Transportation Research Part B 122 (2019) 350–364

Contents lists available at ScienceDirect

Transportation Research Part B

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

Vehicle routing problem with drones

Zheng Wang[a], Jiuh-Biing Sheu[b,*]

[a] School of Maritime Economics and Management, Dalian Maritime University, No.1, Linghai Road, Dalian,116026, China
[b] Department of Business Administration, National Taiwan University, No.1, Section. 4, Roosevelt Road, Taipei 10617, Taiwan, ROC

ARTICLE INFO

Article history:
Received 26 August 2018
Revised 1 March 2019
Accepted 6 March 2019
Available online 13 March 2019

Keywords:
Vehicle routing problem with drones
Branch and price
Column generation

ABSTRACT

The vehicle routing problem with drones (VRPD) is an extension of the classic capacitated vehicle routing problem, where not only trucks but drones are used to deliver parcels to customers. One distinctive feature of the VRPD is that a drone may travel with a truck, take off from its stop to serve customers, and land at a service hub to travel with another truck as long as the flying range and loading capacity limitations are satisfied. Routing trucks and drones in an integrated manner makes the problem much more challenging and different from classical vehicle routing literature. We propose a mixed integer programming model, and develop a branch-and-price algorithm. Extensive experiments are conducted on the instances randomly generated in a practical setting, and the results demonstrate the good computational performance of the proposed algorithm. We also conduct sensitivity analysis on a key factor that may affect the total cost of a solution.

1. Introduction

Urban logistics service providers have always been seeking ways for faster and more cost-efficient last-mile deliveries. An emerging technology of the unmanned aerial vehicles, also known as drones, opens a new window of opportunity for practitioners in urban logistics. Compared with a regular delivery truck, a drone has attractive advantages such as avoiding the congestion of road networks, delivering faster, spending much lower transportation cost per kilometer (Wohlsen, 2014), and operating without a costly human pilot. Despite of the regulatory barriers preventing the widespread adoption of drone delivery, there are trends that countries are relaxing the regulation in response to commercial companies. Currently, half a dozen countries have allowed for deliveries either in designated zones or universally (Jones, 2017). Drones have been regarded as a promising choice for logistics industry applications (French, 2015).

On the other hand, since most drones on the market are powered by batteries, their loading capacity and flying range are much more limited than trucks. Compared with a drone, a truck can take heavier parcels, travel a longer distance, and carry drones closer to customers. Without trucks, drones cannot undertake all the possible deliveries themselves. Delivering with both trucks and drones then becomes an efficient way that can make the best use of their advantages (Wohlsen, 2014). As a result, several pioneer companies, like Amazon, Wal-Mart, Google (French, 2015), Alibaba (Wang, 2015), DHL (DHL, 2014), and SF Express (Shields, 2018) have run practical trials or even put it into large-scale applications. In their practices, several novel attempts can be seen (Oswald, 2017), such as establishing full-time hubs for drone delivery services, using a special device in a nicely spacious area to make a controlled landing, facilitating drones to deliver goods by a parachute airdrop,

* Corresponding author.
E-mail addresses: drwz@dlut.edu.cn (Z. Wang), jbsheu@ntu.edu.tw (J.-B. Sheu).

https://doi.org/10.1016/j.trb.2019.03.005
0191-2615/

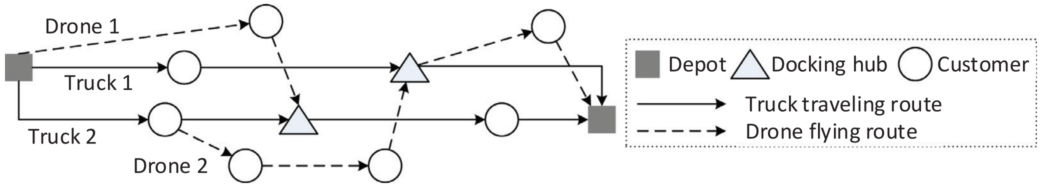

Fig. 1. An example of the VRPD.

and launching a drone from a stationary truck. These attempts outline the future scenarios of VRPD, in which a truck can not only serve customers itself but can carry a drone, launch it for making deliveries, and pick it up at a docking hub.

However, the participation of drones in last-mile delivery with trucks brings challenges to the operations of logistics companies. Fig. 1 shows a simple VRPD and its solution in which two trucks and two drones are employed for delivery. As illustrated, drone 2 is initially taken by truck 2 from the depot, then flies to serve two customers, lands at a docking hub for the subsequent service, and finally flies back to the depot.

Different from the classical vehicle routing problem (VRP), there are two types of vehicles in VRPD: a drone may have multiple times of flying and landing, each of which may be associated with a different truck; and a truck may launch and collect multiple drones at different times and locations. Such integrated and novel routing feature seldom appears in the VRP literature. It is difficult to determine the order of customer service fulfilled by two types of vehicles, the locations where trucks collect or launch drones, and the routes traveled by vehicles. All these decisions need to be made in solving VRPD. The problem becomes more challenging if spatial and capacity constraints, e.g. the flying range and remaining loading capacities of vehicles, are imposed.

Given a set of known customers, how much cost savings can be achieved by using drones for deliveries, and how to schedule trucks and drones together in order to achieve a cost-effective vehicle routing scheme, are important issues faced by every logistics company that plans to launch the drone delivery project. This paper aims at answering these questions for decision makers.

Our contributions are as follows. First, we present the VRPD in a real-world setting. To the best of our knowledge, we are the first to study this problem in urban logistics. Second, we construct an arc-based model for the problem. Third, we exploit the special problem structure, reformulate it as a path-based model and develop a branch-and-price algorithm that can find high-quality solution to the problem. Fourth, we conduct the extensive computational experiments and sensitivity analysis.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 formulates the problem. Section 4 presents an exact algorithm. Section 5 discusses the results of computational experiments. Section 6 concludes the paper and provides the future directions.

2. Literature review

VRPD belongs to the family of the classical VRP whose objective is to determine a set of optimal routes performed by vehicles with limited capacity to serve a given set of customers. A large number of models, exact algorithms, and heuristics have been developed for it (Toth and Vigo, 2014; Golden et al., 2008). Relative studies include the following VRP variants: (1) the VRP with trailers where trucks and trailers are routed to customers who may be served by either a truck or a truck-trailer pair but not a trailer itself (Derigs et al., 2013; Villegas et al., 2013); (2) the pickup and delivery problem in which each heavy resource (e.g., van) can transport a set of light resources (e.g., scooters or foot couriers) for serving customers (Lin, 2011); and (3) the two-echelon VRP in which freight is first transported from a central depot to satellite facilities by large vehicles, from where it is then brought to the final customers by small vehicles (Perboli et al., 2011; Hemmelmayr et al., 2012).

However, VRPD is different from the above VRP variants in that both truck and drone can serve customers independently and the two types of vehicles have the many-to-many relationship. These special problem aspects make the existing VRP models and algorithms difficult to be used for VRPD.

Compared with the classical VRP, the application of drones in last-mile delivery is a relatively new topic that has not been studied extensively. The studies on this topic can be grouped into two types according to the number of the trucks: the traveling salesman problem with drone (TSPD) in which a single truck is employed and the VRPD in which multiple trucks are employed.

Murray and Chu (2015) introduced the TSPD in which a drone collaborates with a truck to distribute customer parcels at the minimum completion time of the two vehicles. A mixed integer linear programming formulation and a heuristic that adopts a “Truck First, Drone Second” idea are proposed and tested on the instances with 10 customers. Ponza (2016) extended the work by proposing an enhancement to the MILP model and solving the TSPD by simulated annealing. Carlsson and Song (2017) generalized the TSPD in that two vehicles can meet anywhere, not only at depot and customers. Agatz et al. (2018) studied TSPD with the objective of minimizing the logistics cost. They constructed an integer programming model and developed route-first, cluster-second heuristics based on local search and dynamic programming.

Table 1
The publications related to the truck and drone routing problem.

