

Drone Delivery Scheduling Optimization Considering Payload-induced Battery Consumption Rates

Maryam Torabbeigi · Gino J. Lim · Seon Jin Kim

the date of receipt and acceptance should be inserted later

Abstract This paper addresses the design of a parcel delivery system using drones, which includes the strategic planning of the system and operational planning for a given region. The amount of payload affects the battery consumption rate (BCR), which can cause a disruption in delivery of goods if the BCR was underestimated in the planning stage or cause unnecessarily higher expenses if it was over-estimated. Hence, a reliable parcel delivery schedule using drones is proposed to consider the BCR as a function of payload in the operational planning optimization. A minimum set covering approach is used to model the strategic planning and a mixed integer linear programming problem (MILP) is used for operational planning. A variable preprocessing algorithm and primal and dual bound generation methods are developed to improve the computational time for solving the operational planning model. The optimal solution provides the least number

of drones and their flight paths to deliver parcels while ensuring the safe return of the drones with respect to the battery charge level. Experimental data show that the BCR is a linear function of the payload amount. The results indicate the impact of including the BCR in drone scheduling, 3 out of 5 (60%) flight paths are not feasible if the BCR is not considered. The numerical results show that the sequence of visiting customers impacts the remaining charge.

Keywords: Battery Consumption Rate, Payload Amount, Path Planning, Drone Scheduling, Delivery Network

1 Introduction

Ground vehicles such as trucks are typically used to deliver parcels across the logistic networks. More recently, large companies such as Amazon, Mercedes-Benz, United Parcel Service, and DHL have plans to utilize drones for delivery purposes [1–5]. The general idea is to deliver packages from a base location to pre-selected destinations by a given fleet of vehicles. UAVs, also known as drones, can deliver packages alone [6–10] or in collaboration with other ground

M. Torabbeigi · G. J. Lim · S. J. Kim
Department of Industrial Engineering, University of Houston,
TX 77204, USA
E-mail: ginolim@uh.edu

M. Torabbeigi
E-mail: maryam.torabbeigi@gmail.com
S. J. Kim
E-mail: sonjin64@gmail.com

transportation vehicles [11–13]. The scheduling model for the drone delivery problem is similar to the traveling salesman problem [14] when each destination location is served by only one vehicle, and vehicles start and finish their path at the same location [8–10, 15].

Although drones are gaining more attraction for delivery purposes in recent years, limited battery endurance and limited payload amount remain to be drawbacks for practical use [16–20]. An accurate estimate of battery endurance during the planning stage is crucial in optimizing drone delivery schedule. For example, a delivery plan based on under-estimated battery endurance may lead to the loss of opportunities to serve more customers. As a result, more drones may be required to satisfy the required delivery demand. The opposite can be much worse because some drones may not be able to return to the base due to lack of battery before completing the planned delivery. The impact of drone failure on the whole network in a delivery application of drones is studied in a few researches [15]. The amount of payload is one of the key factors affecting the flight duration. Therefore, it should be considered in drone scheduling as it impacts the battery endurance.

Some existing studies have considered the limitation on the total flight time or the payload amount in drone scheduling [9, 15, 21–24] and others considered the impact of carried payload on the total flight time calculation [25–27]. However, these approaches did not address the issue of potential failure of drones to return due to lower than expected battery charge. There are studies focusing on characterizing the energy consumption of drones [28–30], but not in the context of drone scheduling. In order to avoid running out of charge and utilize drones efficiently, the battery con-

sumption rate is included in drone routing in this paper. A Phantom 4 Pro+ [31] is tested to collect flight time and remaining battery charge data. We experimentally show that the BCR is a linear function of carried payload, and it is modeled using linear regression.

A group of drones is considered in this paper to deliver small packages to customers to study the impact of BCR on the fleet scheduling. Two optimization planning models are proposed to design drone-based parcel delivery: strategic planning (SP) and operational planning (OP). The number of delivery bases and their locations are determined by solving the strategic planning model. A set covering problem approach is used to model SP by taking into account the distance between customers and base locations to ensure feasible flights. The OP model aims to minimize the number of drones while considering specifications of drones in drone routing optimization. A mixed integer linear programming (MILP) model for drone routing and scheduling is proposed to determine optimal drone delivery assignments and their paths. The optimal solution provides the least number of drones required to serve all the customers. The variable preprocessing technique, primal and dual bound generation schemes are developed to reduce the computational time. Overall, the contributions of this paper include: 1) proposing the BCR concept in drone delivery application to capture the effect of payload amount on battery endurance and estimate parameters based on case study collected data, 2) proposing two optimization planning models: strategic planning to figure out the optimal locations to open depots, and operational planning to determine drone paths by including limitations of payload amount and battery endurance, 3) providing the

solution methodology consisting of a variable preprocessing technique, primal and dual bound generation methods to reduce the computational time.

The rest of this paper is organized as follows: Section 2 explains the collected data from testing a drone and the corresponding linear regression to estimate the battery consumption rate as a function of payload amount. The SP and OP mathematical models are presented in Section 3, and the solution methodology for OP model is provided in Section 4. The numerical results and conclusion are discussed in Section 5 and Section 6, respectively.

2 Drone Battery Consumption Rate

The BCR can be defined as the amount of charge consumption per unit of time (per minute); it is the rate at which battery charge decreases during the flight. The BCR is estimated as a function of payload based on data collected using a Phantom 4 Pro+ drone [31] with specifications stated in Table A in Appendix A. The data include flight time and the state of battery charge over time. Although the power consumption can vary depending on the flight mode (e.g., hovering, forward flight, landing), the difference among different flight modes is negligible [28]. Furthermore, the power consumption in hovering mode is greater than the other flight modes due to effective translational lift [32]. Therefore, the data collected in the hovering mode was used in this paper.

Table B in Appendix B shows the flight time duration in minutes for various combinations of battery charge (from 15% to 95%) and payload amount (from 0 lb. to 0.882 lb.). The data are also plotted in Fig-

ure 1, which shows the State Of Charge (SOC) during the flight for different payload amounts. The payload amount is the amount of weights that a drone carries excluding the weight of the drone and its battery. A drone can consume up to 5% of the fully charged battery until it reaches the hovering mode. Therefore, the initial battery charge was set at 95% to be consistent in all experiments conducted in this paper. Another parameter is the minimum remaining battery charge to ensure a safe landing, which is 15% for the drone we used in our experiments. Therefore, the flight time was recorded in the hovering position with the battery charge between 15% and 95% at 5% interval. The experiment was repeated for different amount of payloads: 0, 0.220, 0.441, 0.661, and 0.882 lb.

Table 1 lists the linear regression lines for the data shown in Figure 1 and the corresponding R-squared values. Since the R-squared values are all higher than 99.9% for all regression lines, we claim that there is approximately a linear relationship between flight time and battery SOC. For each payload amount, the BCR corresponds to the slope of the regression line. For example, the BCR value of 4.39 (%/min) means 4.39% of battery charge decreases in every minute of flight. The regression lines clearly show that the BCR increases as the payload increases. When we increased the payload from 0 to 0.22lb, the BCR was increased by 14.5% (i.e., moved from -3.834 to -4.390). A similar trend was observed for other regression lines as plotted in Figure 2.

In Figure 2, the red dots are the calculated BCR values corresponding to different payload amounts, and the dashed line is the regression model used in the optimization model in OP (see Section 3.2). We further

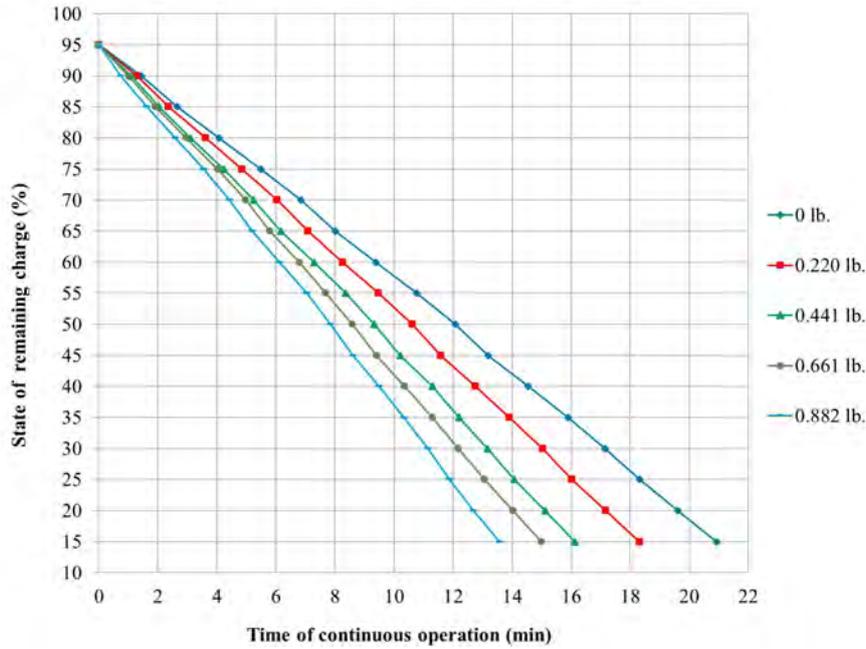


Fig. 1: The remaining battery level (%) along the flight for different amount of payload (lb.).

