Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Exact Algorithm for Heterogeneous Drone-Truck Routing Problem

Munjeong Kang, Chungmok Lee

To cite this article:
Munjeong Kang, Chungmok Lee (2021) An Exact Algorithm for Heterogeneous Drone-Truck Routing Problem. Transportation Science 55(5):1088-1112. https://doi.org/10.1287/trsc.2021.1055

informs
http://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Vol. 55, No. 5, September–October 2021, pp. 1088–1112
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

An Exact Algorithm for Heterogeneous Drone-Truck Routing Problem

Munjeong Kang,[a] **Chungmok Lee**[a]

[a] Department of Industrial and Management Engineering, Hankuk University of Foreign Studies, Yongin-si, Gyeonggi-do 17035, Korea

Contact: mj0225868@gmail.com, https://orcid.org/0000-0002-9742-1324 (MK); chungmok@hufs.ac.kr, https://orcid.org/0000-0002-0274-6928 (CL)

Received: October 2, 2020
Revised: December 14, 2020
Accepted: February 21, 2021
Published Online in Articles in Advance: August 18, 2021

https://doi.org/10.1287/trsc.2021.1055

Abstract. Recently, there are attempts to utilize drones in the logistic application. We consider the case in which there are multiple drones with different characteristics, such as speed and battery capacity. The truck and drone collaborate the delivery to serve all customers, while the drones are carried by and dispatched from the truck. The multiple drones can be deployed simultaneously; however, the truck must wait until all drones return. Therefore, the goal is to minimize the total sum of truck travel and waiting times for drones to return after deliveries. We call the proposed model a heterogeneous drone-truck routing problem (HDTRP), and a mixed-integer programming formulation for the problem is presented. We develop an exact algorithm based on the logic-based Benders decomposition approach, which outperforms the state-of-the-art solvers.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning [Grant NRF-2018R1A1A1A05077775]. C. Lee was also supported by the Hankuk University of Foreign Studies Research Fund of 2020.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/trsc.2021.1055.

Keywords: vehicle routing problem • drone logistics • Benders decomposition

1. Introduction

This paper introduces a heterogeneous drone-truck routing problem (HDTRP). Due to the limited operation ranges, the drones are not appropriate for long-distance delivery. Moreover, the drones have minimal carrying capability, which makes drone-only delivery virtually infeasible. On the other hand, traditional delivery vehicles like trucks can cover very long-ranged deliveries regardless of the volume and weight of the demands. Recently, a combined delivery of the drone and truck was proposed by various studies (Murray and Chu 2015, Boysen et al. 2018, Kim and Moon 2019, Gaba and Winkenbach 2020). The advantage of the drone-truck cooperative delivery is clear: the truck can mitigate the limitations of drones by cruising long distances so that the drones can be used only in the close proximity to the customers.

Most of the previous studies on the drone-truck combined delivery are based on the so-called "flying-sidekick" traveling salesman problem (FSTSP; Murray and Chu 2015, Daknama and Kraus 2017, Ha et al. 2018, Kitjacharoenchai et al. 2019, Sacramento et al. 2019). These studies considered the case in which the drone is dispatched from the truck, and the drone catches up the truck along the truck's route after delivery. One significant advantage of this approach is the truck can travel while the drone is away so that no time is wasted waiting for the drone. However, launching and retrieving the drone while the truck is moving can be impractical and posing a high risk due to the uncertainty of the operation conditions. For example, it is crucial to know the truck's future location for redocking, but precisely predicting the truck's movement is very difficult because of the uncertain traffic condition. Therefore, most of the FSTSP literature assumes a single drone-truck pair because planning real-time cooperations between many drones to the truck can become very complicated.

Another line of research is to use "stations" for drone delivery. The truck visits some fixed stations, from which drones are deployed (Murray and Chu 2015, Kim and Moon 2019). The parallel drone scheduling TSP (PDSTSP; Murray and Chu 2015) allows the drones to serve the customers inside the drone service range from the depot. There may exist multiple drone stations, and in this case, the truck is used to refurbish the demands for drones at the stations (Kim and Moon 2019). This approach needs much less synchronization between the truck and drones, and the use of multiple drones is relatively easy in this approach. However, establishing and maintaining drone stations requires significant financial investment. Moreover, the drone stations can be less flexible to highly dynamic demands because future demands can happen far from the drone stations.

The HDTRP addresses the drawbacks of the previous approaches by replacing the concept of drone stations with truck's temporary waiting. We assume that, contrary to the drone station-based approach, the drones are equipped at the truck, so there is no need to have stationary facilities for the drones. Moreover, contrary to the FSTSP, we consider heterogeneous drones that have different characteristics such as battery capacity and flight speed. The HDTRP does not permit in-moving docking and releasing of the drones, which can make the real-world implementation of the solution of the HDTRP rather feasible.

Figure 1 illustrates a feasible solution of the HDTRP for an 11-customer example problem. There is a single depot node d, and the truck departs from and returns to the depot after visiting some customers. The truck equips multiple drones, for example, two for this example, that can be used simultaneously. At a particular node, the truck deploys the drones for deliveries. A drone may take multiple deliveries as long as the drone's battery allows. In this example, the second drone conducts two deliveries for ⑧ and ⑨ at node ⑦, while the first one only delivers a single demand for ⑩ at the same node ⑦, which makes the truck wait for 6 (= $\max\{2+4, 5\}$). Therefore, at certain nodes (e.g., ② and ⑦), the truck must wait for all dispatched drones to complete assigned deliveries.

This study employs the following assumptions:

Assumption 1. *Each drone can carry a single demand at a time.*

Assumption 2. *Each drone has a specific speed and battery capacity.*

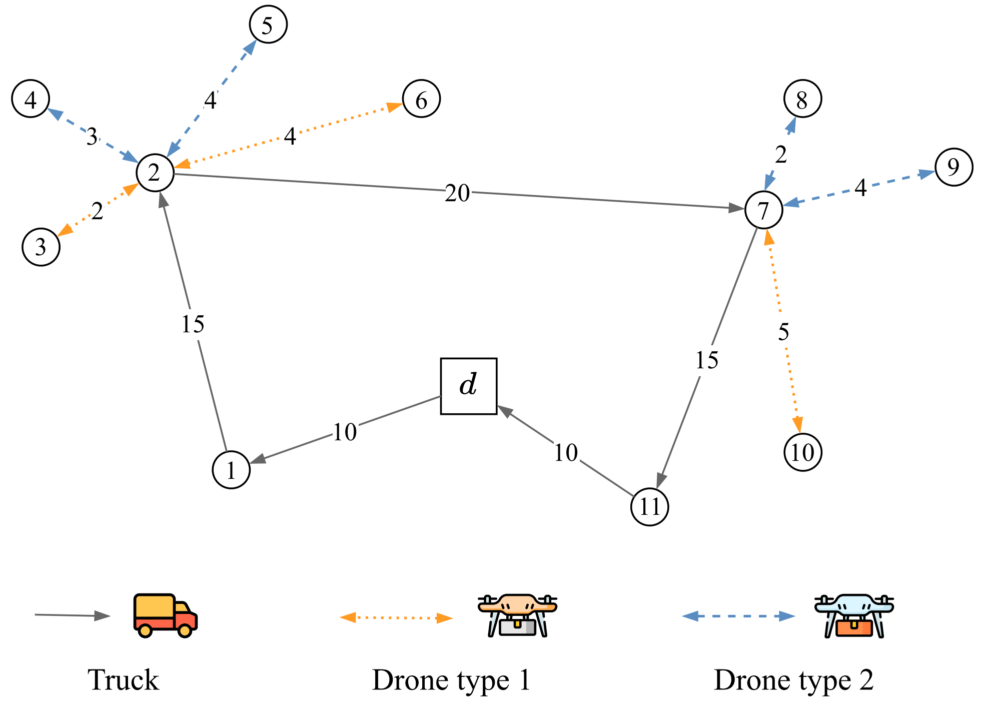
Assumption 3. *The truck has a sufficient capacity to deliver all demands while carrying all drones.*

Assumption 4. *Multiple drones can be dispatched for deliveries at the same time, and the drones must return to the location from which the drones depart.*

Assumption 5. *The truck cannot leave before all drones return to the truck.*

Note that, due to Assumptions 1 and 4, a drone's flight is always a return trip to the node at which it was launched. Moreover, the Assumption 4 enables the drones to be used parallel at a waiting node, as illustrated in Figure 1. We call the nodes "waiting nodes" if any drones are dispatched at the nodes (e.g., ② and ⑦ in the figure). We note that any nodes that the truck visits can be waiting nodes, so in the

Figure 1. (Color online) An Eleven-Node Example



Notes. Solid arrows represent the truck's route. Two drones are dispatched from the truck that is waiting for the drones to return from deliveries. Numbers on the arcs represent the travel time (for the truck) or delivery time (for drones). For the sake of simplicity, we assume the service time is zero for all customers. The total time for complete delivery is 83 (= 70 + 7 + 6). Note that by dispatching two drones at the same time, the truck can reduce the waiting times.

example, the truck may wait at nodes ①, ②, ⑦, ⑪, or *d*. This also means the drones only can be dispatched at these nodes. We see that the use of multiple drones due to Assumption 4 may enable faster deliveries. For example, notice that node ⑩ is closer to node ⑪ than node ⑦. However, dispatching the drone for node ⑩ at node ⑪ will increase the total completion time. Another aspect that differentiates the HDTRP from the previous drone station-based approach is that one needs to take the limited drone battery into account. Typically, the drone stations are well-equipped, and it is assumed that replacing the drone's battery is quick and easy, which implies that drones operating at different stations are independent, that is, the drones are not shared across the drone stations. However, in the HDTRP case, the same drones with fixed batteries are used for deliveries at multiple locations, so we should take the total battery usage at multiple locations into account. For example, the given battery capacity of the drone type 1, in Figure 1, must be large enough to cover all deliveries ② ↔ ③, ② ↔ ⑥, and ⑦ ↔ ⑩.

The benefit of the heterogeneousness of drones is clear. Typical parcel delivery consists of many light-weighted demands, which implies that the heavy demands can be preassigned to the truck. Among the demands assigned to the drones, there can be some demands only that can be delivered by a powerful but slow drone. The use of powerful but slow drones may deteriorate the merit of drone delivery. Therefore, in the homogenous case, the drone's speed and flight time are limited by the heaviest demand *assigned to the drones*. Having heterogeneous drones allows faster drones for lighter demands, which possibly reduces the delivery time and enlarges the set of demands that can be delivered by the drones.

We claim the following contributions to the literature:

- We propose a novel heterogeneous drone-truck joint routing problem.
- We present a mathematical formulation for the problem.
- We develop an exact algorithm based on the logic-based Benders decomposition approach.
- We report an extensive computational study that shows our algorithm outperforms the state-of-the-art MIP solver.

This paper is organized as follows. Section 2 reviews the previous studies related to the presenting problem. The formal problem definition and a mathematical formulation are given in Section 3. Section 4 develops the logic-based Benders decomposition algorithm for the problem. The results of computational experiments are reported in Section 5. Section 6 concludes the paper.

2. Literature Review

Recently there have been many attempts to use drones for deliveries. Giant logistic companies such as Amazon (Popper 2015) and DHL (DHL 2014) announced the use of drones for their parcel delivery service. The drone's ability to fly without human intervention is certainly considerable merit for logistics applications because it can avoid traffic. Moreover, awareness of global climate change pushes environmentally friendly logistics. Traditionally, logistics operations have been considered labor-intensive and the primary source of environmental pollution (Vidová et al. 2012). The "green" logistics aims to establish environmentally and economically sustainable logistics using environmentally friendly methods such as drones and electric vehicles (McKinnon et al. 2015, Cheng et al. 2020).

2.1. Traditional Vehicle Routing Problems

Among the logistics operation problems, the vehicle routing problem (VRP) has been studied extensively in the literature (Toth and Vigo 2002). The VRP is a generalization of the traveling salesman problem (TSP), which makes the VRP being NP-hard (Johnson and Garey 1979). There are a lot of variants of the VRP in the literature. Each variant of the VRP addressed specific business requirements and constraints. Some well-known variants include capacitated VRP (CVRP; Dantzig and Ramser 1959), the VRP with time windows (VRPTW; Russell 1977, Solomon 1987), the VRP with pickup and delivery (VRPPD; Savelsbergh and Sol 1995, Parragh et al. 2008), time-dependent VRP (TDVRP; Malandraki and Daskin 1992), stochastic VRP (SVRP; Laporte and Louveaux 1993; Gendreau et al. 1995, 1996; Laporte et al. 2002), robust VRP (RVRP; Sungur et al. 2008, Ordóñez 2010, Lee et al. 2013), and many more.

2.2. Drone Routing Problems

Most of the existing VRP studies implicitly assumed that the vehicle is a truck-like transportation unit. Unfortunately, replacing the trucks with drones introduces many novel challenges because the drone is not so efficient when it is operated in the same ways as trucks. Table 1 summarizes the advantages and disadvantages of drones compared with traditional trucks. In terms of operation characteristics, the truck and the drone are very different, so the previous assumptions used for the trucks no longer apply to the drones. For example, the most crucial decision in the VRP literature is the visiting sequence (i.e., route), which implicitly requires the vehicle can load the multiple demands. The drone's limited capacity often makes the route of the drone trivial, while the other considerations like short operation range become much more critical. Therefore, there are few studies on the "drone-only" routing problems for the delivery purpose. One exception is the multitrip drone routing problem (MTDRP; Cheng et al. 2018), which

Table 1. Operation Characteristics of Truck and Drone

	Speed	Weight	Capacity	Range	Environmental impact	Reliability
Drone	High	Light	One	Short	Little	Low
Truck	Low	Heavy	Many	Long	Big	High

Note. Updated from Agatz et al. (2018).

addressed the influence of payload and distance on flight duration. This study is based on the previous multitrip VRP (MTVRP; Cattaruzza et al. 2014), considering the explicit energy consumption function. The authors developed an exact algorithm based on the branch-and-cut and demonstrated the necessity and benefits of exact energy calculation via computational experiments. The MTVRP with time windows was considered by Paradiso et al. (2020), in which the column generation and cutting plain were utilized to solve problems with up to 50 customers. The computational study verified the efficiency of the developed algorithm using four variants of MTVRP proposed in the literature.

Other studies on the drone-only routing problems mainly concern the search and covering applications such as surveillance and aerial monitoring (Otto et al. 2018), which do not require "delivering" of physical demands. For example, Tseng et al. (2017) investigated a drone routing problem by considering the battery replacement. Recognizing the energy consumption of drones highly depends on the operation conditions (e.g., wind speed and direction), they presented a mathematical formulation incorporating the regression model for energy consumption and developed an approximation algorithm. There are also drone routing problems for specific applications: railway monitoring (Flammini et al. 2016), maritime surveillance (Bürkle and Essendorfer 2010), and agriculture (Mogili and Deepak 2018). We refer the reader to Otto et al. (2018) for a good survey on the drone-related optimization approaches.

2.3. Drone-Truck Cooperative Routing Problems

The drone-truck cooperative routing attempts to mitigate the drones' disadvantages by making the drone and truck operating together for delivery and collection tasks. Usually, the drones attach to the roof of the truck so that truck becomes the carrier for the drone and demands (Lithia 2017). The drone-truck cooperative routing often appears in parcel delivery applications. The review paper by Macrina et al. (2020) provides a good overview of the last-mile drone-related delivery literature. There are mainly two distinct approaches in the drone-truck cooperative routing literature: (1) the variants of the FSTSP and (2) the drone station-based approaches.

2.3.1. Flying Sidekick Traveling Salesman Problems.

The FSTSP was first introduced by Murray and Chu (2015), in which the drone is used for last-mile delivery. Any customers in close proximity to the depot can be directly served by the drone, while the other customers out of reach of the drone are covered by either the truck or the drone launched from the truck. After launching the drone, the truck should move to the next stop to reduce the total delivery time. The drone is reunited with the truck, after finishing the delivery, at the next truck stop, which forms a triangle-shaped subroute between the truck and drone. The authors presented a mathematical formulation and proposed a route-construction type heuristic.

The multiple TSP with the drone (mTSPD; Kitjacharoenchai et al. 2019) considers the case in which multiple drones and trucks can be used. The mTSPD extends the previous mTSP (Bektas 2006) by utilizing the drones in the same way as the FSTSP. Note that mTSP is equivalent to the classical VRP with an infinity capacity, of which the objective is to minimize the total delivery time. The proposed mathematical formulation is a complicated mixed-integer problem due to the synchronization between the trucks and drones, which is very hard to solve. The authors developed a heuristic, which is called mTSPD-ADI, and showed that the proposed algorithm outperforms Cplex.

Another extension of the FSTSP was proposed by Sacramento et al. (2019), which is called VRP with drones (VRP-D). The original FSTSP and its variants, called TSP-D, did not consider the capacity of the truck. The VRP-D addressed the truck capacity, which results in a problem with multiple, but identical, trucks. Except for the capacity constraint and multiple trucks, the VRP-D is a direct generalization of the FSTSP. Recognizing that the presented mathematical formulation is very hard to solve, an adaptive large neighborhood search based heuristic is developed. A variant of TSP-D, called min-cost TSP-D, was introduced by Ha et al. (2018), which minimizes the sum of transportation cost and time spent waiting for the drone. Along with a mathematical formulation, two heuristic methods were developed by adapting the previous one developed for the FSTSP. Boysen et al. (2018) considered the drone scheduling problem (DSP), which determines the drone route from a given truck route. This study takes multiple drones into account and allows the drones to land at the nodes at which the drones did not take off. The authors provided computational complexity for each case and proposed a mathematical formulation. The k-Multi-Visit Drone Routing Problem (k-MVDRP) proposed by Poikonen and Golden (2019) addressed multiple (k) drones that can deliver one or more packages to customers. The study provided a heuristic method that can consider a specific drone energy drain function defined by the package weights delivered by the drones.