Refs.	Problem	Approach	Drone capacity	Drone and truck	Objective	Solved instances
Murray and Chu (2015)	1-Truck; 1-Drone	Heuristic	1-Customer	1:1	Min returning time of vehicles	10 or 20 nodes
Wang et al. (2017)	M-Truck; N-Drone	Worst-case analysis	1- Customer	M:1	Same as above	–
Ponza (2016)	1-Truck; 1-Drone	Heuristic	1- Customer	1:1	Same as above	Up to 200 nodes
Carlsson and Song (2017)	1-Truck; 1-Drone	Analysis base on heuristics	1- Customer	1:1	Min traveling time	Up to 100 nodes
Agatz et al. (2018)	1-Truck; 1-Drone	Heuristic	1- Customer	1:1	Min logistics cost	10 nodes
Ha et al. (2018)	1-Truck; 1-Drone	Heuristic	1- Customer	1:1	Min operational cost	Up to 100 nodes
Ham (2018)	M-Truck; N-Drone	Heuristic	M- Customer	No rendezvous	Min maximum time	Up to 100 nodes
Chang and Lee (2018)	1-Truck; M-Drone	Heuristic	1- Customer	M:1	Min total time	Up to 100 nodes
This paper	M-Truck; N-Drone	Exact algorithm	M-Customer	M:N	Min logistics cost	15 nodes

Ha et al. (2018) also considered TSPD but with the objective of minimizing operational cost including transportation cost and the cost incurred by the waiting time of a vehicle for another. Chang and Lee (2018) studied the routing problem with a truck and several drones, in which the truck works as a carrier that carries drones to some centers where drones can fly for serving customers. Mathew et al. (2015) studied the heterogeneous delivery problem with a truck and a drone and multiple street vertices are defined for trucks to stop and deploy drones. Dayarian et al. (2018) considered a home delivery system in which a drone is used to resupply a delivery truck regularly. Mourelo Ferrandez et al. (2016) compared the truck-drone delivery system with standalone truck or drone systems.

Wang et al. (2017) and Poikonen et al. (2017) studied the VRPD in which a fleet of trucks, each equipped with a number of drones, delivers parcels to customers. They derived worst case bounds for the ratios of the total delivery times with or without drones. Different from our VRPD, they assumed that a drone must return to the truck that it launches from. The feature that a drone may be associated with multiple trucks is thus prohibited. Ham (2018) considered two different types of drone tasks: drop off and pickup. After a drone finishes a delivery, it can fly to a customer or depot for pickup. He considered multiple depots, multiple trucks, and multiple drones and developed a constraint programming approach. This work did not consider the rendezvous of two types of vehicles, making the problem different from ours. Campbell et al. (2017) provided a strategic analysis for the design of hybrid truck-drone delivery systems using continuous approximation modeling techniques to derive general insights.

The related literature and the work of this paper are concluded in Table 1, from which it can be observed that this paper differs from the literature in vehicle number, solution approach, drone capacity, mapping relationship between trucks and drones. Because of these differences, the existing methods cannot be applied directly.

3. Problem statement and arc-based model

In this section, we define the VRPD and propose an arc-based integer programming model.

3.1. Problem statement

The VRPD is defined in a graph $G=(N, A)$: N is the node set, containing the depot node, a set of customer nodes $C=\{c_1, c_2, \dots, c_n\}$, and a set of docking hub nodes $O=\{o_1, o_2, \dots, o_m\}$; and $A=\{(i, j)|i, j\in N, i \neq j\}$ is the arc set. For the notational convenience, let o^s and o^t represent the origin and termination depot respectively. A set of trucks K and a set of drones D are initially located at the depot and docking hub nodes, ready for serving customers.

A docking hub node serves as a transfer station for drones since it has turned out to be a preferable way in the drone delivery industry (Morgan, 2017). The station is generally designed for storing and maintaining some backup drones and supporting drones to land. In pilot experiments, a drone can fly from any nodes but can only land at a docking hub or depot, not a customer. This is because the landing of a drone usually needs special conditions, such as a spacious area and a special docking device at the space that can exchange information with the drone in order to guide it to land accurately and safely (Hern, 2014). These conditions cannot be easily realized at a customer site because of safety or privacy. Instead of landing at a customer, a drone in VRPD can do deliveries by a parachute airdrop (Mogg, 2017).

A drone can be loaded with up to L^D weight units of customer parcels. A truck can carry at maximum L^R drones through an arc and supply up to L^S drones at a hub with customer parcels and corresponding attachments so that the drones can fly from the hub for deliveries ($L^R > L^S$). Furthermore, a truck can also be loaded with at maximum L^T weight units of customer parcels ($L^T > L^D$). The maximum flying duration of a drone is T^D. Let g_i be the weight units of the parcel demanded by customer $i(\in C)$, which is less than L^T but may be larger than L^D indicating the customer cannot be served by a drone. A customer must be served once by a vehicle. The travel times of an arc for trucks and drones are t_{ij}^T and t_{ij}^D ($< t_{ij}^T$) respectively.

To serve customers, a vehicle needs to start from and return to the depot. If a customer is within the flying range of a drone, the drone can serve it independently. Otherwise, the drone has to be carried by a truck near to the customer before serving it, in which the drone's travel is dependent on the truck's.

It is worth mentioning that if a truck arrives at a docking node before a drone (d_1), to which it will supply customer parcels, the truck does not need to wait for d_1 but can supply the parcels to another backup drone (d_2) that is already there and equipped with a new battery. Thus drone d_2 replaces d_1 by completing the subsequent travel of d_1. We assume enough backup drones at the depot and docking nodes. This is actually not a strong assumption since it is a common practice in logistics companies to purchase sufficient vehicles, such as trucks and drones in this case. In addition, a drone is much cheaper than a truck (Wang, 2015). We neglect the time of swapping battery that can be done beforehand, and the time of loading a drone which is very fast and has not a significant impact on the total cost.

The objective of VRPD is to minimize the total logistics cost that consists of the fixed cost of employing trucks, and the transportation cost of trucks and drones. We define F^Tas the fixed cost for employing a truck, and define C^T and C^D as the transportation cost per unit of travel time for trucks and drones respectively.

3.2. Arc-based model

We present an arc-based model (*ARC-M*) for VRPD based on the following variables.

- x_{ijk}: Equals 1 if the kth truck travels arc $(i, j) \in A$ independently, and 0 otherwise.
- y_{ijd}: Equals 1 if the dth drone travels arc $(i, j) \in A$ independently, and 0 otherwise.
- z_{ijkd}: Equals 1 if the kth truck carries the dth drone through arc $(i, j) \in A$, and 0 otherwise.
- u_{ijk}: Equals 1 as long as the kth truck carries one or more drones through arc $(i, j) \in A$, and 0 otherwise.
- v_{id}: Be the cumulative flying time at node i for the dth drone after its last leave from the depot or a docking node.
- w_{id}^D: Be the cumulative weight units of customer parcels at node i that the dth drone has dropped after its last leave from the depot or a docking node.
- w_{ik}^T: Be the cumulative weight units of customer parcels at node i that the kth truck has delivered.

$$\min F^T\left(\sum_{(i,j)\in A:i=o^s}\sum_{k\in K}x_{ijk}+\sum_{(i,j)\in A:i=o^s}\sum_{k\in K}u_{ijk}\right) \tag{1}$$

$$+\,C^T\sum_{(i,j)\in A}\sum_{k\in K}t_{ij}^T\left(x_{ijk}+u_{ijk}\right)+\,C^D\sum_{(i,j)\in A}\sum_{d\in D}t_{ij}^D y_{ijd} \tag{2}$$

$$\text{s.t.}\quad \sum_{(i,j)\in A:i=o^s}\sum_{k\in K}x_{ijk}+\sum_{(i,j)\in A:i=o^s}\sum_{k\in K}u_{ijk}=\sum_{(i,j)\in A:j=o^t}\sum_{k\in K}x_{ijk}+\sum_{(i,j)\in A:j=o^t}\sum_{k\in K}u_{ijk} \tag{3}$$

$$\sum_{(i,j)\in A:i=o^s}\sum_{d\in D}y_{ijd}+\sum_{(i,j)\in A:i=o^s}\sum_{k\in K}\sum_{d\in D}z_{ijkd}=\sum_{(i,j)\in A:j=o^t}\sum_{d\in D}y_{ijd}+\sum_{(i,j)\in A:j=o^t}\sum_{k\in K}\sum_{d\in D}z_{ijkd} \tag{4}$$

$$\sum_{(i,j)\in A}\sum_{k\in K}x_{ijk}+\sum_{(i,j)\in A}\sum_{d\in D}y_{ijd}+\sum_{(i,j)\in A}\sum_{k\in K}u_{ijk}=1\ \ \forall j\in C \tag{5}$$

$$\sum_{(i,j)\in A}\sum_{k\in K}x_{ijk}+\sum_{(i,j)\in A}\sum_{d\in D}y_{ijd}+\sum_{(i,j)\in A}\sum_{k\in K}u_{ijk}=1\ \ \forall i\in C \tag{6}$$

$$\sum_{j\in N}x_{ijk}+\sum_{j\in N}u_{ijk}=\sum_{j\in N}x_{jik}+\sum_{j\in N}u_{jik}\qquad \forall k\in K, i\in O\cup C \tag{7}$$

$$\sum_{j\in N}y_{ijd}+\sum_{k\in K}\sum_{j\in N}z_{ijkd}=\sum_{j\in N}y_{jid}+\sum_{k\in K}\sum_{j\in N}z_{jikd}\quad \forall d\in D, i\in O\cup C \tag{8}$$