Table 1: The BCR values for different amount of payloads

Payload amount (lb.)	Linear regression line	R^2	BCR (% /min)
0	SOC = -3.834 t + 95.67	0.9997	3.834
0.220	SOC = -4.390 t + 95.88	0.9996	4.390
0.441	SOC = -4.977 t + 95.71	0.9996	4.977
0.661	SOC = -5.388 t + 95.91	0.9996	5.389
0.882	SOC = -5.867 t + 95.32	0.9994	5.867

analyze the resulting linear regression model:

$$BCR = \alpha \cdot \text{payload} + \beta, \quad (1)$$

where α is the slope and β is intercept. According to Table 2, the estimated value of α is $2.297 \left(\frac{\%}{\text{min}\cdot\text{lb}}\right)$ and β is $3.879 \left(\frac{\%}{\text{min}}\right)$. The corresponding p -values of the payload and intercept are below 0.000115, which implies that both parameters are statistically significant. The adjusted R-squared value of the model is greater than 99%. Therefore, we claim that a linear relationship exists between the BCR and the payload amount.

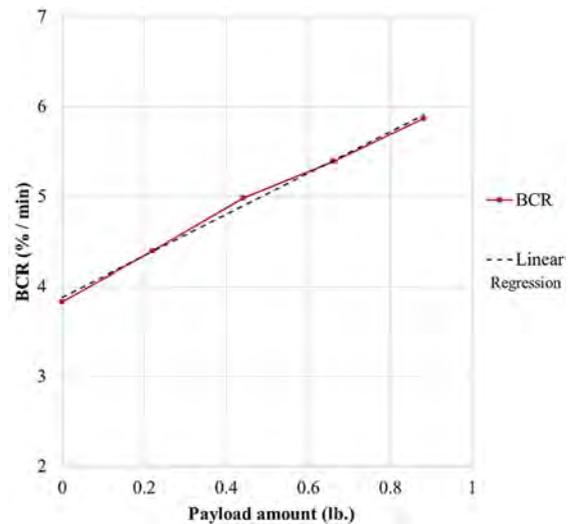


Fig. 2: The relationship between the BCR and the payload amount

Table 2: Linear regression analysis: BCR vs. payload

Coefficients	Estimate	Std. Error	t-value	$Pr(> t)$
Intercept	3.87886	0.04645	83.5	3.79e-06
payload	2.29705	0.08603	26.7	0.000115

*Residual standard error: 0.05999 on 3 degrees of freedom

*Multiple R-squared: 0.9958

*Adjusted R-squared: 0.9944

3 Problem Description and Formulations

The problem description and the mathematical models are presented in this section. This study focuses on the lightweight parcels so that drones can carry them to customers within the weight capacity of drone. Each day, a fleet of drones pick up parcels from the base depots, deliver them to customers, and then return to the [same base](#).

One may explore similar strategy for satisfying a demand in shareable products [33]. If the demand of a customer is greater than the weight capacity, then the demand can be divided into multiple sub-orders for delivery.

In a drone-induced parcel delivery system, a strategic planning should be made in the facility design phase and operational planning decisions are made for parcel delivery schedule. The SP includes the facility planning and decides about the number and location of depots based on customers' location and drones' specifications. The OP includes drone utilization planning and flight path planning, and decides about the number of drones to use for the day, the assignment of customers to drones and order of visiting them. The effect of payload amount on the total flight duration and the remaining battery charge can be considered in the OP.

3.1 Strategic Planning (SP)

A drone starts its flight from a base depot, delivers products to customers, and returns to the depot. In the depot, the batteries are replaced or charged, and the payloads are loaded for future flights. Among a set of potential locations, a few of them are chosen to establish base depots and serve all the customers. A customer can be covered by a candidate location if the customer is located within the flight range of a drone. In order to calculate the SOC at each flight stop, the following notation is used:

SOC_i	state of charge at node i ,
RC_k	drone k remaining charge on returning to the depot,
d_i	customer i 's demand,
t_{ij}	flight time between node i and j (is assumed to be symmetric, $t_{ij} = t_{ji}$),
α_k, β_k	slope and intercept parameters of the linear function to calculate BCR according to Formula (1),
$MinCh_k$	minimum required battery charge for safe landing of drone k , ($k=1,2, \dots$),
$MaxP_k$	maximum payload capacity of drone k , ($k=1,2, \dots$).

Figure 3 shows a flight path consisting of one depot and one customer. The battery's SOC at each stop (depot, customer's location, and again depot) is calculated through equations (2a)-(2c) by using Formula (1):

$$SOC_j = 100, \quad (2a)$$

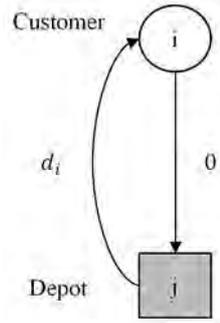


Fig. 3: Covering customer i by candidate location j

$$SOC_i = SOC_j - t_{ji}(\alpha_k d_i + \beta_k)$$

$$\rightarrow SOC_i = 100 - t_{ji}(\alpha_k d_i + \beta_k), \quad (2b)$$

$$RC_k = SOC_i - t_{ij}(\alpha_k \cdot 0 + \beta_k)$$

$$\rightarrow RC_k = 100 - t_{ij}(\alpha_k d_i + \beta_k) - t_{ij}\beta_k. \quad (2c)$$

The drone battery is assumed to be fully charged at the beginning of the flight as stated in equation (2a). Equation (2b) calculates the remaining battery level at location i considering the amount of payload in the flight segment from depot j to customer i to be the customer demand. There is no load on the way back to the depot and the remaining battery level at the end of the path is computed by (2c).

To avoid running out of charge on landing on the depot and have a feasible flight path, the remaining charge at the end of the flight should be greater than the minimum charge requirement, which leads to the following inequality:

$$\begin{aligned} RC_k &> MinCh_k \quad \text{and} \quad t_{ij} = t_{ji}, \\ \rightarrow t_{ij}(\alpha_k d_i + 2\beta_k) &< 100 - MinCh_k, \\ \rightarrow t_{ij} &< \frac{100 - MinCh_k}{\alpha_k d_i + 2\beta_k}. \end{aligned} \quad (3)$$

The demand of customers changes daily. Finally, inequality (4) is obtained by replacement of d_i with the drone weight capacity. If inequality (4) holds, then in-

equality (3) holds for different amounts of load.

$$t_{ij} < \frac{100 - MinCh_k}{\alpha_k MaxP_k + 2\beta_k}. \quad (4)$$

The right-hand side of inequality (4) depends on the drone specifications and it can be used to calculate the maximum flight range of a drone. A depot can cover multiple customers as long as they are located within this drone maximum flight range. We propose the minimum set covering problem to determine the least number of candidate locations to cover all the customers.

The notation used in the SP model is:

Sets:

C Set of customers,

S Set of candidate locations for depots.

Parameters:

f_j Fixed cost of opening depot at candidate location j ($j \in S$),

θ_{ij} 1, if candidate location j can cover customer i , 0 otherwise ($i \in C$, $j \in S$).

Variable:

u_j 1, if location j is chosen to open depot, 0 otherwise ($j \in S$).

The SP model is a binary linear problem (BLP) aiming at minimizing the cost of opening depots and then the mathematical model can be written as the following minimum set covering problem:

$$\text{Min} \quad \sum_{j \in S} f_j u_j \quad (5)$$

$$\text{Subject to:} \quad \sum_{j \in S} \theta_{ij} u_j \geq 1, \quad \forall i \in C \quad (6)$$

$$u_j \in \{0, 1\}. \quad \forall j \in S$$

The objective function (5) minimizes the initial cost of opening a depot, while all the customers are covered by at least one open depot (Constraint (6)). The parameter θ_{ij} is calculated based on drone maximum coverage range.

3.2 Operational Planning (OP)

This section explains assumptions made regarding the types of drones and parcels to deliver, and provides a route planning model. Based on the set of drone center locations given by the SP model in Section 3.1, the OP model finds the optimal assignment of drones to customers and their corresponding flight paths. The limitations on payload amount, battery endurance, BCR and the remaining charge requirement at the end of each flight path are included in the OP model. It is assumed that the batteries are fully charged before departure and battery level will decrease along the path according to BCR calculation provided in Section 2. **The following mathematical notation is defined to formulate the optimization model and obtain the routing planning mathematical model:**

Sets:

- C Set of customers,
- D Set of open depots,
- K Set of drones.