Solution approaches for the variants of FSTSP, as mentioned, are mostly heuristics. The mathematical formulations for the VRPs are known for the weak linear relaxation bounds, which yields poor performance of state-of-the-art MIP solvers like Cplex. The synchronization constraints for the drone-truck coordination makes the problem even harder to solve, which yields scarce studies on the exact algorithms in the literature. Vásquez et al. (2020) developed an exact algorithm for the TSP-D by decomposing the problem into two subproblems for the truck and drone. A Benders cut-generation algorithm was employed, which could solve problems with up to 20 nodes. On the other hand, the single drone assumption naturally encourages the use of the insertion heuristic type approaches (Rosenkrantz et al. 1977), that is, replacing one truck visit with that of a drone and connecting the previous and next visits with the drone, which results in various local search heuristics. Besides the classical heuristic methods, a mathematical programming based heuristic was proposed by Yurek and Ozmutlu (2018). They decomposed the problem into two separate but depending decision problems. The two decomposed problems were solved iteratively to find a feasible solution.

2.3.2. Drone-Station-Based Approaches.

A drone station is a dedicated facility equipped with (possibly many) drones. In the drone-station based approaches, the drones operate at the drone stations, not at the truck. The truck visits the drone stations to refurbish the demands for the drones, only after which the drones fly to the customers within the operation range. Therefore, in this approach, there are no route synchronization constraints between the drones and the truck. Moreover, it is typically assumed that the drones have unlimited battery capacity because the battery can be replaced easily at the drone stations. However, the use of the many drones at the drone station invites another challenge: parallel drone scheduling with precedence constraints.

The parallel drone scheduling TSP (PDSTSP) proposed by Murray and Chu (2015), along with the FSTSP, is an extension of the well-known parallel machine scheduling (PMS). The customers within the drone operation range can be served by either the drone or truck, while the remaining customers are visited by the truck. The parallel drone scheduling problem occurs at the depot, in which multiple drones should be operated in parallel to reduce the makespan. Two heuristic algorithms for the PDSTSP were developed by Dell'Amico et al. (2020), which were compared with the MIP formulation solved by Gurobi. The computational experiments with generated problem instances showed that the proposed heuristics could obtain good quality solutions for up to 90 customer problems.

The original PDSTSP assumed a single drone station located at the depot, which was generalized by Kim and Moon (2019), called TSP with drone stations (TSP-DS), to allow multiple drone stations. One significant advantage of the TSP-DS is the drone station may be relatively cheaper to maintain than additional depots. With a given set of drone stations, the authors developed an optimization algorithm by deriving a decomposition method.

Determining the locations of the drone stations is crucial in designing an efficient drone-truck logistics network. There are a large number of studies on facility location problems (FLP; Laporte et al. 2019). Notably, the location-routing problem (LRP; Drexl and Schneider 2015) combines the decisions for the route and locations, which can produce suboptimal solutions when solved separately. In the VRP literature context, the LRP is a generalization of the VRP with intermediate depots (VRPwID; Perboli et al. 2011), in which the vehicle is refurbished at the intermediate depots. Even though the drone stations can be seen as a special case of the intermediate depots, there is scant literature for this line of approach for the drone truck joint routing. One exception is the traveling salesman drone station location problem (TSDSLP) by Schermer et al. (2019), in which a set of candidates for the drone stations is given. A mathematical formulation was presented, which could solve small and medium-sized problems by the MILP solver Gurobi.

2.3.3. Heterogeneous Drone-Truck Routing Problem.

It is noteworthy that our problem HDTRP has a close connection with the two-echelon location routing problem (2E-LRP; Perboli et al. 2011, Cuda et al. 2015), in which two hierarchical vehicle routes are interconnected at some nodes, called "satellites," to be determined. In the HDTRP, the truck route and drone delivery routes are linked hierarchically at waiting nodes. We note that, in the 2E-LRP, there is a clear distinction between the potential satellites and customers. Hence, the upper echelon route only involves the satellite nodes, while lower echelon routes serve only the customers. This implies that the TSDSLP is a special case of the 2E-LRP, while the lower echelon routes are operated by the drones. Another thing to note is that typically the 2E-LRP does not take into account the temporal aspects because it is essentially a combination of the FLP and the VRP. However, in the HDTRP, the drone only can be used after the truck arrives at a particular node. Moreover, in the HDTRP, the drones' operations at different waiting nodes are not independent due to the sharing of the drone battery. This means that one cannot deal with the drone scheduling decisions separately, which complicates the problem. We note that in the TSDSLP (or the 2E-LRP), there is a relatively small set of potential drone stations (or satellites) whose locations are predetermined carefully. Therefore, it can be said that both TSDSLP and 2E-LRP readily make use of well-determined locations of the potential drone stations. On the other hand, the HDTRP does not assume possible drone stations, which means any nodes can be the temporary waiting nodes. Therefore, the HDTRP can have different waiting nodes in the same area for any given demand. This can be an advantage of the HDTRP because it may be costly to change the drone stations in use for the daily delivery operation.

3. Problem Definition

The HDTRP is defined as follows. Consider a set of customers N and a given depot d. The depot node is duplicated as two identical nodes s and t. Let N_s denote a set $N \cup \{s\}$. Similarly, N_t and N_{st} are defined as $N \cup \{t\}$ and $N \cup \{s,t\}$, respectively. We consider a directed graph $G(N_{st}, A)$, where the set of arcs is defined as $A := \{(i,j) | i \in N_s, j \in N_t, i \neq j\}$. We denote L as the set of drones. For any drone $l \in L$, the battery capacity B^l is given, which represents how long the drone can be used for deliveries. The truck travel time of an arc $(i,j) \in A$ is given as t^v_{ij}. Without loss of generality, we assume that the travel time t^v_{ij} for the truck includes the service time at node j. The delivery of a drone can be seen as three-leg operations: (1) a drone is dispatched at node i with attached the demand for another node j; (2) the drone lands at the node j to release the demand; and (3) the drone returns to node i. Because we know the distance of arc (i, j) and demand for j, we can precalculate the required energy for a given drone. We denote b^l_{ij} as the total battery consumptions required to complete the delivery operations of drone $l \in L$ from waiting node $i \in N_s$ for the demand of $j \in N$. Similarly, we denote τ^l_{ij} as the total required time for the drone l to complete the delivery from i to j. The HDTRP is NP-hard because it contains the TSP as a special case when $L = \emptyset$.

Now we present the mathematical formulation (P) for the HDTRP as follows:

$$\text{(P)} \min \sum_{(i,j)\in A} t^v_{ij} x_{ij} + \sum_{i \in N_s} w_i \tag{1}$$

$$\text{s.t.} \sum_{j \in N} x_{sj} = 1, \tag{2}$$

$$\sum_{i \in N} x_{it} = 1, \tag{3}$$

$$\sum_{j \in N_t : j \neq i} x_{ij} = \sum_{j \in N_s : j \neq i} x_{ji}, \quad \forall\, i \in N, \tag{4}$$

$$v_i - v_j \leq M(1 - x_{ij}) - 1, \quad \forall\, (i,j) \in A, \tag{5}$$

$$\sum_{i \in N_s : i \neq j} x_{ij} + \sum_{i \in N_s : i \neq j} \sum_{l \in L} h^l_{ij} = 1, \quad \forall\, j \in N, \tag{6}$$

$$M \sum_{j \in N_t : j \neq i} x_{ij} \geq \sum_{j \in N : j \neq i} \sum_{l \in L} h^l_{ij}, \quad \forall\, i \in N_s, \tag{7}$$

$$\sum_{i \in N_s} \sum_{j \in N : j \neq i} b^l_{ij} h^l_{ij} \leq B^l, \quad \forall\, l \in L, \tag{8}$$

$$w_i \geq \sum_{j \in N : j \neq i} \tau^l_{ij} h^l_{ij}, \quad \forall\, i \in N_s, l \in L, \tag{9}$$

$$v_s = 0, \tag{10}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, \tag{11}$$

$$h^l_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, l \in L, \tag{12}$$

where M is a big-number. The binary decision variable x_{ij} is 1 if the truck moves from i to j, 0 otherwise. The continuous variable v_i represents the "visiting order" of the truck at the node i. If a drone $l \in L$ is dispatched to the node j from node i, the binary variable h^l_{ij} is 1, 0 otherwise. The truck waiting time at node i is determined by the continuous decision variable w_i.

The objective function (1) minimizes the sum of the travel times (including the total service times) and the total waiting times. The truck must depart from and returns to the depot, which is ensured by the constraints (2) and (3). The constraints (4) are well-known flow-balance constraints. The truck route must not contain any subtours. Therefore, the so-called subtour elimination constraints (5) proposed by Miller et al. (1960) are added. The constraints (6) ensure that exactly one of the truck or drones must serve (i.e., visit) all customer nodes. The drones can be dispatched from the node i only if the truck visits the node, which is represented in the constraints (7). The use of a drone consumes the drone's battery. The total battery consumption cannot exceed the given battery capacity, which is ensured by the constraints (8). The truck must wait for all dispatched drones to return from the deliveries. Therefore, the waiting time should be greater than or equal to the total time spent by the drone deliveries, which is ensured by the constraints (9). The constraints (11)–(12) represent the binary requirements of the decision variables.

The problem (P) has many binary decision variables that make solving the formulation by the MIP solvers very challenging. Especially, the subtour elimination constraints (5) are known for poor linear relaxation bounds.

3.1. Extensions of HDTRP

The proposed model can be modified to address extra business restrictions or requirements. Here, we present some possible extensions that can be addressed easily.

3.1.1. No-Fly-Zone. In many countries, the use of drones is subject to strict regulations due to safety and privacy concerns (Jones 2017). The no-fly-zone does not allow any use of drones inside the area. There can be two cases in which the no-fly-zone hinders the drone operations for customer j: (1) the customer j is located inside the no-fly-zone; or (2) the customer j is not in the no-fly-zone but the drone's flight path from some node i intersects the no-fly-zone. The first case can be easily modeled by letting $h_{ij}^l = 0$ for all $i \in N_s$ such that $i \neq j$. On the other hand, the second case can be modeled by considering a detour to avoid the no-fly-zone, which will increase τ_{ij}^l.

3.1.2. Battery Replacement. Recharging the drone's battery would take significant time. A more practical method for extending the drone usage time is replacing the drone's battery with a spare one. Let R^l denote the set of spare batteries for drone $l \in L$. We replace the constraints (8) with the following set of constraints:

$$\sum_{j \in N: j \neq i} b_{ij}^l h_{ij}^l \leq B^l u_i^{l,r}, \quad \forall\, i \in N_s, r \in R^l, l \in L, \tag{13}$$

$$\sum_{i \in N_s} u_i^{l,r} \leq 1, \quad \forall\, r \in R^l, l \in L, \tag{14}$$

$$u_i^{l,r} \geq 0, \quad \forall\, i \in N_s, r \in R^l, l \in L, \tag{15}$$

where the newly added decision variable $u_i^{l,r}$ determines the usage of the battery r's capacity for drone l at node i. We note that having two identical batteries is not the same as having a single battery with twice capacity, which is why the decision variable $u_i^{l,r}$ and the constraints (13)–(14) are required.

3.1.3. Weighty Demands. If there are some demands that cannot be delivered by a drone, for example, too heavy demands, we should ensure that the nodes with these demands are served by the truck. This can be done easily by adding $\sum_{l \in L} \sum_{i \in N_s: i \neq j} h_{ij}^l = 0$, for all j with the heavy demands.

3.1.4. Weather Conditions. The energy consumption and speed of drones significantly depend on weather conditions (Tseng et al. 2017). To address the weather effect, one can make the adjustment on the value of b_{ij}^l and τ_{ij}^l according to the given weather conditions. For example, if the speed and direction of the wind are known, we can calculate the adjusted b_{ij}^l and τ_{ij}^l by considering the direction of arc (i, j) against the wind direction.

4. Logic-Based Benders Decomposition Approach

The problem (P) consists of two distinct decisions: the truck route (i.e., x_{ij}) and drone deliveries (i.e., h_{ij}^l). Without the coupling constraints (6) and (7), the problem can be separated into two independent decision problems. Based on this observation, we develop a decomposition approach in this section.

The Benders decomposition, originally proposed by Benders (1962), divides the problem into two distinct parts: the Benders master and subproblems. The classical Benders approach can be applied when the problem has both discrete and continuous decision variables. The Benders master problem finds the best discrete decision, which in turn is provided to the Benders subproblem. The Benders subproblem is solved after "fixing" the discrete decision variables to assert the validity of the provided solution of the Benders master problem. After solving, the Benders subproblem produces two types of cuts: Benders optimality and Benders feasibility cuts. The former cuts state that the given master solution cannot be optimal, while the latter ones mean that it is not possible to have a feasible solution with the given master solution. Any identified Benders cuts are added to the Benders master problem so that the current master solution cannot be reproduced. The Benders decomposition algorithm proceeds by iteratively solving the Benders master problem and the Benders subproblem. The algorithm terminates if there are no Benders cuts to add.

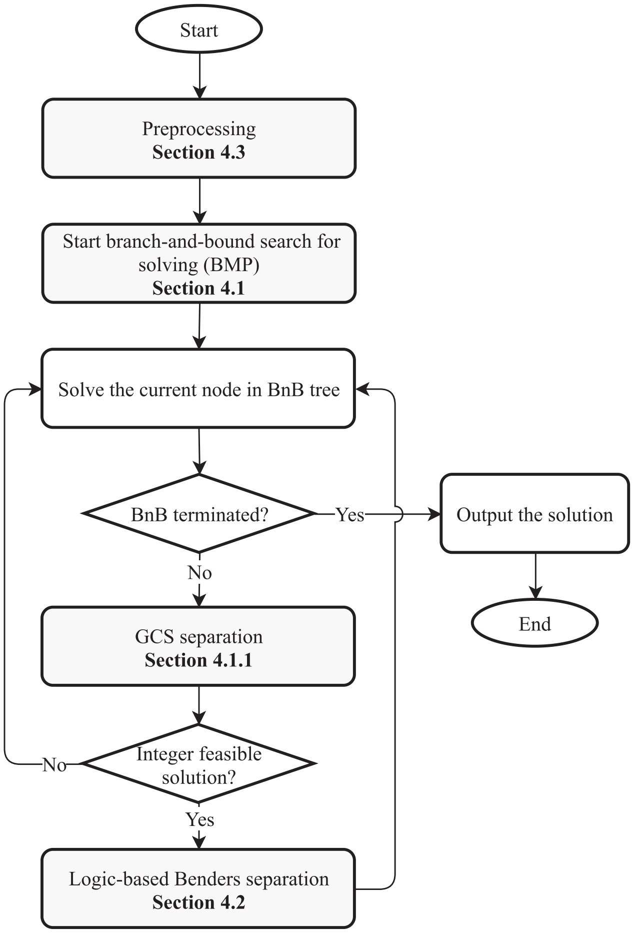
The Benders decomposition approach has been used in various applications, including network design (Balakrishnan et al. 1995, Lee et al. 2013), transportation (Errico et al. 2017, Lee and Han 2017), vehicle routing (Corréa et al. 2007), and stochastic optimization (Infanger 1992, Nielsen and Zenios 1997). We refer readers to Rahmaniani et al. (2020) for a good survey on the Benders decomposition.

The classical Benders decomposition requires the Benders subproblem to be a convex optimization problem (e.g., linear programming) because the Benders cuts are obtained from the duality of the Benders subproblem. Regarding the problem (P), the Benders subproblem should give the Benders cuts with given truck route solution x_{ij}, which implies that the Benders subproblem is an MIP problem that determines the drone deliveries subject to the battery restrictions. Therefore, we cannot employ the classical Benders decomposition approach for solving the problem (P).

The logic-based Benders decomposition is an extension of the classical Benders approach (Hooker and Ottosson 2003, Hooker 2007, Roshanaei et al. 2017). The main idea of the logic-based Benders decomposition is to utilize "inference dual" that provides a valid lower bound of the objective value for the Benders master solutions. The inference dual problem can be any optimization problems that provide a valid low bound for all feasible solutions of the Benders master problem, which is why it is called "dual" even though the MIP Benders subproblem does not warrant the strong duality.

Figure 2 shows the flowchart of the proposed algorithm. The grayed processes are of importance and explained in the sections noted. In the classical Benders decomposition algorithm, the Benders master problem is solved to optimality every iteration. However, the Benders cut generation can be incorporated into the branch-and-bound search (Rahmaniani et al. 2020). In this case, the Benders master problem is solved once, and the Benders subproblem is solved to generate the Benders cuts whenever an integer feasible solution is found, which constitutes a *branch-and-cut* algorithm.

Figure 2. Flowchart of the Proposed Algorithm


Start
Preprocessing
Section 4.3
Start branch-and-bound search for solving (BMP)
Section 4.1
Solve the current node in BnB tree
BnB terminated?
Yes
Output the solution
End
No
GCS separation
Section 4.1.1
Integer feasible solution?
No
Yes
Logic-based Benders separation
Section 4.2

Note. The steps of importance are grayed with section numbers for detailed explanations.