$$\sum_{d\in D}z_{ijkd}\le L^R\qquad \forall k\in K,\ (i,j)\in A \tag{9}$$

$$\sum_{(j,i)\in A}\sum_{d\in D}y_{jid}+\sum_{(j,i)\in A}\sum_{k\in K}\sum_{d\in D}z_{jikd}\le\sum_{(i,j)\in A}\sum_{k\in K}L^S\left(x_{ijk}+u_{ijk}\right)\quad \forall j\in O\cup C \tag{10}$$

$$v_{id}\le T^D\qquad \forall d\in D,\ i\in N \tag{11}$$

$$v_{jd} \geq v_{id} + t_{ij}^{D} + \left(y_{ijd} - 1\right)M \quad \forall d \in D, (i, j) \in A, i \in C \tag{12}$$

$$v_{jd} \geq t_{ij}^{D} + \left(y_{ijd} - 1\right)M \qquad \forall d \in D, (i, j) \in A, i \in O \cup \{o^s\} \tag{13}$$

$$w_{id}^{D} \leq L^{D} \qquad \forall d \in D, \; i \in C \tag{14}$$

$$w_{jd}^{D} \geq w_{id}^{D} + g_j + \left(y_{ijd} - 1\right)M \qquad \forall d \in D, (i, j) \in A, j \in C \tag{15}$$

$$w_{ik}^{T} \leq L^{T} \qquad \forall k \in K, \; i \in C \tag{16}$$

$$w_{jk}^{T} \geq w_{ik}^{T} + g_j + \left(x_{ijk} + u_{ijk} - 1\right)M \qquad \forall k \in K, (i, j) \in A, j \in C \tag{17}$$

$$w_{jk}^{T} \geq w_{ik}^{T} + \left(x_{ijk} + u_{ijk} - 1\right)M \qquad \forall k \in K, (i, j) \in A, j \in O \tag{18}$$

$$1 + \left(u_{ijk} - 1\right)M \leq \sum_{d \in D} z_{ijkd} \leq u_{ijk}M \qquad \forall k \in K, (i, j) \in A \tag{19}$$

$$x_{ijk} + u_{ijk} \leq 1 \quad \forall k \in K, (i, j) \in A \tag{20}$$

$$y_{ijd} + \sum_{k \in K} z_{ijkd} \leq 1 \quad \forall d \in D, (i, j) \in A \tag{21}$$

$$\sum_{(n,i) \in A} y_{nid} \leq \left(1 - z_{ijkd}\right)M \quad \forall (i, j) \in A, i \in C, k \in K, d \in D \tag{22}$$

$$x_{ijk}, \; y_{ijd}, \; z_{ijkd}, \; u_{ijk} \in \{0, 1\}, \; v_{id}, \; w_{id}^{D}, \; w_{ik}^{T} \geq 0 \tag{23}$$

In the objective function, term (1) denotes the fixed cost of trucks and (2) denotes the transportation cost. Constraint (3) and (4) ensure respectively the trucks and drones that leave depot must return to depot. Constraints (5) and (6) ensure both the in-degree and out-degree of a customer node are 1. Constraints (7) and (8) ensure the equal numbers of the incoming and outgoing drones of each station. The conservation of drone numbers at each station is introduced because there are limited resources at each station, e.g. staffs and docks, to conduct drone maintenance. It is worthy of mentioning that we do not force exactly same drones stationed at each site. Therefore, one drone path can be covered by different drones. Constraints (9) and (10) force that a truck cannot be loaded with more drones than restricted. Constraints (11)-(13) guarantee the feasibility of the drone flying duration. Constraints (14) and (15) satisfy the drone capacity feasibility. Constraints (16)-(18) guarantee the truck capacity feasibility. Constraints (19)-(21) define the relationships of binary variables. Constraint (22) prevents the landing of a drone at a customer node.

It is easy to show that the model is NP-hard: if there is no drones available, the problem turns out to be a classical capacitated VRP, which has proved to be NP-hard; thus the model is also NP-hard.

4. Path-based model and branch-and-price algorithm

In this section, we propose a path-based model and develop a branch-and-price algorithm.

4.1. Path-based model

In VRPD, the dependent and independent travels of vehicles and the special requirements result in a large number of constraints ((7)-(15) and (19)-(22)). These constraints can be effectively handled by reformulation.

We next reformulate the VRPD based on all the feasible paths traveled by trucks or drones. The model aims at finding the optimal path subset that satisfies some constraints. To reformulate it, possible nodes and arcs in a feasible path are first defined in the following. As illustrated in Fig. 2, there are four types of nodes (depot, docking node, truck node, and drone node) and two types of arcs (truck arc and drone arc) in a feasible path. A truck node is a customer node served by a truck that may carry drones. A drone node is a customer node served by a drone itself. A truck arc is an arc traveled by a truck that may carry drones. A drone arc is an arc traveled by a drone itself. A feasible truck path may contain depot, docking nodes, truck nodes, and truck arcs. A feasible drone path may contain all types of nodes and arcs.

Let R^T and R^D be the sets of all feasible truck paths and drone paths respectively, each of which is from o^s to o^t and satisfies the capacity and duration constraints of a vehicle. Let c_r be the transportation cost of path $r\,(\in R^T \cup R^D)$. Let δ_i^r be

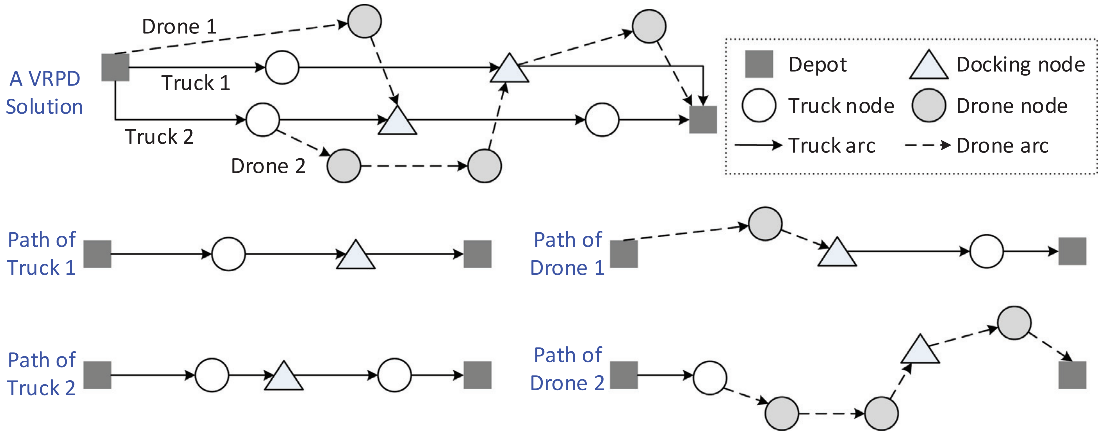

Fig. 2. Illustration of splitting a solution into truck paths and drone paths.

1 if $r\,(\in R^T)$ visits truck node i or $r\,(\in R^D)$ visits drone node i, and be 0 otherwise. Let θ_i^r be 1 if $r\,(\in R^T \cup R^D)$ visits docking or truck node i, and be 0 otherwise. Let ϑ_a^r be 1 if $r\,(\in R^T)$ travels arc a, and be 0 otherwise. Let φ_a^r be 1 if $r\,(\in R^D)$ travels truck arc a, and be 0 otherwise. Let A^O be the set of truck arcs whose endpoints are not drone nodes. We define two binary variables $x_r\,(r \in R^T)$ and $y_r\,(r \in R^D)$, representing if path r is selected in the final solution.

The difficulty of constructing a path-based model lies in ensuring the feasibility of drone paths. A drone path will be infeasible if there are no truck paths matching its truck arcs and nodes. The following lemmas and theorem are concluded for modeling.

Lemma 1. *$\forall\, r\,(\in R^D)$, $a\,(\in A^O)$ that has $y_r = 1$ and $\varphi_a^r = 1$, $\exists\, r'\,(\in R^T)$, $x_{r'} = 1$ and $\vartheta_a^{r'} = 1$. It indicates any truck arc in a drone path of the final solution must be traveled by a truck path of the solution, otherwise no truck transports the drone through the arc.*

Lemma 2. *$\forall\, r\,(\in R^D)$, $i\,(\in r)$ that has $y_r = 1$ and $\theta_i^r = 1$, $\exists\, r'\,(\in R^T)$, $x_{r'} = 1$ and $\theta_i^{r'} = 1$. It indicates any docking and truck node in a drone path of the final solution must be traveled by a truck path of the solution, otherwise no truck supplies customer parcels to the drone at the node.*

Theorem 1. *If Lemma 1 is satisfied and Lemma 2 is satisfied for any docking node, the feasibility of drone paths can be ensured.*

Proof. There are two situations that a drone path depends on a truck's: (1) a drone is transported by a truck through an arc, in which the two vehicles visit both the arc and its endpoints; and (2) a drone lands at a node, picks up customer parcels, and flies for deliveries, in which the two vehicles visit only the node. Lemma 1 ensures the feasibility of situation (1). And situation (2) can be ensured by that Lemma 2 is satisfied for any docking node because the situation can only happen at a docking node where a drone can land.