Parameters:

- M A large number,
- $MaxP_k$ Payload capacity of drone k ($k \in K$),
- $MinCh_k$ Minimum remaining battery level requirement for drone k ($k \in K$),
- t_{ijk} Flight time from node i to node j by drone k , ($i, j \in \{C \cup D\}, k \in K$),
- α_k, β_k Slope and intercept parameters of the BCR linear regression function for drone k ($k \in K$).

Variables:

- x_{ijk} 1 if drone k goes from node i to node j , 0 otherwise ($i, j \in \{C \cup D\}, k \in K$),
- h_k 1 if drone k is utilized in the network, 0 otherwise ($k \in K$),
- l_{ij} Payload from node i to node j , ($i, j \in \{C \cup D\}$),
- SOC_i State of charge (remaining battery level) at customer i location ($i \in C$),
- RC_k Remaining charge of drone k at return to depot ($k \in K$),
- y_c The order of sequence of visiting customer c in the path ($c \in C$).

$$\text{Min} \quad \sum_{k \in K} h_k \quad (7)$$

$$\text{s.t.} \quad \sum_{j \in \{C \cup D\}} \sum_{k \in K} x_{ijk} = 1, \quad \forall i \in C \quad (8)$$

$$\sum_{i \in \{C \cup D\}} \sum_{k \in K} x_{ijk} = 1, \quad \forall j \in C \quad (9)$$

$$\sum_{i \in C \cup D} x_{ijk} = \sum_{i \in \{C \cup D\}} x_{jik}, \quad \forall j \in C, \forall k \in K \quad (10)$$

$$\sum_{i \in C} x_{ijk} = \sum_{i \in C} x_{jik}, \quad \forall j \in D, \forall k \in K \quad (11)$$

$$\sum_{i \in D} \sum_{j \in C} x_{ijk} = h_k, \quad \forall k \in K \quad (12)$$

$$\sum_{i \in C} \sum_{j \in D} x_{ijk} = h_k, \quad \forall k \in K \quad (13)$$

$$\sum_{i \in \{C \cup D\}} l_{ij} - \sum_{i \in \{C \cup D\}} l_{ji} = d_j, \quad \forall j \in C \quad (14)$$

$$\sum_{i \in C} \sum_{j \in \{C \cup D\}} d_i x_{ijk} \leq MaxP_k h_k, \quad \forall k \in K \quad (15)$$

$$SOC_j \leq SOC_i - t_{ijk}(\alpha_k l_{ij} + \beta_k) + M(1 - x_{ijk}), \quad \forall i \in \{C \cup D\}, \forall j \in C, i \neq j, \forall k \in K \quad (16)$$

$$RC_k \leq SOC_i - t_{ijk}(\beta_k) + M(1 - x_{ijk}), \quad \forall i \in C, \forall j \in D, \forall k \in K \quad (17)$$

$$RC_k \geq MinCh_k, \quad \forall k \in K \quad (18)$$

$$y_i - y_j + n \sum_{k \in K} x_{ijk} \leq n - 1, \quad \forall i, j \in C \quad (19)$$

$$x_{ijk} \in \{0, 1\}, l_{ij}, SOC_i, RC_k \geq 0, l_{im} = 0, \quad \forall i, j \in \{C \cup D\}, \forall k \in K, \quad (20)$$

$$SOC_m = 100\%. \quad \forall m \in D.$$

The objective function (7) is to minimize the number of drones used in the network. Constraints (8) and (9) ensure that each customer is served only once by exactly one drone. Flow conservation is guaranteed via constraints (10) and (11) by which when a drone enters a node, it must leave the node and visit another one until it completes its delivery tour. Constraints (12) and (13) show the utilization of drones. Constraint (14) is to satisfy customer demand. Constraint (15) limits the total payload assigned to a drone up to its capacity. The state of charge of a drone during the flight and at the end of the path is calculated by constraints (16) and (17). At the beginning of the path, drone battery is completely charged (100%) and during the path, it will decrease based on travel time and the payload weight carried between each pair of nodes (Constraint (16)). The remaining battery level at returning to the depot is also stated in Constraint (17). Parameter M is a sufficiently large positive number that user specifies. On one hand, if the value of M is too large, it can increase the solution time. On the other hand, if the value is too small, the model can lose its optimality. Therefore, finding an appropriate value of M is important. In this paper, the value of M is determined by Formula (21) (explained in Appendix C).

$$M = 100 - \min_{k \in K} \{MinCh_k\} + \max_{k \in K} (\alpha_k MaxP_k + \beta_k) \cdot \max_{i, j \in \{C \cup D\}} t_{ijk}. \quad (21)$$

A threshold value for the battery level is considered in Constraint (18) to ensure a safe return to the depot from a flight without running out of battery. Constraint (19) is to eliminate any sub-tours in the network [34].

4 Solution Approach

The OP model in Section 3.2 is an extended version of Vehicle Routing Problem (VRP), which is known to be hard to solve [35]. Therefore, this section introduces methods to solve the OP model faster. By preprocessing in Section 4.1 we can fix some of the variables to 0 before solving the model so that the solution search space will be reduced, and it reduces the computational time. Section 4.2 introduces a primal bound generation method and Section 4.3 presents multiple dual bound generation methods for the OP model. Whenever the primal and dual bounds are equal, the optimal solution is obtained, otherwise, when the gap between them is less than a threshold value $\varepsilon > 0$, the objective function value is close to the optimal value within an accuracy of ε .

4.1 Variable Preprocessing Algorithm

The variable preprocessing procedure is implemented on variable x_{ijk} . Variable x_{ijk} has three indexes; i and j are for nodes in the network, and k represents a drone. The dimension of variable x_{ijk} is $(|C| + |D|)(|C| + |D|)(|K|)$. However, our experiments showed that many of them are zeros at the optimal solution. Consequently, the total number of non-zero variables can be less than $2|C|$ in the case each customer is served by one drone. The preprocessing procedure is introduced through statements (22), (23), and (25).

1) The model is an extension of VRP and Constraint (19) prevents sub-tours in the solution. A self-loop is an edge between a node and itself. Therefore, it is clear there is no self-loop in the solution.

Rule 1: $\forall i, j \in \{C \cup D\}, \forall k \in K :$

$$\text{if } i = j, \text{ then } x_{ijk} = 0. \quad (22)$$

2) The total payload capacity is limited to $MaxP_k$ for drone k . Thus, if the total demand of two customers exceeds the capacity, then they should be assigned to different drones.

Rule 2: $\forall i, j \in C, \forall k \in K :$

$$\text{if } d_i + d_j > MaxP_k, \text{ then } x_{ijk} = 0. \quad (23)$$

Note that if the path “Depot \rightarrow Customer $i \rightarrow$ Customer $j \rightarrow$ Depot” is infeasible due to weight capacity, then path “Depot \rightarrow Customer $j \rightarrow$ Customer $i \rightarrow$ Depot” is also infeasible because the summation of de-

mand in both paths is $d_i + d_j$, which is greater than drone weight capacity.

3) A feasible path should satisfy Constraint (18). The remaining battery level at the end of the path depends on time to travel and payload in each segment of the flight. Two customers can be assigned to drone k if the battery level is at least equal to $MinCh_k$. In an optimistic case, these two customers are the only customers to be served by drone k . Figure 4 shows a path consisting of two customers. The battery level in each step of the path can be determined by (24a)-(24e). According to Equation (24d), the remaining charge depends on time to travel between locations, payload, and drone specifications.

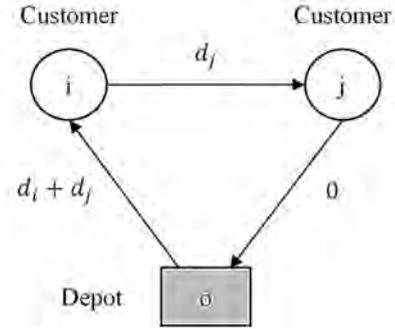


Fig. 4: Example of a path consisting of two customers

$$SOC_o = 100, \quad (24a)$$

$$\begin{aligned} SOC_i &= SOC_o - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] \\ &= 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k], \end{aligned} \quad (24b)$$

$$\begin{aligned} SOC_j &= SOC_i - t_{ijk}(\alpha_k d_j + \beta_k) \\ &= 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] - t_{ijk}(\alpha_k d_j + \beta_k), \end{aligned} \quad (24c)$$

$$\begin{aligned} RC_k &= SOC_j - t_{jok}\beta_k \\ &= 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] \end{aligned}$$

$$-t_{ijk}(\alpha_k d_j + \beta_k) - t_{jok}\beta_k, \quad (24d)$$

$$RC_k \geq MinCh_k$$

$$\begin{aligned} &\rightarrow 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] \\ &-t_{ijk}(\alpha_k d_j + \beta_k) - t_{jok}\beta_k \geq MinCh_k. \end{aligned} \quad (24e)$$

The SP model may choose multiple locations to establish depots. Therefore, the travel time between a depot and a customer should be checked for all the open depots. The feasibility of assigning each pair of customers to open depots regarding the remaining battery level is tested by (25). The rule 3 states if there is no drone to serve a pair of customers, the corresponding variable x_{ijk} should be fixed to zero.