In our branch-and-cut algorithm, two separation procedures are incorporated. The first separation procedure is for ensuring no subtour of the truck's route (see Section 4.1.1), and the second separation solves the Benders subproblem to find the logic-based Benders cuts (see Section 4.2). We implemented the separation procedures by using the callback interfaces provided by Cplex (CPLEX-User's-Manual 1987). The preprocessing stage employs both primal heuristic and variable fixing to accelerate the branch-and-cut (see Section 4.3). In the following, we explain details of the logic-based Benders decomposition, followed by a description of the preprocessing procedure.

4.1. Benders Master Problem

The logic-based Benders master problem is stated as follows.

$$\text{(BMP)}\ \min \sum_{(i,j)\in A} t^v_{ij} x_{ij} + W \tag{16}$$

$$\text{s.t.} \sum_{j\in N} x_{sj} = 1, \tag{17}$$

$$\sum_{i\in N} x_{it} = 1, \tag{18}$$

$$\sum_{j\in N_t: j\neq i} x_{ij} = \sum_{j\in N_s: j\neq i} x_{ji}, \quad \forall\, i \in N, \tag{19}$$

$$\sum_{j\in N_t: j\neq i} x_{ij} = z_i, \quad \forall\, i \in N, \tag{20}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, \tag{21}$$

$$z_i \in \{0,1\}, \quad \forall\, i \in N, \tag{22}$$

$$W \geq 0. \tag{23}$$

The purpose of the problem (BMP) is to find a feasible truck route. Note that the problem (BMP) has binary decision variables x_{ij} and constraints (17)–(19) for the truck route. However, there are no subtour elimination constraints in (BMP). The binary decision variable z_i is 1 if the truck visits node i, 0 otherwise. We note that the constraints (20) may be replaced with $\sum_{j\in N_t: j\neq i} x_{ij} \leq 1$ for all $i \in N$ so that the binary variable z_i can be removed. The binary variable z_i is introduced to clearly represent the master problem's decision to the Benders subproblem. The continuous variable W represents the total sum of waiting times of the truck.

The Benders master problem (BMP) only concerns the truck's route decision, so the obtained solutions, especially at early iterations, may not be good in terms of the original objective function (1). A well-known trick is to add a subset of the constraints and decision variables of the Benders subproblem to the Benders master problem. We add the following set of constraints to the problem (BMP).

$$\sum_{i \in N_s : i \neq j} \sum_{l \in L} h^l_{ij} = 1 - z_j, \qquad \forall\, j \in N, \tag{24}$$

$$z_i \geq h^l_{ij}, \qquad \forall\, i \in N, j \in N_t, l \in L, \tag{25}$$

$$\sum_{i \in N_s, j \in N, j \neq i} b^l_{ij} h^l_{ij} \leq B^l, \qquad \forall\, l \in L, \tag{26}$$

$$W \geq \sum_{i \in N_s} w_i, \tag{27}$$

$$w_i \geq \sum_{j \in N : j \neq i} \tau^l_{ij} h^l_{ij}, \qquad \forall\, i \in N_s, l \in L, \tag{28}$$

$$h^l_{ij} \geq 0, \qquad \forall\, (i,j) \in A, l \in L. \tag{29}$$

The decision variables for drone deliveries h^l_{ij}, which are continuous, and the constraints (24)–(28) are not necessary but added to "guide" the Benders master problem for better solutions at the expense of increased problem size.

Note that the constraints (26)–(28) have no integer decision variables. Therefore, the Cplex solver cannot exploit these constraints to add valid inequalities to strengthen the linear relaxation bound. We consider the following constraints.

$$w_i \geq \min_{l \in L}\{\tau^l_{ij}\} \sum_{l \in L} h^l_{ij}, \qquad \forall\, (i,j) \in A. \tag{30}$$

Proposition 1. *The constraints (30) exclude no integer optimal h^l_{ij} solutions while strengthening the linear relaxation bound.*

Proof. It is clear that the above constraints exclude no integer optimal h^l_{ij} solutions because any optimal solutions should satisfy $\sum_{l \in L} h^l_{ij} \leq 1$ for any $(i,j) \in A$ unless $\tau^l_{ij} = 0$ for some $l \in L$. It is sufficient to show there is a fractional feasible solution of the problem (BMP), that violates the constraints (30). Consider a feasible solution of (BMP) satisfying $\hat{z}_i = 1, w_{\hat{i}} = 0.5 \max\left\{\tau^{l_1}_{\hat{i}\hat{j}}, \tau^{l_2}_{\hat{i}\hat{j}}\right\}$ and $h^{l_1}_{\hat{i}\hat{j}} = h^{l_2}_{\hat{i}\hat{j}} = 0.5$. This solution does not satisfy the constraints (30) when $0.5 \max\left\{\tau^{l_1}_{\hat{i}\hat{j}}, \tau^{l_2}_{\hat{i}\hat{j}}\right\} < \min\left\{\tau^{l_1}_{\hat{i}\hat{j}}, \tau^{l_2}_{\hat{i}\hat{j}}\right\}$. □

Figure 3 illustrates the case in which the constraints (30) strengthen the linear relaxation bound of the problem (BMP). The constraints (30) are added to the problem (BMP) before solving the Benders master problem. As stated earlier, the separation procedures for both subtour elimination (Section 4.1.1) and Benders cuts (Section 4.2) are executed during the branch-and-bound tree search. Therefore, the proposed logic-based Benders decomposition algorithm terminates when the branch-and-bound for the problem (BMP) finishes.

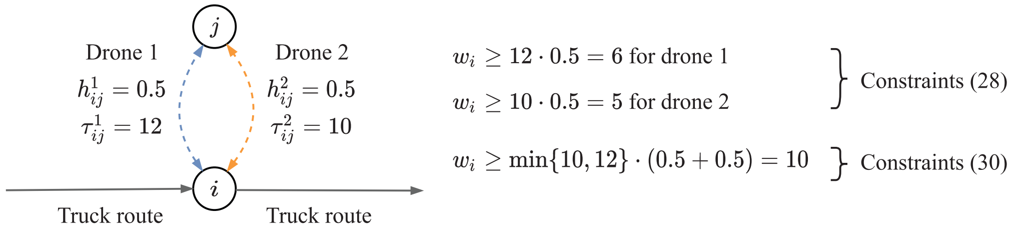
4.1.1. Subtour Elimination Constraints.

The problem (BMP) lacks the subtour elimination constraints. One option is to add the so-called MTZ subtour elimination constraints (5) to the problem (BMP). The MTZ formulation is compact but has a relatively weak linear relaxation bound. A stronger linear relaxation bound can be obtained by using exponentially many constraints such as generalized cut-set inequalities (GCS; Taccari 2016). The GCS is an extension of the well-known cut-set inequalities often used in the traveling salesman problem (TSP) literature. Unlike the TSP, the truck of the problem (BMP) does not need to visit all nodes, which results in the following valid inequalities:

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq \sum_{(i,j) \in \delta^+(\{k\})} x_{ij}, \qquad \forall\, k \in S, S \subseteq N_s, |S| \geq 2, \tag{31}$$

where $\delta^+(S) := \{(i,j) \in A | i \in S, j \notin S\}$, that is, a set of arcs leaving the set S. The GCS cuts ensure that the truck must leave from a node subset S if the truck visits any nodes in S. Because there are exponentially many constraints (31), we use the branch-and-cut method, as shown in Figure 2. At every node during the branch-and-bound search for solving the problem (BMP), we solve the separation problem presented in Algorithm 1, which is adopted from Taccari (2016).

Figure 3. (Color online) A Feasible Solution to the Constraints (24)–(29)



Notes. The truck passes node *i*, and two (fractional) drone routes visit node *j* at the same time. The added constraints (30) can strengthen the LP relaxation bound of the problem (BMP).

Algorithm 1 (Separation of GCS for Subtour Elimination)

```
1: $\mathbf{x}^* \leftarrow$ solution of the (BMP) at the current BnB node
2: $\epsilon \leftarrow 0.8$
3: Construct graph $G(N_{st}, A^*)$, where $A^* := \{(i,j) \in A | x^*_{ij} > 0$ or $x^*_{ji} > 0\}$
4: $\mathcal{S} \leftarrow \{S \subseteq N_s | S$ is a strongly connected component on G$\}$ ▷ Depth-first search on $G(N_{st}, A^*)$
5: $C \leftarrow \emptyset$
6: for $S \in \mathcal{S}$ do
7:     for $k \in S$ do
8:         $v \leftarrow \sum_{(i,j)\in\delta^+(\{k\})} x^*_{ij} - \sum_{(i,j)\in\delta^+(S)} x^*_{ij}$
9:         if $v \geq \epsilon$ then
10:            $C \leftarrow C \cup \{(v, S, k)\}$
11:        end if
12:    end for
13: end for
14: return $C$
```

The separation of GCS constraints (31) should determine a node subset S and a node $k \in S$. Algorithm 1 first finds the set of strongly connected components on the support graph G. For each strongly connected component S, the node $k \in S$ giving a violated constraint among (31) is found in lines 7–12. The tolerance ϵ is introduced to make sure only "strongly violated" cuts are found. The set of violated cuts C is added to the problem (BMP). The tolerance ϵ is required when we want to separate any fractional solution $\mathbf{x}^*$. That is, any tolerance $\epsilon > 0$ is sufficient to exactly separate the integer solution $\mathbf{x}^*$. We set $\epsilon = 0.8$ because this is the recommended value by Taccari (2016) from an extensive computational study.

Note that one can obtain a subtour free route if the separation is solved only when the branch-and-bound node has an integer solution because Algorithm 1 is exact when $\mathbf{x}^*$ is integral. However, our computational experiments show that solving the separation at every node, regardless integrality of the solution, performs better.

4.2. Benders Subproblem

The goal of the Benders subproblem is to identify Benders cuts with a given Benders master solution. Let $(\mathbf{x}^*, \mathbf{z}^*, W^*)$ be a feasible truck solution to the problem (BMP). Being an integer feasible solution of (BMP) means that the solution $\mathbf{x}^*$ satisfies all GCS constraints, that is, the solution $\mathbf{x}^*$ is a valid truck route without subtours. Let $N_0(\mathbf{z}^*) := \{i \in N | z^*_i = 0\}$ denote the set of customers not visited by the truck route. Similarly, we denote $N_1(\mathbf{z}^*) := \{i \in N | z^*_i = 1\}$ as the set of possible waiting nodes of the truck route. The Benders subproblem is given as follows.

$$
\begin{aligned}
\text{(BSP)} \min \quad & \sum_{i\in N_s} w_i && \quad (32)\\
\text{s.t.} \quad & \sum_{i\in N_s}\sum_{j\in N: j\neq i} b^l_{ij} h^l_{ij} \leq B^l, \quad \forall\, l \in L, && \quad (33)\\
& w_i \geq \sum_{j\in N: j\neq i} \tau^l_{ij} h^l_{ij}, \quad \forall\, i \in N_s, l \in L, && \quad (34)\\
& \sum_{i\in N_s: i\neq j}\sum_{l\in L} h^l_{ij} \geq 1 - z^*_j, \quad \forall\, j \in N, && \quad (35)\\
& \sum_{j\in N: j\neq i}\sum_{l\in L} h^l_{ij} \leq |N| z^*_i, \quad \forall\, i \in N, && \quad (36)\\
& h^l_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, l \in L, && \quad (37)\\
& w_i \geq 0, \quad i \in N_s. && \quad (38)
\end{aligned}
$$

The binary decision variable h^l_{ij} is 1 if the drone $l \in L$ is dispatched at node i for delivery of j, 0 otherwise. The waiting time at node i is represented by the decision variable w_i. The objective function (32) minimizes the total waiting times. The constraints (33) restrict each drone battery's total consumption. The truck's waiting time at node i depends on the drones deployed at the node, which is determined by the constraints (34). The constraints (35) make all customers in $N_0(\mathbf{z}^*)$ served by the drones. The constraints (36) prevent the dispatch of the drones if the truck does not visit the node, that is, $z^*_i = 0 \rightarrow h^l_{ij} = 0, \ \forall\, j \in N, l \in L$.

The problem (BSP) is the 0-1 Multiple Knapsack Problem with side constraints (Martello and Hoboken 1990), which is NP-hard. However, the state-of-the-art MIP solvers like Cplex can solve the problem (BSP) very quickly, unless the problem size is very large. Moreover, a significant number of binary decision variables are fixed to zero due to the constraints (36). With a given $\mathbf{z}^*$, solving (BSP) can result in two cases: infeasibility and an optimal solution.

4.2.1. Case 1: (BSP) Is Infeasible.

This means that, with the current $\mathbf{z}^*$, the drones cannot complete the deliveries due to the shortage of battery. In this case, we need to eliminate the current solution $(\mathbf{x}^*, \mathbf{z}^*, W^*)$ from the Benders master problem. The Benders *feasibility* cut is defined as follows:

$$\sum_{i\in N_1(\mathbf{z}^*)} (1 - z_i) + \sum_{i\in N_0(\mathbf{z}^*)} z_i \geq 1. \tag{39}$$

We add the above constraint to the problem (BMP). Note that the current Benders master solution $\mathbf{z}^*$ does not satisfy the constraint (39). One may concern that the above cut is too weak because it seems only to exclude a single feasible solution $\mathbf{z}^*$. However, the cut actually excludes all feasible routes visiting nodes in $N_1(\mathbf{z}^*)$ regardless of the visiting sequence.

4.2.2. Case 2: (BSP) Has an Optimal Solution.

Let $\tilde{W}_{\mathbf{z}^*}$ denote the objective value of the problem (BSP) with a given $\mathbf{z}^*$. We note that $W^* \leq \tilde{W}_{\mathbf{z}^*}$ because W^* is a lower bound of the total waiting time for the master solution $\mathbf{z}^*$. If $W^* = \tilde{W}$, no Benders cut is found, and the current master solution $(\mathbf{x}^*, \mathbf{z}^*, W^*)$ combined with the optimal solution of (BSP) is a feasible drone-truck routing solution. If $W^* < \tilde{W}_{\mathbf{z}^*}$, the Benders *optimality* cut must be added to the Benders master problem. Unlike the Benders feasibility cut, building the Benders optimality cut is a little more involved.

We consider any feasible solutions to the Benders master problem that make the problem (BSP) feasible. The feasible solutions of the Benders master problem can be partitioned with the length of routes, which is defined as the number of visits, that is, $\sum_{i \in N} z_i^*$. Let Z^p denote a set of all feasible solutions of the Benders master problem (BMP) with length p and giving feasible (BSP) solutions. Each solution $\mathbf{z} \in Z^p$ would give a different minimum waiting time $\tilde{W}_{\mathbf{z}}$ for drone delivery, that is, the objective value of the problem (BSP) with the given $\mathbf{z}$. We assume that, in the preprocessing stage, we obtained a *lower bound* of waiting times $\underline{W}^p$ for all solutions in Z^p, that is, for any $p \in \{\underline{p}, \ldots, \overline{p}\}$, $\underline{W}^p$ satisfies

$$\underline{W}^p \leq \tilde{W}_{\mathbf{z}}, \quad \forall\, \mathbf{z} \in Z^p, \tag{40}$$

where $\underline{p}$ and $\overline{p}$ are the lower and upper bounds of the number of truck visits (see Section 4.3).

Now we define the Benders optimality cut as follows.

$$W \geq \tilde{W}_{\mathbf{z}^*} - \Omega(\mathbf{z}^*) \sum_{i \in N_0(\mathbf{z}^*)} z_i, \tag{41}$$

where

$$\Omega(\mathbf{z}^*) = \begin{cases} \max_{q=p^*+1,\ldots,\overline{p}} \left\{ \dfrac{\tilde{W}_{\mathbf{z}^*} - \underline{W}^q}{q - p^*} \right\}, & \text{if } p^* < \overline{p} \\ \tilde{W}_{\mathbf{z}^*} - \underline{W}^{p^*}, & \text{otherwise} \end{cases}$$

and $p^* = \sum_{i \in N} z_i^*$.

Theorem 1. *The constraint (41) is a valid Benders optimality cut, if the following condition is satisfied:*

$$(C) \quad \underline{W}^p \geq \underline{W}^q, \quad \forall\, \underline{p} \leq p < q \leq \overline{p}.$$

Proof. Let $\beta_{\mathbf{z}^*}(\mathbf{z})$ denote the right-hand-side of the constraint (41), that is,

$$\beta_{\mathbf{z}^*}(\mathbf{z}) = \tilde{W}_{\mathbf{z}^*} - \Omega(\mathbf{z}^*) \sum_{i \in N_0(\mathbf{z}^*)} z_i,$$

where $\mathbf{z}^*$ is the Benders master solution defined the Benders cut, and $\mathbf{z}$ is a feasible solution of the Benders master problem (BMP). As shown by Hooker and Ottosson (2003), if the Benders optimality cut satisfies the following conditions, the Bender branch-and-cut method finds an optimal solution.

