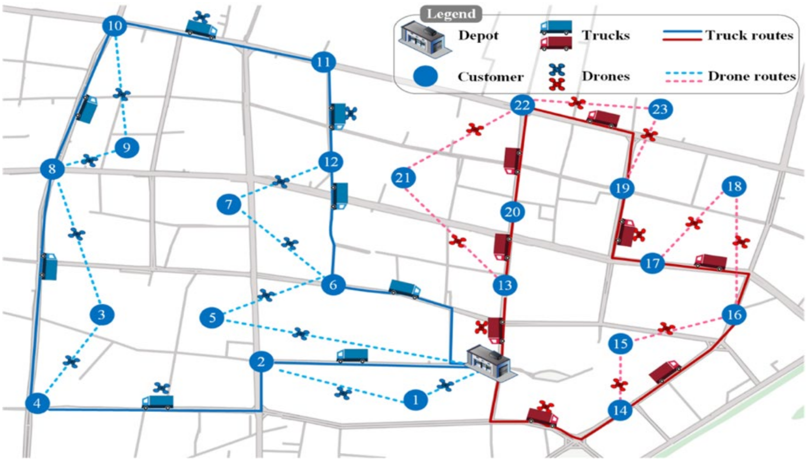


**Figure 2** An example with two truck groups and 23 customers

The aim of the model proposed in this study is to schedule all of the trucks' routes and select suitable trips from the above set $F$ for all of the drones. This minimizes the total cost, which includes the fixed and travel costs of trucks and drones, and the waiting cost of trucks for drones.

Before we address the model, we summarize the underlying assumptions as follows.

(1) Each drone is dedicated to a truck.

(2) A drone can only be launched from and return to its dedicated truck when the truck stops at a customer node or depot, which can occur at different customer points. In addition, this study first assumes a drone visits one customer in one trip; however it is relaxed in the model extension at Appendix 12.

(3) Drone batteries are replaced rather than charged, as this achieves more efficient operation. We assume that there are sufficient batteries for each drone in its truck and that the time for replacing the battery is short, and this is ignored in the model formulation.

(4) The required setup time before the launching of drones is ignored, as it is relatively short in comparison to their travel time.

(5) Each customer location has a space suitable for drone landing and take-off. The power consumed when drones hover in the air is much greater than that required to start their engines, so we assume that the drones stop their engines when they need wait for their trucks. We ignore the waiting cost of drone.

(6) A drone's maximum flying time is influenced by the weight of the goods it carries.

(7) The total demand delivered by a truck route does not exceed the capacity of the truck associated with the route.

(8) The trucks with drones are sufficient so that all the customers could be served.

The first assumption can be relaxed to that multiple drones are dedicated a truck by adding some constraints into the model proposed in this study. For example, when two drones are dedicated to a truck, we define two truck groups in the model to denote the two drones and the truck. The two "virtual" trucks in the two truck groups are actually one "physical" truck. We just need to add some constraints to make the routing and timing decision variables' values of the two "virtual" trucks equal to each other in the model. Then our proposed model and algorithm are still applicable.

## 4. Mathematical formulation

In this section, an MIP model is established for the problem. The model determines the routes of trucks and drones for minimizing the sum of the fixed cost and travel cost of trucks and drones, and the waiting cost of trucks for drones. Below, we first introduce the notation used, followed by the mathematical formulation.

### 4.1. Notation

**Indices and sets:**

$K$ set of the trucks and drones, indexed by $k$.

$N$ set of the nodes, indexed by $i$, $j$ and $h$; $N = \{0,1,2 \dots n, n+1\}$; $0$ and $n+1$ represent the starting and ending depots, respectively.

$N_c$ set of the customers, $N_c = N \backslash \{0, n+1\} = \{1,2 \dots n\}$.

$N_0$ set of the nodes from which a truck or a drone may depart, $N_0 = N \backslash \{n+1\} = \{0,1,2 \dots n\}$.

$N_+$ set of the nodes from which a truck or a drone may visit, $N_+ = N \backslash \{0\} = \{1,2 \dots n, n+1\}$.

$F$ set of the possible path for drones, $(i,j,h) \in F$, $i \in N_0$, $j \in \{N_c : j \neq i\}$, $h \in \{N_+ : h \neq j, h \neq i, t_{ijk}^D + t_{jhk}^D \leq e_{kj}^D\}$.

**Parameters:**

$q_i$ weight of the goods delivered to customer $i$.

$f$ net weight of a drone without carrying any goods.

$m_k^K$ maximum load capacity of truck $k$.

$m_k^D$ maximum load capacity of the drone on truck $k$.

$e_k^D$ maximum flying duration of the drone on truck $k$ without carrying any goods.

$e_{ki}^D$ maximum flying duration when the drone on truck $k$ loads the goods required by customer $i$. If $q_i \leq m_k^D$, $e_{ki}^D = e_k^D \frac{f}{f+q_i}$; else $e_{ki}^D = 0$

$t_{ijk}^K$ time for truck $k$ to travel from node $i$ to node $j$.

$t_{ijk}^D$ time for the drone on truck $k$ to fly from node $i$ to node $j$.

$s^K$ unit transportation cost of a truck per unit of time.

$s^D$ unit transportation cost of a drone per unit of time.

$s^W$ unit waiting cost of a truck per unit of time.

$s^G$ fixed cost of a truck group.

$M$ a sufficiently large positive number.

**Decision variables:**

$\alpha_{ijk}$ binary, equals one if truck $k$ travels from node $i \in N_0$ to node $j \in N_+$, and zero otherwise.

$\beta_{ijhk}$ binary, equals one if the drone on truck $k$ is launched from node $i \in N_0$, and then flies to customer $j \in N_c$, finally returns to truck $k$ or the ending depot at node $h \in N_+$, and zero otherwise.

$\mu_{ik}$ integer, represents the order of node $i$ in the path of truck $k$.

$\gamma_{ijk}$ binary, equals one if truck $k$ serves node $i$ before serving node $j$, and zero otherwise.

$\tau_{ik}^K$ nonnegative continuous, the arrival time of truck $k$ at node $i$.

$\tau_{ik}^D$ nonnegative continuous, the arrival time of the drone on truck $k$ at node $i$.

$\rho_{ik}$ nonnegative continuous, the latest time for truck group $k$ arriving at node $i$.

$\varepsilon_k$ binary, equals one if truck group $k$ is used, and zero otherwise.

### 4.2. An MIP model

Based on the above definitions, a mathematical model is formulated as follows.

$$\text{Minimize } \{\sum_{k\in K} s^K \sum_{i\in N_0} \sum_{j\in N_+} t_{ijk}^K \alpha_{ijk} + \sum_{k\in K} s^D \sum_{i\in N_0} \sum_{j\in N_c} \sum_{h\in N_+} (t_{ijk}^D + t_{jhk}^D) \beta_{ijhk} + \sum_{k\in K} s^G \varepsilon_k + \sum_{k\in K} s^W (\rho_{n+1,k} - \sum_{i\in N_0} \sum_{j\in N_+} t_{ijk}^K \alpha_{ijk})\} \quad (4\text{-}1)$$

subject to

$$\sum_{j\in N_+} \alpha_{0jk} = 1 \qquad \forall k \in K \quad (4\text{-}2)$$

$$\sum_{i\in N_0} \alpha_{i,n+1,k} = 1 \qquad \forall k \in K \quad (4\text{-}3)$$

$$\sum_{\substack{i\in N_0 \\ i\neq j}} \alpha_{ijk} = \sum_{\substack{h\in N_+ \\ h\neq j}} \alpha_{jhk} \leq 1 \qquad \forall j \in N_c, k \in K \quad (4\text{-}4)$$

$$\sum_{k\in K} \sum_{\substack{i\in N_0 \\ i\neq j}} \sum_{\substack{h\in N_+ \\ (i,j,h)\in F}} \beta_{ijhk} \leq 1 \qquad \forall j \in N_c \quad (4\text{-}5)$$

$$\sum_{k\in K} \sum_{\substack{j\in N_c \\ i\neq j}} \sum_{\substack{h\in N_+ \\ (i,j,h)\in F}} \beta_{ijhk} \leq 1 \qquad \forall i \in N_0 \quad (4\text{-}6)$$

$$\sum_{k\in K} \sum_{\substack{i\in N_0 \\ i\neq h}} \sum_{\substack{j\in N_c \\ (i,j,h)\in F}} \beta_{ijhk} \leq 1 \qquad \forall h \in N_+ \quad (4\text{-}7)$$

$$\sum_{k\in K}\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ijk}+\sum_{k\in K}\sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijhk}=1 \qquad \forall j\in N_c \tag{4-8}$$

$$2\beta_{ijhk}\leq\sum_{\substack{j_1\in N_0\\ j_1\neq i}}\alpha_{j_1ik}+\sum_{\substack{i_1\in N_c\\ i_1\neq h}}\alpha_{i_1hk} \qquad \forall i\in N_c, j\in\{N_c: i\neq j\}, k\in K, h\in\{N_+:(i,j,h)\in F\} \tag{4-9}$$

$$\beta_{0jhk}\leq\sum_{\substack{i\in N_0\\ i\neq h}}\alpha_{ihk} \qquad \forall j\in N_c, k\in K, h\in\{N_+:(0,j,h)\in F\} \tag{4-10}$$

$$\tau_{jk}^{K}\geq\rho_{ik}+t_{ijk}^{K}-M\left(1-\alpha_{ijk}\right) \qquad \forall i\in N_0, k\in K, j\in\{N_+: j\neq i\} \tag{4-11}$$

$$\tau_{jk}^{D}\geq\rho_{ik}+t_{ijk}^{D}-M\left(1-\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijhk}\right) \qquad \forall j\in N_c, k\in K, i\in\{N_0: i\neq j\} \tag{4-12}$$

$$\tau_{hk}^{D}\geq\rho_{jk}+t_{jhk}^{D}-M\left(1-\sum_{\substack{i\in N_0\\ (i,j,h)\in F}}\beta_{ijhk}\right) \qquad \forall j\in N_c, k\in K, h\in\{N_+: h\neq j\} \tag{4-13}$$

$$\mu_{hk}-\mu_{ik}\geq 1-(n+2)\left(1-\sum_{\substack{j\in N_c\\ (i,j,h)\in F}}\beta_{ijhk}\right) \qquad \forall i\in N_c, k\in K, h\in\{N_+: h\neq i\} \tag{4-14}$$