Rule 3: $\forall i, j \in C, \quad \forall k \in K :$

$$\begin{aligned} &\text{if } \nexists o \in D : \quad t_{oik}[\alpha_k(d_i + d_j) + \beta_k] \\ &\quad + t_{ijk}(\alpha_k d_j + \beta_k) + t_{ojk}\beta_k \leq 100 - MinCh_k, \\ &\text{then } x_{ijk} = 0. \end{aligned} \quad (25)$$

Drones have to return to the launch depot at the end of the flight path. Note that even if the path “Depot \rightarrow Customer $i \rightarrow$ Customer $j \rightarrow$ Depot” is infeasible as a result of insufficient remaining battery level, then path “Depot \rightarrow Customer $j \rightarrow$ Customer $i \rightarrow$ Depot” can be feasible or infeasible as the battery consumption rates in these two paths are not necessarily the same, and depend on travel time between path segments and the payload.

4.2 Primal (Upper) Bound Generation

The objective function of the OP model is the minimization of number of drones so every feasible solution provides a primal bound. The location of base

centers is determined by solving the SP model. Then, Algorithm 1 is proposed here to find a feasible solution (a primal bound) for OP model.

Algorithm 1 Primal bound on the number of drones (a feasible solution)

- 1: **Inputs:**
 - 2: The number and location of base depots (result of SP model with objective function= ω)
 - 3: Parameters in OP problem
 - 4: **Step 1:**
 - 5: Assign each customer to the nearest depot.
 - 6: **Step 2:**
 - 7: Solve OP problem for each depot.
-

Step 1 assigns customers to the nearest open depot. As we solved a minimum set covering problem in Section 3.1 to find the location of centers; at least one customer is assigned to each depot. In Step 2, the OP problem is solved for each depot to find the number of required drones and their flight paths. The primal bound of the objective function for the OP problem is $\sum_{r=1}^{\omega} z_r$, where z_r is the optimal number of drones for depot r , $r = \{1, 2, \dots, \omega\}$.

4.3 Dual Bounds Generation

The OP model is to minimize the objective function. Hence, a lower bound is referred to as dual bound. The dual bounds for a minimization problem are usually found by a relaxation of the original model to a simpler one. We can either optimize over a larger feasible set or substitute the objective function by a term with a lower value everywhere. In the following, we introduce three dual bound generation methods: Lagrangian relaxation (Section 4.3.1), network configuration (Section 4.3.2) based bound, and weight capacity based bound (Section 4.3.3). Each of the methods has its own strengths

and weakness in terms of finding a tight dual bound and/or being able to solve the model much faster.

4.3.1 Lagrangian Relaxation

Some constraints of an optimization model are harder to satisfy than others. The general idea of a Lagrangian relaxation [36] is to remove the constraints that make the problem hard to solve and move them to the objective function with a penalty cost associated with them. The hard constraints are the most time-consuming constraints and depend on the model structure. Hence, it discourages constraint violation, while the resulting model is much easier to solve. Constraint (14) is a hard constraint in the OP model and we move this to the objective function. Accordingly, we relax the corresponding constraints with associated Lagrangian multipliers $\mu \in \{\mu_1, \dots, \mu_{|C|}\} \geq 0$. The resulting model is stated as:

$$L(\mu) = \text{Min} \sum_{k \in K} h_k + \sum_{j \in C} \mu_j (d_j - \sum_{i \in C \cup D} l_{ij} + \sum_{i \in C \cup D} l_{ji}), \quad (26)$$

Subject to: (8) – (13), (15) – (20).

The Lagrangian relaxation model is solved iteratively by updating μ in each iteration. As $L(\mu)$ is not differentiable at all points, a subgradient algorithm described in Algorithm 2, is used.

The objective function states the number of drones to use, and the value cannot be fractional. Hence, the dual bound obtained by Lagrangian relaxation should be rounded up to the nearest integer that is larger than the resulting optimal objective value if it is fractional.

Algorithm 2 Subgradient algorithm

Inputs:

The number and location of open depots (result of SP problem), Parameters in OP problem, Primal bound obtained from Algorithm 1, μ_0 , θ , max_iteration, threshold-value.

While($k < \text{max_iteration}$ or $\Delta\mu > \text{threshold-value}$)

Do {

Step 1: solve the Lagrangian dual problem (26) with Constraints (8)-(13), (15)-(20) to obtain the optimal solution;

Step 2: direction = $d_j - \sum_{i \in C \cup D} l_{ij} + \sum_{i \in C \cup D} l_{ji}$ (gradient of $L(\mu_k)$);

Step 3: step size = $\theta \cdot \frac{\text{upper_bound} - L(\mu_k)}{\|\text{direction}\|}$;

Step 4: update μ :

$$\mu_{k+1} = \max\{0, \mu_k + \text{direction} \cdot \text{step size}\};$$

Step 5: converged or not? $\Delta\mu = |\mu_k - \mu_{k+1}|$;

Step 6: $k = k + 1$;

} End While

4.3.2 Network Configuration

Two customers are considered to be incompatible if they cannot be assigned to one drone due to any limitations. This situation happens if either Rule 2 or Rule 3 in Section 4 is true for a pair of customers i and j , and it is denoted as $i||j$. The incompatibility graph represented by $G_{inc} = (C, E_C)$, where C is the set of customers and $E_C = (i, j) \in C \cdot C : i||j$ is constructed by rules 2 and 3. In graph G_{inc} , an arc represents a pair of incompatible customers, in which different drones are needed to serve them. Therefore, a complete subgraph with m number of vertices in graph G_{inc} shows m incompatible customers, who need different drones to be served. In such a subgraph, m drones are needed to serve the customers in the subgraph because each pair of these customers are incompatible with each other. Therefore, the largest complete subgraph in G_{inc} represents the largest subset of customers that all of them are incompatible with each other.

In an undirected graph, a complete subgraph is called a clique and a maximum clique is a clique with the largest possible number of vertices [36]. Therefore,

the maximum clique in graph G_{inc} shows the largest subset of incompatible customers, that each customer needs a separate drone to be served. The size of the maximum clique in graph G_{inc} is a dual bound for the number of required drones. This problem is a well-known problem and there are some algorithms to solve it fast. Here, we use Bron-Kerbosch maximal clique finding algorithm [37].

4.3.3 Drone Weight Capacity

The assignment of customers to drones has to meet drone weight capacity. Constraint (15) is related to drone capacity limitation in the OP model. This section introduces a dual bound generation based on the drone weight capacity limitation. A new variable is defined and the OP model is relaxed as follow: We define the variable $\rho_{ik} = \sum_{j \in \{CUD\}} x_{ijk}$ as a binary variable that gets value 1 if drone k serves customer i , and 0 otherwise. Constraint (8) and Constraint (15) in the OP model are simplified and rewritten by variable ρ_{ik} :

$$\begin{aligned} \text{Constraint (8): } \sum_{j \in \{CUD\}} \sum_{k \in K} x_{ijk} &= \sum_{k \in K} \left(\sum_{j \in \{CUD\}} x_{ijk} \right) \\ &= \sum_{k \in K} \rho_{ik} = 1. \end{aligned} \quad (27)$$

$$\begin{aligned} \text{Constraint (15): } \sum_{i \in C} \sum_{j \in \{CUD\}} d_i x_{ijk} &\leq \text{MaxP}_k h_k \\ \rightarrow \sum_{i \in C} d_i \left(\sum_{j \in \{CUD\}} x_{ijk} \right) &\leq \text{MaxP}_k h_k, \\ \rightarrow \sum_{i \in C} d_i \rho_{ik} &\leq \text{MaxP}_k h_k. \end{aligned} \quad (28)$$

Here, customers have different amounts of demand, and their demand should be prepared and load to drones considering the weight capacity to minimize the number of drones. This problem can be cast as a Bin

Packing Problem (BPP) [38] that consists of drones as bins with a capacity of MaxP_k , and a customer demand as an object with a size of d_i , and it is stated as follows:

$$\text{BPP-weight: } \text{Min} \sum_{k \in K} h_k \quad (29)$$

$$\text{Subject to: } \sum_{k \in K} \rho_{ik} = 1 \quad \forall i \in C \quad (30)$$

$$\sum_{i \in C} d_i \rho_{ik} \leq \text{MaxP}_k h_k \quad \forall k \in K \quad (31)$$

$$\rho_{ik}, h_k \in \{0, 1\} \quad \forall i \in C, \forall k \in K$$

The objective function (29) minimizes the number of drones used for delivery. Constraint (30) guarantees each customer is served by one drone while the weight capacity of a drone is satisfied through Constraint (31). Feasible region of the OP is a subset of the feasible region of the BPP-weight because the BPP-weight is a relaxed form of the OP model. Hence, the optimal objective function value of the OP model is at least as large as the optimal objective function value of the BPP-weight.