Based on the theorem, we construct a path-based model (*PATH-M*) for VRPD, where the feasibility of drone paths is ensured by constraints (26) and (27).

$$\min \sum_{r \in R^T} (c_r + F^T) x_r + \sum_{r \in R^D} c_r \cdot y_r \tag{24}$$

$$s.t. \sum_{r \in R^T} \delta_i^r x_r + \sum_{r \in R^D} \delta_i^r y_r \geq 1 \quad \forall i \in C \tag{25}$$

$$\sum_{r \in R^T} L^S \theta_i^r x_r \geq \sum_{r \in R^D} \theta_i^r y_r \qquad \forall i \in O \tag{26}$$

$$\sum_{r \in R^T} L^R \vartheta_a^r x_r \geq \sum_{r \in R^D} \varphi_a^r y_r \quad \forall a \in A^O \tag{27}$$

$$\sum_{r \in R^T} x_r \leq |K| \tag{28}$$

$$x_r,\ y_{r'} \in \{0, 1\}\ \forall r \in R^T, r' \in R^D \tag{29}$$

Objective (24) is to minimize the total logistics cost. Constraint (25) ensures each customer is served once. Constraints (26) and (27) force that a truck cannot be associated with more drones than restricted at a node or along an arc. Constraint (28) restricts the number of employed trucks.

Table 2
Reduced costs of different drone arcs.

No.	Starting node (i)	Ending node (j)	Arc type	Reduced cost of arc type ($\bar{c}^D_{ij}$)	No.	Starting node (i)	Ending node (j)	Arc type	Reduced cost of arc type ($\bar{c}^D_{ij}$)
1	Depot	Drone	Drone Arc	$c^D_{ij}-\pi_j$	9	Drone	Drone	Drone Arc	$c^D_{ij}-\pi_j$
2	Depot	Truck	Truck Arc	ρ_{ij}	10	Drone	Truck	–	–
3	Depot	Docking	Truck Arc	$\mu_j+\rho_{ij}$	11	Drone	Docking	Drone Arc	$c^D_{ij}+\mu_j$
4	Depot	Depot	–	–	12	Drone	Depot	Drone Arc	$F^D+c^D_{ij}$
5	Truck	Drone	Drone Arc	$c^D_{ij}-\pi_j$	13	Docking	Drone	Drone Arc	$c^D_{ij}-\pi_j$
6	Truck	Truck	Truck Arc	ρ_{ij}	14	Docking	Truck	Truck Arc	ρ_{ij}
7	Truck	Docking	Truck Arc	$\mu_j+\rho_{ij}$	15	Docking	Docking	Truck Arc	$\mu_j+\rho_{ij}$
8	Truck	Depot	Truck Arc	$F^D+\rho_{ij}$	16	Docking	Depot	Truck Arc	$F^D+\rho_{ij}$

4.2. Branch-and-price algorithm

To solve the path-based model, we develop a branch-and-price algorithm (e.g. see Barnhart et al., 1998; Lübbecke and Desrosiers, 2005; Guedes and Borenstein, 2018). The framework is detailed in the following.

Step 1: Solve the linear relaxation of a restricted master problem, defined as (24)-(29), with a subset of columns in an initial solution, generated by the procedure detailed below;

Step 2: Solve a pricing sub-problem to generate columns with negative reduced cost, in which a stabilization strategy is applied; go to step 1 if the columns are found, otherwise go to step 3;

Step 3: If an integral solution is obtained, return it; otherwise, use the branch-and-bound framework to explore every node of the search tree by repeating the above steps.

The branch-and-price algorithm starts with an initial solution that is obtained by the following procedure. The customer set is first split into two subsets: one contains the customers that can be reached by the drones from depot; the other contains the other customers. Drones and trucks are used to serve the customers in the two subsets respectively. Then each subset is associated with a routing problem with only one type of vehicle. We obtain a solution for each subset by the classical saving heuristic and combine the solutions of the two subsets into an initial solution.

4.2.1. Pricing

Let dual variables π, μ, ρ and σ correspond to constraints (25), (26), (27) and (28) respectively. The reduced costs of a truck path ($\bar{c}^T_r$) and a drone path ($\bar{c}^D_r$) are formulated by (30) and (31) respectively.

$$\bar{c}^T_r = c_r + F^T - \sum_{i\in C}\pi_i\delta^r_i - \sum_{i\in O}\mu_i L^S\theta^r_i - \sum_{a\in A}\rho_a L^R\vartheta^r_a + \sigma \ \ \forall r\in R^T \tag{30}$$

$$\bar{c}^D_r = c_r - \sum_{i\in C}\pi_i\delta^r_i + \sum_{i\in C}\mu_i\theta^r_i + \sum_{a\in A}\rho_a\varphi^r_a \ \forall r\in R^D \tag{31}$$

The reduced cost of an arc (i, j) in a truck path, $\bar{c}^T_{ij}$, can be formulated by (32).

$$\bar{c}^T_{ij} = \begin{cases} c^T_{ij} - \pi_j - \rho_{ij}L^R, & j\in C \\ c^T_{ij} - \mu_j L^S - \rho_{ij}L^R, & j\in O \\ c^T_{ij} + F^T - \rho_{ij}L^R + \sigma, & j = o^t \end{cases} \tag{32}$$

The reduced cost of an arc (i, j) in a drone path, $\bar{c}^D_{ij}$, depends on the types of the nodes i and j, which can be depot, truck node, drone node, or docking node, as illustrated in Fig. 2. The reduced costs of the drone arcs with different starting and ending nodes are formulated in Table 2.

The pricing sub-problem consists of finding a truck or drone path from o^s to o^t with the minimum reduced cost. We design a network from o^s to o^t. Each customer node has three copies in the network, representing truck node in a truck path, and truck node and drone node in a drone path respectively. Each docking node has two copies in the network for truck and drone paths respectively. In the network, the nodes of truck path do not connect to those of drone path, the copied nodes of a customer are not connected since a customer cannot be visited twice, and a drone node does not connect to a truck node since a drone cannot land at a customer.

Let N^T and N^D be the node sets of truck path and drone path in the designed network respectively. Let N^R be the set of drone nodes in N^D. Let A^N be the arc set of the designed network. Let a variable p_{ij} ($(i,j)\in A^N$) be 1 if node i is connected to node j in the solution, and be 0 otherwise. For a drone path, let the variable v_i be the cumulative flying time at node i after its last leave from depot or docking node; let w^D_i be the cumulative weight units of customer parcels at node i that a drone has dropped after its last leave from a depot or docking node. For a truck path, let w^T_i be the cumulative weight units of customer parcels at node i that a truck has delivered. The pricing sub-problem can be formulated as follows.

$$\min \sum_{(i,j)\in A^N} \bar{c}^D_{ij} p_{ij} \tag{33}$$

$$\text{s.t.} \sum_{(i,j)\in A^N: i=o^s} p_{ij} = 1 \tag{34}$$

$$\sum_{(i,j)\in A^N: j=o^t} p_{ij} = 1 \tag{35}$$

$$\sum_{(i,j)\in A^N} p_{ij} = \sum_{(j,k)\in A^N} p_{jk} \ \forall j \in N^T \cup N^D \backslash \left\{o^s, o^t\right\} \tag{36}$$

$$v_i \le T^D \qquad \forall\ i \in N^D \tag{37}$$

$$v_j \ge v_i + t_{ij}^D + \left(p_{ij} - 1\right)M \qquad \forall (i, j) \in A^N, i \in N^R \tag{38}$$

$$v_j \ge t_{ij}^D + \left(p_{ij} - 1\right)M \quad \forall (i, j) \in A^N, i \notin N^R, j \in N^R \tag{39}$$

$$w_i^D \le L^D \qquad \forall i \in N^D \tag{40}$$

$$w_j^D \ge w_i^D + g_j + \left(p_{ij} - 1\right)M \qquad \forall (i, j) \in A^N, j \in N^R \tag{41}$$

$$w_i^T \le L^T \qquad \forall i \in N^T \tag{42}$$

$$w_j^T \ge w_i^T + g_j + \left(p_{ij} - 1\right)M \qquad \forall (i, j) \in A^N, i, j \in N^T \tag{43}$$

$$p_{ij} \in \{0, 1\},\ v_i,\ w_i^D,\ w_i^T \ge 0 \tag{44}$$

It can be shown that a special case of the pricing sub-problem is the constrained shortest-path problem, which is NP-hard (Garey and Johnson, 1979). Therefore, the pricing sub-problem is also NP-hard.