$$\mu_{ik}-\mu_{jk}+1\leq(n+2)\left(1-\alpha_{ijk}\right) \qquad \forall i\in N_0, k\in K, j\in\{N_+: j\neq i\} \tag{4-15}$$

$$\mu_{jk}\leq(n+2)\sum_{i\in N_0}\alpha_{ijk} \qquad \forall k\in K, j\in\{N_+: j\neq i\} \tag{4-16}$$

$$\mu_{ik}\geq\mu_{jk}-(n+2)\gamma_{ijk} \qquad \forall i\in N_c, k\in K, j\in\{N_c: j\neq i\} \tag{4-17}$$

$$\mu_{ik}-\mu_{jk}\leq -1+(n+2)\left(1-\gamma_{ijk}\right) \qquad \forall i\in N_c, k\in K, j\in\{N_c: j\neq i\} \tag{4-18}$$

$$\gamma_{0jk}=\sum_{i\in N_0}\alpha_{ijk} \qquad \forall j\in N_+, k\in K \tag{4-19}$$

$$\rho_{i_2k}\geq\tau_{h_1k}^{D}-M\left(3-\sum_{\substack{j_1\in N_c\\ j_1\neq i_2\\ (i_1,j_1,h_1)\in F}}\beta_{i_1j_1h_1k}-\sum_{\substack{j_2\in N_c\\ j_2\neq i_1\\ j_2\neq h_1\\ j_2\neq i_2}}\sum_{\substack{h_2\in N_+\\ (i_2,j_2,h_2)\in F\\ h_2\neq i_1\\ h_2\neq h_1}}\beta_{i_2j_2h_2k}-\gamma_{i_1i_2k}\right)$$

$$\forall i_1\in N_0, h_1\in\{N_+: h_1\neq i_1\}, i_2\in\left\{N_c: i_2\neq i_1, i_2\neq h_1\right\}, k\in K \tag{4-20}$$

$$\tau_{hk}^{D}-\rho_{jk}+t_{ij}^{D}\leq e_{kj}^{D}+M\left(1-\beta_{ijhk}\right) \qquad \forall h\in N_+, j\in\{N_c: j\neq h\}, k\in K, i\in\{N_0:(i,j,h)\in F\} \tag{4-21}$$

$$\beta_{ijhk}\left(t_{ij}^{D}+t_{jh}^{D}\right)\leq e_{kj}^{D} \qquad \forall h\in N_+, j\in\{N_c: j\neq h\}, k\in K, i\in\{N_0:(i,j,h)\in F\} \tag{4-22}$$

$$\rho_{hk}-\rho_{ik}\leq e_{kj}^{D}+M\left(1-\beta_{ijhk}\right) \qquad \forall h\in N_+, j\in N_c, k\in K, i\in\{N_0:(i,j,h)\in F\} \tag{4-23}$$

$$\sum_{i\in N_c}\sum_{j\in N_0}q_i\alpha_{jik}+\sum_{i\in N_c}\sum_{j\in N_0}\sum_{\substack{h\in N_+\\ (j,i,h)\in F}}q_i\beta_{jihk}\leq m_k^{K} \qquad \forall k\in K \tag{4-24}$$

$$q_i\sum_{j\in N_0}\sum_{\substack{h\in N_+\\ (j,i,h)\in F}}\beta_{jihk}\leq m_k^{D} \qquad \forall i\in N_c, k\in K \tag{4-25}$$

$$\varepsilon_k\geq\alpha_{ijk} \qquad \forall i\in N_0, k\in K, j\in\{N_c: j\neq i\} \tag{4-26}$$

$$\varepsilon_k\geq\beta_{ijhk} \qquad \forall h\in N_+, j\in\{N_c: j\neq h\}, k\in K, i\in\{N_0:(i,j,h)\in F\} \tag{4-27}$$

$$\tau_{0k}^{K}=0 \qquad \forall k\in K \tag{4-28}$$

$$\tau_{0k}^{D}=0 \qquad \forall k\in K \tag{4-29}$$

$$\rho_{ik}=max\left\{\tau_{ik}^{D},\tau_{ik}^{K}\right\} \qquad \forall i\in N, k\in K \tag{4-30}$$

$\tau_{ik}^{K} \geq 0 \qquad \forall i \in N, k \in \mathrm{K} \qquad (4\text{-}31)$

$\tau_{ik}^{D} \geq 0 \qquad \forall i \in N, k \in \mathrm{K} \qquad (4\text{-}32)$

$\rho_{ik} \geq 0 \qquad \forall i \in N, k \in K \qquad (4\text{-}33)$

$1 \leq \mu_{ik} \leq n+2, \mu_{ik} \in \mathbb{Z}^{+} \qquad \forall i \in N, k \in K \qquad (4\text{-}34)$

$\alpha_{ijk} \in \{0,1\} \qquad \forall i \in N_0, j \in N_+, k \in K \qquad (4\text{-}35)$

$\beta_{ijhk} \in \{0,1\} \qquad \forall i \in N_0, j \in N_c, k \in K, h \in N_+ \qquad (4\text{-}36)$

$\gamma_{ijk} \in \{0,1\} \qquad \forall i \in N, k \in K, j \in N \qquad (4\text{-}37)$

$\varepsilon_k \in \{0,1\} \qquad \forall k \in K. \qquad (4\text{-}38)$

Objective (4-1) minimizes the total cost, which includes the fixed cost of trucks and drones, the travel cost of trucks, the travel cost of drones, and the waiting cost of trucks. It is the usual practice of the VRP variants to minimize the travel cost related to travel time; the minimization of the fixed cost is to make the obtained plan use as fewer trucks and drones as possible. In addition, the waiting cost (time) that was considered in related literature is also taken account in the objective so as to increase the cooperative degree between the trucks and drones when executing the delivery tasks. Constraints (4-2) and (4-3) guarantee that truck $k$ starts from the depot and eventually returns to the depot. Constraints (4-4) ensure flow conservation and each customer can only be served by a truck at most once. Constraints (4-5) require that each customer can only be served by one drone at most. Constraints (4-6) indicate that at a particular node $i$, only one drone may be launched from that point, at most once. Constraints (4-7) indicate that at a particular node $j$, at most one drone may return to the truck at this point. Constraints (4-8) ensure that each customer needs to be served. Constraints (4-9) state that if the drone on truck $k$ is launched from node $i$ and returns to truck $k$ at node $h$, then nodes $i$ and $h$ must be assigned to truck $k$. Constraints (4-10) state that if the drone on truck $k$ is launched from the depot and returns to truck $k$ at node $h$, then node $h$ must be assigned to truck $k$. Constraints (4-11) ensure the time that truck $k$ arrives at node $j$. Constraints (4-12) and (4-13) limit the time that the drone on truck $k$ arrives at node $j$ and node $h$. In Constraints (4-14), if the drone on truck $k$ is launched from node $i$ and returns to truck $k$ at node $h$, then truck $k$ must serve node $i$ before node $h$. Constraints (4-15) and (4-16) are *subtour elimination* constraints. Constraints (4-17)~(4-19) determine the value of $\gamma_{ijk}$. Constraints (4-20) require that the drone of truck $k$ will be launched from point $i_2$ for the next time is later than the last time the drone returned to truck $k$ at point $i_1$. Constraints (4-21)~(4-23) guarantee the maximum flight time of the drone. In Constraints (4-24), if truck group $k$ forms a route, the total weight of the goods that customers need to be delivered on this route does not exceed the maximum load of truck $k$ serving the route. Constraints (4-25) state that the weight of the goods delivered to each customer by the drone does not exceed the maximum capacity of the drone. Constraints (4-26) and (4-27) state whether truck group $k$ is used. Constraints (4-28) state that the time that truck $k$ departs from the depot is zero. Constraints (4-29) state that the time that drone on truck $k$ departs from the depot is zero.

Constraints (4-30) ensure that truck group $k$'s departure time from node $i$ should be the maximum of the time when truck $k$ arrives at node $i$ and the time when drone on truck $k$ arrives at node $i$. Constraints (4-31)~(4-38) define the decision variables.

In practice, the above formulation cannot be solved to optimality using general purpose MIP solvers even for small-sized instances. Therefore, in the next section we investigate an exact method based on an alternative mathematical formulation for the problem.

**4.3. Computing lower bounds**

In this section, we describe two lower bounds (LBs) on the optimal solution cost of the mathematical formulation. The first lower bound, called LBP (lower bound problem), is obtained by relaxing some of the constraints of the MIP model described in the previous section. The second lower bound requires additional information on the customers distribution and computes a lower estimate on the value of the objective function using the continuous approximation paradigm (see Carlsson and Song, 2018).

The first LB is tighter to the original problem than the second LB; while the computation time of the former one is a bit longer than the time of the latter one. Thus, the first LB will be used as an benchmark in the comparative experiments to validate the quality of solutions solved by our proposed algorithm; while the second LB is embedded in a dynamic programming for accelerating the process of solving the pricing problem.

***4.3.1 Lower bound LBP***

Lower bound LBP is obtained by simply relaxing the integrality Constraints (4-34)～(4-37) as follows:

$$\alpha_{ihk}, \beta_{ijhk}, \gamma_{ihk} \leq 1 \quad \forall i \in N_0, j \in \{N_c: i \neq j\}, h \in \{N_+: (i,j,h) \in F\}, k \in K \tag{4-39}$$

$$\alpha_{ihk}, \beta_{ijhk}, \gamma_{ihk} \geq 0 \quad \forall i \in N_0, j \in \{N_c: i \neq j\}, h \in \{N_+: (i,j,h) \in F\}, k \in K \tag{4-40}$$

$$1 \leq \mu_{ik} \leq n+2 \quad \forall i \in N_+, k \in K. \tag{4-41}$$

The resulting model, which is still an MIP model due to Constraints (4-38), will be used in the computational experiments to compute lower bounds for instances by means of a general-purpose MIP solver.