5 Numerical Results

This section begins with a case study to demonstrate how the proposed methods work (Section 5.1). Further experiments are conducted to understand the impact of considering BCR with respect to the payload amount in drone scheduling (Section 5.2), and investigate the computational efficiency of the proposed solution approach (Section 5.3). The Bron-Kerbosch Maximal clique finding algorithm [37] is implemented in MATLAB [39]. The other algorithms and the proposed SP and OP models are implemented in GAMS [40] and solved by CPLEX 12.6.3. [41]. All computational ex-

periments were conducted using a Linux server with 24 cores and 384GB RAM.

5.1 A Case Study

A random test network having 20 customers and 5 depot candidate locations is shown in Figure 5, where square nodes are candidate depots and circle nodes represent the customers. A homogeneous fleet of drones similar to the drone used in Section 2 is used to serve the customers. According to the results of Section 2, the linear relationship between BCR and the payload amount is $BCR = (2.297)payload + 3.879$.

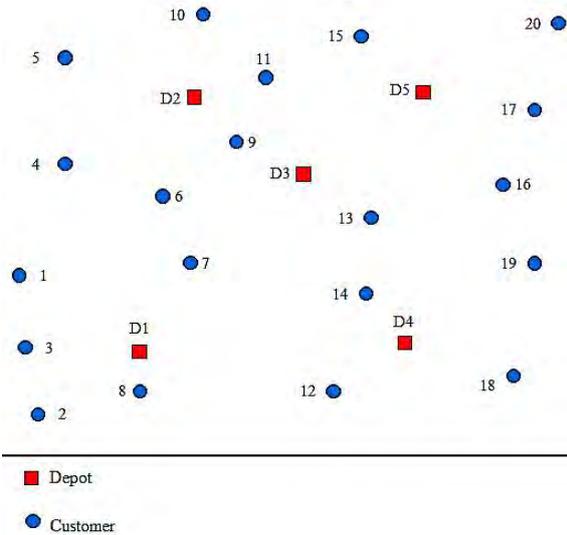


Fig. 5: A random test network with 20 customers and 5 candidate locations

First, the covering range of each depot candidate location is identified as shown in Figure 6 and Table D in Appendix D. This is determined by the drone battery specifications, the minimum required battery level of 15% and the maximum payload capacity of 1 lb. Second, a subset of candidate locations is determined by the proposed SP model in Section 3.1. As can be seen from Figure 6, some customers can be covered by

just one candidate location (e.g., Customer 2), some others (e.g., Customer 1) can be covered by multiple locations (D1, D2, and D3), and yet others can be covered by all depot candidates such as Customer 9. According to the result of the SP model in Section 3.1, two (D1 and D3) out of five candidate locations are selected to establish depots there. Third, the optimal as-

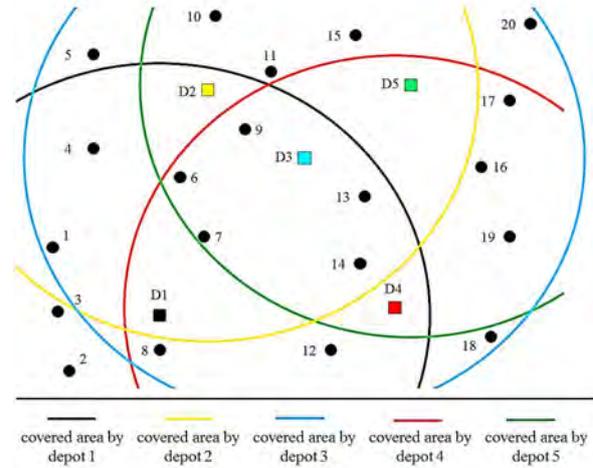


Fig. 6: Covering customers by depot candidate locations

segment of customers and drone paths are determined by the OP problem as discussed in Section 3.2. We apply both the primal bound and dual bound generation methods on the objective function (Section 4). This step is important in reducing the computational time; our initial test run of more than 24 hours returned an objective function value of 10 for the problem instance with 18% relative optimality gap, i.e., $relative\ gap = \frac{PrimalBound - DualBound}{PrimalBound} \cdot 100$.

– Primal Bound Generation

Step 1: Assign customers to the nearest depot using Algorithm 1. Hence, customers 1, 2, 3, 7, 8, and 12 are assigned to D1 and the rest of them are assigned to D3.

Step 2: Solve OP problem for each depot to get the optimal paths.

The experiments took 20.82 minutes with 0% relative optimality gap to solve the OP problem for both depots. The results are presented in Table 3, in which four drones are assigned to Depot 1 and six drones are needed in D3.

Table 3: Primal bound calculation (a feasible solution) for the case study

Depot	Path
Depot 1	$D1 \rightarrow 1 \rightarrow 3 \rightarrow D1$
	$D1 \rightarrow 2 \rightarrow D1$
	$D1 \rightarrow 7 \rightarrow 12 \rightarrow D1$
	$D1 \rightarrow 8 \rightarrow D1$
Depot 3	$D3 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow D3$
	$D3 \rightarrow 6 \rightarrow 14 \rightarrow D3$
	$D3 \rightarrow 9 \rightarrow 19 \rightarrow 13 \rightarrow D3$
	$D3 \rightarrow 11 \rightarrow 17 \rightarrow 16 \rightarrow D3$
	$D3 \rightarrow 15 \rightarrow 20 \rightarrow D3$
	$D3 \rightarrow 18 \rightarrow D3$
Primal Bound	10

– Dual Bound Generation

Different dual bounds are calculated according to Section 4.3 and the final dual bound on the objective function of OP is the maximum value of them. Table 4 shows the values of the proposed dual bounds. The dual bound based on Lagrangian relaxation is 8.54, which is rounded up because the objective function (number of drones) should be an integer value. The solution of the maximum clique problem has a size of 7, consisting of customers 2, 3, 5, 8, 14, 19, and 20. The final dual bound is $\max\{9, 7, 8\} = 9$.

– Discussions on the Results from the OP Model

The optimal assignment of customers and drone paths is determined as shown in Figure 7, in which 3 drones are needed in D1 and 6 drones in D3. The

drone flight paths start and finish in the same depot and along the flight, they serve 2 or 3 customers. Although a customer might be covered by more than one depot, just one of them can serve the customer in the optimal solution and it is not the closest depot necessarily. For example, in the optimal solution of our case study, customer 6 and 7 are assigned to D1 and D3, respectively; however, they are closer to the D3 and D1, respectively. The primal bound generation assigns each customer to the nearest depot and 10 drones are needed to serve all customers. Using this bound, the optimal solution lowered the drone count to 9, which is 10% improvement from the original bound. Table 5 shows the details of optimal flight paths, which includes the total demand, total travel time, and the remaining battery level at the end of each flight path.

Table 4: Dual bound calculation for the case study

Dual bound generation method	Dual bound value
Lagrangian relaxation	$\lceil 8.54 \rceil = 9$
Network configuration	7
BPP-weight	8
Dual Bound	$\max\{9, 7, 8\} = 9$

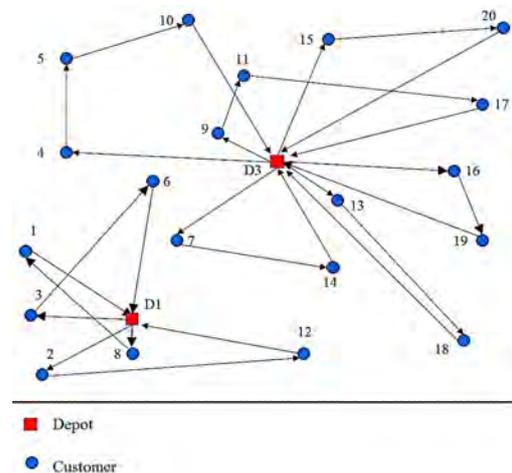


Fig. 7: Optimal flight paths- result of OP model for the case study network

5.2 Impact of BCR on the Drone Flight Scheduling

We investigated the impact of considering the BCR on drone flight paths using a reverse path concept and different philosophies of estimating the BCR.