As a dynamic programming algorithm, label correcting has been widely used to solve the constrained shortest-path problem (e.g., see Feillet et al., 2004; Righini and Salani, 2006). Recently, Lozano and Medaglia (2013) proposed a pulse algorithm that was proved to outperform the label correcting by using a depth-first search strategy and pruning strategies. This algorithm was extended through a novel bounding scheme that narrowed the search space (Lozano et al., 2015). To handle the pricing sub-problem, we present an improved pulse algorithm using the following pruning and extending strategies.

(1) Pruning by infeasibility. Whenever a partial path reaches a node, the algorithm checks if visiting the node creates a cycle or violates the constraints of vehicle capacity and drone flying duration. If any of these events happen, the partial path is infeasible and thus can be safely pruned.
(2) Pruning by dominance. We define a label (i, N_i,c_i,w_i) for a node i in a truck path, where N_i is the set of the nodes in the path from o^s to i, and c_i and w_i are the reduced cost and the accumulative weight units of customer parcels of the path respectively. We define another label (i, N_i,c_i,w_i,v_i) for a node i in a drone path, where v_i is the accumulative flying time since the last non-drone node of the path. Consider two labels (i, N_i,c_i,w_i,[v_i]) and (i, $N_i{}'$, c_i', w_i', [v_i']) with the same last reached node i, the former label dominates the later if $N_i \subseteq N_i{}'$, $c_i \le c_i{}'$, $w_i \le w_i{}'$, and $v_i \le v_i{}'$.
(3) A greedy extending strategy. When extending a node i to its "child" nodes that can be connected to i, we need to decide their extending order. During the depth-first search, different extending order determines the order of the partial paths we obtain. Generally, a good partial path is expected to be obtained as early as possible so that it can be used to dominate the other non-optimal paths. Thus we first extend the termination depot o^t and then extend the other child nodes in an ascending order of the arc reduced cost. Our preliminary experiments show that this strategy is helpful for accelerating the whole searching process.

4.2.2. Stabilization and lower bounding

A major difficulty in column generation may be slow convergence when the solution is near the optimum, which is called the tailing-off effect. A widely used approach that mitigates the tailing-off effect is through computing alternative Lagrangian lower bound and applying early termination Wentges, 1997). By associating dual variables $\bar{\pi}$, $\bar{\mu}$ and $\bar{\rho}$ with constraints ((25)-(27) respectively, the Lagrangian dual problem can be formulated as follows.

$$\max_{\bar{\pi},\ \bar{\mu},\bar{\rho}} \min \left\{ \begin{array}{l} \sum_{r\in R^T} \left((c_r + F^T - \sum_{i\in C} \bar{\pi}_i \delta_i^r - \sum_{j\in O} \bar{\mu}_j L^S \theta_j^r - \sum_{a\in A} \bar{\rho}_a L^R \vartheta_a^r \right) x_r \\ + \sum_{r\in R^D} \left(c_r - \sum_{i\in C} \bar{\pi}_i \delta_i^r + \sum_{j\in O} \bar{\mu}_j \theta_j^r + \sum_{a\in A} \bar{\rho}_a \varphi_a^r \right) y_r + \sum_{i\in C} \bar{\pi}_i \end{array} \right\} \tag{45}$$

$$\text{s.t.} \sum_{r\in R^T} x_r \le |K| \tag{46}$$

$$x_r,\ y_{r'} \in \{0, 1\}\ \forall r \in R^T, r' \in R^D \tag{47}$$

If the optimal value of the pricing sub-problem $\bar{c}_r^*$ is negative, the lower bound on the linear relaxation of the path-based model is $\text{Obj}_{\text{RMP}} + |K|\bar{c}_r^*$ $(r \in R^T)$, where Obj_{RMP} is the optimal value of the current restricted master problem (e.g., see Farley, 1990; Dell'Amico et al., 2006). If this lower bound is greater than or equal to the best incumbent integer solution, the current node is fathomed.

Another important method for mitigating the tailing-off effect is column generation stabilization. Notable approaches include the primal-dual strategy (du Merle et al. 1999), the interior point method (Rousseau et al., 2007), and the weighted Dantzig-Wolfe decomposition (Wentges, 1997; Li, 2014) which has been successfully used in facility location (Klose and Drexl, 2005; Klose and Görtz, 2007), machine scheduling (Pessoa et al., 2010), production order batching (Tang et al., 2011), and transit bus scheduling (Li, 2014). We use the weighted Dantzig-Wolfe decomposition to combine the dual solution from the restricted master problem and Lagrangian multipliers from subgradient search. The overall stabilization approach is outlined as follows.

Step 1: Run a given number of iterations on the Lagrangian dual problem by solving the pricing sub-problem and updating the Lagrangian dual by sub-gradient search. Let $\bar{\pi}^*$, $\bar{\mu}^*$ and $\bar{\rho}^*$ correspond to the dual with the best Lagrangian lower bound.

Step 2: Solve the restricted master problem in the path-based model and let $\hat{\pi}$, $\hat{\mu}$ and $\hat{\rho}$ correspond to the optimal dual solutions.

Step 3: Let $\pi = \gamma\hat{\pi} + (1-\gamma)\bar{\pi}^*$, $\mu = \gamma\hat{\mu} + (1-\gamma)\bar{\mu}^*$ and $\rho = \gamma\hat{\rho} + (1-\gamma)\bar{\rho}^*$, where γ is between 0 and 1 (0.6 in this paper). If some columns generated have negative costs at dual prices π, μ and ρ, update $\bar{\pi}^*$, $\bar{\mu}^*$ and $\bar{\rho}^*$ by solving the pricing sub-problem and updating the Lagrangian dual by sub-gradient search from π, μ and ρ.

Step 4: Increase γ by a small amount (0.1 in this paper); go to step 3 until γ approaches 1.

Step 5: Return $\bar{\pi}^*$, $\bar{\mu}^*$ and $\bar{\rho}^*$, which are used to replace $\hat{\pi}$, $\hat{\mu}$ and $\hat{\rho}$ in the pricing of the restricted master problem.

4.2.3. Branching

To handle the non-integral solution of the restricted master problem, we use a constraint-based branching rule proposed by Ryan and Foster (1981). The rule first converts the solution of the path-based model into an equivalent solution of the arc-based model, then identifies fractional arc flows, and finally branches on one of them. The underlying network is changed according to if the fractional arc is visited by a truck or drone. Using such a branching scheme, the pricing sub-problem structure is preserved after branching. To obtain a feasible solution quickly, we apply the depth-first node selection strategy that prefers to extend the arc flow with a value close to 1.

5. Computational experiments

In this section, we evaluate the performance of the branch-and-price algorithm by using randomly generated instances and conduct sensitivity analysis under different parameter settings. We implement the proposed algorithm in C# on a computer with Intel I7 2.69 GHz processor and 16GB RAM. Gurobi 8.0.0 is chosen as the MIP solver.

5.1. Instance generation and parameter estimation

We randomly generate some locations distributed in a 15 square kilometer area. The distances for a truck and a drone between any pair of locations equal to the Manhattan distance and Euclidean distance respectively. Both the two types of vehicles are assumed to travel at a speed of 40 km/h.

We generate four types of instances. In the first two types, the x and y coordinates of customer and docking nodes are generated uniformly from {0, 0.5, 1, 1.5, ..., 15}. The depot is located by the same rule for the first-type instances and at the location averaging the x and y coordinates of customers for the second. In the third-type instances, the x and y coordinates of customer and docking nodes are generated in a clustering manner that equal to $r\sin\alpha$ and $r\cos\alpha$, where r is a distance from a normal distribution with mean 0 and standard deviation 10 and α is an angle from $[0, 2\pi]$ uniformly. And the depot is located at the origin of the axis. In the fourth-type instances, we locate customers, docking nodes, and depot in a USA city by an electronic map and obtain the travel time of a pair of locations from the map. The Euclidean distance of a pair of locations is obtained based on their longitudes and latitudes. We generate 6 instances for the first three types and 2 instances for the fourth type and thus totally 20 instances are obtained. Among these instances, 11 have 8 customers and 2 docking nodes and 9 have 13 customers and 2 docking nodes.

Table 3
Results of the arc-based model and the branch-and-price algorithm.