***4.3.2 A lower bound based on continuous approximation***

This section describes a continuous approximation model to compute a lower bound on the objective function. We assume that there are $n$ customers in the Euclidean plane to be served by a truck group $k$. These customers are assumed to follow an absolutely continuous probability distribution $f(x)$ which is defined on a compact planar region. The computation of the lower bound requires the following information for the truck group $k$:

| | |
|---|---|
| $\Omega_k \subseteq R^2$ | distribution area of the customers served by truck group $k$. |
| $l_k$ | truck route in $\Omega_k$ represented by a loop on the compact planar region. |
| $length(l_k)$ | the length of truck route $l_k$. |
| $v^K$ | speed of the truck. |
| $v^D$ | speed of the drone. |
| $c_k$ | total cost of truck group $k$. |

$f(x)$ continuous probability distribution of customer $x$ being served; it is defined on the compact planar region.

The following proposition holds.

**Proposition 1**. A lower bound ($LB$) on the cost of all truck groups can be computed as follows:

$$LB = \sum_{k\in K}\left\{\frac{s^K}{v^D}\sqrt{\frac{n}{2v^K v^D}}\frac{\iint_{\Omega_k}\sqrt{f(x)}dx}{\left(1-\left(\frac{v^K}{v^D}\right)^2\right)^{1/4}} + \frac{s^D}{v^D}\sqrt{\frac{n}{2}}\frac{\iint_{\Omega_k}\sqrt{f(x)}dx}{\left(1-\left(\frac{v^K}{v^D}\right)^2\right)^{1/4}\sqrt{\frac{v^K}{v^D}}} + \frac{s^W}{v^D}\left[\sqrt{\frac{n}{2v^K v^D}}\frac{\iint_{\Omega_k}\sqrt{f(x)}dx}{\left(1-\left(\frac{v^K}{v^D}\right)^2\right)^{1/4}} - \sqrt{\frac{n}{2v^K v^D}}\frac{\iint_{\Omega_k}\sqrt{f(x)}dx}{\left(1-\left(\frac{v^K}{v^D}\right)^2\right)^{1/4}}\right] + s^G\right\} = \sum_{k\in K}\left\{\frac{length(l_k)^*\left(\frac{s^K}{v^D}+s^D\right)}{v^K} + s^G\right\}. \quad (4\text{-}42)$$

**Proof**: The proof is provided in Appendix 1. ■

The above expression will be used in Section 5.7.2 to speed up the computation of a dynamic programming used to solve the pricing problem of an alternative formulation of the problem, described in the next section.

## 5. B&P&C based solution method

In this section, we describe a B&P&C based exact solution method, which combines column generation, cut generators, and a branch-and-bound solution framework. In the following, Sections 5.1~5.4 describe a set-covering model for the problem and the corresponding column generation algorithm. Section 5.5 presents the branching and node selection strategy, followed by Section 5.6 where valid inequalities are introduced to strengthen the lower bound obtained from the LP-relaxation of the set-covering formulation. Section 5.7 designs a dynamic programming algorithm for solving the pricing problem, followed by Section 5.8 describing some accelerating strategies. For a thorough description of the column generation technique and corresponding solution approaches, the reader is referred to the book of Desaulniers et al. (2005) and the reviews of Barnhart et al. (1998) and Lübbecke and Desrosiers (2005).

### 5.1 Set-covering based model for the problem

The B&P&C method is based on the following set-covering based model for the problem. Let $R_k$ be the set of the possible routes for truck group $k$. Each route includes a truck and a drone. Here, $R = \cup_{k\in K} R_k$ is defined as a set of the possible routes. The following notation is used:

$x_{ijr_k}$ 0-1 coefficient, equal to one if truck $k$ travels from node $i \in N_0$ to node $j \in N_+$ in route $r_k$, and zero otherwise.

$y_{ijhr_k}$ 0-1 coefficient, equal to one if the drone on truck $k$ is launched from node $i \in N_0$, flies to customer $j \in N_c$, and returns to truck $k$ or the ending depot at node $h \in N_+$ in route $r_k$, and zero otherwise.

$c_{r_k}$ cost of route $r_k$.

For each feasible route, $r_k \in R_k$, $\forall k \in K$, let $\xi_{r_k}$ be a binary decision variable, which equals one if route

$r_k$ is selected in solution, and zero otherwise. Based on the above definitions, the set partitioning model of the problem is as follows:

[MP] Minimize $\sum_{k\in K}\sum_{r_k\in R_k} c_{r_k}\xi_{r_k}$ (5-1)

subject to

$$\sum_{k\in K}\sum_{r_k\in R_k}\sum_{\substack{i\in N_0\\ i\neq j}} x_{ijr_k}\,\xi_{r_k} + \sum_{k\in K}\sum_{r_k\in R_k}\sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}} y_{ijhr_k}\,\xi_{r_k} \geq 1 \qquad \forall j\in N_c \qquad (5\text{-}2)$$

$$\sum_{r_k\in R_k}\xi_{r_k} \leq 1 \qquad \forall k\in K \qquad (5\text{-}3)$$

$$\xi_{r_k}\in\{0,1\} \qquad \forall r_k\in R_k, k\in K. \qquad (5\text{-}4)$$

Objective (5-1) minimizes the cost of the routes used in the solution. Constraints (5-2) state each customer is served at least once. Constraints (5-3) ensure each truck group selects at most one route. Constraints (5-4) define the decision variable.

In the above model, the route cost parameter $c_{r_k}$ is calculated as follows.

$$c_{r_k} = s^K\sum_{i\in N_0}\sum_{j\in N_+} t_{ijk}^K x_{ijr_k} + s^D\sum_{i\in cN_0}\sum_{j\in N_c}\sum_{h\in N_+}\left(t_{ijk}^D + t_{jhk}^D\right)y_{ijhr_k} + s^G + s^W\Big(\rho_{n+1,k} - \sum_{i\in N_0}\sum_{j\in N_+} t_{ijk}^K x_{ijr_k}\Big) \qquad \forall r_k\in R_k, k\in K \qquad (5\text{-}5)$$

**5.2 Restricted master problem (RMP)**

The RMP is defined as the LP-relaxation of formulation MP where, in addition, a subset of all feasible routes is selected and represented by $R' = \cup_{k\in K} R'_k \subseteq R$. The initial feasible solution $R'$ is computed using a heuristic algorithm, which is described in Section 5.4. The RMP in then formulated as follows.

[RMP] Minimize $\sum_{k\in K}\sum_{r_k\in R'_k} c_{r_k}\xi_{r_k}$ (5-7)

subject to

$$\sum_{k\in K}\sum_{r_k\in R'_k}\sum_{\substack{i\in N_0\\ i\neq j}} x_{ijr_k}\,\xi_{r_k} + \sum_{k\in K}\sum_{r_k\in R'_k}\sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}} y_{ijhr_k}\,\xi_{r_k} \geq 1 \qquad \forall j\in N_c \qquad (5\text{-}8)$$

$$\sum_{r_k\in R'_k}\xi_{r_k} \leq 1 \qquad \forall k\in K \qquad (5\text{-}9)$$

$$\xi_{r_k}\geq 0 \qquad \forall r_k\in R'_k, k\in K. \qquad (5\text{-}10)$$

In the above model, the route cost parameter $c_{r_k}$ is calculated as follows, here $r_k$ belongs to the above defined set $R'_k$, i.e., $r_k\in R'_k$.

$$c_{r_k} = s^K\sum_{i\in N_0}\sum_{j\in N_+} t_{ijk}^K x_{ijr_k} + s^D\sum_{i\in N_0}\sum_{j\in N_c}\sum_{h\in N_+}\left(t_{ijk}^D + t_{jhk}^D\right)y_{ijhr_k} + s^G + s^W\Big(\rho_{n+1,k} - \sum_{i\in N_0}\sum_{j\in N_+} t_{ijk}^K x_{ijr_k}\Big) \qquad \forall r_k\in R'_k, k\in K \qquad (5\text{-}11)$$

At each iteration of the column generation, a new column is added to the RMP until optimality is proven. The dual variables of the RMP are transferred to the pricing problem (PP), which is used to generate new columns. The dual variables of the RMP are defined as follows:

$\pi_j^1$ dual variables for Constraint (5-8), $\forall j\in N_c$.

$\pi_k^2$ dual variables for Constraint (5-9), $\forall k\in K$.

### 5.3 Pricing problem (PP)

The aim of the pricing problem (PP) is to obtain feasible route plans having negative reduced cost. At each iteration of the algorithm, $|K|$ pricing problems need to be solved, each of which corresponds to one truck group and generates a feasible route plan. In the following, we define the pricing problem $PP_k$ , for a generic truck $k$. For sake of the exposition, the parameters and variables used in the following model $PP_k$ will omit the subscript $k$.

The mathematical model of the $PP_k$ corresponding to truck group $k$ is formulated as follows:

$$[PP_k]\ \text{Minimize}\ \sigma_k = c_{r_k} - \sum_{j\in N_c}\left(\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh}\right)\pi_j^1 - \pi_k^2 \qquad (5\text{-}12)$$

subject to

$$\sum_{j\in N_+}\alpha_{0j} = 1 \qquad (5\text{-}13)$$

$$\sum_{i\in N_0}\alpha_{i,n+1} = 1 \qquad (5\text{-}14)$$

$$\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} = \sum_{\substack{h\in N_+\\ h\neq j}}\alpha_{jh} \leq 1 \qquad \forall j\in N_c \qquad (5\text{-}15)$$

$$\sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh} \leq 1 \qquad \forall j\in N_c \qquad (5\text{-}16)$$

$$\sum_{\substack{j\in N_c\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh} \leq 1 \qquad \forall i\in N_0 \qquad (5\text{-}17)$$

$$\sum_{\substack{i\in N_0\\ i\neq h}}\sum_{\substack{j\in N_c\\ (i,j,h)\in F}}\beta_{ijh} \leq 1 \qquad \forall h\in N_+ \qquad (5\text{-}18)$$

$$\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh} \leq 1 \qquad \forall j\in N_c \qquad (5\text{-}19)$$

$$\sum_{j\in N_c}\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{j\in N_c}\sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh} \geq 1 \qquad (5\text{-}20)$$

$$2\beta_{ijh} \leq \sum_{\substack{j_1\in N_0\\ j_1\neq i}}\alpha_{j_1 i} + \sum_{\substack{i_1\in N_c\\ i_1\neq h}}\alpha_{i_1 h} \quad \forall i\in N_c, j\in\{N_c: i\neq j\}, h\in\{N_+:(i,j,h)\in F\} \qquad (5\text{-}21)$$