(B1) $\tilde{W}_{\mathbf{z}} \geq \beta_{\mathbf{z}^*}(\mathbf{z})$ for all feasible solution $\mathbf{z}$ of the problem (BMP)

(B2) $\tilde{W}_{\mathbf{z}^*} = \beta_{\mathbf{z}^*}(\mathbf{z}^*)$

The above conditions state that $\beta_{\mathbf{z}^*}(\mathbf{z})$ should be a (tight) lower bound function of $\mathbf{z}$. For the condition (B1), assume that there is $\hat{\mathbf{z}}$, which is a feasible solution of (BMP), such that $\tilde{W}_{\hat{\mathbf{z}}} < \beta_{\mathbf{z}^*}(\hat{\mathbf{z}})$. Let $p^* = \sum_{i \in N} z_i^*$, $\hat{p} = \sum_{i \in N} \hat{z}_i$ and $r = \sum_{i \in N_0(\mathbf{z}^*)} \hat{z}_i$. It is easy to see that $\hat{p} \leq p^* + r$ holds. We have three cases:

1. $r = 0$: This case implies $N_0(\mathbf{z}^*) \subseteq N_0(\hat{\mathbf{z}})$, which is equivalent to $N_1(\hat{\mathbf{z}}) \subseteq N_1(\mathbf{z}^*)$. Due to the constraints (35) and (36), the set of feasible solutions of (BSP) with $\mathbf{z}^*$ includes the set of feasible solutions of (BSP) with $\hat{\mathbf{z}}$, which implies that $\tilde{W}_{\mathbf{z}^*} \leq \tilde{W}_{\hat{\mathbf{z}}}$, by the assumption, we also have $\tilde{W}_{\hat{\mathbf{z}}} < \beta_{\mathbf{z}^*}(\hat{\mathbf{z}}) = \tilde{W}_{\mathbf{z}^*}$, which gives a contradiction.

2. $r > 0$ and $p^* + r \leq \overline{p}$: We have

$$\begin{aligned} \beta_{\mathbf{z}^*}(\hat{\mathbf{z}}) &= \tilde{W}_{\mathbf{z}^*} - \Omega(\mathbf{z}^*) r \\ &\leq \tilde{W}_{\mathbf{z}^*} - \frac{\tilde{W}_{\mathbf{z}^*} - \underline{W}^{p^*+r}}{r} r \quad \left(\because \Omega(\mathbf{z}^*) \geq \frac{\tilde{W}_{\mathbf{z}^*} - \underline{W}^{p^*+r}}{r} \right) \\ &= \underline{W}^{p^*+r} \\ &\leq \underline{W}^{\hat{p}} \qquad (\because \hat{p} \leq p^* + r \text{ and } (C)) \\ &\leq \tilde{W}_{\hat{\mathbf{z}}} \quad (\because (40)), \end{aligned}$$

which gives a contradiction.

3. $r > 0$ and $p^* + r > \overline{p}$: Let $r' = \overline{p} - p^*$. If $r' = 0$, we have

$$\begin{aligned} \beta_{\mathbf{z}^*}(\hat{\mathbf{z}}) &= \tilde{W}_{\mathbf{z}^*} - \Omega(\mathbf{z}^*) r \\ &= \tilde{W}_{\mathbf{z}^*} - (\tilde{W}_{\mathbf{z}^*} - \underline{W}^{\overline{p}}) r \ (\because \overline{p} = p^*) \\ &\leq \underline{W}^{\overline{p}} \quad (\because r \geq 1) \\ &\leq \underline{W}^{\hat{p}} \quad (\because \hat{p} \leq \overline{p} \text{ and } (C)) \\ &\leq \tilde{W}_{\hat{z}} \quad (\because (40)) \end{aligned}$$

which gives a contradiction. If $r' > 0$, we have

$$\begin{aligned} \beta_{\mathbf{z}^*}(\hat{\mathbf{z}}) &= \tilde{W}_{\mathbf{z}^*} - \Omega(\mathbf{z}^*) r \\ &\leq \tilde{W}_{\mathbf{z}^*} - \Omega(\mathbf{z}^*) r' \quad (\because r' < r) \\ &\leq \tilde{W}_{\mathbf{z}^*} - \frac{\tilde{W}_{\mathbf{z}^*} - \underline{W}^{\overline{p}}}{r'} r' \quad \left(\because \Omega(\mathbf{z}^*) \geq \frac{\tilde{W}_{\mathbf{z}^*} - \underline{W}^{\overline{p}}}{r'} \right) \\ &= \underline{W}^{\overline{p}} \\ &\leq \underline{W}^{\hat{p}} \quad (\because \hat{p} \leq \overline{p} \text{ and } (C)) \\ &\leq \tilde{W}_{\hat{\mathbf{z}}} \quad (\because (40)), \end{aligned}$$

which gives a contradiction.

Therefore, the condition (B1) is satisfied by the Benders cut (41). Moreover, the condition (B2) is obvious, which completes the proof. □

The idea of Theorem 1 can be understood with a numerical example. Assume a master solution $\mathbf{z}^* = (1, 1, 1, 0, \ldots, 0)$ with $\underline{p} = 2 \leq p^* = 3 \leq \overline{p} = 5$, which means the length of the truck route can be any value between 2 and 5. Benders subproblem's optimal objective value would give the minimum waiting time $\tilde{W}_{\mathbf{z}^*} = 100$ while $\underline{W}^4 = 90$ and $\underline{W}^5 = 70$ are already known in the preprocessing stage. The right-hand side of the Benders cut (41), defined by $\beta_{\mathbf{z}^*}(\mathbf{z})$, must not be greater than the minimum waiting time $\tilde{W}_{\mathbf{z}}$, due to condition (B1), for any truck route $\mathbf{z}$. With $\Omega(\mathbf{z}^*) = \max\{10, 15\} = 15$, we have the right-hand side of the Benders cut $\beta_{\mathbf{z}^*}(\mathbf{z}) = 100 - 15\sum_{i\in N\setminus\{1,2,3\}} z^i$. Consider, for example, any truck route $\mathbf{z}$ with length four. It implies that $\sum_{i\in N\setminus\{1,2,3\}} z^i \geq 1$, which makes $\beta_{\mathbf{z}^*}(\mathbf{z}) \leq 85$. Therefore, $\beta_{\mathbf{z}^*}(\mathbf{z})$ cannot be greater than the minimum waiting time $\tilde{W}_{\mathbf{z}}$ for the master solution $\mathbf{z}$, because $\beta_{\mathbf{z}^*}(\mathbf{z}) \leq 85 < \underline{W}^4 = 90 \leq \tilde{W}_{\mathbf{z}}$. We can easily show the condition holds for any other truck routes if the truck route's length is within 2 and 5.

To construct a Benders optimality cut (41), it is required to know $\underline{W}^p$ for all $p \in \{\underline{p}, \ldots, \overline{p}\}$. Because $\underline{W}^p$ is a lower bound of the waiting times, trivially setting $\underline{W}^p = 0$ for all $p \in \{\underline{p}, \ldots, \overline{p}\}$ would work because the condition (C) is satisfied. However, using poor lower bounds may produce weak Benders optimality cuts, because it makes Ω bigger. In the preprocessing step explained in the following section, we present a procedure that gives tighter lower bounds.

4.3. Preprocessing

The purpose of the preprocessing is threefold: (1) to accelerate the branch-and-bound search by providing a good incumbent solution, (2) to limit the length of the truck route so that the unnecessary search is avoided, and (3) to provide the lower bound of the total waiting times $\underline{W}^p$ for the Benders optimality cuts.

4.3.1. Primal Heuristic. The performance of the proposed algorithm relies on the efficiency of the branch-and-bound of the Benders master problem (BMP). Here, we present a primal heuristic algorithm for providing an incumbent solution. First, for a given p, consider the following problem:

$$(\text{LW}-p)\ \min \sum_{i\in N_s} w_i \tag{42}$$

$$\text{s.t.} \sum_{i\in N_s}\sum_{j\in N: j\neq i} b^l_{ij} h^l_{ij} \leq B^l, \quad \forall\, l \in L, \tag{43}$$

$$w_i \geq \sum_{j\in N: j\neq i} \tau^l_{ij} h^l_{ij}, \quad \forall\, i \in N_s, l \in L, \tag{44}$$

$$y_{ij} = \sum_{l\in L} h^l_{ij}, \quad \forall\, (i,j) \in A, \tag{45}$$

$$z_i \geq y_{ij} \quad \forall\, i \in N_s, j \in N, i \neq j, \tag{46}$$

$$\sum_{i\in N} z_i = p, \tag{47}$$

$$\sum_{i\in N_s: i\neq j} y_{ij} + z_j \geq 1, \quad \forall\, j \in N, \tag{48}$$

$$z_s = 1, \tag{49}$$

$$z_i \in \{0,1\}, \quad \forall\, i \in N_s, \tag{50}$$

$$y_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, \tag{51}$$

$$h^l_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, l \in L. \tag{52}$$

Let $\underline{W}^p$ denote the optimal objective value of the problem (LW-p), which is a classical p-median problem with side constraints. The problem (LW-p) finds the minimum total waiting times when there are $p + 1$ potential waiting nodes including the depot. In other words, $\underline{W}^p$ is a lower bound of the total waiting times when the truck visits p customers.

The primal heuristic is based on the idea that the problem (LW-p) is relatively easy to solve when p is small enough, and a feasible drone-truck routing solution can be obtained by solving a small TSP with p-median nodes. Therefore, it can be seen as a decomposition-based heuristic. Algorithm 2 describes the primal heuristic algorithm. The primal heuristic solution's objective value $\hat{Z}$ is fed to the Benders master problem (BMP) as an initial incumbent solution. The TSP in line 7 of Algorithm 2 is solved by a formulation similar to the (BMP) with the branch-and-cut and the GCS separation.

Algorithm 2 also returns $\underline{p}$, which represents the minimum possible length of the truck route. Notice that any truck routes with length $p < \underline{p}$ are infeasible because there are no feasible drone delivery method solutions for p.

Algorithm 2 (Primal Heuristic)

```
 1: $P \leftarrow \{1, \ldots, |N| - 1\}$
 2: for $p \in P$ do
 3:   Solve (LW-$p$)
 4:   if (LW-$p$) is feasible then
 5:     $\underline{W}^p \leftarrow$ objective value of (LW-$p$)
 6:     $\mathbf{z}^p \leftarrow$ solution of (LW-$p$)
 7:     Solve TSP with $\mathbf{z}^p$. Let $T^p$ be the objective
        value.
 8:     $\hat{Z} \leftarrow T^p + \underline{W}^p$      ▷ Primal heuristic solution
 9:     $\underline{p} \leftarrow p$                    ▷ Minimum truck route length
10:     Break
11:   end if
12: end for
13: return $\hat{Z}, \underline{p}$
```

4.3.2. Truck Route Length Bounding. The next preprocessing step is bounding the length of the truck route by using the incumbent value $\hat{Z}$. For a given p, we consider the problem (P-p) given as follows.

$$\begin{aligned} &(\text{P}-p)\ \min\ (1) \\ &\text{s.t.}\ (2)-(12), \\ &\sum_{i\in N}\sum_{j\in N_t: j\neq i} x_{ij} = p. \end{aligned} \tag{53}$$

The problem (P-p) is the HDTRP with restricting the length of the truck route to p. Solving the problem

(P-p) is still very hard, so we solve the root node relaxation with Cplex. The state-of-the-art solvers like Cplex can considerably enhance the root relaxation bound by adding various valid inequalities. Therefore, the objective value, denoted by $\underline{Z}^p$ of the root relaxation of the problem (P-p), gives a lower bound of the HDTRP when the length of the truck route is p. This implies that it is unnecessary to consider any truck routes of length p if $\underline{Z}^p \geq \hat{Z}$.

Algorithm 3 (Truck Route Length Bounding)

```
Require: $\hat{Z}, \underline{p}$ from Algorithm 2
$p \leftarrow \underline{p} + \text{round}((|N| - \underline{p})/3)$
loop
    Solve the root relaxation of (P-$p$)
    $\underline{Z}^p \leftarrow$ objective value of the root relaxation of (P-$p$)
    if $\underline{Z}^p \leq \hat{Z}$ then
        Break
    else
        $p \leftarrow \underline{p} + \text{round}((p - \underline{p} + 1)/3)$
    end if
end loop
$\hat{p} \leftarrow p$
for $p \in \{\hat{p}+1, \hat{p}+2, \ldots, |N|-1\}$ do
    Solve the root relaxation of (P-$p$)
    $\underline{Z}^p \leftarrow$ objective value of the root relaxation of (P-$p$)
    if $\underline{Z}^p > \hat{Z}$ then
        $\overline{p} \leftarrow p - 1$ ▷ Maximum truck route length
        Break
    end if
end for
return $\overline{p}$
```

Algorithm 3 returns the upper bound $\overline{p}$ of the truck route length, while the lower bound $\underline{p}$ is obtained from Algorithm 2. Algorithm 3 first finds $\hat{p}$ such that $\underline{Z}^{\hat{p}} \leq \hat{Z}$ by the bisectional search. Then, we repeatedly increase p by 1 until the upper bound of truck length $\overline{p}$ is found. We add the following constraint to the problem (BMP)

$$\underline{p} \leq \sum_{i\in N} \sum_{j\in N_t: j\neq i} x_{ij} \leq \overline{p}, \qquad (54)$$

which limits the length of the truck route between $\underline{p}$ and $\overline{p}$.

4.3.3. Calculation of $\underline{W}^p$. Recall that the Benders optimality cuts are defined with lower bounds of the waiting times $\underline{W}^p$ for all $p \in \{\underline{p}, \ldots, \overline{p}\}$. Obviously, the optimal solution of the problem (LW-p) would give a tightest lower bound for any p. However, solving the problem (LW-p) to optimality, for all $p \in \{\underline{p}, \ldots, \overline{p}\}$, might take too much time. By noticing that any lower bounds can be used as shown in Theorem 1, we simply solve the root node relaxation of the problem (LW-p) and set $\underline{W}^p$ as the objective value of the root relaxation for all $p \in \{\underline{p}, \ldots, \overline{p}\}$. Then, it is easy to show that the lower bounds of waiting times $\underline{W}^p$ satisfy the following.

Lemma 1. *Consider $\underline{W}^p$, which is the objective value of the root relaxation of the problem (LW-p). Then, $\underline{W}^p \geq \underline{W}^q$ holds, for all $\underline{p} \leq p < q \leq \overline{p}$.*

Proof. Let $(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \hat{\mathbf{h}})$ be a solution to the root relaxation of the problem (LW-p). Due to the constraint (47), $\sum_{i\in N} \hat{z}_i = p$ holds. Because $p < q \leq \overline{p}$, it is clear that there exists $\tilde{\mathbf{z}}$ such that $\sum_{i\in N} \tilde{z}_i = q$ and $\hat{z}_i \leq \tilde{z}_i$ for all $i \in N$. The solution $(\tilde{\mathbf{z}}, \hat{\mathbf{y}}, \hat{\mathbf{h}})$ should be feasible to (LW-q), which implies that $\underline{W}^p \geq \underline{W}^q$. □

5. Computational Study

The algorithm was implemented by Python 3.7, and Cplex 12.10 was used for solving the mathematical formulations. All experiments were conducted on a Linux machine equipped with Intel i9-9900KS 5GHz CPU and 64GB RAM. Unless specified, the Cplex uses the default parameters in our experiments.

The GCS separation procedure shown in Algorithm 1 enumerates all strongly connected components on the support graph, for which we used a faster implementation from Networkit (Staudt et al. 2016). The branch-and-cut algorithm was implemented with callback interfaces of Cplex. Specifically, CutGenCallback, which is called at every node, executes the GCS separation (Algorithm 1), while LazyCutGenCallback, which is called whenever an integer solution is found, performs both the GCS separation and the Benders separation (Section 4.2).

5.1. Instances

We used three sets of problems for computational experiments. The first set is randomly generated instances from real-life use cases, and the last two are benchmark problems widely used in the vehicle routing literature.

UPS (refer to the online appendix): These are randomly generated problems from two real-life use cases. There are two types of problems (CVS and WM) depending on the areas the problem was generated from. The sizes of problems are 20, 30, 40, and 50 nodes, and each size contains 10 instances. Therefore, there are total 80 instances (= 2 areas × 4 sizes × 10 instances). The problem generation method is explained in the online appendix.

Solomon (Solomon 1987): These problems were originally made for the VRP with time windows, in which there are three types of problems depending on generation methods: randomly generated (R), clustered (C), and semiclustered (RC) problems. There are about 20 problems for each type. However, these problems for the same type have the same node locations, while the time windows are different. Therefore, the first problem of each type was used. We used 25 and 50 node problems for our computational experiments.

Augerat (Augerat 1995): These problems were originally proposed for the capacitated VRP. There are three types of problems (A, B, and P) depending on the generation methods. There are problems with the same node locations but with a different number of vehicles. Because our study assumes a single vehicle, we ignored the problems that have a different number of vehicles with the same node positions. We included any problems having no more than 50 nodes for the experiments.

5.2. Parameter Setting

The computational experiments require reasonable parameters for the truck and drones. We first present "standard" drone parameters for each problem set, which define the speed, flight duration, and service time of typical parcel drones. Then, to take the heterogeneousness of drones account, we introduce a new parameter α to represent how much the corresponding drone is faster than the standard one. Unfortunately, faster drones consume more energy, which yields shorter flight durations. The relation between the drone's power drain and speed is governed by the equation (A) (see online appendix) (Liu et al. 2019), which implies that the maximum flight duration, say D, of a drone with speed parameter α is reduced to D/α assuming that the battery capacity is the same. Let $\hat{b}_{ij}$, $\hat{t}_{ij}$ and $\hat{s}$ denote the required total energy consumption, the flight time of the standard drone from i to j, and the standard service time, respectively. Without loss of generality, we can treat the battery constraints (8) like duration constraints if we redefine $\hat{b}_{ij}$ and B^l as the total required delivery time launched at i to customer j and the maximum flight duration time of drone l, respectively. Therefore, we let

$$B^l \leftarrow (\text{the maximum duration time}), \quad \forall\, l \in L, \tag{55}$$

$$\hat{b}_{ij} \leftarrow 2\hat{t}_{ij} + \hat{s}, \quad \forall\, i \in N_s, j \in N, i \neq j. \tag{56}$$

Then, we set

$$b^l_{ij} \leftarrow \alpha_l \hat{b}_{ij}, \quad \forall\, i \in N_s, j \in N, l \in L, i \neq j, \tag{57}$$

$$\tau^l_{ij} \leftarrow 2\hat{t}_{ij}/\alpha_l + \hat{s}, \quad \forall\, i \in N_s, j \in N, l \in L, i \neq j, \tag{58}$$

where α_l is the speed parameter for the drone $l \in L$.