– Reverse of a Path

The reverse of a path has an opposite direction of the primary flight path and moves backward. For example, the flight path of drone 1 in Table 5, is $D1 \rightarrow 8 \rightarrow 1 \rightarrow D1$ and the reverse path is $D1 \rightarrow 1 \rightarrow 8 \rightarrow D1$. Although a flight path and its reverse path have the same total flight time and total assigned demand, they are not the same in terms of the BCR

calculations. It is due to the fact that the flight distance between location i and j is the same as flight distance between location j and i , but the carrying payload and therefore the BCR on the flight segment can be different by the flight direction. Table E in Appendix E shows the optimal flight paths in the case study (Figure 7) and the reverse of them. By taking into account the threshold value of 15% for the final remaining charge, the reverse of a feasible path might be infeasible as it is for drones 2, 4, 6, 7, and 8, which means more than half of the optimal flight paths (55.6%) are infeasible if the reverse paths are used.

Table 5: Optimal drone flight path for the case study

Drone	Path	Total demand (lb.)	Total travel time (min)	Remaining battery level (%)
1	$D1 \rightarrow 8 \rightarrow 1 \rightarrow D1$	1 (0.3+0.7)	11.92	42.03
2	$D1 \rightarrow 2 \rightarrow 12 \rightarrow D1$	0.9 (0.5+0.4)	16.88	19.93
3	$D1 \rightarrow 3 \rightarrow 6 \rightarrow D1$	1 (0.6+0.4)	12.23	42.21
4	$D3 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow D3$	0.7 (0.3+0.3+0.1)	18.07	17.25
5	$D3 \rightarrow 7 \rightarrow 14 \rightarrow D3$	1 (0.6+0.4)	12.53	39.05
6	$D3 \rightarrow 9 \rightarrow 11 \rightarrow 17 \rightarrow D3$	1 (0.3+0.4+0.3)	15.81	26.84
7	$D3 \rightarrow 13 \rightarrow 18 \rightarrow D3$	0.6 (0.2+0.4)	18.35	19.38
8	$D3 \rightarrow 15 \rightarrow 20 \rightarrow D3$	1 (0.7+0.3)	17.28	18.93
9	$D3 \rightarrow 16 \rightarrow 19 \rightarrow D3$	0.8 (0.3+0.5)	14.11	33.34

– Fixed Total Flight Time Regardless of the Payload Amount

Drones can fly for a limited time before needing to land to recharge. BCR is the rate at which battery charge decreases during the flight (see Section 2) so the higher amount of BCR means the battery level decreases faster and the total flight time will be lower. Therefore, the total flight time has a reverse relationship with the BCR. As mentioned in Section 1, most of the studies in the literature do not consider BCR in scheduling and a fixed value for the

limitation of total flight time is included. In this section, the solutions provided by a fixed value for the total flight time are evaluated. Two extreme cases for the total flight time are considered here: flight time based on maximum and minimum carried payload. On one hand, for a specific drone, the BCR has the lowest value and it can fly longer when it has no payload. On the other hand, the BCR has the highest value if it is fully loaded. For the drone used in Section 2, the total flight time is 13.76 minutes with a payload amount of 1 lb. (maximum payload amount)

and it is 21.92 minutes when it does not carry any payload. A subset of nodes in Figure 7 are taken as a test case (D3 and customers 4, 5, 6, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, and 20) to expedite the computational experiments for this section. The optimal objective function value of the OP model is 6 for this test case.

This problem is infeasible with the total flight time of 13.76 minutes because the drone is not able to complete the path “ $D3 \rightarrow customer \rightarrow D3$ ” for some of the customers (e.g., customer 18 and 20) within

the total flight time. In this case, although drones can return to the depot before running out of battery, the energy consumption is considered to be pessimistic and prevents us from serving all the customers. In the other case, the objective function value is 5 with the total flight time of 21.92 minutes and the optimal flight paths are presented in Table 6. These paths can meet the limitation of the total flight time but regarding the remaining charge, most of them (60%) are infeasible and 6 drones will run out of battery before landing at the depot.

Table 6: Optimal paths with the minimum BCR

Drone	Path	Total demand (lb.)	Total travel time (min)	Remaining battery level (%)
1	$D3 \rightarrow 15 \rightarrow 20 \rightarrow D3$	1	17.28	18.93
2	$D3 \rightarrow 19 \rightarrow 18 \rightarrow D3$	0.9	19.36	8.12 *
3	$D3 \rightarrow 4 \rightarrow 9 \rightarrow 11 \rightarrow D3$	1	14.54	23.21
4	$D3 \rightarrow 16 \rightarrow 17 \rightarrow 14 \rightarrow D3$	1	20.12	-0.6 *
5	$D3 \rightarrow 13 \rightarrow 6 \rightarrow 5 \rightarrow 10 \rightarrow D3$	1	21.40	-2.18 *

* Infeasible flight path

5.3 Computational Efficiency of Proposed Solution

Method

– Dual Bound Generation Methods

In this section, the performance of the proposed dual bounds (Section 4.3) is tested on six different problems that have two depots and 11 customers and a homogeneous fleet of drones stated in Table 7. The difference between the test case 1 and the 5 other test cases is the travel time and demand parameters which are a ratio of the test case 1 parameters. The ratio of each test case parameters to the test case 1 are presented in the first section of Table 7. The second section of the table shows the number of variable x_{ijk} reductions by using prepro-

cessing rules (Section 4.1). Rule 2 and Rule 3 show the incompatibility among customers regarding the drone weight capacity and the remaining charge, respectively. Some of the variable x_{ijk} are fixed by more than one rule in the preprocessing so the total number of variables fixed by preprocessing algorithm is not necessarily the summation of reductions by three rules.

The obtained dual bounds by using three different dual bound generation methods are presented in Table 8. The larger value for the dual bound is better because it provides a smaller gap between primal and dual bounds. The bold numbers in the table show the largest dual bound value for each

Table 7: Test cases used in comparing the dual bound methods

Test case characteristics	Test case					
	1	2	3	4	5	6
Ratio to the test case 1 for:						
Customer demand	1	1.5	2	1	1.5	2
Travel time	1	1	1	0.74	0.74	0.74
Number of variables fixed by:						
Rule 1 (self-loop)	11	11	11	11	11	11
Rule 2 (weight)	0	30	66	0	30	66
Rule 3 (battery)	84	95	101	16	28	47
Total	95	112	117	27	60	95

test case. As it can be seen, Lagrangian relaxation algorithm has a good performance and can provide the largest dual bound in all the cases except test

case number 5. The network configuration and the BPP-weight method give the largest dual bound in 66.7% and 50% of these cases, respectively.

Table 8: The bounds obtained by dual bound methods for different test cases

Dual bound generation method	Test case					
	1	2	3	4	5	6
Lagrangian relaxation	5	8	9	4	5	7
Network configuration	5	8	9	2	5	7
BPP-weight	4	6	7	4	6	7
Dual bound	5	8	9	4	6	7
Primal bound	7	8	9	4	7	8

Table 9 presents the total computational time in seconds to get the dual bound by each dual bound generation method and solve the test case by using that bound. The bold numbers in the table show the lowest total computational time for each test case.

Note that for these problems the preprocessing algorithm is not used in running each test case to be able to capture the effect of bound generation methods.

Table 9: Computational time (second) with different dual bound generation methods

Dual bound generation methods	Solved problem	Test case					
		1	2	3	4	5	6
Lagrangian relaxation	Dual bound	6,901.44	36.17	0.45	25,210.70	15,906.14	30.60
	Test case	880.87	0.33	0.24	18.01	12,722.80	0.22
	Total	7,782.31	36.50	0.69	25,228.71	28,628.94	30.82
Network configuration	Dual bound	0.023	0.019	0.023	0.018	0.018	0.018
	Test case	880.87	0.33	0.24	5.41	12,722.80	0.22
	Total	880.89	0.33	0.26	5.42	12,722.81	0.23
BPP-weight	Dual bound	0.28	0.16	0.15	0.34	0.21	0.11
	Test case	1280.94	0.30	0.30	18.01	0.88	0.22
	Total	1281.22	0.46	0.45	18.35	1.09	0.33

Although Lagrangian relaxation method provides good dual bounds, its computational time is higher than the two other methods for all the test cases and make it ineffective in practice. As it can be seen, the computational time by [Lagrangian relaxation increased](#) if the total incompatibility among the customers decreases. For example test cases 4, 5, and 1 that have the highest computational time by Lagrangian relaxation algorithm in Table 9, have the lowest number of variables fixed by the preprocessing algorithm in Table 8 too. In total, the Lagrangian relaxation algorithm is not suggested to be used due to the high computational time especially for the data parameters with low incompatibility.