$$\beta_{0jh} \leq \sum_{\substack{i\in N_0\\ i\neq h}}\alpha_{ih} \qquad \forall j\in N_c, h\in\{N_+:(0,j,h)\in F\} \qquad (5\text{-}22)$$

$$\tau_j^K \geq \rho_i^K + t_{ij}^K - M(1-\alpha_{ij}) \qquad \forall i\in N_0, j\in\{N_+: j\neq i\} \qquad (5\text{-}23)$$

$$\tau_j^D \geq \rho_i^D + t_{ij}^D - M\left(1 - \sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh}\right) \qquad \forall j\in N_c, i\in\{N_0: i\neq j\} \qquad (5\text{-}24)$$

$$\tau_h^D \geq \rho_j^D + t_{jh}^D - M\left(1 - \sum_{\substack{i\in N_0\\ (i,j,h)\in F}}\beta_{ijh}\right) \qquad \forall j\in N_c, h\in\{N_+: h\neq j\} \qquad (5\text{-}25)$$

$$\mu_h - \mu_i \geq 1 - (n+2)\left(1 - \sum_{\substack{j\in N_c\\ (i,j,h)\in F}}\beta_{ijh}\right) \qquad \forall i\in N_c, h\in\{N_+: h\neq i\} \qquad (5\text{-}26)$$

$$\mu_i - \mu_j + 1 \leq (n+2)(1-\alpha_{ij}) \qquad \forall i\in N_0, j\in\{N_+: j\neq i\} \qquad (5\text{-}27)$$

$$\mu_j \leq (n+2)\sum_{i\in N_0}\alpha_{ij} \qquad \forall j\in\{N_+: j\neq i\} \qquad (5\text{-}28)$$

$$\mu_i \geq \mu_j - (n+2)\gamma_{ij} \qquad \forall i\in N_c, j\in\{N_c: j\neq i\} \qquad (5\text{-}29)$$

$$\mu_i - \mu_j \le -1 + (n+2)\big(1-\gamma_{ij}\big) \qquad \forall i \in N_c, j \in \{N_c : j \ne i\} \qquad (5\text{-}30)$$

$$\gamma_{0j} = \textstyle\sum_{i \in N_0} \alpha_{ij} \qquad \forall j \in N_+ \qquad (5\text{-}31)$$

$$\rho_{i_2}^D \ge \tau_{h_1}^D - M\left(3 - \sum_{\substack{j_1 \in N_c \\ j_1 \ne i_2 \\ (i_1, j_1, h_1) \in F}} \beta_{i_1 j_1 h_1} - \sum_{\substack{j_2 \in N_c \\ j_2 \ne i_1 \\ j_2 \ne h_1 \\ j_2 \ne i_2}} \sum_{\substack{h_2 \in N_+ \\ (i_2, j_2, h_2) \in F \\ h_2 \ne i_1 \\ h_2 \ne h_1}} \beta_{i_2 j_2 h_2} - \gamma_{i_1 i_2}\right)$$

$$\forall i_1 \in N_0, h_1 \in \{N_+ : h_1 \ne i_1\}, i_2 \in \big\{N_c : i_2 \ne i_1, i_2 \ne h_1\big\} \qquad (5\text{-}32)$$

$$\tau_h^D - \rho_j^D + t_{ij}^D \le e_j^D + M\big(1 - \beta_{ijh}\big) \qquad \forall h \in N_+, j \in N_c, i \in \{N_0 : (i,j,h) \in F\} \qquad (5\text{-}33)$$

$$\beta_{ijh}\big(t_{ij}^D + t_{jh}^D\big) \le e_j^D \qquad \forall h \in N_+, j \in N_c, i \in \{N_0 : (i,j,h) \in F\} \qquad (5\text{-}34)$$

$$\rho_h^D - \rho_i^D \le e_j^D + M\big(1 - \beta_{ijh}\big) \qquad \forall h \in N_+, j \in N_c, i \in \{N_0 : (i,j,h) \in F\} \qquad (5\text{-}35)$$

$$\textstyle\sum_{i \in N_c} \sum_{j \in N_0} q_i \alpha_{ji} + \sum_{i \in N_c} \sum_{j \in N_0} \sum_{\substack{h \in N_+ \\ (j,i,h) \in F}} q_i \beta_{jih} \le m^K \qquad (5\text{-}36)$$

$$q_i \textstyle\sum_{j \in N_0} \sum_{\substack{h \in N_+ \\ (j,i,h) \in F}} \beta_{jih} \le m^D \qquad \forall i \in N_c \qquad (5\text{-}37)$$

$$\tau_0^K = 0 \qquad (5\text{-}38)$$

$$\tau_0^D = 0 \qquad (5\text{-}39)$$

$$\rho_i = max\big\{\tau_i^D, \tau_i^K\big\} \qquad \forall i \in N \qquad (5\text{-}40)$$

$$1 \le \mu_i \le n+2, \mu_i \in \mathbb{Z}^+ \qquad \forall i \in N \qquad (5\text{-}41)$$

$$\tau_i^K \ge 0 \qquad \forall i \in N \qquad (5\text{-}42)$$

$$\tau_i^D \ge 0 \qquad \forall i \in N \qquad (5\text{-}43)$$

$$\rho_i \ge 0 \qquad \forall i \in N \qquad (5\text{-}44)$$

$$\alpha_{ij} \in \{0,1\} \qquad \forall i \in N_0, j \in N_+ \qquad (5\text{-}45)$$

$$\beta_{ijh} \in \{0,1\} \qquad \forall i \in N_0, j \in N_c, h \in N_+ \qquad (5\text{-}46)$$

$$\gamma_{ij} \in \{0,1\} \qquad \forall i \in N, j \in N \qquad (5\text{-}47)$$

$$c_{r_k} = s^K \textstyle\sum_{i \in N_0} \sum_{j \in N_+} t_{ij}^K \alpha_{ij} + s^D \sum_{i \in N_0} \sum_{j \in N_c} \sum_{h \in N_+} \big(t_{ij}^D + t_{jh}^D\big) \beta_{ijh} + s^G + s^W \big(\rho_{n+1} - \sum_{i \in N_0} \sum_{j \in N_+} t_{ij}^K \alpha_{ij}\big). \qquad (5\text{-}48)$$

Objective (5-12) minimizes the reduced cost. Constraints (5-13)~(5-18) and Constraints (5-21)~(5-47) are similar to Constraints (4-2)~(4-7), Constraints(4-9)~(4-25) and Constraints (4-28)~(4-37), respectively. Thus, the explanation is omitted here for conciseness. Constraints (5-19) and (5-20) ensure that all customers should be served at most once, and at least one customer needs to be served. Constraint (5-48) is the calculation of the route's cost, i.e., value $c_{r_k}$, which is used in the objective function.

### 5.4 Generation of the initial solution

A set of initial feasible route plans for the RMP needs to be generated to initialize the column generation procedure. To generate the initial solution, we designed a greedy heuristic that performs the following steps.

**Step 1**: Sort the customers according to the increasing order of their demands. Initialize an empty truck route.

**Step 2**: Following the customer ordering, add the customers to the emerging route until the vehicle capacity constraint is satisfied. If no additional customers can be added, initialize a new truck route, and repeat Step 2 until all customers have been served.

**Step 3**: For each truck route defined at Step 2, apply the well-known 2-opt and 3-opt local search procedures.

**Step 4**: For each truck route, evaluate the insertion of drone routes by considering different drone constraints.

**5.5 Branching and node selection strategy**

Based on the branching scheme used by Zhen et al. (2018), the branching and node selection strategy used in this paper are designed as follows.

We decide the assignment of customer $j$ to truck group $k$ according to the expression $\varsigma_{jk} = \frac{\sum_{r_k \in R_k}\left(\sum_{i \in N_0, i \neq j} x_{ijr_k} + \sum_{i \in N_0, i \neq j} \sum_{h \in N_+, (i,j,h) \in F} y_{ijhr_k}\right)\xi^*_{r_k}}{\sum_{k \in K}\sum_{r_k \in R_k}\left(\sum_{i \in N_0, i \neq j} x_{ijr_k} + \sum_{i \in N_0, i \neq j} \sum_{h \in N_+, (i,j,h) \in F} y_{ijhr_k}\right)\xi^*_{r_k}}$. We calculate the value $\sum_{r_k \in R_k}\left(\sum_{i \in N_0, i \neq j} x_{ijr_k} + \sum_{i \in N_0, i \neq j} \sum_{h \in N_+, (i,j,h) \in F} y_{ijhr_k}\right)\xi^*_{r_k}$ and the value $\sum_{k \in K}\sum_{r_k \in R_k}\left(\sum_{i \in N_0, i \neq j} x_{ijr_k} + \sum_{i \in N_0, i \neq j} \sum_{h \in N_+, (i,j,h) \in F} y_{ijhr_k}\right)\xi^*_{r_k}$, where $\xi^*_{r_k}$ is the optimal solution of RMP. If the calculated value of $\varsigma_{jk}$ is the closest to 0.5, we assign the customer $j$ to the truck group $k$. Then, we divide the parent node into two child nodes. For the branch that generates the left child node, truck group $k$ must serve customer $j$. For the branch that generates the right child node, customer $j$ must be served by other truck groups.

The branch constraints are added to the RMP and pricing problem in the following way. For the first branch, which requires that truck group $k$ must serve customer $j$, we remove all routes of truck group $k$ that do not serve customer $j$, and delete all routes in the RMP that assign customer $j$ to other truck groups. For problem PP and truck group $k$, we require that the newly generated route must include customer $j$. For the remaining groups, the newly generated route cannot include customer $j$. For the second branch, which requires that truck group $k$ cannot serve customer $j$, we remove all routes of truck group $k$ that serve customer $j$. For problem PP and truck group $k$, we require that the newly generated route cannot include customer $j$. For the remaining groups, no constraints are added. When all values of $\varsigma_{jk}$, $\forall j \in N_c$, $k \in K$ are integers, the optimal solutions of the RMP are integral and feasible, as shown below.