For the sake of simplicity, we used the same drone service time for all drones, that is, $s^l_i = \hat{s}$ for all $i \in N$ and $l \in L$. The fleet of drones is represented by a tuple $[\alpha_l]_{l\in L}$. For example, a drone fleet with speed parameter tuple $[1, 0.5]$ consists of a standard drone and a slow but long-flight-duration drone.

We note that a more sophisticated scheme for preparing the above data are also viable. For example, the round-trip may have asymmetric flight time due to the release of the demand. Nonetheless, our algorithm will work as long as the required data are given.

5.2.1. Parameter Setting for UPS Problems.

The necessary parameters for the UPS problems are calculated by the following values.

- Truck speed: 8 m/sec (or 28.8 km/hour)
- Truck service time: 60 seconds (or 1 minute)
- (Standard) Drone flight duration: 1,800 seconds (or 30 minutes)
- (Standard) Drone speed: 10 m/sec (or 36 km/hour)
- (Standard) Drone service time: 30 seconds

The detailed rationale behind these values is presented in the online appendix.

5.2.2. Parameter Setting for Solomon and Augerat Problems.

All problems were generated by placing nodes on a square plain, and the positions of nodes are given as two integers in $[0, 99]$. Because the Solomon and Augerat problems are without any explicit units, we determine the following parameters to have a similar relative scale to the UPS problems' parameters.

- Truck speed: 1
- Truck service time: 10
- (Standard) Drone flight duration: 100
- (Standard) Drone speed: 2.5
- (Standard) Drone service time: 5

The travel time of the truck t^v_{ij} between two nodes i and j are defined as the Euclidean distance between them, that is, the truck speed is 1. Internally, we multiplied all numbers by 100 and rounded to the nearest integer to avoid any numerical instability.

5.3. Performance Comparison of Benders Approach

Tables 2–4 compare the computational results of the proposed algorithm with the formulation (P) solved by Cplex. All times in the tables are in seconds. The asterisk (*) means the algorithm could not solve the problem within the 1-hour time limit. The number of branch-and-bound nodes is shown in the column "BnB". The heading "GAP" is defined as $\frac{\text{best incumbent value - best lower bound}}{\text{best incumbent value}} \times 100$. Having a zero GAP means the algorithm found and proved the optimal solution. The heading "Time ratio" represents the ratio of computational times between Cplex and the proposed algorithm. If either of both algorithms could not solve the problem, the time ratio cannot be calculated. In these cases, we put either ">" or "<" at the time ratios to represent the values are either lower bound or upper bound. The time ratios indicate how much time our algorithm is faster than Cplex. The heading “Drone ratio” shows how many customers are served by the drones. The use of drones depends on drone parameters. For these experiments, we used a fixed combination of parameters $L = \{0, 1\}$ and $\alpha = [1, 1.5]$, which implies that (1) there are two drones; (2) one drone is standard; and (3) the other drone is faster but has a shorter flight duration.

The results clearly show that the proposed algorithm outperforms Cplex with the formulation (P). Note that when our algorithm could not solve the problems within the time limit, our algorithm’s GAP values are much smaller for most of the cases. It is known that the GAP is a good indicator of the solvability of the problems. Our algorithm requires much less branch-and-bound nodes to prove the optimality unless the problem sizes are too small (e.g., CVS 20 node problems). Overall, the Benders approach performs better for the WM problems, while the Cplex with the formulation (P) performs comparably for the Augerat A-type problems. We found that the Augerat A-type problems have less clustered customer nodes and yield low drone ratios, which implies that these problems result in long truck routes. Because we used fixed drone parameters for all problems, the number of drone deliveries (Drone ratio) varies with the

Table 2. Computational Results for UPS (CVS) Problems

		Cplex				Benders					
Problem	$\lvert N\rvert$	Time	BnB	GAP (%)	Obj.	Time	BnB	GAP (%)	Obj.	Time ratio	Drone ratio (%)
CVS-21-1	20	2.4	3,515	0.0	1,645.0	3.9	143	0.0	1,645.0	0.61	90.0
CVS-21-2	20	8.1	17,864	0.0	1,730.0	2.2	127	0.0	1,730.0	3.74	95.0
CVS-21-3	20	1.6	2,016	0.0	1,635.0	2.5	90	0.0	1,635.0	0.64	95.0
CVS-21-4	20	2.0	2,420	0.0	1,718.0	2.2	87	0.0	1,718.0	0.89	95.0
CVS-21-5	20	0.2	103	0.0	1,650.0	2.2	56	0.0	1,650.0	0.09	95.0
CVS-21-6	20	3.0	5,849	0.0	1,694.0	1.9	251	0.0	1,694.0	1.62	95.0
CVS-21-7	20	3.7	4,729	0.0	1,892.0	3.3	208	0.0	1,892.0	1.11	90.0
CVS-21-8	20	4.1	9,986	0.0	1,632.0	2.2	241	0.0	1,632.0	1.88	90.0
CVS-21-9	20	6.1	14,807	0.0	1,750.0	2.2	216	0.0	1,750.0	2.73	90.0
CVS-21-10	20	1.9	2,322	0.0	1,684.0	2.8	99	0.0	1,684.0	0.68	95.0
CVS-31-1	30	32.8	23,277	0.0	2,194.0	19.0	547	0.0	2,194.0	1.73	90.0
CVS-31-2	30	278.6	203,898	0.0	2,372.0	23.0	1,978	0.0	2,372.0	12.10	86.7
CVS-31-3	30	783.4	481,974	0.0	2,273.0	31.1	2,201	0.0	2,273.0	25.21	90.0
CVS-31-4	30	179.7	154,087	0.0	2,224.0	19.2	1,335	0.0	2,224.0	9.34	86.7
CVS-31-5	30	61.5	58,843	0.0	2,277.0	25.0	1,079	0.0	2,277.0	2.46	86.7
CVS-31-6	30	814.9	726,281	0.0	2,171.0	28.2	2,214	0.0	2,171.0	28.88	90.0
CVS-31-7	30	1,450.1	752,939	0.0	2,593.0	41.0	3,174	0.0	2,593.0	35.34	86.7
CVS-31-8	30	71.6	49,105	0.0	2,076.0	11.1	708	0.0	2,076.0	6.44	90.0
CVS-31-9	30	334.5	215,476	0.0	2,292.0	22.2	2,601	0.0	2,292.0	15.05	86.7
CVS-31-10	30	509.1	361,819	0.0	2,220.0	24.0	2,890	0.0	2,220.0	21.24	86.7
CVS-41-1	40	3,600.0*	877,906	10.8	2,915.0	196.0	11,573	0.0	2,908.0	>18.37	82.5
CVS-41-2	40	3,600.0*	1,173,164	7.2	2,903.0	138.8	9,265	0.0	2,880.0	>25.95	82.5
CVS-41-3	40	3,600.0*	1,282,779	5.1	2,897.0	207.1	17,619	0.0	2,873.0	>17.39	82.5
CVS-41-4	40	3,600.0*	1,532,249	2.7	2,905.0	88.4	8,747	0.0	2,905.0	>40.72	82.5
CVS-41-5	40	890.6	480,221	0.0	2,846.0	62.6	4,895	0.0	2,846.0	14.22	82.5
CVS-41-6	40	2,498.3	1,209,135	0.0	2,771.0	89.4	8,721	0.0	2,771.0	27.95	82.5
CVS-41-7	40	3,600.0*	1,494,612	12.4	3,171.0	336.3	37,781	0.0	3,099.0	>10.71	80.0
CVS-41-8	40	2,752.9	1,449,073	0.0	2,757.0	99.7	9,622	0.0	2,757.0	27.61	85.0
CVS-41-9	40	3,040.6	1,329,313	0.0	2,748.0	101.4	8,358	0.0	2,748.0	29.98	85.0
CVS-41-10	40	3,600.0*	1,537,211	6.6	2,722.0	234.0	15,776	0.0	2,721.0	>15.39	85.0
CVS-51-1	50	3,600.0*	1,069,426	17.9	3,737.0	1,073.8	91,543	0.0	3,507.0	>3.35	82.0
CVS-51-2	50	3,600.0*	1,167,387	15.8	3,495.0	579.9	46,865	0.0	3,433.0	>6.21	78.0
CVS-51-3	50	3,600.0*	824,282	7.1	3,372.0	387.3	23,592	0.0	3,372.0	>9.30	80.0
CVS-51-4	50	3,600.0*	1,007,005	6.0	3,480.0	545.9	46,379	0.0	3,480.0	>6.60	80.0
CVS-51-5	50	3,600.0*	1,066,532	3.1	3,474.0	916.0	42,370	0.0	3,474.0	>3.93	82.0
CVS-51-6	50	3,600.0*	1,277,155	0.7	3,326.0	401.5	23,125	0.0	3,326.0	>8.97	80.0
CVS-51-7	50	3,600.0*	1,488,806	15.1	3,714.0	566.6	35,780	0.0	3,540.0	>6.36	78.0
CVS-51-8	50	3,600.0*	1,122,917	11.8	3,449.0	2,354.4	175,950	0.0	3,445.0	>1.53	82.0
CVS-51-9	50	3,600.0*	1,139,466	13.9	3,327.0	3,600.0*	100,064	1.2	3,274.0	–	82.0
CVS-51-10	50	3,600.0*	888,396	9.3	3,342.0	1,764.8	129,043	0.0	3,331.0	>2.04	80.0

Note. Drone speed parameters: $\alpha = [1, 1.5]$.
*Time limit (3,600 seconds) reached.

Table 3. Computational Results for UPS (WM) Problems

		Cplex				Benders					
Problem	$\lvert N \rvert$	Time	BnB	GAP (%)	Obj.	Time	BnB	GAP (%)	Obj.	Time ratio	Drone ratio (%)
WM-21-1	20	33.3	102,545	0.0	1,619.0	3.6	852	0.0	1,619.0	9.25	90.0
WM-21-2	20	58.9	112,994	0.0	1,628.0	5.2	677	0.0	1,628.0	11.21	90.0
WM-21-3	20	24.0	48,820	0.0	1,599.0	4.0	321	0.0	1,599.0	5.94	90.0
WM-21-4	20	80.6	124,604	0.0	1,793.0	4.2	502	0.0	1,793.0	19.14	90.0
WM-21-5	20	56.0	93,066	0.0	1,539.0	6.6	939	0.0	1,539.0	8.49	90.0
WM-21-6	20	17.6	26,695	0.0	1,784.0	4.0	394	0.0	1,784.0	4.36	90.0
WM-21-7	20	33.2	54,484	0.0	1,528.0	6.4	838	0.0	1,528.0	5.21	85.0
WM-21-8	20	92.4	179,947	0.0	1,822.0	4.5	642	0.0	1,822.0	20.36	90.0
WM-21-9	20	16.0	20,969	0.0	1,610.0	4.7	330	0.0	1,610.0	3.45	90.0
WM-21-10	20	38.8	71,596	0.0	1,940.0	6.3	608	0.0	1,940.0	6.14	90.0
WM-31-1	30	3,600.0*	1,700,961	6.5	2,084.0	56.5	9,697	0.0	2,084.0	>63.81	86.7
WM-31-2	30	3,600.0*	2,074,603	6.4	1,978.0	27.2	4,431	0.0	1,978.0	>132.59	90.0
WM-31-3	30	3,600.0*	2,058,904	1.8	2,159.0	32.0	3,624	0.0	2,149.0	>112.36	90.0
WM-31-4	30	3,600.0*	3,095,415	0.7	2,242.0	18.0	2,019	0.0	2,232.0	>200.22	90.0
WM-31-5	30	2,846.3	2,931,569	0.0	1,934.0	14.9	1,986	0.0	1,934.0	191.03	90.0
WM-31-6	30	3,600.0*	1,912,302	5.2	2,290.0	20.0	3,274	0.0	2,284.0	>179.95	86.7
WM-31-7	30	3,600.0*	2,083,167	8.4	1,966.0	34.6	7,976	0.0	1,966.0	>104.08	90.0
WM-31-8	30	3,600.0*	2,112,336	5.8	2,420.0	20.8	1,994	0.0	2,376.0	>173.02	86.7
WM-31-9	30	3,600.0	2,145,254	5.8	2,222.0	23.5	3,863	0.0	2,220.0	>153.06	86.7
WM-31-10	30	3,384.6	2,918,192	0.0	2,260.0	29.7	5,091	0.0	2,260.0	113.92	90.0
WM-41-1	40	3,600.0*	1,260,063	16.5	2,523.0	649.6	29,480	0.0	2,513.0	>5.54	87.5
WM-41-2	40	3,600.0*	1,577,555	16.8	2,483.0	477.5	44,193	0.0	2,467.0	>7.54	87.5
WM-41-3	40	3,600.0*	2,174,522	16.2	2,529.0	290.6	19,267	0.0	2,529.0	>12.39	85.0
WM-41-4	40	3,600.0*	1,633,476	15.5	2,662.0	198.1	17,608	0.0	2,640.0	>18.18	85.0
WM-41-5	40	3,600.0*	1,546,549	10.8	2,508.0	293.2	25,820	0.0	2,508.0	>12.28	87.5
WM-41-6	40	3,600.0*	2,710,142	13.9	2,829.0	230.1	16,657	0.0	2,786.0	>15.65	85.0
WM-41-7	40	3,600.0*	1,777,252	15.2	2,460.0	122.6	8,177	0.0	2,460.0	>29.38	87.5
WM-41-8	40	3,600.0*	1,487,572	17.2	3,053.0	473.3	38,780	0.0	2,983.0	>7.61	80.0
WM-41-9	40	3,600.0*	1,796,366	16.5	2,635.0	625.6	50,536	0.0	2,631.0	>5.76	87.5
WM-41-10	40	3,600.0*	1,400,628	15.6	2,846.0	3,600.0*	71,768	4.5	2,858.0	–	82.5
WM-51-1	50	3,600.0*	1,298,493	16.8	3,318.0	3,366.7	136,163	0.0	3,254.0	>1.07	80.0
WM-51-2	50	3,600.0*	766,769	14.2	3,016.0	3,600.0*	112,216	1.1	2,984.0	–	80.0
WM-51-3	50	3,600.0*	1,363,804	12.7	3,119.0	1,525.9	68,098	0.0	3,098.0	>2.36	82.0
WM-51-4	50	3,600.0*	870,931	23.3	3,167.0	3,600.0*	108,352	0.4	3,016.0	–	84.0
WM-51-5	50	3,600.0*	1,661,038	13.6	3,125.0	585.4	19,300	0.0	3,076.0	>6.15	80.0
WM-51-6	50	3,600.0*	1,362,638	11.5	3,260.0	2,481.1	113,786	0.0	3,245.0	>1.45	80.0
WM-51-7	50	3,600.0*	931,995	16.5	3,055.0	3,600.0*	55,476	8.9	3,012.0	–	80.0
WM-51-8	50	3,600.0*	1,215,962	14.8	3,504.0	1,639.1	95,628	0.0	3,488.0	>2.20	74.0
WM-51-9	50	3,600.0*	1,403,691	22.3	3,363.0	3,600.0*	152,099	7.0	3,329.0	–	80.0
WM-51-10	50	3,600.0*	1,549,617	15.5	3,340.0	3,600.0*	94,855	5.4	3,350.0	–	76.0

Note. Drone speed parameters: $\alpha = [1, 1.5]$.
*Time limit (3,600 seconds) reached.

problems. The problems with clustered nodes (e.g., UPS, C, RC, and B-type) show that the drones serve more customers.