Ins	Type	$\lvert C\rvert$, $\lvert O\rvert$	ARC-M by Gurobi		Branch-and-Price		Ins	Type	$\lvert C\rvert$, $\lvert O\rvert$	ARC-M by Gurobi		Branch-and-Price	
			Obj	CPU	Obj	CPU				Obj	CPU	Obj	CPU
1	1	8, 2	(49.452)	3600	47.277	306	11	2	13, 2	(65.757)	18,000	60.836	18,304
2	1	8, 2	(50.804)	3600	49.376	358	12	2	13, 2	(63.005)	18,000	56.074	16,013
3	1	8, 2	(48.267)	3600	46.247	326	13	3	8, 2	(51.547)	3600	48.559	410
4	1	13, 2	(64.152)	18,000	60.56	16,910	14	3	8, 2	(50.243)	3600	48.471	535
5	1	13, 2	(63.613)	18,000	58.007	15,692	15	3	8, 2	(55.834)	3600	52.356	477
6	1	13, 2	(59.625)	18,000	54.925	17,804	16	3	13, 2	(62.56)	18,000	59.826	18,427
7	2	8, 2	(47.982)	3600	45.028	459	17	3	13, 2	(66.064)	18,000	62.24	17,060
8	2	8, 2	(49.378)	3600	46.468	435	18	3	13, 2	(61.105)	18,000	57.063	14,526
9	2	8, 2	49.236	2839	49.236	432	19	4	8, 2	(52.239)	3600	49.631	336
10	2	13, 2	(61.436)	18,000	57.122	16,116	20	4	8, 2	(55.348)	3600	52.398	422

*The objective value in a parenthesis is obtained after 1 or 5 h' running of Gurobi.

Customer demands are randomly generated from {10, 20, 30, 40, 50} weight units. The parameters of L^D, L^R, L^S, and L^T are set to 20, 2, 5, and 100 respectively. The maximum flying duration of a drone T^D is set to 30 min and F^T is set to \$20. The parameter of C^T is set to \$0.083 per min because fuel consumption is about 0.15 liter/km for a truck and oil cost is about \$0.83 per liter (Global Petrol Prices, 2018). The parameter of C^D is set to \$0.021 per min because a drone generally costs \$0.05 per mile (Kim, 2016).

5.2. Computational results

In this section, we compare the branch-and-price algorithm with the arc-based model (ARC-M) solved by Gurobi 8.0.0. The results are reported in Table 3, where the columns provide the instance, type, numbers of customer and docking nodes ($|C|$, $|O|$), the objective values (Obj) and running times (CPU) in seconds by Gurobi and the branch-and-price algorithm respectively.

It can be observed from Table 3 that the branch-and-price algorithm can obtain the optimal solution by using much less time for all instances than MIP solver of Gurobi. Because of the problem aspect that trucks and drones can be routed in various ways to serve all the customers, there is a huge number of the feasible paths for vehicles, which cannot be generated in a limited time and space. However, the branch-and-price algorithm can solve it to optimality in an average of 7 min for 10-node instances and 4 h and a half for the 15-node instances. On the other hand, among 20 instances, Gurobi only finds the optimal solution to instance 9 in an hour. For the other instances, Gurobi generates best-found solutions, which have an average gap of 6.1% with the optimal solutions. The running times of the branch-and-price algorithm also show that the scale of the model increases exponentially with the problem size, and a small increase in size quickly leads to much more time and memory consuming. Compared with a 10-node instance, solving a 15-node instance takes 4 h more.

To provide more insights into the solutions obtained by the branch-and-price algorithm, we show the details of the solutions in Table 4. The table also shows the VRP solutions with no drones in order to facilitate the observation of the benefit of using drones for delivery. The VRP solutions are generated by setting an empty set of available drones and docking nodes to customer nodes with the demand 10. The first two columns in Table 4 provide the instance and type respectively. Columns 3–8 show, for the VRPD solutions, the number of drones employed (*Drones*), the number of customers served by drones (*D-Cust*), the total distance traveled by drones (*D-Dist*), the transportation cost per customer by drones (*D-Cost/Cust*), the number of trucks employed (*Trucks*), and the total distance traveled by trucks (*T-Dist*) respectively. Columns 9–11 show, for the VRP solutions, the number of trucks employed (*Trucks*), the distance traveled by trucks (*T-Dist*), and the transportation cost per customer by trucks (*T-Cost/Cust*) respectively. Column 12 is calculated by $\frac{C_2-C_1}{C_1}\times 100\%$ where C_1 and C_2 represent the total logistics costs of the VRPD and VRP solutions respectively. The last column is calculated by $(t_2 - t_1)$ where t_1 and t_2 represent the average delivery time of customers in VRPD and VRP solutions respectively. Each column is averaged over an instance type and all the instances.

We can obtain several observations from Table 4. First, the VRPD solutions employ at most 2 drones for delivery. These drones travel an average distance of 47.4 km to serve about 5 customers. The average transportation costs of serving a customer by a VRPD drone and a VRP truck are 0.316 and 1.933 respectively, indicating a truck costs over six times that of a drone for serving a customer. The big difference well demonstrates the effectiveness of using drones for delivery.

Second, the four types of instances show different numbers of customers served by drones. In the first two types of instances, drones serve over 5 customers averagely; while in the third type, the number of customers served by drones is obviously less. It indicates trucks are easier to find good delivery paths in a clustering distribution, where customers are relatively close to each other, than in a uniform distribution.

Third, the second last column shows that the average saving in the total cost of the VRPD solution is over 20%. We can find that some customers served by trucks in a VRP solution are served by drones in the corresponding VRPD solution. The cost incurred by using drones to serve the customers is much less than the cost incurred by using trucks to serve them. Thus, the 20% cost saving mainly results from the delivery of drones. Furthermore, the last column shows the average

Table 4
Details of the solutions from the branch-and-price algorithm.

Ins	Type	VRPD solution						VRP solution				Avg. delivery time gap (min)
		Drones	D-Cust	D-Dist	D-Cost/Cust	Trucks	T-Dist	Trucks	T-Dist	T-Cost/ Cust	Cost%	
1	1	1	4	36.698	0.289	2	49.165	2	135.225	1.684	20.22%	4.606
2	1	2	4	41.270	0.325	2	64.867	2	160.502	1.998	21.48%	6.562
3	1	1	4	38.476	0.303	2	40.442	2	145.293	1.809	25.61%	7.385
4	1	2	6	53.524	0.281	2	151.598	2	241.383	2.003	15.67%	3.376
5	1	2	6	55.238	0.290	2	130.659	2	234.272	1.944	19.24%	5.536
6	1	2	7	75.333	0.339	2	100.819	2	218.636	1.815	22.39%	6.608
Avg. of type 1	1.67	5.17	50.090	0.305	2	89.592	2	189.219	1.876	20.77%	5.679	
7	2	2	4	36.952	0.291	2	31.036	2	120.625	1.502	22.19%	4.454
8	2	1	5	52.222	0.329	2	38.739	2	133.057	1.657	21.73%	6.509
9	2	2	4	57.270	0.451	2	59.695	2	148.140	1.844	18.70%	5.950
10	2	2	6	76.571	0.402	2	118.153	2	221.837	1.841	18.38%	5.194
11	2	2	6	59.238	0.311	2	152.369	2	230.755	1.915	12.97%	5.221
12	2	2	6	63.619	0.334	2	113.012	2	218.951	1.817	19.95%	6.204
Avg. of type 2	1.83	5.17	57.646	0.353	2	85.501	2	178.894	1.763	18.99%	5.589	
13	3	2	4	38.222	0.301	2	59.076	2	156.817	1.952	22.58%	3.035
14	3	2	4	40.508	0.319	2	57.791	2	146.393	1.823	20.13%	5.403
15	3	1	2	18.730	0.295	2	94.506	2	200.627	2.498	24.11%	7.127
16	3	2	5	42.381	0.267	2	148.522	2	239.857	1.991	16.78%	5.705
17	3	2	5	43.968	0.277	2	167.510	2	242.187	2.010	12.71%	4.070
18	3	2	4	32.889	0.259	2	128.731	2	210.345	1.746	15.99%	4.822
Avg. of type 3	1.83	4.00	36.116	0.286	2	109.356	2	199.371	2.003	18.72%	5.027	
19	4	1	4	36.571	0.288	2	68.104	2	192.747	2.400	28.95%	7.694
20	4	1	4	47.619	0.375	2	87.534	2	193.988	2.415	22.43%	4.513
Avg. of type 4	1	4	42.095	0.332	2	77.819	2	193.367	2.407	25.69%	6.104	
Avg. of 4 types	1.7	4.7	47.365	0.316	2	93.116	2	189.582	1.933	20.11%	5.499	

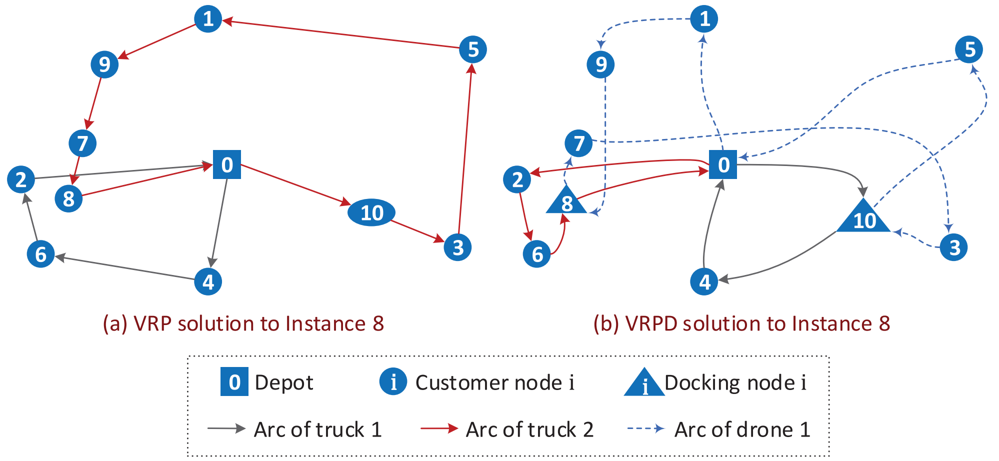

Fig. 3. Illustration of the routes of VRP and VRPD optimal solutions.