After the optimal solutions of RMP are obtained, if there is no fractional value of $\varsigma_{jk}$, $\forall j \in N_c$, for a given truck group $k$, there are two cases for the values of $\varsigma_{jk}$, $\forall j \in N_c$. In the first case, no customer is assigned to truck group $k$, $\varsigma_{jk} = 0$, $\forall j \in N_c$. In the second case, there is at least one customer assigned to truck group $k$, all $\varsigma_{jk}$, $\forall j \in N_c$ are integral. When considering the selection of routes in the second case, only two possible selections exist: (1) one route is precisely selected for customer $j$, thereby indicating that $\xi^*_{r_k}$, $\forall r_k \in R$ are integer; and (2) more than one fractional route is chosen for truck group $k$. In this case, assume that two fractional routes are chosen and denoted by $r_{k1}$ and $r_{k2}$, respectively. The value of $\left(\sum_{i\in N_0, i\neq j} x_{ijr_{k1}} + \sum_{i\in N_0, i\neq j}\sum_{h\in N_+, (i,j,h)\in F} y_{ijhr_{k1}}\right)$ and $\left(\sum_{i\in N_0, i\neq j} x_{ijr_{k2}} + \sum_{i\in N_0, i\neq j}\sum_{h\in N_+, (i,j,h)\in F} y_{ijhr_{k2}}\right)$ must be exactly the same, and $\frac{\left(\xi^*_{r_{k1}}+\xi^*_{r_{k2}}\right)}{\sum_{k\in K}\left(\xi^*_{r_k}\right)}$ must be equal to an integer, in order to ensure that all values of $\varsigma_{jk},\ \forall j\in N_c$ are integral. These two routes represent the same route with an integral value of $\frac{\left(\xi^*_{r_{k1}}+\xi^*_{r_{k2}}\right)}{\sum_{k\in K}\left(\xi^*_{r_k}\right)}$. Thus, for this selection, one route is precisely chosen for truck group $k$. Thus, if there is no fractional value of $\varsigma_{jk},\ \forall j\in N_c$, all values of $\xi^*_{r_k}\ \ \forall r_k\in R_k, k\in K$ are integral, and the integral routes have been chosen, which means that the optimal solutions of RMP are integral.

If the current node has not been explored, the depth-first-search rule is used; otherwise, the best-lower-bound rule is used, which means the node with the smallest lower bound is selected from the unexplored nodes as the next node to be explored. If all nodes have been explored, the whole process stops, and the algorithm terminates with an optimal solution.

### 5.6 Strengthening the lower bound value

To speed up the algorithm, the lower bounds provided by CG can be improved by adding valid inequalities to the RMP. This section presents the valid inequalities for the RMP. The valid inequalities are added to the RMP in a cutting plane fashion.

We add the following *rounded capacity inequalities* and *subset-row inequalities* to the RMP.

Let $C$ be a subset of $N_c$, $C\subseteq N_c$, $R'_k(C)$ be the set of routes of truck group $k$ that visits at least one customer in set $C$, and $p^C_{r_k}$ be a 0-1 parameter. If $r_k\in R'_k(C)$, $p^C_{r_k}$ equals one, otherwise zero. *The rounded capacity inequalities* are inspired to similar inequalities designed for the Capacitated VRP (see Toth and Vigo, 2014) and are defined as:

$$\sum_{k\in K}\sum_{r_k\in R'_k} p^C_{r_k}\xi_{r_k} \geq \left\lceil \frac{\sum_{j\in C} q_j}{m^K_k} \right\rceil \qquad \forall C\subseteq N_c. \tag{5-49}$$

The separation problem for capacity constraint (5-49) is NP-complete (see Augerat et al. 1998). Augerat et al. (1998) and Ralphs et al. (2003) designed several separation heuristics for these constraints, and in our implementation, we used the heuristic, called the *greedy randomized algorithm*, proposed by Augerat et al. (1998) to separate inequalities (5-49). We refer he reader to Augerat et al. (1998) for the corresponding details.

Constraints (5-49) are added to the RMP. The dual variable for Constraints (5-49) and a given set $C$ is defined as $\pi^3_C$. In the $PP_k$, $p^C$ defined as a binary decision variable, which is equal to one if the route of truck group that visits at least one customer in set $C$, and zero otherwise. The objective of pricing problem is revised as the formula (5-50); and the variable $p^C$ related Constraints (5-51) and (5-52) are also added as follows.

$$\text{Minimize } \sigma_k = c_{r_k} - \sum_{j\in N_c}\left(\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh}\right)\pi_j^1 - \sum_{C\subseteq N_c} p^C\pi_C^3 - \pi_k^2. \tag{5-50}$$

$$p^C \geq \sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh} \qquad \forall j\in C, C\subseteq N_c \tag{5-51}$$

$$p^C \in \{0,1\} \qquad \forall C\subseteq N_c. \tag{5-52}$$

In addition, Jepsen et al. (2008) introduced subset-row inequalities in the VRPTW for the first time and proved that subset-row inequalities are valid for the set partitioning model. For any $B \subseteq N_C$ and an integer $p \in \mathbb{N}$ such that $0 < p \leq |B|$, the valid inequalities obtained by the Chvatal-Gomory rounding of the partitioning constraints are as follows, also called Subset Rows Inequalities (SRI):

$$\sum_{k\in K}\sum_{r_k\in R_k'}\left\lfloor\frac{1}{p}\sum_{j\in B}\left(\sum_{i\in N_0}x_{ijr_k} + \sum_{i\in N_0}\sum_{h\in N_+}y_{ijhr_k}\right)\right\rfloor\xi_{r_k} \leq \left\lfloor\frac{|B|}{p}\right\rfloor. \tag{5-53}$$

In order to have a good tradeoff between the quality of the lower bound obtained by adding inequality (5-53) to the RMP and the complexity of the resulting pricing problem, we decided to use the subset row inequalities with $|B| = 4$ and $p = 3$, i.e.:

$$\sum_{k\in K}\sum_{r_k\in R_k'}\left\lfloor\frac{1}{3}\sum_{j\in B}\left(\sum_{i\in N_0}x_{ijr_k} + \sum_{i\in N_0}\sum_{h\in N_+}y_{ijhr_k}\right)\right\rfloor\xi_{r_k} \leq 1. \tag{5-54}$$

The inequalities (5-54) are separated by complete enumeration of the possible sets *B*.

Constraints (5-54) are added to the RMP. The dual variable for constraint (5-54) is defined as $\pi_B^4$. After adding the inequalities to the RMP, the objective function of the $PP_k$ is rewritten as follows:

$$\text{Minimize } \sigma_k = c_{r_k} - \sum_{j\in N_c}\left(\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh}\right)\pi_j^1 - \pi_k^2 - \sum_{C\subseteq N_c} p^C\pi_C^3 -$$

$$\sum_{B\subseteq N_c:\,|B|=4}\left\lfloor\frac{1}{3}\sum_{j\in B}\left(\sum_{\substack{i\in N_0\\ i\neq j}}\alpha_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}}\sum_{\substack{h\in N_+\\ (i,j,h)\in F}}\beta_{ijh}\right)\right\rfloor\pi_B^4. \tag{5-55}$$

## 5.7 Solving the pricing problem

PP is a new variant of Elementary Shortest Path Problem with Resource Constraints (ESPPRC). For solving the PP efficiently, this section proposes a novel algorithm, which is accelerated by using the dynamic programming and calculus approximation principle.

### *5.7.1 Dynamic programming*

Dynamic programming can be used to solve the PP. As mentioned in Section 3, a solution for the problm can be broken down into a concatenation of 3-tuple of the form and 2-tuple of the form. Each 3-ruple of the form is a sequence of combined arcs of a drone, and each 2-tuple of the form is an arc of a truck. Let → connect to the truck routes and ⇢ connect to the drone routes. The solution illustrated in Figure 2 is the routes for two truck groups, which can be represented as ( [Depot → 2, Depot ⇢ 1 ⇢ 2], [2 → 4], [4 → 8,4 ⇢ 3 ⇢ 8], [8 → 10,8 ⇢ 9 ⇢ 10], [10 → 11], [11 → 12], [12 → 6,12 ⇢ 7 ⇢ 6], [6 → Depot, 6 ⇢ 5 ⇢ Depot] ), ( [Depot →

14], [14 → 16,14 ⇢ 15 ⇢ 16], [16 → 17,16 ⇢ 18 ⇢ 17], [17 → 19], [19 → 22,19 ⇢ 23 ⇢ 22], [22 → 20], [20 → 13,22 ⇢ 21 ⇢ 13], [13 → Depot]). Each stage can be generated by first generating the 2-tuple of the form for trucks, and then adding the drone routes. The dynamic programming we describe in this section are based on the idea that a solution for the problem can be decomposed into a set of 3-tuple of the form and 2-tuple of the form. Therefore, a complete solution can be generated by sequentially adding a 3-tuple of the form or 2-tuple of the form at a time to a partial solution.

The PP starts from the source state $(\{v_0\}, v_0)$. The truck group travels from $v_0$, visits all the nodes that meet the capacity constraints, and then ends at $v_0$. The PP is to find the path with the smallest reduced cost in the above process. Let Stage $m$ be the $|m|$ sub-states that have been explored. Since there are multiple nodes in the sub-state, the end state $m$ is less than the total number of nodes. Let the maximum value of $m$ be $Z$, $0 \leq m \leq Z \leq n+1$; $m=0$ represents the starting point, and $m=Z$ is the destination point.

Given the set of partial $PP_k$ solutions, we define the function for the value of the reduced cost at Stage $m$:

$$f_m(j, d, V_{m+1}) \tag{5-56}$$

where $j$ is the end node of the customer that the truck will serve, $d$ is the customer node that the drone will serve at this stage, and $V_{m+1}$ is the set of nodes that the truck or the drone may pass through from Stage 0 to Stage $m$.

Let $x_m = (i, d', V_m)$ be the state of the stage from 0 to $m$, where $i$ represents the customer that the truck serves, $d'$ represents the customer that the drone serves, and $V_m$ is the set of all the nodes that the truck or the drone may pass through from node 0 to node $i$ and node $d'$; $V_m$ excludes node $i$ and $d'$. $V_1 = \emptyset$, $V_Z = \{0,1,\dots,n\}$, $x_1 = (0,0,\emptyset)$.

**Proposition 2**. The optimal path satisfying all the capacity constraints and the minimum reduced cost are obtained through the recursive equation.

$$f_m(j, d, V_{m+1}) = \begin{cases} \infty, & \text{if } j \notin S_{m+1} \\ t_{0j}^{K} s^{K}, & \text{if } S_{m+1} = \{j\} \\ \min\{D_{op}(i,j,d) + f_{m-1}(x_m)\}, & \text{otherwise} \end{cases} \tag{5-57}$$

$$f_0(x_1) = f_0(0,0,\emptyset) = 0$$

where $D_{op}(i,j,d)$ is the reduced cost of the path where the truck starts from node $i$, ends at node $j$, and the drone services node $d$.