Figure 4 shows the optimal solutions to the selected problems. The solutions to the real-life problems shown in Figure 4, (a) and (b) illustrate typical routes of the truck and drones. The solid black lines represent the routes of the truck. We can see that the fast drone (dashed green line) takes longer flight routes, while the standard drone (dashed orange line) visits customers nearby the truck's waiting node. We also notice that real-life problems tend to have clustered customer nodes. In the last four figures, the numbers indicate which drone is used for the node's delivery. The truck should deploy the drones and wait at the gray nodes. As expected, the use of drones tends to happen in clustered areas. For example, Figure 4, (c) and (f) clearly show that many short distance deliveries are executed by the drones, while the truck travels longer arcs. A particular case shown in Figure 4(f) reveals the importance of cooperation with the truck. Because the depot is located far from the customers, the drones cannot be used at the depot. The truck can bring the drones to the area where the customers are

Table 4. Computational Results for Solomon and Augerat Problems with $|N| \leq 50$

		Cplex				Benders					
Problem	$\|N\|$	Time	BnB	GAP (%)	Obj.	Time	BnB	GAP (%)	Obj.	Time ratio	Drone ratio (%)
C	25	869.1	1,140,026	0.0	214.16	13.2	2,775	0.0	214.16	65.99	80.0
R	25	39.3	49,810	0.0	395.27	8.3	3,280	0.0	395.27	4.72	48.0
RC	25	3,600.0	2,738,053	16.2	315.87	22.6	6,804	0.0	315.87	>158.94	76.0
C	50	3,600.0*	1,427,006	2.6	493.14	3,600.0*	329,536	1.2	537.19	–	48.0
R	50	3,600.0*	886,017	0.9	766.88	3,600.0*	75,657	0.8	766.88	–	30.0
RC	50	3,600.0*	1,225,232	21.3	689.81	3,600.0*	505,866	2.6	675.48	–	48.0
A-n32-k5	31	204.3	180,527	0.0	612.48	53.3	15,173	0.0	612.48	3.83	45.2
A-n33-k5	32	3,600.0	2,213,332	1.5	588.32	3,600.0*	301,073	0.2	588.05	–	43.8
A-n33-k6	32	3,600.0*	2,202,901	1.0	605.64	3,600.0*	60,909	2.5	605.64	–	40.6
A-n34-k5	33	3,600.0*	2,103,173	4.4	628.56	3,600.0*	100,234	1.9	626.00	–	42.4
A-n36-k5	35	3,600.0*	1,384,577	1.3	653.60	3,600.0*	102,951	2.1	653.63	–	40.0
A-n37-k5	36	367.4	312,250	0.0	677.77	2,074.1	297,203	0.0	677.77	0.18	30.6
A-n37-k6	36	3,600.0*	786,353	0.7	684.50	3,600.0*	105,283	1.4	684.50	–	36.1
A-n38-k5	37	2,860.9	1,499,275	0.0	660.87	3,600.0*	319,401	0.8	660.87	<0.79	40.5
A-n39-k5	38	3,600.0*	1,499,328	0.8	735.49	1,121.3	160,941	0.0	735.49	>3.21	31.6
A-n39-k6	38	3,600.0*	817,173	0.3	735.66	3,600.0*	142,194	1.6	735.66	–	31.6
A-n44-k7	43	3,600.0*	1,194,227	2.3	816.70	3,600.0*	264,126	1.2	816.70	–	30.2
A-n45-k6	44	3,600.0*	1,446,936	2.5	829.42	3,600.0*	107,178	1.3	829.42	–	31.8
A-n45-k7	44	3,600.0*	1,103,407	0.8	779.29	3,600.0*	175,900	1.0	779.29	–	36.4
A-n46-k7	45	3,600.0*	886,884	2.7	818.18	3,600.0*	147,413	1.3	816.98	–	35.6
A-n48-k7	47	2,765.3	978,713	0.0	826.69	3,600.0*	147,975	1.1	826.69	<0.77	31.9
B-n31-k5	30	3,600.0	2,438,694	13.5	415.28	86.1	14,681	0.0	413.51	>41.81	80.0
B-n34-k5	33	3,600.0*	3,307,747	17.7	467.94	3,600.0*	209,925	0.6	467.94	–	66.7
B-n35-k5	34	3,600.0*	3,839,401	20.6	535.42	3,600.0*	455,382	0.7	534.87	–	61.8
B-n38-k6	37	3,600.0*	2,411,248	18.9	531.29	3,600.0*	124,281	2.6	528.50	–	59.5
B-n41-k6	40	3,600.0*	2,533,666	18.3	595.10	3,600.0*	276,463	0.4	589.75	–	55.0
B-n43-k6	42	3,600.0*	1,019,860	4.1	556.27	3,600.0*	223,428	0.6	554.54	–	54.8
B-n44-k7	43	3,600.0*	1,411,184	19.5	546.73	3,600.0*	140,930	1.7	543.22	–	55.8
B-n45-k5	44	3,600.0*	1,445,567	21.1	709.92	3,600.0*	22,362	4.7	703.89	–	50.0
B-n45-k6	44	3,600.0*	1,129,300	11.1	568.47	3,600.0*	184,520	1.2	561.19	–	56.8
B-n50-k7	49	3,600.0*	784,053	25.3	686.03	3,600.0*	200,464	1.9	673.97	–	49.0
B-n50-k8	49	3,600.0*	1,133,702	17.7	741.88	3,600.0*	8,326	8.7	764.26	–	36.7
B-n51-k7	50	3,600.0*	801,434	29.3	810.70	3,600.0*	165,238	3.2	751.07	–	48.0
P-n16-k8	15	3.4	8,402	0.0	177.20	1.8	304	0.0	177.20	1.93	73.3
P-n19-k2	18	18.6	42,083	0.0	215.93	4.2	786	0.0	215.93	4.43	72.2
P-n20-k2	19	27.6	54,687	0.0	229.39	5.4	918	0.0	229.39	5.08	73.7
P-n21-k2	20	24.1	54,463	0.0	239.36	3.8	1,118	0.0	239.36	6.34	70.0
P-n22-k2	21	61.4	129,429	0.0	244.66	4.0	1,120	0.0	244.66	15.15	71.4
P-n22-k8	21	1,722.5	2,090,328	0.0	342.09	7.6	2,945	0.0	342.09	225.75	52.4
P-n23-k8	22	368.8	670,690	0.0	251.44	7.8	1,222	0.0	251.44	47.58	72.7
P-n40-k5	39	474.5	248,771	0.0	565.38	2,530.4	160,942	0.0	565.38	0.19	35.9
P-n45-k5	44	2,332.3	639,167	0.0	649.38	3,600.0*	105,666	0.3	649.38	<0.65	34.1
P-n50-k7	49	3,600.0*	645,154	0.9	689.37	3,600.0*	31,871	1.6	689.99	–	34.7
P-n51-k10	50	3,600.0*	1,012,916	1.3	732.94	3,600.0*	47,009	1.8	735.20	–	30.0

Note. Drone speed parameters: $\alpha = [1, 1.5]$.
*Time limit (3,600 seconds) reached.

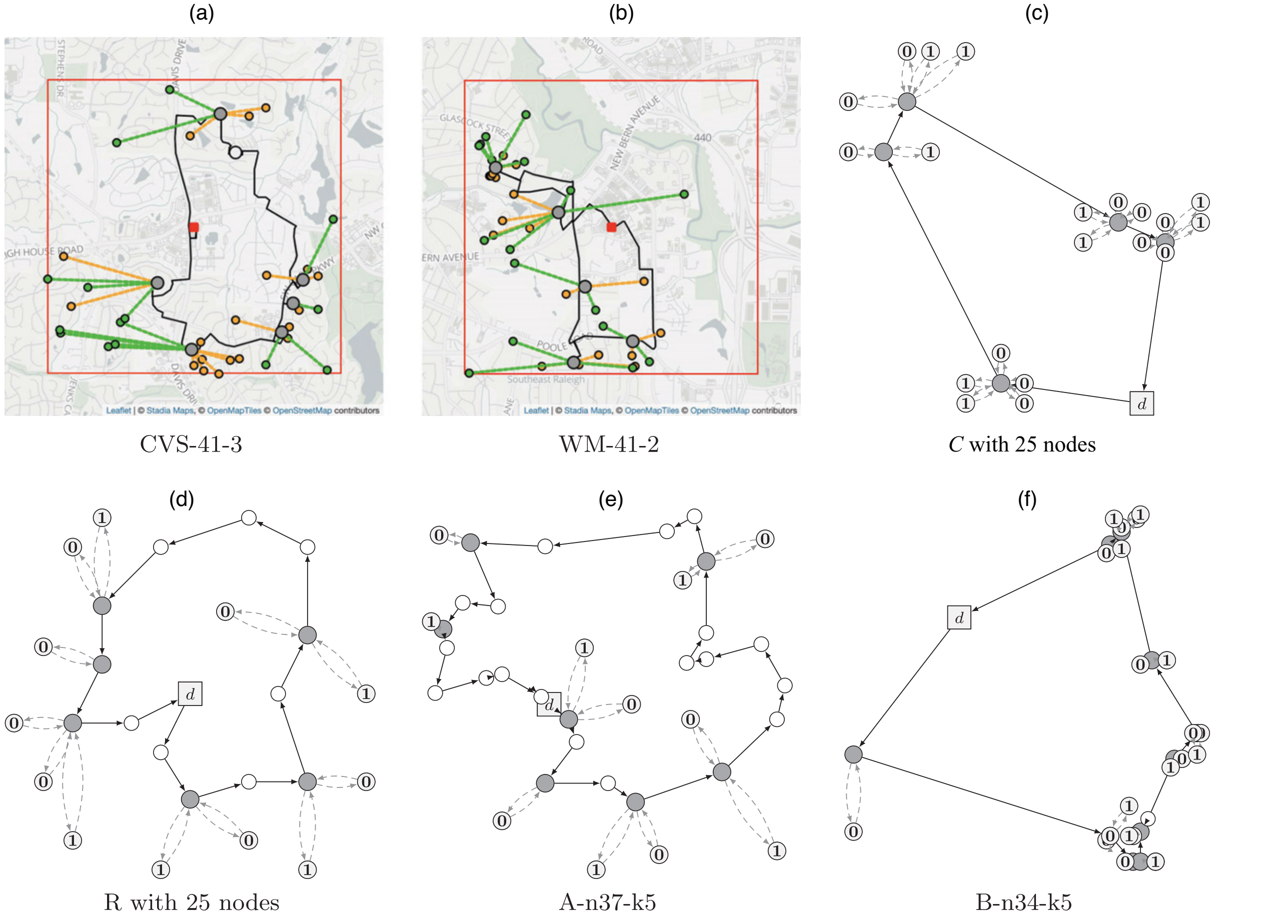
located so that the drones can visit many customers, even with the limited flight time. The heterogeneousness of drones is shown as the asymmetric use of two drones.

The solutions also show the impact of node distributions. In general, when nodes are clustered as Figure 4 (a), (b), (c), and (f), each drone took multiple deliveries at a small number of waiting nodes. On the other hand, the problems R and A-n37-k5 have more evenly distributed nodes, which result in many waiting nodes with shorter waiting times at each waiting node.

5.4. Analysis of Benders Algorithm

Table 5 reports the detailed results of our algorithm for the UPS problems solved within the time limit. The computational time of our algorithm consists of four parts: preprocessing (Section 4.3), solving the Benders master problem (Section 4.1), GCS separation (Section 4.1.1), and solving the Benders subproblem (Section 4.2). Average times spent in each procedure are presented in columns "Prep.", "MasterP", "GCS", and "SubP", respectively. Our algorithm utilizes three types of cuts: the GCS inequalities, the Benders

Figure 4. (Color online) Optimal Solutions to Selected Problems


(a)
(b)
(c)
(d)
(e)
(f)
Leaflet | © Stadia Maps, © OpenMapTiles © OpenStreetMap contributors
Leaflet | © Stadia Maps, © OpenMapTiles © OpenStreetMap contributors
d
d
d
d

CVS-41-3

WM-41-2

C with 25 nodes

R with 25 nodes

A-n37-k5

B-n34-k5

Notes. The first two are UPS problems, in which the standard drone (orange) and the fast drone (green) are used. The next two are Solomon C and R problems. The last two are Augerat problems. A node with a number indicates which drone serves the node. The gray nodes are where the truck waits (waiting nodes). The solid arrows show the route of the truck, while dashed arrows represent the drones' route. Notice that the difference of node distributions over the problems. (a) CVS-41-3. (b) WM-41-2. (c) C with 25 nodes. (d) R with 25 nodes. (e) A-n37-k5. (f) B-n34-k5.

feasibility cuts, and the Benders optimality cuts, whose generated numbers are listed in the columns "GCS", "Feas", and "Opt", respectively. The results of the primal heuristic algorithm are presented in the columns "GAP", and "$\underline{p}, \overline{p}$", which are the gap from the optimal solution, and the bounding of the truck route length, respectively.

It turns out that the most time-consuming part is solving the Benders master problem (BMP). Recall that we solve the Benders master problem once because the Benders cut generation is integrated into the branch-and-bound search. Nonetheless, the number of LP relaxations to solve in the branch-and-bound tree is enormous, as shown in Tables 2–4. Regarding

Table 5. Detailed Results of Benders Algorithm for the UPS Problems

		Time					# Cuts			Primal Heuristic	
Problem	$\|N\|$	Total	Prep.	MasterP	GCS	SubP	GCS	Feas	Opt	GAP	$\underline{p}, \bar{p}$
CVS-20	20	2.54	1.67	0.66	0.07	0.15	80.3	0.0	4.3	6.61	1.0, 6.4
CVS-30	30	24.39	8.97	10.90	2.33	2.18	985.4	0.0	10.7	10.94	2.1, 14.1
CVS-40	40	155.38	21.60	94.41	14.99	24.38	2,659.1	0.1	95.7	14.73	4.1, 21.5
CVS-50	50	954.44	54.77	650.60	102.80	146.27	5,156.3	0.1	461.6	14.73	6.6, 29.4
WM-20	20	4.96	2.32	1.77	0.40	0.47	419.4	0.0	10.5	10.42	1.0, 11.0
WM-30	30	27.72	8.44	14.97	3.64	0.67	2,740.0	0.0	10.8	8.52	2.0, 17.0
WM-40	40	373.40	26.60	287.55	34.48	24.77	8,375.4	0.0	83.8	13.52	3.6, 23.6
WM-50	50	1,919.63	73.50	1,548.44	137.77	159.92	12,271.0	0.0	530.4	21.57	6.2, 33.2

the number of generated cuts, a significant number of the GCS inequalities were used because the separation for the GCS is called at every branch-and-bound node. Note that the computational time for the GCS separation was relatively small, which implies that Algorithm 1 is very fast.

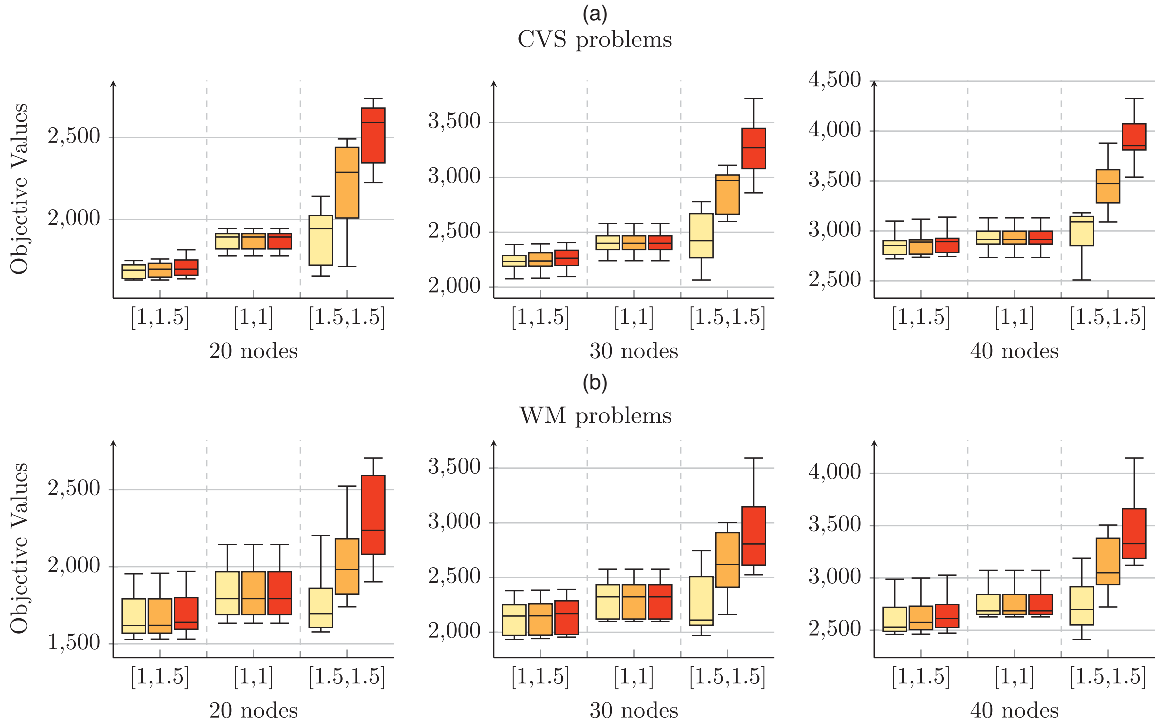
On the other hand, considerable time was spent on solving the Benders subproblem, considering the Benders subproblem is solved only if the linear relaxation solution of the (BMP) is integral, and the Benders subproblem (BSP) is an MIP problem. It is interesting to see that almost no Benders feasibility cuts are needed. Note that the truck route length bounding, see Algorithm 3, ensures that the truck route not being too short. Moreover, the optional constraints (26)–(28) in the problem (BMP) prevent any “absurd” integer feasible solutions of the problem (BMP).

5.5. Benefit of Heterogeneousness

We assume that there are two types of drones: the standard ($\alpha = 1$) and fast ($\alpha = 1.5$). To assert the benefit of the drone types’ heterogeneousness, we consider the cases in which there are some “heavy” demands that cannot be carried by fast drones. Therefore, the heavy demands must be delivered by either truck or the standard drone. Depending on the ratios of heavy demands, we consider three scenarios, in which 10%, 20%, and 30% of total demands are heavy, respectively. The heavy demands are randomly chosen from the set of customers. For each scenario, we solved the UPS problems by varying the number of nodes from 20 to 40. Here, we want to compare three drone configurations: (1) heterogeneous drones with the speed parameters $[1, 1.5]$; (2) homogeneous standard drones with $[1, 1]$; and (3) homogeneous fast drones with $[1.5, 1.5]$.