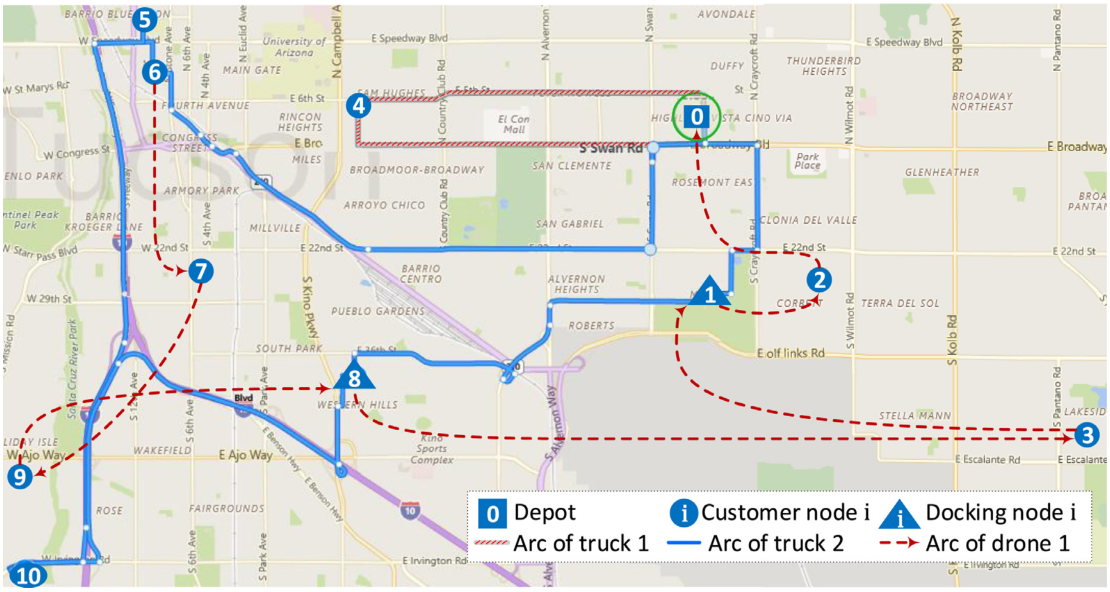

Fig. 4. Illustration of the routes in the optimal solution to instance 19.

delivery time gap is about 5.5 min, which indicates each customer can obtain her parcel 5 min earlier, on an average, if drone is used for delivering. The truck-drone delivery service can not only reduce the logistics cost but provide better service for customers.

From the obtained solutions, we found various patterns of collaborations between trucks and drones. We draw some routes in Fig. 3 in order to provide an intuitive impression of these collaborations and help understand the cost savings when using drones for delivery. The two trucks in the VRP solution are routed as “0-4-6-2-0” and “0-10-3-5-1-9-7-8-0” respectively. The two trucks and the drone in the VRPD solution are routed as “0-10-4-0”, “0-2-6-8-0”, and “0-1-9-8-7-3-10-5-0” respectively. As shown in Fig. 3(b), the drone departs from depot itself, lands at docking nodes twice, and returns back to depot. This solution contains not only short drone flights, e.g. from node 1 to 9 and 8 to 7, but also long flights, e.g. from node 7 to 3 and 10 to 5. If these flights are traveled by trucks, the transportation cost will increase obviously. Compared with the VRP solution in Fig. 3(a), the routing scheme in (b) is completely changed: some customers are still served by trucks while others are left for drones so that the cost saving can be achieved to the maximum degree of 21.73%.

We show in Fig. 4 the routing scheme of instance 19 on an electric map (Bing Map). From the figure, we can see that trucks have to travel along the roads on the ground that may be congested but a drone can fly over the roads and thus greatly reduce the travel time of a pair of nodes. The two types of vehicles achieve a well collaboration. In the solution, a truck passes through nodes 6, 5, 10, 8, 1, and 2 sequentially so that it can launch a drone at node 6 and meet the drone

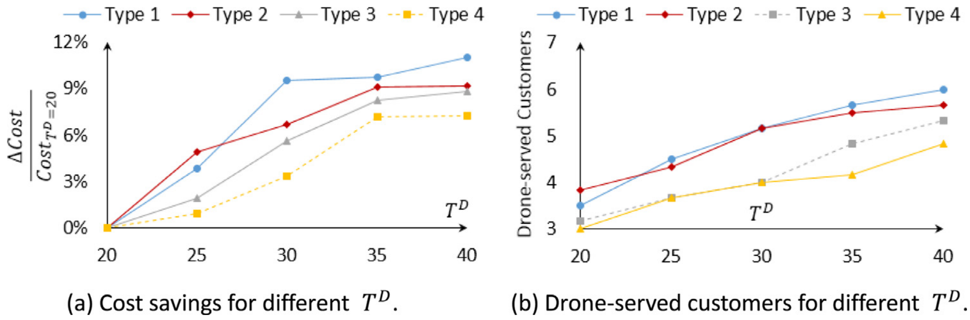

Fig. 5. Illustration of the results for different T^D.

at two docking nodes. As a sidekick of the truck, the drone delivers four customers within its flying range and loading constraint, including the most remote customer 3 and customers 7, 9, and 2. Without the participation of the truck, the drone cannot serve these customers itself. Without the drone, the truck has to spend much more cost serving the four customers. Compared with an independent truck delivery, such a collaboration achieves a 28.95% cost saving.

5.3. Sensitivity analysis

In this section, we conduct sensitivity analysis by varying the maximum flying duration of drones. The flying duration, denoted as T^D, determines the flying range of a drone after a battery swap. When battery techniques are improved, the flying duration can be enlarged. Given a set of customers and a takeoff node for a drone, the changes in the flying duration may result in different number of the customers served by a drone. We then experiment on this parameter by setting the parameter to 20, 25, 30, 35, and 40 min respectively.

Fig. 5(a) illustrates the ratio of the cost saving with different T^D to the total cost when T^D is 20. The ratio is the average over each instance type. We find from the results that the cost saving ratio is nearly 10% when T^D changes from 20 to 40. It can be observed that the cost saving mainly results from the transportation cost, not the fixed cost, since the numbers of the employed vehicles have small changes for different T^D. The cost saving shown in Fig. 5(a) can also be verified by Fig. 5(b), which illustrates the changes of drone-served customers with the changing T^D. In summary, an advanced battery technique that doubles the flying duration of drones will reduce the total logistics cost by nearly 10%.

6. Conclusions

This paper studies the vehicle routing problem with drones (VRPD), in which both trucks and drones are used for delivery. A drone may travel with a truck, take off from its stop to serve customers, and land at a service hub to travel with another truck as long as the flying range and loading capacity limitations are satisfied. Routing trucks and drones in an integrated manner makes the problem much more challenging and different from classical VRP literature.

We propose an arc-based integer programming model for VRPD. Since the special problem structure results in a large number of constraints, we reformulate it as a path-based model and developing a branch-and-price algorithm. In the pricing sub-problem, we design a special network that can distinguish different types of paths and nodes, and present an improved pulse algorithm by customized pruning and extending strategies. To mitigate the tailing-off effect from the slow convergence in column generation, we compute the alternative Lagrangian lower bound and apply column generation stabilization.

We obtain the optimal solutions by the branch-and-price algorithm for a set of instances in an average of 2 h, while the MIP solver of Gurobi fails to find the optimal solutions of 19 instances over 20 ones in a time comparable to the branch-and-price. The average cost gap is over 6% between the feasible solutions from Gurobi and the optimal solutions. Compared with a VRP solution, a VRPD solution not only saves an average cost of 20% but also advances the delivery time of each customer by an average of 5 min, which proves the effectiveness of using drones for delivery. In the sensitivity analysis, we observe that an advanced battery technique that doubles the flying duration of drones will reduce the logistics cost by nearly 10%.