$D_{op}(i,j,d)$ is related to the traveling time of the truck and which customer can be served by drone on the truck route. Thus, we determine the traveling time of the truck firstly, then look for customers that can be served by drone on the truck route, and obtain the reduce cost and sub-path of the sub-state.

**Proof**: See Appendix 2. ■

**Algorithm 1: Determine the traveling time of the truck**

This algorithm computes the total time $T(P_v, w)$ of path $P_v$ as the sum of the travel time of the arcs composing the path. $T(P_v, w)$ is the time of a path starting at node $v$, visiting nodes in $P_v$ and ending at $w$. The path $P_v$ can be represented as $(v, u_1, u_2, \cdots, w)$.

$$T(P_v, w) = \begin{cases} \infty, & \text{if } w \notin P_v \\ t_{vw}^K, & \text{if } P_v = \{w\} \\ T(P_v \setminus \{w\}, u) + t_{uw}^K & \text{otherwise}. \end{cases} \tag{5-58}$$

The details of Algorithm 1 are provided in Appendix 3.

**Algorithm 2: Compute the reduce cost**

Based on the truck paths calculated, drone paths are added into truck paths. In the sub-state, the optimization problem of the truck group starting from node $v$, serving node $d$ and ending at node $w$ can represent the shortest path problem, in which the truck starts from node $v$, ends at node $w$, and the drone serves node $d$, returns to the truck at node $w$. For the point where the drone takes off, it is denoted by $h$, which is in the truck route that the truck group has traveled. The truck path is the shortest path starting from node $v$ and ending at node $w$. For the takeoff point $h$ of the drone, the constraints of the flying duration time of the drone and the non-crossing routes of the drone need to be satisfied. Among the set of possible takeoff points, we choose the takeoff point $h$ corresponding to the path with the smallest reduced cost.

The reduced cost of a truck group in the sub-state is calculated according to Formula (5-59), where $P_v$ represents a set of customers that the truck serves sequentially starting from node $v$, $P_v \subset N$, $w \in P_v \backslash \{v\}$. $D_{op}(v, w, d)$ means the reduced cost of the path where the truck starts from node $v$, ends at node $w$, and the drone services node $d$.

$$D_{op}(v, w, d) = \begin{cases} \infty, & v = w \\ t_{vw}^K s^K - \pi_w^1 & (v, w, d) \notin F \text{ and } v \neq w \\ \min\limits_{h \in P_0 \backslash \{w\}} \left\{ \begin{array}{c} [max\{(t_{hd}^D + t_{dw}^D) - T(P_h \backslash \{d\}, w), 0\}] s^w \\ + T(P_v \backslash \{d\}, w) s^K + s^D (t_{hd}^D + t_{dw}^D) \end{array} \right\} - \pi_w^1 - \pi_d^1 & \text{if} (v, w, d) \in F. \end{cases} \tag{5-59}$$

Details of Algorithm 2 are provided in Appendix 4. According to all the descriptions above, the whole flow of dynamic programming (named by Algorithm 3) for solving the pricing problem is provided in Appendix 5.

### *5.7.2 Accelerating the dynamic programming*

This subsection presents a novel method to accelerate the dynamic programming by using a lower bound, which is based on the principle of calculus approximation (Carlsson and Song, 2018). Before presenting this accelerating tactic, some required parameters are defined first as follows.

$\Omega_k$ distribution area of the customers served in the $PP_k$.

$\Omega_C$ distribution area of the customers in the set $C$ of the RMP Constraints (5-49).

$\Omega_B$ distribution area of the customers in the set $B$ of the RMP Constraints (5-54).

$f(x)$ continuous probability distribution of customer $x$ being served in the $PP_k$.

$l_k$ truck route in $\Omega_k$.

Based on these parameters, Section 4.3, and Section 5.6, a formula is obtained to calculate the lower bound of reduced cost. The formula is shown in the following proposition.

**Proposition 3**. At stage $m$ of dynamic programming, some required parameters are defined as follows. $\Omega_{km}$ is he distribution area of the customers served in the PP$_k$ from stage 0 to stage $m$. $l_{km}$ is the truck route in $\Omega_{km}$. $P_{km}$ the set of the customers served in the PP$_k$ from stage 0 to stage $m$. Let $\sigma_{km}$ be the reduced cost in the PP$_k$ from stage 0 to stage $m$. A lower bound of reduced cost in the PP$_k$ from stage 0 to stage $m$ shows that:

$$LB_{\sigma_{km}} = \frac{length(l_{km})^{*}\left(\frac{s^K}{v^D}+s^D\right)}{v^K} + s^G - \pi_k^2 - \sum_{j\in N_c}\pi_j^1 - \sum_{C\subseteq P_{km}}\pi_C^3. \quad (5\text{-}60)$$

**Proof**: See Appendix 6. ■

We combine the dynamic programming with the principle of calculus approximation to solve the PP$_k$ efficiently. Let $z_c$ be the reduced cost from stage 0 to the current stage. Before performing the dynamic programming, we calculate the lower bound ($LB$) according to formula (5-60) for each stage $m$. If $LB$ is less than $z_c$, the dynamic programming is performed and the reduced cost is $f_m(j, d, V_{m+1}) + s^G - \pi_k^2$; otherwise, the next node is selected. After traversing all the nodes that satisfy the truck capacity constraint, the optimal solution of the PP$_k$ is obtained. The above new dynamic programming algorithm (named by Algorithm 4) with the acceleration based on the calculus approximation is elaborated in Appendix 7.

### 5.8. A rounding heuristic

To improve the value of the incumbent primal solution, we apply a rounding heuristic at the different nodes of the numeration tree. At a generic iteration of the column generation procedure, let $p$ be the current fractional solution of cost $z_{lp}(p)$. Based on solution $p$, we apply the rounding heuristic to obtain integer solution. Besides fractional solution $p$ and cost $z_{lp}(p)$, the input of the rounding heuristic also contains the assignment values of each customer to truck group, denoted by $node\_solution_{ik}$. The output contains $UB$, updated assignment value $best\_solution_{ik}$, other decision variables' values ($\alpha_{ij}$, $\beta_{ijh}$, $\gamma_{ij}$, $\mu_i$, $\tau_i^K$, $\tau_i^D$, $\rho_i$), and global optimal objective value $best$. The core idea of the rounding heuristic is to assign customers to truck groups according to the priority related to the assignment value, i.e., $node\_solution_{ik}$; the capacity of each truck group should also be taken into account during the above assignment decision. The detailed pseudo code of the rounding heuristic is presented in Appendix 8.

## 6. Numerical experiments

In this section, we present numerical experiments for evaluating the performance of the B&P&C algorithm. Experiments were 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 model and algorithm were implemented in C# (VS2019) concert technology and CPLEX 12.6.1 was used as the MIP and LP solver. The time limit for all of the test instances was set to three hours (10,800 seconds).

### 6.1 Benchmark sets

We generated instances to simulate a real truck and drone distribution scenario based on the set introduced in Poikonen et al. (2019), i.e., poi-10-1~ poi-10-10 and poi-15-1~ poi-15-10. We used the Manhattan distance to define the distance between any two nodes for a truck, as they are driven in a real traffic network, and the Euclidean distance to define that between any two nodes for a drone. If $(i, j)$ and $(i', j')$ are the coordinates of node $i$ and node $j$, the Manhattan distance between the two nodes is $|i - i'| + |j - j'|$, and the Euclidean distance between the two nodes is $\sqrt{(i - i')^2 + (j - j')^2}$. We assume that the speed of trucks is 30 km/h and the speed of drones is 48 km/h, and the parameters of $m^k$, $m^D$ and $e_f$ are set to 200 kg, 2.3 kg and 0.5 hours (based on Mayerowitz, 2013; Wang and Sheu, 2019), respectively. Customer demands are randomly generated in the interval (0, 50] kg. The gasoline price on September 9, 2021 was 6.93 CNY/L in Shanghai, and the average fuel consumption of a truck is 0.10 L/km, so $s^K$ is 20.79 CNY/h. We defined $s^D$ as 12.35 CNY/h, as the cost incurred by a drone is typically 0.19 CNY/km (Wang and Sheu, 2019). The fuel consumption of a truck for one minute of idling is one third of its normal traveling consumption, so the parameter $s^W$ is set to 6.93 CNY/h, and $s^G$ to 100 CNY per truck group. We considered two instance groups (ISGs) in the computational experiments, ISG1 and ISG2, as shown in Table 2. Ten instances were generated per group.

**Table 2** Instance groups

| Group ID | Number of customers ($\|N_c\|$) | Number of truck groups ($\|K\|$) |
|---|---|---|
| ISG1 | 9 | 2 |
| ISG2 | 14 | 3 |

### 6.2 Effectiveness of the valid inequalities

We conducted experiments to evaluate the performance of the B&P&C algorithm using different combinations of valid inequalities or cuts. We first considered a variant in which the cuts are used only at the root node of the B&P&C. We evaluated the following cases: Constraints (5-49); (5-54); (5-49); and (5-54). The numerical results are shown in Table 3.