Figure 5 plots the distributions of the objective values of 10 problems for each heavy demand scenario and drone configuration. As expected, the objective values (i.e., delivery completion time) increase with higher ratios of heavy demands. However, the case with the heterogeneous drones showed the least objective values for all scenarios. With the homogeneous standard drones, we had the same objective value regardless of the ratios of heavy demands because the standard drone can carry the heavy demand. We can see that steep increases of objective values for the homogeneous fast drones case as the ratios of heavy demands become higher. Because fast drones cannot deliver heavy demands, the truck must visit the customers with heavy demands, which results in increased delivery times. Obviously, it is desired that as many as demands are subject to potential drone delivery. Therefore, the use of homogeneous drone fleets may cause one of two outcomes: (1) high

Figure 5. (Color online) Distributions of Objective Values when There Are Some Heavy Demands that Cannot Be Carried by the Fast Drone with $\alpha = 1.5$



Notes. Heavy demand percentages: 10% (light gray), 20% (medium gray), and 30% (dark gray). Three cases of drone configurations are compared: (1) heterogeneous drones with the speed parameters [1,1.5]; (2) homogeneous standard drones with [1,1]; and (3) homogeneous fast drones with [1.5,1.5]. (a) CVS problems. (b) WM problems.

applicability of drone delivery with slow drones; or (2) low applicability of drone delivery with fast drones. The former is the case of drone configuration [1,1] while the latter is the case of [1.5,1.5].

5.6. Benefit of Drone-Truck Cooperative Routing

Here we assert the benefit of drone-truck cooperative routing. We solved the UPS problems with 20, 30, 40 nodes by the following three approaches:

- Truck-only delivery (T-Only)
- Drone-only delivery (D-Only)
- Drone-Truck cooperative routing (this study, D-T)

The first approach, T-Only, is equivalent to the traditional vehicle routing problem, which is solved by the same method as line 7 in Algorithm 1. The second approach, D-Only, is the original operation method of UPS in the considered areas. All customers are served by the drones dispatched from the depot with assumed infinity battery capacity. We consider two drones with the speed parameters [1,1.5]. Note that we can easily solve the drone only delivery problem by the formulation (BSP) with relaxing constraints (33), and letting $z_s = 1$ and $z_i = 0$ for all $i \in N$. The last approach, D-T, is the proposed method in this study which uses the same set of drones as the D-Only approach.

Figure 6 compares the results of the three different operation approaches. As expected, the truck only method showed the worst objective values. Though the truck only routing method seems too inefficient, it may be the only choice when the demands cannot be delivered by the drone. The drone only delivery method showed better results than the truck only approach. Even though a drone is much faster than the truck to deliver a demand, its limited carrying capacity makes the drone come back to the depot to fetch another demand. On the other hand, the drone-truck cooperative routing performed the best among the tested approaches in terms of the objective value. Moreover, the drone-truck routing showed the least variance of the objective values, which indicates the operational robustness of the approach. Contrary to the fixed depot, the truck can provide flexible drone operating locations, which yields adaptive solutions.

5.7. Sensitivity Analysis

We solved the same problems with various drone parameters to assert the changes in the solutions for different drone configurations. We consider the following scenarios:

Maximum flight duration of the standard drone: 1,200, 1,500, 1,800, and 2,100 seconds

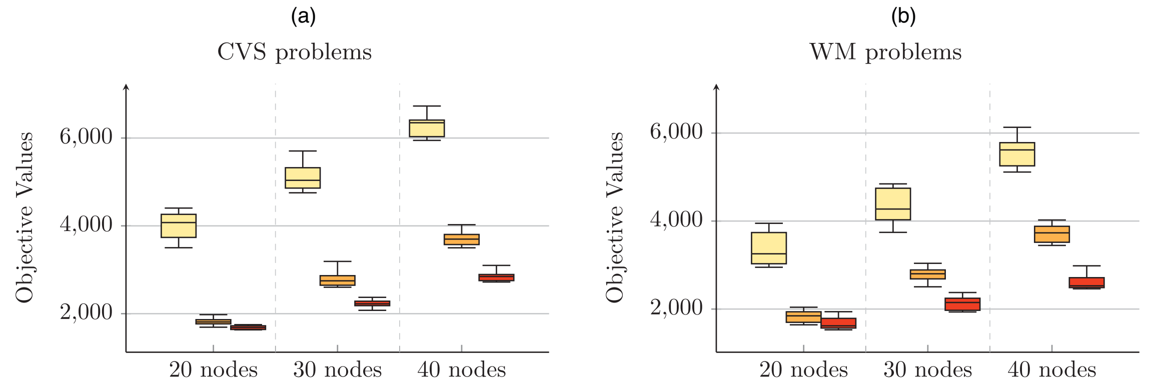
Fleet of drones to use: [1], [1, 1], and [1, 1, 1]

Therefore, we have 12 (4 × 3) cases to consider. For each case, we solved UPS problems with 20, 30, and 40 customer nodes. Note that the change on the maximum flight duration of the standard drone would impact all drones in the fleet because each drone's capability is defined relative to the standard drone by the speed parameter.

Figure 7 shows the changes in objective values for the tested cases. Each plot displays the distribution of objective values for the given problem set. Clearly, we see the decreases in the objective values with longer flight durations. The flight duration is proportional to the battery capacity, which means the plots show benefits of having larger battery capacity. When the number of deliveries is small (e.g., 20), it is shown that the benefit of larger battery capacity is not significant. On the other hand, the benefit of using multiple drones is significant regardless of problem sizes, especially when the number of drones becomes two from one. We also see that reduction of the objective values by deploying more than two drones is less significant.

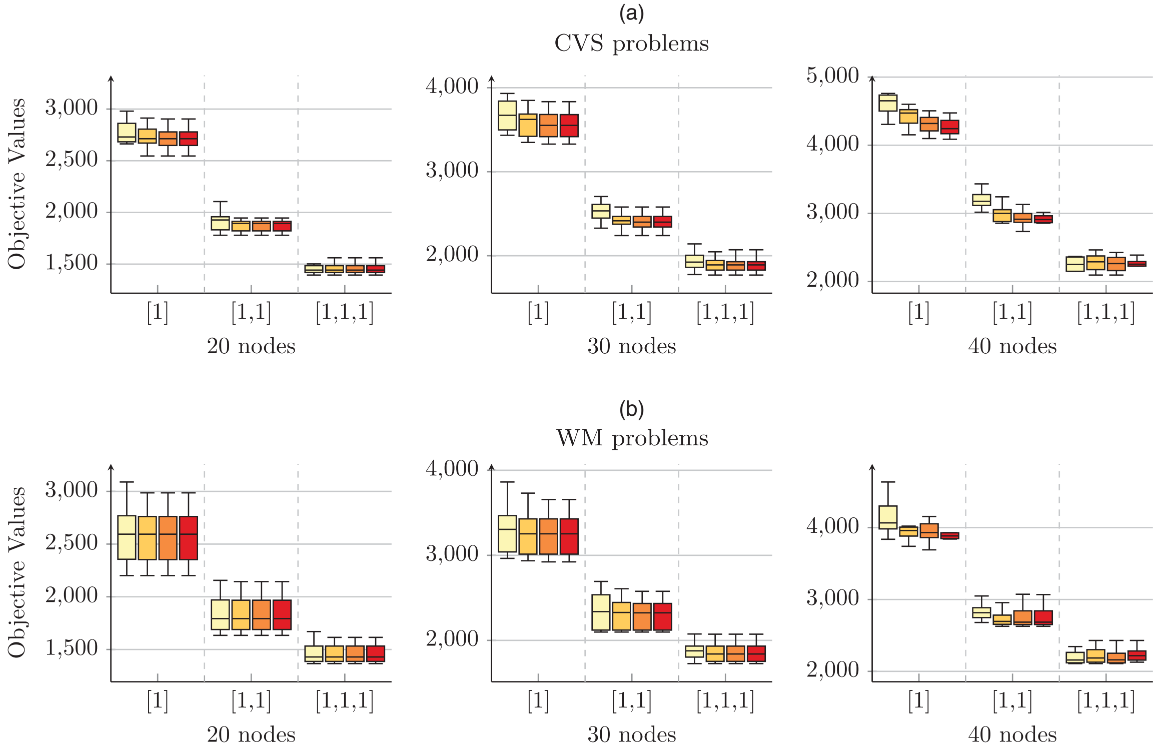
Figure 8 presents the changes of drone ratios for different drone configuration cases. We can see that the drone serves more customers with increased flight duration and the number of drones. Actually, the drone ratios' plots seem strongly negatively correlated with

Figure 6. (Color online) Distributions of Objective Values for the Three Different Delivery Methods



Notes. (1) Truck-only (light gray); (2) Drone-only (medium gray); and (3) Drone-Truck cooperative routing (dark gray). (a) CVS problems. (b) WM problems.

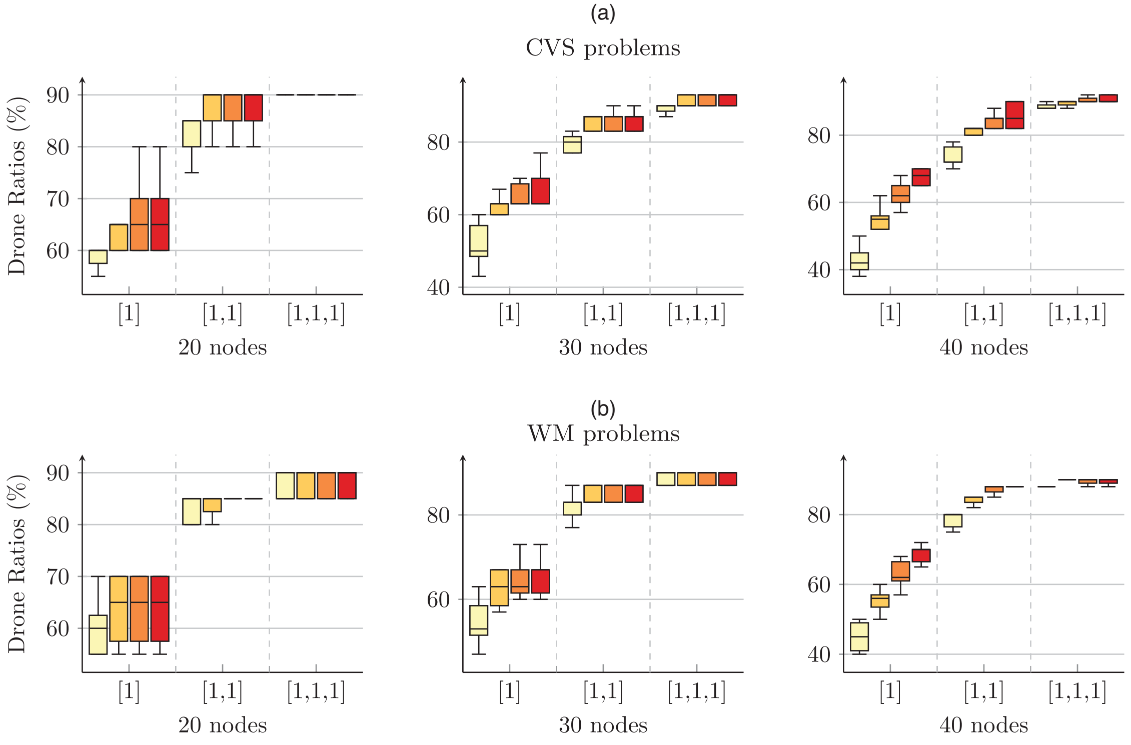
Figure 7. (Color online) Distributions of Objective Values for Different Combinations of Drone Fleets and Maximum Flight Durations


(a)
CVS problems

(b)
WM problems

Notes. 1,200 seconds (light gray), 1,500 seconds (medium gray), 1,800 seconds (dark gray), and 2,100 seconds (darkest gray). (a) CVS problems. (b) WM problems.

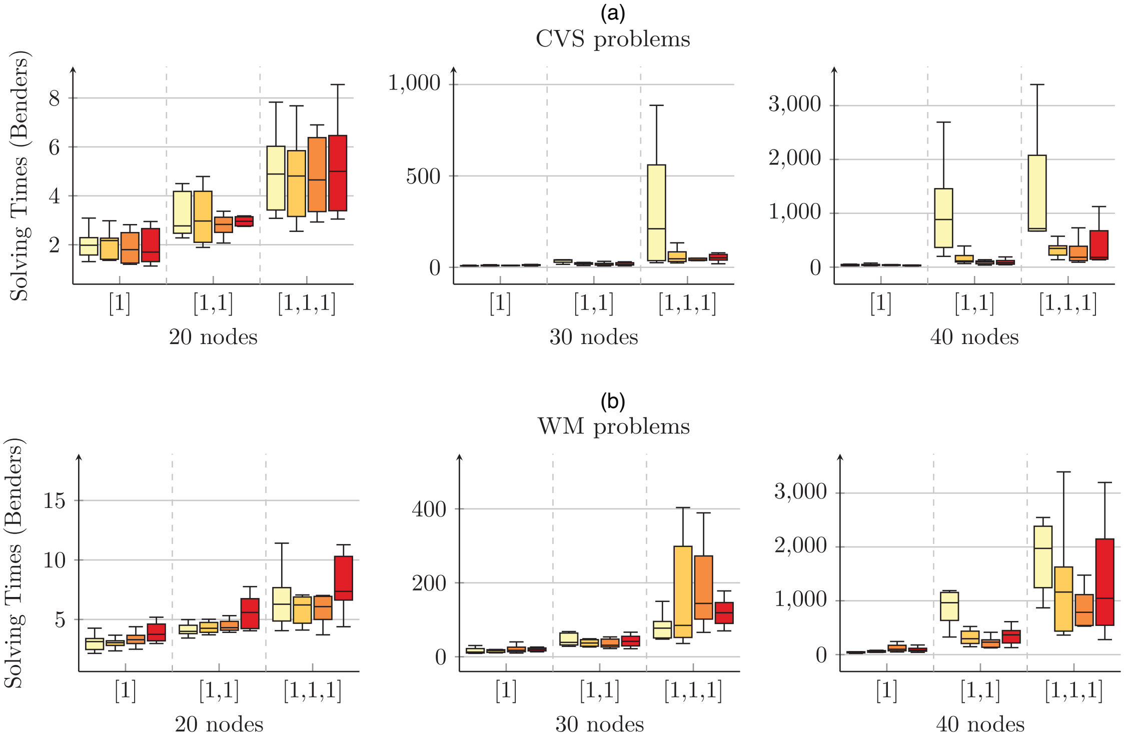
Figure 8. (Color online) Distributions of Drone Ratios for Different Combinations of Drone Fleets and Maximum Flight Durations


(a)
CVS problems

(b)
WM problems

Notes. 1,200 seconds (light gray), 1,500 seconds (medium gray), 1,800 seconds (dark gray), and 2,100 seconds (darkest gray). (a) CVS problems. (b) WM problems.

Figure 9. (Color online) Distributions of Solving Times of the Benders Approach for Different Combinations of Drone Fleets and Maximum Flight Durations



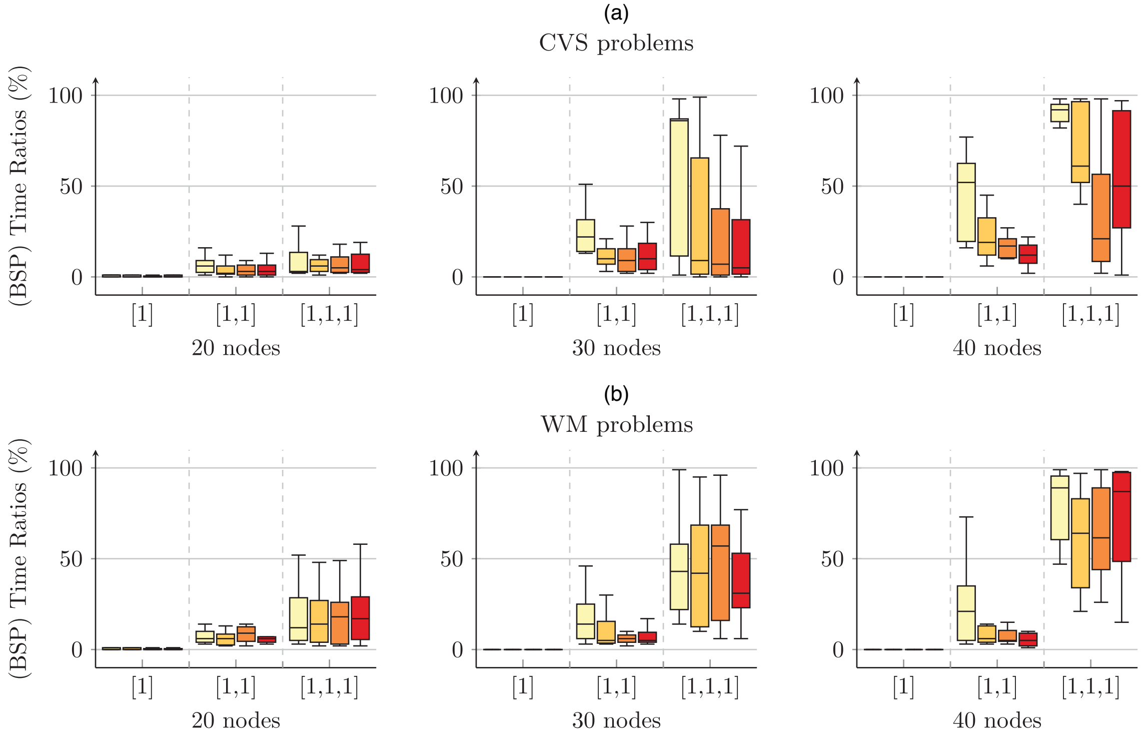
Notes. 1,200 seconds (light gray), 1,500 seconds (medium gray), 1,800 seconds (dark gray), and 2,100 seconds (darkest gray). (a) CVS problems. (b) WM problems.

the objective values shown in Figure 7. When the drones are highly available (e.g., long flight duration and multiple drones), almost 90% of demands are delivered by the drones. Actually, it seems that the drone ratios are saturated with 90%. As shown in Section 5.6, letting the drones deliver 100% of total demands is not necessarily better because the drone's flight distances increase. This observation implies that to achieve efficient delivery results, the truck should act as a mobile drone station to minimize the truck deliveries. Therefore, the truck can maximize drone coverage and reduce the drone's flight distances.