The dual bound generation methods based on network configuration and the BBP-weight provide better dual bounds in a reasonable time. The computational time for the network configuration problem to find the dual bound has a low variability and depends less on the parameters. Regarding the total computational time, it has the lowest runtime in all the cases except for test case 5 (83.3% of the cases). The computational time of BPP-weight problem is related to the incompatibility among customers regarding the weight capacity of drones. In test case 1 and 4, no variable is fixed by Rule 2 in Table 9 and these cases have the highest computational time by the BPP-weight problem in Table 8. As the number of reduced variables by Rule 2 increases the BPP-weight computational time decreases.

Overall, the computational time of the bound generation methods decreases if the incompatibility among customers increases. The required time to

solve each test case based on the obtained dual bound depends on the quality of the dual bound. The higher dual bound is better and results in less computational time to solve the test case. The network configuration dual bound generation method outperforms the two other methods because it provides good dual bounds (the highest bound for 66.7% of the cases), has low computational time (the lowest time in 83.3% of the cases), and depends less on the demand and travel time parameters.

– Computational Efficiency of Preprocessing and Bound Generation Methods

This section examines the computational efficiency of preprocessing and bound generation methods proposed in Section 4. Three randomly generated problems with different sizes are tested here: 1 depot- 6 customers, 2 depots-10 customers, and 1 depot-14 customers. The results are presented in Table 10, with the optimality relative gap of 5% and the CPU run time of 3600 seconds (i.e., 1 hour). The name of each test problem shows the number of depots and number of customers, respectively. The second and third columns show whether bounds on the objective function and variable preprocessing algorithm is used for the problem or not. The CPLEX solver could not find an optimal solution within 1 hour of running time for the last two test problems. However, both the variable preprocessing and bound generation helped reduce the computational time significantly for all three test problems (70.1% for the case problem with one depot and 6 customers). The first test problem is small

Table 10: Effect of preprocessing and bound generation methods on the computational time

Problem	Bounds	Preprocessing	Computational time (s)	MIP objective function value	Relative optimality gap
1-6	N	N	3.11	4 **	0 %
	Y	N	3.10	4 **	0 %
	N	Y	1.30	4 **	0 %
	Y	Y	0.93	4 **	0 %
2-10	N	N	3600.00	7 *	14.28 %
	Y	N	2783.74	7 **	0 %
	N	Y	42.12	7 **	0 %
	Y	Y	30.85	7 **	0 %
1-14	N	N	3600.00	7 *	14.28 %
	Y	N	3600.00	6 *	14.28 %
	N	Y	1568.43	6 **	0 %
	Y	Y	12.96	6 **	0 %

* Objective function value for an integer feasible solution

** Objective function value for the integer optimal solution

size, in which all four problems are able to reach an optimal solution with 0% optimality relative gap. Note that problem “1-6-Y-Y” returned the lowest computational time. For the second case, all four problems gave the objective function of 7; however, problem “2-10-N-N” cannot recognize the optimality and there is still an optimality gap (14.28%). In the third test case, problem “1-14-N-N” and “1-14-Y-N” are not able to meet the stopping criteria (relative gap $\leq 5\%$) within 1 hour of running. According to the results of Table 10, as the size of the problem increases, it is more important to use the proposed solution algorithm to reduce the computational time and obtain the optimal solution. For the smallest test case (problem “1-6”) even without the bound and the preprocessing algorithms, the optimal solution can be obtained in a few seconds (3.11 seconds). But when the size of problem increases (problems “2-10” and “1-14”), we are not able to get the optimal solution within 1 hour of running without the proposed solution algorithm.

Furthermore, it can be seen that the impact of preprocessing on the computational time reduction is more than the impact of primal and dual bound generation methods. For all three test problems, the computational time for “N-Y” cases (just using the preprocessing algorithm) is lower than the “Y-N” cases (just using the bound generation algorithm). Using variable preprocessing in comparison to using the bound generation algorithm reduces the computational time 58.06% for the test problem “1-6” and 98.48% for the “2-10” test problem. In problem “1-14”, we are not able to get the optimal solution within 1 hour if we do not use the variable preprocessing technique.

6 Conclusion

A delivery application of drones was studied in this paper, in which a group of drones was considered to deliver parcels to customers. A primary focus was given to understand the impact of the drone battery consumption on the design of a drone-based parcel delivery sys-

tem. Among factors affecting the drone battery consumption, the payload amount and flight time were two factors studied in this paper. Based on actual experiments using a drone, we showed that there is a linear relationship between the BCR and the payload amount. Based on the linear regression model, two planning optimization models were proposed to find the depot locations and drone flight paths for delivery. The SP model was to find the depot locations by optimizing a set covering problem and the OP model was proposed to determine the assignment of customers to depots and flight paths by including the drone battery endurance as constraints in the routing optimization problem. The preprocessing algorithm and several bound generation methods were proposed to improve the computational time. The proposed models and the solution method were implemented in a case study. The numerical results showed that (1) up to 60% of the flight paths generated without considering the BCR ended up fail to complete the delivery trips due to insufficient battery duration, (2) reversing the flight paths for visiting the same subset of customers could result in insufficient battery duration to complete the deliveries.

Our initial test runs revealed that solving the OP model can be computationally challenging as there are more customers to cover. Hence, a primal bound generation algorithm, as well as three different dual bound methods, are developed and their performance was compared. The efficiency of the proposed solution algorithm in reducing the computational time was shown through several randomly generated network. The total computational times of Lagrangian relaxation and the BPP-weight method depend on the incompatibility among the customers. The dual bound by network con-

figuration method computationally outperformed the other two methods. Furthermore, for all test problems, the impact of preprocessing algorithm coupled with the bound generation methods enabled us to solve all test problems, which was not possible without these methods. One can extend this work by including other factors affecting the BCR such as flight speed and environmental conditions.

References

1. Amazon Inc., "Amazon prime air," (access date: July, 2018). [Online]. Available: www.amazon.com/primeair
2. T. Keeney, "How can amazon charge \$1 for drone delivery?" (access date: July, 2018). [Online]. Available: <https://ark-invest.com/research/drone-delivery-amazon>
3. Mercedes-Benz co., "Vans & drones in zurich," (access date: July, 2018). [Online]. Available: <https://www.mercedes-benz.com/en/mercedes-benz/vehicles/transporter/vans-drones-in-zurich/>
4. M. McFarland, "Ups drivers may tag team deliveries with drones," 2007 (access date: July, 2018). [Online]. Available: <http://money.cnn.com/2017/02/21/technology/ups-drone-delivery/index.html>
5. DHL, "Successful trial integration of dhl parcelcopter into logistics chain," 2016 (access date: July, 2018). [Online]. Available: http://www.dhl.com/en/press/releases/releases_2016/all/parcel_ecommerce/successful_trial_integration_dhl_parcelcopter_logistics_chain.html

6. S. Omidshafiei, A.-a. Agha-mohammadi, C. Amato, S.-Y. Liu, J. P. How, and J. L. Vian, "Health-aware multi-uav planning using decentralized partially observable semi-markov decision processes," in *AIAA Infotech@ Aerospace*, 2016, p. 1407.
7. J. Enright, E. Frazzoli, K. Savla, and F. Bullo, "On multiple uav routing with stochastic targets: Performance bounds and algorithms," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, p. 5830.
8. P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *IEEE robotics & automation magazine*, vol. 17, no. 4, pp. 70–77, 2010.
9. S. J. Kim, G. J. Lim, J. Cho, and M. J. Côté, "Drone-aided healthcare services for patients with chronic diseases in rural areas," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 163–180, 2017.
10. Y. Kim, D.-W. Gu, and I. Postlethwaite, "Real-time optimal mission scheduling and flight path selection," *IEEE Transactions on Automatic control*, vol. 52, no. 6, pp. 1119–1123, 2007.
11. C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.
12. Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 597–621, 2018.
13. J. G. Carlsson and S. Song, "Coordinated logistics with a truck and a drone," *Management Science*, 2017.
14. C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.
15. M. Torabbeigi, G. J. Lim, and S. J. Kim, "Drone delivery schedule optimization considering the reliability of drones," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 1048–1053.
16. S. J. Kim and G. J. Lim, "Drone-aided border surveillance with an electrification line battery charging system," *Journal of Intelligent & Robotic Systems*, pp. 1–14, 2018.
17. S. Kim and G. Lim, "A hybrid battery charging approach for drone-aided border surveillance scheduling," *Drones*, vol. 2, no. 4, p. 38, 2018.
18. I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 198–212, 2018.
19. E. E. Yurek and H. C. Ozmutlu, "A decomposition-based iterative optimization algorithm for traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 249–262, 2018.
20. V. Olivares, F. Cordova, J. M. Sepúlveda, and I. Derpich, "Modeling internal logistics by using drones on the stage of assembly of products," *Procedia Computer Science*, vol. 55, pp. 1240–1249, 2015.