**Table 3** Effectiveness of the different cuts once used at the root node of the B&P&C

| Instances | | Solution and time for different combinations cuts | | | | | | | | Gap | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scale | ID | $F_0$ | $t_0$(s) | $F_1$ | $t_1$(s) | $F_2$ | $t_2$(s) | $F_3$ | $t_3$(s) | $\Delta_{F1}$ | $\Delta_{t1}$ | $\Delta_{F2}$ | $\Delta_{t2}$ | $\Delta_{F3}$ | $\Delta_{t3}$ |
| ISG1 | 1 | 324 | 73 | 324 | 67 | 324 | 64 | 324 | 70 | 0.00% | -8.96% | 0.00% | -12.33% | 0.00% | -4.11% |
| | 2 | 331 | 131 | 331 | 143 | 331 | 119 | 331 | 122 | 0.00% | 8.39% | 0.00% | -9.16% | 0.00% | -6.87% |
| | 3 | 332 | 77 | 332 | 68 | 332 | 82 | 332 | 73 | 0.00% | -13.24% | 0.00% | 6.49% | 0.00% | -5.19% |
| | 4 | 312 | 63 | 312 | 63 | 312 | 61 | 312 | 58 | 0.00% | 0.00% | 0.00% | -3.17% | 0.00% | -7.94% |
| | 5 | 306 | 79 | 306 | 98 | 306 | 93 | 306 | 79 | 0.00% | 19.39% | 0.00% | 17.72% | 0.00% | 0.00% |
| | 6 | 343 | 55 | 343 | 38 | 343 | 43 | 343 | 53 | 0.00% | -44.74% | 0.00% | -21.82% | 0.00% | -21.82% |
| | 7 | 299 | 74 | 299 | 80 | 299 | 79 | 299 | 69 | 0.00% | 7.50% | 0.00% | 6.76% | 0.00% | -6.76% |
| | 8 | 290 | 78 | 290 | 71 | 290 | 86 | 290 | 75 | 0.00% | -9.86% | 0.00% | 10.26% | 0.00% | -3.85% |
| | 9 | 355 | 75 | 355 | 74 | 355 | 72 | 355 | 66 | 0.00% | -1.35% | 0.00% | -4.00% | 0.00% | -12.00% |
| | 10 | 305 | 89 | 305 | 73 | 305 | 102 | 305 | 89 | 0.00% | -20.27% | 0.00% | 14.61% | 0.00% | -2.25% |
| ISG2 | 11 | 352 | 4225 | 352 | 3545 | 352 | 5152 | 352 | 3285 | 0.00% | -16.09% | 0.00% | 21.94% | 0.00% | -22.25% |
| | 12 | 355 | 3773 | 355 | 3999 | 355 | 4268 | 355 | 4037 | 0.00% | 5.99% | 0.00% | 13.12% | 0.00% | 7.00% |
| | 13 | 363 | 4189 | 363 | 3725 | 363 | 3920 | 363 | 3454 | 0.00% | -11.08% | 0.00% | -6.42% | 0.00% | -17.55% |
| | 14 | 347 | 4928 | 347 | 4941 | 347 | 5661 | 347 | 5444 | 0.00% | 0.26% | 0.00% | 14.87% | 0.00% | 10.47% |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 329 | 4342 | 329 | 3803 | 329 | 3549 | 329 | 3358 | 0.00% | -12.41% | 0.00% | -18.26% | 0.00% | -22.66% |
| 16 | 340 | 6687 | 340 | 6284 | 340 | 5658 | 340 | 6186 | 0.00% | -6.03% | 0.00% | -15.39% | 0.00% | -7.49% |
| 17 | 356 | 6189 | 356 | 5619 | 356 | 7911 | 356 | 4439 | 0.00% | -9.21% | 0.00% | 27.82% | 0.00% | -28.28% |
| 18 | 371 | 4357 | 371 | 4122 | 371 | 4085 | 371 | 4232 | 0.00% | -5.39% | 0.00% | -6.24% | 0.00% | -2.87% |
| 19 | 348 | 9875 | 348 | 8112 | 348 | 9607 | 348 | 8678 | 0.00% | -17.85% | 0.00% | -2.71% | 0.00% | -12.12% |
| 20 | 330 | 1256 | 330 | 1967 | 330 | 1257 | 330 | 363 | 0.00% | 56.61% | 0.00% | 0.08% | 0.00% | -71.10% |
| Average | | | | | | | | | 0.00% | -1.52% | 0.00% | 2.88% | 0.00% | -16.68% |

***Notes:*** (1) $F_0$, $F_1$, $F_2$, $F_3$ is solution of the method without cuts in RMP of CG before BB, with Constraints (5-49) in RMP of CG before BB, with Constraints (5-54) in RMP of CG before BB, with both Constraints (5-49) and Constraints (5-54) in RMP of CG before BB, respectively. And $t_0$, $t_1$, $t_2$, $t_3$ is computation time of the method without cuts in RMP of CG before BB, with Constraints (5-49) in RMP of CG before BB, with Constraints (5-54) in RMP of CG before BB, with both Constraints (5-49) and Constraints (5-54) in RMP of CG before BB, respectively. (2) $\Delta_{F1}= (F_1 - F_0)/F_0$, $\Delta_{F2}= (F_2 - F_0)/F_0$, $\Delta_{F3}= (F_3 - F_0)/F_0$, $\Delta_{t1}= (t_1 - t_0)/t_0$, $\Delta_{t2}= (t_2 - t_0)/t_0$, $\Delta_{t3}= (t_3 - t_0)/t_0$.

Table 3 shows that the average values of $\Delta_{F1}$, $\Delta_{F2}$, and $\Delta_{F3}$ are 0.00%, demonstrating that the optimal solution can be obtained by methods with different combinations of cuts. The average value of $\Delta_{t2}$ is 2.88%, indicating that the solution is less efficient when adding subset-row inequalities in the RMP of CG before BB than that without cuts. However, the average values of $\Delta_{t1}$ and $\Delta_{t3}$ are -1.52% and -16.68%, respectively. $\Delta_{t1}$ and $\Delta_{t3}$ are negative, and $|\Delta_{t3}|$ is greater than $|\Delta_{t1}|$, which indicates the effectiveness of adding round capacity inequalities and subset-row inequalities at the root node.

We then considered the addition of these Constraints during the enumeration. For the interest of space, the results are shown in Appendix 9. The results indicate that using cuts at the root node is more efficient than using cuts at all nodes, which suggests that Constraints (5-49) and Constraints (5-54) should be used at the root node only, and that the separation of cuts during the enumeration should be disabled. .

We also investigate the effectiveness of the dynamic programming algorithm and of the calculus based approximation for solving PP. For the interest of space, the results are shown in Appendix 10, and indicate that both the dynamic programming and the principle of calculus based on dynamic programming can improve the efficiency of the PP solution. We therefore used the combination of dynamic programming and calculus-based approximation in the PP solving process.

By summarizing the above experiments, we tuned a suitable combination of strategies for our proposed B&P&C algorithm by adding round capacity inequalities and subset-row inequalities to the RMP of CG before BB, by adding no cuts when branching RMP, and by using dynamic programming and the calculus-based LB updating method to solve PP. The above tuned B&P&C algorithm was then used in the following experiments and the sensitivity analysis.

### 6.3 Evaluating performance of the B&P&C algorithm

In this section, we conduct experiments to compare the solutions and computation time of solving the original MIP model using the CPLEX solver, the original MIP model using the proposed B&P&C algorithm, and the LBP model (see Section 4.3.1) using the CPLEX solver. The results of on group ISG1 instances are shown in Table 4. The table shows that the B&P&C algorithm obtain the optimal solution; but the computation time of the proposed B&P&C algorithm is longer than the CPLEX solver. Thus, CPLEX may be more proper for solving group ISG1 instances.

**Table 4** Algorithm performance for group ISG1 instances

| Instances Scale | ID | CPLEX $F_{CPLEX}$ | $t_{CPLEX}$(s) | $\Delta_{CPLEX}$ | LBP $F_{LBP}$ | $\Delta_{LBP}$ | B&P&C $F_{BPC}$ | $t_{BPC}$ | $\Delta_{BPC}$ | $\Delta_{tBPC}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| ISG1 | 1 | 324 | 20 | 0.00% | 252 | 22.22% | 324 | 70 | 0.00% | 250.00% |
| | 2 | 331 | 19 | 0.00% | 253 | 23.56% | 331 | 122 | 0.00% | 542.11% |
| | 3 | 332 | 28 | 0.00% | 238 | 28.31% | 332 | 73 | 0.00% | 160.71% |
| | 4 | 312 | 25 | 0.00% | 250 | 19.87% | 312 | 58 | 0.00% | 132.00% |
| | 5 | 306 | 19 | 0.00% | 229 | 25.16% | 306 | 79 | 0.00% | 315.79% |
| | 6 | 343 | 70 | 0.00% | 230 | 32.94% | 343 | 53 | 0.00% | -24.29% |
| | 7 | 299 | 112 | 0.00% | 253 | 15.38% | 299 | 69 | 0.00% | -38.39% |
| | 8 | 290 | 56 | 0.00% | 241 | 16.90% | 290 | 75 | 0.00% | 33.93% |
| | 9 | 355 | 60 | 0.00% | 257 | 27.61% | 355 | 66 | 0.00% | 10.00% |
| | 10 | 305 | 58 | 0.00% | 248 | 18.69% | 305 | 89 | 0.00% | 53.45% |
| Average | | | | | | 23.07% | | | 0.00% | 143.53% |

***Notes:*** (1) $F_{CPLEX}$, $F_{LBP}$ and $F_{BPC}$ denote the objective value of solving the original model by CPLEX, solving LBP model by CPLEX, and solving the original model B&P&C algorithm, respectively. And $t_{CPLEX}$ and $t_{BPC}$ denote the computation time of CPLEX and B&P&C algorithm to solve the original model, respectively. (3) $\Delta_{CPLEX}$ is the gap when the optimal solution is obtained, or the maximum solution time is reached during the solving process of CPLEX. (4) $\Delta_{LBP}$= $(F_{CPLEX} - F_{LBP})/F_{CPLEX}$. (5) $\Delta_{BPC}$= $(F_{UB} - F_{LB})/F_{LB}$, here $F_{LB}$ be the lower bound value and $F_{UB}$ be the upper bound value of B&P&C algorithm. (6) $\Delta_{t_{BPC}}$= $(t_{BPC} - t_{CPLEX})/t_{CPLEX}$.

The data column $\Delta_{LBP}$ in Table 4 also demonstrates the optimality gap of the LBP model, and indicates that the proposed lower bound (LB) is on average 23.07% below the optimal value. This result was used in further comparative experiments in group ISG2 instances. The results in Table 5 indicate that CPLEX cannot solve all of the instances to optimality within three hours. We can therefore simply use the LB to evaluate the solution quality of the B&P&C algorithm.