The impact on the computational times for different drone configurations is shown in Figure 9. The general trend is the computational time decreases with increased flight durations. This is because the less drone flight duration becomes, the longer the truck's route gets. Generally, the problem is harder to solve when the truck has to visit many nodes, which can be seen in the problems with uniformly distributed nodes such as Augerat A-type problems. We found that increasing the number of drones also makes the problem much more challenging.

Figure 10 compares the ratios of times spent in solving the Benders subproblem. Because the Benders subproblem (BSP) should be solved repeatedly during the algorithm, the Benders subproblem's solving time becomes dominant as the size of the Benders subproblem grows. As shown in the figure, almost all computational time is spent on solving the Benders subproblem when there are many drones to consider (e.g., 40 node problems with [1, 1, 1]). We also notice that the solving time for the (BSP) is relatively not consistent, which means that there are some peculiar cases in which solving the Benders subproblem takes a long time. Note that the formulation (BSP) has symmetry over drones $l \in L$ when there are identical drones. It is known that the existence of symmetry may deteriorate the performance of the branch-and-bound search. Cplex provides an option for symmetry breaking strategy, with which we have tried for the (BSP) formulation. Unfortunately, we observed little improvement in setting the symmetry breaking option to Cplex. We believe that the Benders subproblem can be solved by a specialized algorithm such as a dynamic programming approach that might be much faster than Cplex. Developing a faster algorithm for the (BSP) would be worth serious investigation, which we will leave for future research. Nonetheless, we also argue that using more than two drones might be impractical because it will complicate the delivery personnel's operations. Moreover, the truck should be much larger to carry many drones, increasing the truck's operation cost.

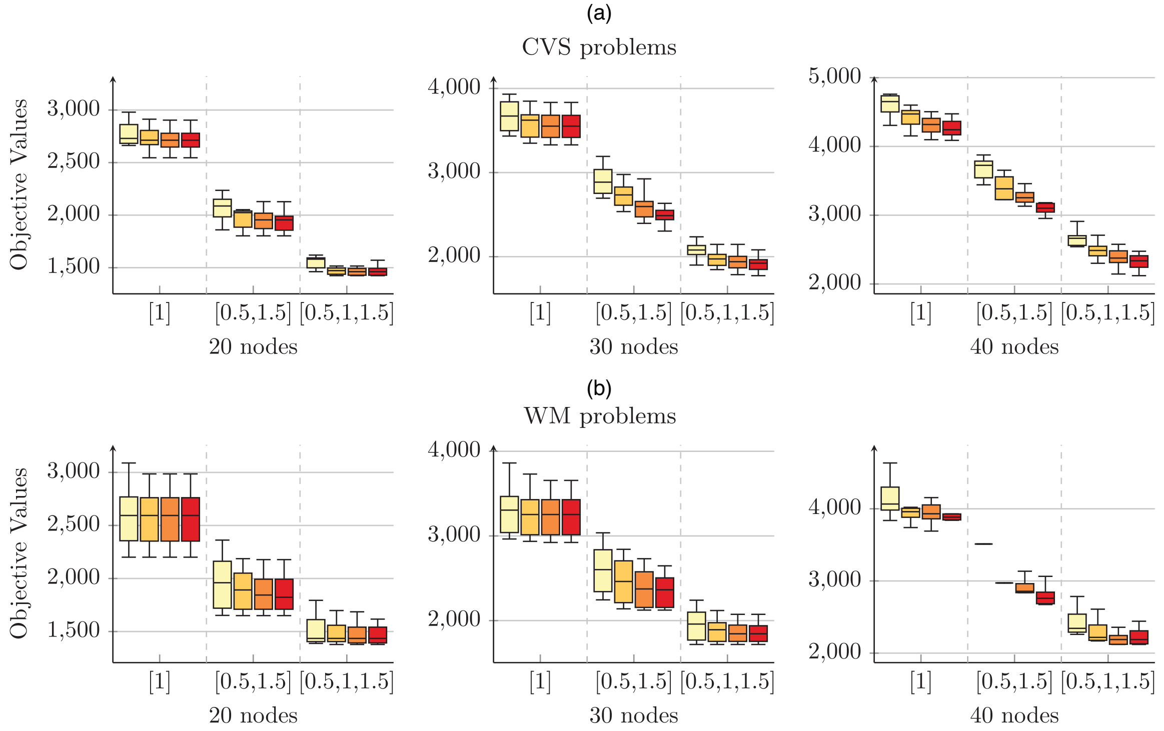
Figure 10. (Color online) Distributions of (BSP) Solving Time Ratios for Different Combinations of Drone Fleets and Maximum Flight Durations


(a)
CVS problems

(b)
WM problems

Notes. 1,200 seconds (light gray), 1,500 seconds (medium gray), 1,800 seconds (dark gray), and 2,100 seconds (darkest gray). (a) CVS problems. (b) WM problems.

Figure 11. (Color online) Distributions of Objective Values for Different Combinations of Drone Fleets ([1], [0.5, 1.5], and [0.5, 1, 1.5]) and Maximum Flight Durations


(a)
CVS problems

(b)
WM problems

Notes. 1,200 seconds (light gray), 1,500 seconds (medium gray), 1,800 seconds (dark gray), and 2,100 seconds (darkest gray). (a) CVS problems. (b) WM problems.

Figure 11 presents the results of another experiment with different drone fleets of [1], [0.5, 1.5], and [0.5, 1, 1.5]. The results show a very similar pattern to that of Figure 7. However, when the flight duration is relatively shorter (e.g., 1,200 seconds), the objective values of Figure 11 are worse than those of Figure 7. Moreover, we can see a fast decrease of the objective values with more flight durations. This behavior implies that the flight duration (or battery capacity) is a bottleneck in the case of Figure 11. Similarly, we may conclude that, in the case of Figure 7, it would be better to increase the number of drones rather than investing in more battery capacity.

6. Conclusion

This paper addressed a heterogeneous drone truck routing problem, which we call HDTRP. We considered the case in which the drones are carried by truck and can be used for delivery while the truck is parked and waiting. To solve the problem, one should make three-stage decisions: the truck's route, the waiting nodes to deploy the drones, and to assign drones for delivery. The proposed model does not require precise synchronization between the drones and truck because the truck is simply waiting at a safe location for the drones to return. Moreover, the use of drones without dedicated drone stations would be another advantage over the existing approach.

The presented mixed integer programming problem is very hard to solve even with a state-of-the-art MIP solver. We developed a decomposition algorithm by separating the decision on the truck route from the problem. The resulting algorithm involves solving the drone delivery problem subject to the given truck route, unfortunately, which is another integer programming problem. We adopted the logic-based Benders decomposition to address the discrete subproblem by proposing the custom Benders cuts, and we showed that the proposed Bender cuts are valid. To accelerate the proposed Benders algorithm, we also developed a set of preprocessing steps, including primal heuristics and variable bounding.

The extensive computational study clearly shows that the proposed algorithm outperformed the Cplex with the MIP formulation. The results also revealed that the performance of the proposed algorithm could depend on the spatial distribution of nodes. The more clustered the nodes, the proposed algorithm seems to perform better. This can be understood by noting that clustered nodes may make the truck route solutions more apparent from which Benders iterations are benefited.

A relevant issue is that uncertain environments can hinder the efficiency of drone operations. For example, wind direction and speed can affect the delivery time and battery usage (Cheng et al. 2020). How to address the operation uncertainty is worth investigation.

References

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

Augerat P (1995) Approche polyèdrale du problème de tournées de véhicules. Ph.D. thesis, Institut National Polytechnique de Grenoble-INPG.

Balakrishnan A, Magnanti TL, Wong RT (1995) A decomposition algorithm for local access telecommunications network expansion planning. *Oper. Res.* 43(1):58–76.

Bektas T (2006) The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* 34(3):209–219.

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4:238–252.

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

Bürkle A, Essendorfer B (2010) Maritime surveillance with integrated systems. *2010 Internat. WaterSide Security Conf.* (IEEE, Piscataway, NJ), 1–8.

Cattaruzza D, Absi N, Feillet D, Vidal T (2014) A memetic algorithm for the multi trip vehicle routing problem. *Eur. J. Oper. Res.* 236(3):833–848.

Cheng C, Adulyasak Y, Rousseau LM (2018) Formulations and exact algorithms for drone routing problem. Technical Report. CIRRELT-2018-31, CIRRELT .

Cheng C, Adulyasak Y, Rousseau LM, Sim M (2020) Robust drone delivery with weather information. http://www.optimization-online.org/DB_HTML/2020/07/7897.html.

Corréa AI, Langevin A, Rousseau LM (2007) Scheduling and routing of automated guided vehicles: A hybrid approach. *Comput. Oper. Res.* 34(6):1688–1707.

CPLEX-User's-Manual (1987) IBM ILOG CPLEX Optimization Studio. *Version*. 12:1987–2020.

Cuda R, Guastaroba G, Speranza MG (2015) A survey on two-echelon routing problems. *Comput. Oper. Res.* 55:185–199.

Daknama R, Kraus E (2017) Vehicle routing with drones. Preprint submitted May 18, https://arxiv.org/abs/1705.06431.

Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Management Sci.* 6(1):80–91.

Dell'Amico M, Montemanni R, Novellani S (2020) Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. *Ann. Oper. Res.* 289:211–226.

DHL (2014) DHL parcelcopter launches initial operations for research purposes. Accessed December 10, 2020, https://www.dhl.com/en/press/releases/releases_2014/group/dhl_parcelcopter_launches_initial_operations_for_research_purposes.html.

Drexl M, Schneider M (2015) A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.* 241(2):283–308.

Errico F, Crainic TG, Malucelli F, Nonato M (2017) A Benders decomposition approach for the symmetric TSP with generalized latency arising in the design of semiflexible transit systems. *Transportation Sci.* 51(2):706–722.

Flammini F, Naddei R, Pragliola C, Smarra G (2016) Toward automated drone surveillance in railways: State-of-the-art and future directions. *Internat. Conf. Adv. Concepts Intelligent Vision Systems* (Springer), 336–348.

Gaba F, Winkenbach M (2020) A systems-level technology policy analysis of the truck-and-drone cooperative delivery vehicle system. Working paper, MIT Center for Transportation & Logistics, Cambridge, MA.

Gendreau M, Laporte G, Séguin R (1995) An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Sci.* 29(2):143–155.

Gendreau M, Laporte G, Séguin R (1996) Stochastic vehicle routing. *Eur. J. Oper. Res.* 88(1):3–12.

Ha QM, Deville Y, Pham QD, Ha MH (2018) On the min-cost traveling salesman problem with drone. *Transportation Res. Part C: Emerging Tech.* 86:597–621.

Hooker JN (2007) Planning and scheduling by logic-based Benders decomposition. *Oper. Res.* 55(3):588–602.

Hooker JN, Ottosson G (2003) Logic-based Benders decomposition. *Math. Programming* 96(1):33–60.

Infanger G (1992) Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs. *Ann. Oper. Res.* 39(1):69–95.

Johnson DS, Garey MR (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (WH Freeman, New York).

Jones T (2017) International Commercial Drone Regulation and Drone Delivery Services (Rand Corporation, Santa Monica, CA).

Kim S, Moon I (2019) Traveling salesman problem with a drone station. *IEEE Trans. Systems Man Cybernetic Systems* 49(1):42–52.

Kitjacharoenchai P, Ventresca M, Moshref-Javadi M, Lee S, Tanchoco JM, Brunese PA (2019) Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Comput. Indust Engrg.* 129:14–30.

Laporte G, Louveaux FV (1993) The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* 13(3):133–142.

Laporte G, Louveaux FV, Van Hamme L (2002) An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Oper. Res.* 50(3):415–423.

Laporte G, Nickel S, da Gama F, eds. (2019) *Location Science* (Springer, Cham, Switzerland).

Lee C, Han J (2017) Benders-and-Price approach for electric vehicle charging station location problem under probabilistic travel range. *Transportation Res. Part B: Methodological* 106:130–152.

Lee C, Lee K, Park S (2013) Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks* 62(1):1–16.

Lithia F (2017) UPS tests residential delivery via drone launched from atop package car. https://www.globenewswire.com/news-release/2017/02/21/925955/0/en/UPS-Tests-Residential-Delivery-Via-Drone-Launched-From-Atop-Package-Car.html.

Liu Rl, Zhang Zj, Jiao Yf, Yang Ch, Zhang Wj (2019) Study on flight performance of propeller-driven UAV. *Internat. J. Aerospace Engrg.*, ePub ahead of print April 21, http://dx.doi.org/10.1155/2019/6282451.

Macrina G, Pugliese LDP, Guerriero F, Laporte G (2020) Drone-aided routing: A literature review. *Transportation Res. Part C: Emerging Tech.* 120:102762.

Malandraki C, Daskin MS (1992) Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Sci.* 26(3):185–200.

Martello S, Hoboken PT (1990) *Knapsack Problems* (Wiley-Interscience, NJ).

McKinnon A, Browne M, Whiteing A, Piecyk M (2015) *Green Logistics: Improving the Environmental Sustainability of Logistics* (Kogan Page Publishers, London).

Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problems. *J. ACM* 7(4):326–329.

Mogili UR, Deepak B (2018) Review on application of drone systems in precision agriculture. *Procedia Comput. Sci.* 133:502–509.

Murray CC, Chu AG (2015) The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Res. Part C: Emerging Tech.* 54:86–109.

Nielsen SS, Zenios SA (1997) Scalable parallel Benders decomposition for stochastic linear programming. *Parallel Comput.* 23(8):1069–1088.

Ordóñez F (2010) Robust vehicle routing. Hasenbein JJ, ed. *Risk and Optimization in an Uncertain World*, INFORMS Tutorials on Operations Research (INFORMS, Catonsville, MD), 153–178.

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

Paradiso R, Roberti R, Laganá D, Dullaert W (2020) An exact solution framework for multitrip vehicle-routing problems with time windows. *Oper. Res.* 68(1):180–198.

Parragh SN, Doerner KF, Hartl RF (2008) A survey on pickup and delivery problems. *J. Betriebswirtschaft* 58(1):21–51.

Perboli G, Tadei R, Vigo D (2011) The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Sci.* 45(3):364–380.

Poikonen S, Golden B (2019) Multi-visit drone routing problem. *Comput. Oper. Res.* 113:104802.

Popper B (2015) Drones could make Amazon's dream of free delivery profitable. http://www.theverge.com/2015/6/3/8719659/amazon-prime-air-drone-delivery-profit-free-shipping-small-items.

Rahmaniani R, Ahmed S, Crainic TG, Gendreau M, Rei W (2020) The Benders dual decomposition method. *Oper. Res.* 68(3):878–895.

Rosenkrantz DJ, Stearns RE, Lewis PM II (1977) An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* 6(3):563–581.

Roshanaei V, Luong C, Aleman DM, Urbach D (2017) Propagating logic-based Benders' decomposition approaches for distributed operating room scheduling. *Eur. J. Oper. Res.* 257(2):439–455.

Russell RA (1977) An effective heuristic for the m-tour traveling salesman problem with some side conditions. *Oper. Res.* 25(3):517–524.

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

Savelsbergh MW, Sol M (1995) The general pickup and delivery problem. *Transportation Sci.* 29(1):17–29.

Schermer D, Moeini M, Wendt O (2019) The traveling salesman drone station location problem. *World Congress on Global Optimization* (Springer, Cham, Switzerland), 1129–1138.

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35(2):166–324.

Staudt CL, Sazonovs A, Meyerhenke H (2016) Networkit: A tool suite for large-scale complex network analysis. *Network Sci.* 4(4):508–530.

Sungur I, Ordónez F, Dessouky M (2008) A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Trans.* 40(5):509–523.

Taccari L (2016) Integer programming formulations for the elementary shortest path problem. *Eur. J. Oper. Res.* 252(1):122–130.

Toth P, Vigo D (2002) *The Vehicle Routing Problem* (SIAM, Philadelphia).

Tseng CM, Chau CK, Elbassioni KM, Khonji M (2017) Flight tour planning with recharging optimization for battery-operated autonomous drones. Preprint, submitted March 29, https://arxiv.org/abs/1703.10049v1.

Vásquez SA, Angulo G, Klapp MA (2020) An exact solution method for the TSP with drone based on decomposition. *Comput. Oper. Res.* 127:105127.

Vidová H, Babčanová D, Witkowski K, Saniuk S (2012) Logistics and its environmental impacts. *7th Internat. Sci. Conf. Bus. Management*, 1007–1014.

Yurek EE, Ozmutlu HC (2018) A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Res. Part C: Emerging Tech.* 91:249–262.