21. S. J. Kim, N. Ahmadian, G. J. Lim, and M. Torabbeigi, "A rescheduling method of drone flights under insufficient remaining battery duration," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 468–472.
22. G. J. Lim, S. Kim, J. Cho, Y. Gong, and A. Kho- daei, "Multi-uav pre-positioning and routing for power network damage assessment," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3643–3651, 2018.
23. J. Scott and C. Scott, "Drone delivery models for healthcare," in *Hawaii International Conference on System Sciences*, 2017, pp. 3297–3304.
24. S. J. Kim, G. J. Lim, and J. Cho, "Drone flight scheduling under uncertainty on battery duration and air temperature," *Computers & Industrial Engineering*, vol. 117, pp. 291–302, 2018.
25. E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "The load-dependent vehicle routing problem and its pick-up and delivery extension," *Transportation Research Part B*, vol. 71, pp. 158–181, 2015.
26. Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu, "Development of a fuel consumption optimization model for the capacitated vehicle routing problem," *Computers and Operation Research*, vol. 39, no. 7, pp. 1419–1431, 2012.
27. K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 1–16, 2017.
28. A. Abdilla, A. Richards, and S. Burrow, "Power and endurance modelling of battery-powered rotorcraft," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 675–680.
29. Z. Liu, B. California, and A. Kurzhanskiy, "A power consumption model for multi-rotor small unmanned aircraft systems," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 310–315.
30. F. Cheng, W. Hua, and C. Pin, "Rotorcraft flight endurance estimation based on a new battery discharge model," *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1561–1569, 2017.
31. L. SZ DJI Technology Co., "Phantom 4 pro," (access date: July, 2018). [Online]. Available: <https://www.dji.com/phantom-4-pro>
32. Handbook, Helicopter Flying, "FAA-H-8083-21A," 2012.
33. M. Pirhooshyaran and L. V. Snyder, "Optimization of inventory and distribution for hip and knee joint replacements via multistage stochastic programming," in *Modeling and Optimization: Theory and Applications*. Springer, Cham, 2017, pp. 139–155.
34. C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM*, vol. 7, no. 4, pp. 326–329, 1960.
35. T. Caric and H. Gold, *Vehicle routing problem*. In- Teh, 2008.
36. L. A. Wolsey, "Integer programming," *IIE Transactions*, vol. 32, no. 273-285, pp. 2–58, 2000.

37. C. Bron and J. Kersch, “Algorithm 457: finding all cliques of an undirected graph,” *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.
38. B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. Algorithms and Combinatorics, 2006.
39. *MATLAB and Statistics Toolbox Release R2016a* The MathWorks, Inc., Natick, MA, USA.
40. GAMS Development Corporation. *General Algebraic Modeling System (GAMS) Release 24.7.3*. Washington, DC, USA, 2016. [Online]. Available: <http://www.gams.com/>
41. IBM ILOG, “CPLEX reference manual,” vol. 12.6.3.0, Released: July 2016. [Online]. Available: <http://www.ilog.com>

Appendices

Appendix A

Table A: Phantom 4 Pro+ specifications used in data collection

Specification	Value
Drone type	Phantom 4 Pro+
Battery type	LiPo 4S
Net weight of drone (including one battery and 4 propellers)	3.06 pounds

Appendix B

The data was collected when the battery’s SOC was between 95% and 15%. For example, it took 2.35 minutes for the battery charge to drop from 95% to 85% when the payload was 0.22 lb.

Table B: Flight time data (in minutes) collected with Phantom 4 Pro+

Battery’s SOC (%)	Payload amount				
	0 lb.	0.220 lb.	0.441 lb.	0.661 lb.	0.882 lb.
95	0.00	0.00	0.00	0.00	0.00
90	1.45	1.28	1.10	1.01	0.74
85	2.65	2.35	2.03	1.90	1.61
80	4.06	3.61	3.11	2.96	2.60
75	5.48	4.85	4.23	4.02	3.54
70	6.84	6.05	5.23	4.97	4.42
65	8.01	7.08	6.16	5.80	5.20
60	9.39	8.26	7.30	6.80	6.11
55	10.77	9.46	8.35	7.69	7.04
50	12.07	10.62	9.33	8.59	7.84
45	13.19	11.58	10.22	9.41	8.61
40	14.54	12.75	11.30	10.35	9.50
35	15.90	13.90	12.19	11.30	10.33
30	17.15	15.03	13.16	12.18	11.15
25	18.32	16.02	14.07	13.07	11.88
20	19.62	17.18	15.12	14.03	12.69
15	20.93	18.32	16.12	15.00	13.57

Appendix C

The parameter M should be large enough that does not eliminate any feasible solution. This parameter appears

in Constraints (16) and (17).

$$\text{Constraints(16)} : \forall i \in \{C \cup D\}, \forall j \in C, i \neq j, \forall k \in K :$$

$$SOC_j \leq SOC_i - t_{ijk}(\alpha_k l_{ij} + \beta_k) + M(1 - x_{ijk}),$$

$$\rightarrow M \geq \frac{SOC_j - SOC_i + t_{ijk}(\alpha_k l_{ij} + \beta_k)}{1 - x_{ijk}}$$

if $x_{ijk} = 1$, then $M \geq 0$

if $x_{ijk} = 0$, then

$$M \geq \max_{i,j,k} \{SOC_j - SOC_i + t_{ijk}(\alpha_k l_{ij} + \beta_k)\}.$$

$$\rightarrow M \geq \max_j \{SOC_j\} - \min_i \{SOC_i\} +$$

$$\max_{i,j,k} \{t_{ijk}\} \cdot \max_{i,j,k} \{\alpha_k l_{ij} + \beta_k\}$$

$$\rightarrow M \geq 100 - \min_k \{MinCh_k\} +$$

$$\max_{i,j,k} \{t_{ijk}\} \cdot \max_k \{\alpha_k MaxP_k + \beta_k\}$$

The same for the Constraint (17).

Appendix D

In the test case problem, there are 5 candidate locations to open depots. Table Table D shows whether a customer is within the covering range of each candidate location (value of 1) or not (value of 0).

Table D: Coverage area by each candidate location in the test case problem

Customer	Candidate locations				
	1	2	3	4	5
1	1	1	1	0	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	1	1	0	0
5	0	1	1	0	0
6	1	1	1	1	1
7	1	1	1	1	1
8	1	0	1	1	0
9	1	1	1	1	1
10	0	1	1	0	1
11	0	1	1	0	1
12	1	0	1	1	0
13	1	1	1	1	1
14	1	1	1	1	1
15	0	1	1	0	1
16	0	0	1	1	1
17	0	0	1	1	1
18	0	0	1	1	0
19	0	0	1	1	1
20	0	0	1	0	1

Appendix E

Table E: Path reverse of optimal solution for the case study

Path	Remaining charge (%)
Drone 1: $D1 \rightarrow 8 \rightarrow 1 \rightarrow D1$	42.03
Reverse: $D1 \rightarrow 1 \rightarrow 8 \rightarrow D1$	38.14
Drone 2: $D1 \rightarrow 2 \rightarrow 12 \rightarrow D1$	19.93
Reverse: $D1 \rightarrow 12 \rightarrow 2 \rightarrow D1$	14.22 *
Drone 3: $D1 \rightarrow 3 \rightarrow 6 \rightarrow D1$	42.21
Reverse: $D1 \rightarrow 6 \rightarrow 3 \rightarrow D1$	35.03
Drone 4: $D3 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow D3$	17.25
Reverse: $D3 \rightarrow 10 \rightarrow 5 \rightarrow 4 \rightarrow D3$	13.52 *
Drone 5: $D3 \rightarrow 7 \rightarrow 14 \rightarrow D3$	39.05
Reverse: $D3 \rightarrow 14 \rightarrow 7 \rightarrow D3$	34.92
Drone 6: $D3 \rightarrow 9 \rightarrow 11 \rightarrow 17 \rightarrow D3$	26.84
Reverse: $D3 \rightarrow 17 \rightarrow 11 \rightarrow 9 \rightarrow D3$	14.2 *
Drone 7: $D3 \rightarrow 13 \rightarrow 18 \rightarrow D3$	19.38
Reverse: $D3 \rightarrow 18 \rightarrow 13 \rightarrow D3$	12.93 *
Drone 8: $D3 \rightarrow 15 \rightarrow 20 \rightarrow D3$	18.93
Reverse: $D3 \rightarrow 20 \rightarrow 15 \rightarrow D3$	7.32 *
Drone 9: $D3 \rightarrow 16 \rightarrow 19 \rightarrow D3$	33.34
Reverse: $D3 \rightarrow 19 \rightarrow 16 \rightarrow D3$	31.29

* Infeasible flight path