As this is one of the first papers to address the mixed use of trucks and drones for delivery, there are many potential areas for future research. The introduction of docking hubs facilitates the coordination of vehicles but requires a big investment on a variety of resources and management. How to locate, size, and configure docking hubs at the minimum cost, and how to relocate drones beforehand at each hub in order to achieve an effective truck-drone routing schedule remain as significant problems. Other promising and challenging topics include investigating the heuristics for large-scale problems, and solving the problems with special aspects such as customer time windows and responses to real-time orders.

Acknowledgements

The authors are grateful to the two anonymous reviewers for their constructive comments and invaluable contributions to enhance the presentation of this paper. This work is also supported by the grants from NSF of China (No. 71421001 and 71531002), the Ministry of Science and Technology of Taiwan, Republic of China (MOST 106-2410-H-002-055-MY3), and Chang Jiang Scholars Program of China.

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transport. Sci. 52 (4), 965–981.
Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998. Branch-and-price: column generation for solving huge integer programs. Oper. Res. 46 (3), 316–329.
Campbell, J.F., Sweeney II, D.C., Z, J., 2017. Strategic Design for Delivery with Trucks and Drones Technical Report.
Carlsson, J.G., Song, S., 2017. Coordinated logistics with a truck and a drone. Manag. Sci. 64 (9), 4052–4069.
Chang, Y.S., Lee, H.J., 2018. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst. Appl. 104, 307–317.
Dayarian, I., Savelsbergh, M., Clarke, J.P., 2018. Same-Day Delivery with Drone Resupply Technical Report. 2018. Available at: http://www.optimization-online.org/DB_FILE/2017/09/6206.pdf.
Dell'Amico, M., Righini, G., Salani, M., 2006. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transport. Sci. 40 (2), 235–247.
Derigs, U., Pullmann, M., Vogel, U., 2013. Truck and trailer routing-problems, heuristics and computational experience. Comput. Oper. Res. 40 (2), 536–546.
DHL Trend Research, 2014. Unmanned Aerial Vehicles in Logistics. HYPERLINK. (Accessed Aug. 21st, 2018).
Farley, A.A., 1990. A note on bounding a class of linear programming problems, including cutting stock problems. Oper. Res. 38 (5), 922–923.
Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks 44 (3), 216–229.
French, S., 2015. Drone Delivery is Already Here – And it Works https://www.marketwatch.com/story/drone-delivery-is-already-here-and-it-works-2015-11-30.
Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco.
Global petrol prices. 2018. https://www.globalpetrolprices.com/gasoline_prices/. (Accessed Aug. 5th, 2018).
Golden, B.L., Raghavan, S., & Wasil, E.A. (Eds.). (2008). The Vehicle Routing Problem: Latest Advances and New Challenges (Vol. 43). Springer Science & Business Media.
Guedes, P.C., Borenstein, D., 2018. Real-time multi-depot vehicle type rescheduling problem. Transport. Res. Part B 108, 217–234.
Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. Transport. Res. Part C 86, 597–621.
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. Transport. Res. Part C 91, 1–14.
Hemmelmayr, V.C., Cordeau, J.F., Crainic, T.G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput. Oper. Res. 39 (12), 3215–3228.
Hern, A., 2014. DHL Launches First Commercial Drone 'Parcelcopter' Delivery Service https://www.theguardian.com/technology/2014/sep/25/german-dhl-launches-first-commercial-drone-delivery-service. (Accessed Nov. 5th, 2018).
Jones, T., 2017. International Commercial Drone Regulation and Drone Delivery Services. RAND Corporation, Santa Monica, CA https://www.rand.org/pubs/research_reports/RR1718z3.html. (Accessed Nov. 6th, 2018).
Kim, E., 2016. The Most Staggering Part About Amazon's Upcoming Drone Delivery Service https://www.businessinsider.com/cost-savings-from-amazon-drone-deliveries-2016-6. (Accessed Aug. 5th, 2018).
Klose, A., Drexl, A., 2005. Lower bounds for the capacitated facility location problem based on column generation. Manag. Sci. 51 (11), 1689–1705.
Klose, A., Görtz, S., 2007. A branch-and-price algorithm for the capacitated facility location problem. Eur. J. Oper. Res. 179 (3), 1109–1125.
Li, J.Q., 2014. Transit bus scheduling with limited energy. Transport. Sci. 48 (4), 521–539.
Lin, C.K.Y., 2011. A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources. Comput. Oper. Res. 38 (11), 1596–1609.
Lozano, L., Duque, D., Medaglia, A.L., 2015. An exact algorithm for the elementary shortest path problem with resource constraints. Transport. Sci. 50 (1), 348–357.
Lozano, L., Medaglia, A.L., 2013. On an exact method for the constrained shortest path problem. Comput. Oper. Res. 40 (1), 378–384.
Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. Oper. Res. 53 (6), 1007–1023.
Mathew, N., Smith, S.L., Waslander, S.L., 2015. Planning paths for package delivery in heterogeneous multirobot teams. IEEE Trans. Autom. Sci. Eng. 12 (4), 1298–1308.
du Merle, O., Villeneuve, D., Desrosiers, J., Hansen, P., 1999. Stabilized column generation. Discrete Math. 194 (1–3), 229–237.
Mogg, T., 2017. Amazon's Delivery Drones May Use Parachutes to Drop Packages in Your Yard https://www.digitaltrends.com/cool-tech/amazon-delivery-drone-parachutes/. (Accessed Nov. 5th, 2018).
Morgan, R., 2017. Drone Food Delivery Service is off the Ground in Iceland https://nypost.com/2017/08/24/drone-food-delivery-service-is-off-the-ground-in-iceland/. (Accessed Nov. 6th, 2018).
Mourelo Ferrandez, S., Harbison, T., Webwer, T., Sturges, R., Rich, R., 2016. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J. Ind. Eng. Manag. 9 (2), 374–388.
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transport. Res. Part C 54, 86–109.
Oswald, E., 2017. Here's Everything You Need to Know About Amazon's Drone Delivery Project. Prime Air https://www.digitaltrends.com/cool-tech/amazon-prime-air-delivery-drones-history-progress/. (Accessed Nov. 6th, 2018).
Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transport. Sci. 45 (3), 364–380.
Pessoa, A., Uchoa, E., de Aragão, M.P., Rodrigues, R., 2010. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. Math. Programm. Comput. 2 (3–4), 259–290.
Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: extended models and connections. Networks 70 (1), 34–43.
Ponza, A., 2016. Optimization of Drone-Assisted Parcel Delivery (Master's thesis).
Righini, G., Salani, M., 2006. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optim. 3 (3), 255–273.
Rousseau, J.M., Gendreau, M., Feillet, D., 2007. Interior point stabilization for column generation. Oper. Res. Lett. 35 (5), 660–668.
Ryan, D.M., Foster, B.A., 1981. An integer programming approach to scheduling. In: Wren, A (Ed.), Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling. North-Holland, Amsterdam, pp. 269–280.
Shields, N., 2018. China's Largest Courier is Starting Drone Deliveries https://www.businessinsider.com/chinas-largest-courier-to-start-drone-deliveries-2018-4. (Accessed Aug. 21st, 2018).
Tang, L., Wang, G., Liu, J., Liu, J., 2011. A combination of Lagrangian relaxation and column generation for order batching in steelmaking and continuous–casting production. Naval Res. Logist. 58 (4), 370–388.

Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications. Vehicle Routing: Problems, Methods, and Applications. Society for Industrial and Applied Mathematics.
Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2013. A matheuristic for the truck and trailer routing problem. Eur. J. Oper. Res. 230 (2), 231–244.
Wang, D., 2015a. The Economics of Drone Delivery https://www.flexport.com/blog/drone-delivery-economics/. (Accessed Nov. 15, 2018).
Wang, X., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: several worst-case results. Optim. Lett. 11 (4), 679–697.
Wang, Y., 2015b. Alibaba Starts Drone Delivery Test in Three-Day Program https://www.forbes.com/sites/ywang/2015/02/03/alibaba-starts-drone-delivery-test-in-three-day-program/. (Accessed Aug. 21st, 2018).
Wentges, P., 1997. Weighted Dantzig–Wolfe decomposition for linear mixed-integer programming. Int. Trans. Oper. Res. 4 (2), 151–162.
Wohlsen, M., 2014. The Next Big Thing You Missed: Amazon's Delivery Drones Could Work–they Just Need Trucks https://www.wired.com/2014/06/the-next-big-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-of-shipping/. (Accessed Aug. 15th, 2018).