**Table 5** Algorithm performance for group ISG2 instances

| Instances Scale | ID | CPLEX $F_{CPLEX}$ | $t_{CPLEX}$(s) | $\Delta_{CPLEX}$ | LBP $F_{LBP}$ | $\Delta_{LBP}$ | B&P&C $F_{BPC}$ | $t_{BPC}$ | $\Delta_{BPC}$ | $\Delta_{tBPC}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| ISG2 | 1 | 352 | >10800 | 16.46% | 261 | 25.85% | 352 | 3285 | 0.00% | - |
| | 2 | 355 | >10800 | 14.37% | 273 | 23.10% | 355 | 4037 | 0.00% | - |
| | 3 | 363 | >10800 | 13.15% | 270 | 25.62% | 363 | 3454 | 0.00% | - |
| | 4 | 347 | >10800 | 12.64% | 257 | 25.94% | 347 | 5444 | 0.00% | - |
| | 5 | 329 | >10800 | 13.89% | 239 | 27.36% | 329 | 3358 | 0.00% | - |
| | 6 | 340 | >10800 | 18.34% | 264 | 22.35% | 340 | 6186 | 0.00% | - |
| | 7 | 356 | >10800 | 18.54% | 257 | 27.81% | 356 | 4439 | 0.00% | - |
| | 8 | 371 | >10800 | 20.91% | 247 | 33.42% | 371 | 4232 | 0.00% | - |
| | 9 | 348 | >10800 | 14.96% | 247 | 29.02% | 348 | 8678 | 0.00% | - |
| | 10 | 330 | >10800 | 10.77% | 260 | 21.21% | 330 | 363 | 0.00% | - |
| Average | | | | | | 26.17% | | | 0.00% | - |

***Notes:*** (1) $F_{CPLEX}$, $F_{LBP}$ and $F_{BPC}$ denote the objective value of solving the original model by CPLEX, solving LBP model by CPLEX, and solving the original model B&P&C algorithm, respectively. (2) $t_{CPLEX}$ and $t_{BPC}$ denote the computation time of CPLEX and B&P&C algorithm to solve the original model, respectively. (3) $\Delta_{CPLEX}$ is the gap when the optimal solution is obtained, or the maximum solution time is reached during the solving process of CPLEX. (4) $\Delta_{LBP}$= $(F_{CPLEX} - F_{LBP})/F_{CPLEX}$. (5) $\Delta_{BPC}$= $(F_{UB} - F_{LB})/F_{LB}$, here $F_{LB}$ be the lower bound value and $F_{UB}$ be the upper bound value of B&P&C algorithm. (6) $\Delta_{t_{BPC}}$= $(t_{BPC} - t_{CPLEX})/t_{CPLEX}$

CPLEX cannot obtain a result within 3 hours for group ISG2 instances, but the B&P&C algorithm can solve them in about 9,000 seconds. This validates the solution quality and the efficiency of the proposed B&P&C algorithm. The data column $\Delta_{LBP}$ in Table 5 indicates that the average gap for solutions provided by the B&P&C algorithm from the LBs is 26.17%. Recall that the average LB gap from the optimal results is about 23.07%, which is estimated from the experiments in the group ISG1 instances. These generally consistent results demonstrate the good performance of the B&P&C algorithm. Table 5 also indicates that the average value of

$\Delta_{BPC}$ is 0.00%, which validates the optimality of the solution provided by the B&P&C algorithm.

Besides the above numerical experiments for validating the efficiency of the algorithm, this study also presents various sensitivity analyses conducted on selected instances to provide managerial implications. Due to the limitation of space, the details are listed in Appendix 11.

## 7. Conclusions

This paper studies a variant of VRP for a novel cooperative delivery mode based on trucks and drones. An MIP model is formulated for the problem and a B&P&C based exact solution method is designed to solve the proposed model to optimality. Numerical experiments are also conducted to validate the effectiveness and efficiency of the proposed solution method. The main contributions of this study are outlined as follows.

(1) From modeling perspective, the proposed model considers many practical factors, such as one-on-one collaboration between trucks and drones, truck road network, the truck can serve multiple customers, the flying duration of a drone is affected by demands of customers, the non-customer nodes for drones' launching/returning/charging are needless, and etc. An MIP model is established for the complex and practical problem context.

(2) From algorithmic perspective, a B&P&C based exact solution method algorithm is designed. For accelerating the solving process, we embed some tailored tactics such as a dynamic programming with the calculus approximation based LB updating for solving the PP, round capacity inequalities and subset-row inequalities for solving the RMP, and some heuristics for updating the UB in the branch-and-bound procedure. The experimental results show that the PP solving strategy combining dynamic programming and calculus approximation can accelerate the solving speed by 97.69%.

The increasing interest in the various applications of drones offers several potential and challenging future research directions, in terms of risk evaluation, number of drone batteries for replacement, batteries' charging on trucks, and uncertainty in delivery (Macrina et al., 2020). In addition, more fast and practical algorithms may also be needed for this problem as well as some further extended problems, which may be more concerned by practitioners. All of the above could be the future research directions.

## Funding

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

## Notes on contributors

***Lu Zhen*** is a Professor and Dean at School of Management, Shanghai University, Shanghai, China. His research interests include operations management and optimization; mixed-integer linear programming and algorithms; port operations and maritime transportation; urban logistics and supply chain management. He has served as an associate editor or an editorial board member of five journals such as Transportation Research Part B, Journal of the Operational Research Society, Computers & Operations Research; and he is also the Fellow of the Operational Research Society (U.K.). He has been awarded the National Funds for Distinguished Young Scientists, for Outstanding Young Scientists in China, the Changjiang Young Scholar in China, and etc.

***Jiajing Gao*** is a Ph.D. student at School of Management, Shanghai University, Shanghai, China. Her research interests include vehicle routing problem's variants. She has published six papers on SCI international journals.

***Zheyi Tan*** is a Research Engineer at School of Management, Shanghai University, Shanghai, China. His research interests include warehouse operations management and optimization, vehicle routing problem's variants. He has published 17 papers on SCI international journals such as European Journal of Operational Research, INFORMS Journal on Applied Analytics.

***Shuaian Wang*** is a Professor at Faculty of Business, The Hong Kong Polytechnic University, Hong Kong, China. His research interests include big data in shipping, green shipping, shipping operations management, port planning and operations, urban transport network modeling, and logistics and supply chain management. He is an editor-in-chief of Cleaner Logistics and Supply Chain and Communications in Transportation Research, an associate editor of Transportation Research Part E, Flexible Services and Manufacturing Journal, Transportmetrica A, and Transportation Letters, a handle editor of Transportation Research Record, an editorial board editor of Transportation Research Part B, and an editorial board member of Maritime Transport Research.

***Roberto Baldacci*** is an Associate Professor at the College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar. His research interests include transportation planning, logistics and distribution, and the solution of vehicle routing and scheduling problems over street networks. He is on the editorial board of Operations Research, and he has published more than 50 papers in reputable journals such as Mathematical Programming, Operations Research, Transportation Science, and INFORMS Journal on Computing.

## References


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

Augerat, P., Belenguer, J. M., Benavent, E., Corberán, A., and Naddef, D. 1998. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research* 106(2–3), 546–557.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. and Vance, P. H. (1998), Branchand-price: Column generation for solving huge integer programs, *Operations Research* 46(3), 316–329.

Bouman, P., Agatz, N., and Schmidt, M. (2018) Dynamic programming approaches for the traveling salesman problem with drone. *Networks* 72(4), 528−542.

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

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

Desaulniers, G. (2010), Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows, *Operations Research* 58(1), 179–192.

El-Adle, A. M., Ghoniem, A., and Haouari, M. (2019) Parcel delivery by vehicle and drone. *Journal of the Operational Research Society* 72(2), 1−19.

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., and Rich, R. (2016) Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management* 9(2), 374−388.

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

Ham, A. M. (2018) Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C-Emerging Technologies* 91, 1−14.

Jeong, H. Y., Song, B. D., and Lee, S. (2019) Truck-drone hybrid delivery routing: payload-energy dependency and no-fly zones. *International Journal of Production Economics* 214, 220−233.

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008) Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* 56(2), 497−511.

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

Laporte, G. (1992) The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research* 59(3), 345−358.

Laporte, G. (2000) Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7(4-5), 285−300.

Li, Y., Fang, X., and Chen, W. (2019) A novel scheduling algorithm to improve SUPT for multi-queue multi-server system. *Wireless Networks* 25(8), 5173-5185.

Lübbecke, M. E. and Desrosiers, J. (2005), Selected topics in column generation, *Operations Research* 53(6), 1007–1023.

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

Mayerowitz S. (2013) Amazon.com Sees Delivery Drone as Future. https://phys.org/news/2013-12-amazon-unveils-futuristic-mini-drone-delivery.html.

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

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

Omagari, H., and Higashino, S.-I. (2018) Provisional-Ideal-Point-Based multi-objective optimization method for drone delivery problem. *International Journal of Aeronautical and Space Sciences* 19(1), 262−277.

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

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

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

Ralphs, T. K., Kopman, L., Pulleyblank, W. R., and Trotter, L. E. (2003). On the capacitated vehicle routing problem. *Mathematical Programming* 94(2-3), 343−359.

Roberti, R., Ruthmair, M. (2021). Exact Methods for the Traveling Salesman Problem with Drone. *Transportation Science* 55, 315–335.

Sacramento, D., Pisinger, D., and 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.

Sirdey, R., and Kerivin, H. L. M. (2007) A branch-and-cut algorithm for a resource-constrained scheduling problem. *RAIRO-Operations Research* 41(3), 235−251.

Toth, P., and Vigo, D. (2014) *Vehicle Routing: Problems, Methods, and Applications* (SIAM).

Wang, B., and Tian, R. (2019) Study on fluctuation feature and breakdown characteristic of water film on the wall of corrugated plate. *International Journal of Heat and Mass Transfer* 143, 118501.

Wang, D., Hu, P., Du, J., Zhou, P., Deng, T., and Hu, M. (2019) Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones. *IEEE Internet of Things Journal* 6(6), 10483−10495.

Wang, K., Yuan, B., Zhao, M., and Lu, Y. (2020) Cooperative route planning for the drone and truck in delivery services: a bi-objective optimisation approach. *Journal of the Operational Research Society* 71(10), 1657−1674.

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

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

Yi, W., and Sutrisna, M. (2021) Drone scheduling for construction site surveillance. *Computer-Aided Civil and Infrastructure Engineering* 36(1), 3−13.

Yurek, E. E., and Ozmutlu, H. C. (2018) A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C-Emerging Technologies* 91, 249−262.

Zhen, L., Wang, K., Wang, S., and Qu, X. (2018) Tug scheduling for hinterland barge transport: a branch-and-price approach. *European Journal of Operational Research* 265(1), 119−132.