An Optimization Approach to Minimize the Expected Loss of Demand Considering Drone Failures in Drone Delivery Scheduling

Maryam Torabbeigi[1], Gino J. Lim[2], Navid Ahmadian[3], Seon Jin Kim[4]

***Abstract*—This study proposes a drone-based delivery scheduling method considering drone failures to minimize the expected loss of demand (ELOD). An optimization model (DDS-F) is developed to determine the assignment of each drone to a subset of customers and the corresponding delivery sequence. Because solving the optimization model is computationally challenging, a Simulated Annealing (SA) heuristic algorithm is developed to reduce the computational time. The proposed SA features a fast initial solution generation based on the Petal algorithm, a binary integer programming model for path selection, and a local neighborhood search algorithm to find better solutions. Numerical results showed that the proposed approach outperformed the well-known Makespan problem in reducing the ELOD by 23.6% on a test case. Several case studies are conducted to illustrate the impact of the failure distribution function on the optimal flight schedules. Furthermore, the proposed approach was able to obtain the exact solutions for the test cases studied in this paper. Numerical results also showed the efficiency of the proposed algorithm in reducing the computational time by 44.35%, on average, compared with the exact algorithm.**

***Index Terms*—Drone, Unmanned Aerial Vehicle, Failure, Expected Loss Of Demand, Simulated Annealing**

I. Introduction

Drones are able to perform tasks that were traditionally operated by manned systems. These drone applications include various civilian fields, such as security enhancement [1], damage assessment [2, 3], health-care services [4, 5], border patrol operations [6, 7], and communication relay [8]. Among different applications, the delivery of lightweight parcels is one of the most rapidly growing civilian applications observed in recent years [9]. Unlike ground transportation vehicles, drones can operate regardless of the existence or accessibility of roads. The use of drones can protect humans from exposure to dangerous areas. Drone-based delivery will be a faster alternative to ground transportation and it can be more cost-effective as well [10–12]. Therefore, parcel delivery by drones is gaining more attention among courier companies, such as Amazon, DHL, and UPS [13–15].

As the research community has focused on path planning and logistics using drones, the reliability of drone-based delivery has not received its well-deserved attention [16]. Drone failures are inevitable, and they occur mainly due to mechanical issues, environmental conditions, cyber-physical attack, collision, or human errors [17, 18]. The mishap rate of unmanned vehicles is much higher than manned vehicles [16, 19], and this prevents them from being operated widely in civilian applications. A drone malfunction can result in mission interruptions and loss of packages, as well as the drone itself, which will lead to customer dissatisfaction [20]. Unreliable drones should not be operated in congested areas. Because of the overall weight and substantial power, civilian injury could occur if they were to fall from the sky [21]. They pose a great risk for people on the ground, and therefore, safety and reliability in drone delivery must be placed at high priority.

The current research on drone failures has revolved around component-based fault diagnosis studies and entire drone health evaluation studies [22]. The safety evaluation and safety enhancement for particular components of drones are extensively studied. The desired flight path is compared with the actual flight path in the existence of a fault in drone components. The residuals obtained from the mathematical model are usually used as an evaluation metric to detect a fault. The fault detection is studied on different drone components, such as the sensor [23, 24], actuator [25–27], accelerometers and inclinometers [28], communication system [29] and battery [30]. However, these studies not only focus solely on drone health evaluation, but use a uniform health indicator for determining faults in all drone components [22, 31].

Studies have typically focused on one drone to capture faults and failures [22–31]. However, the impact of drone failures in the delivery network has not been well investigated. To fill this gap in the literature, this paper focuses on delivery networks operated by a fleet of drones. The primary goal is to develop a reliable delivery schedule considering drone failures to minimize failed package delivery to customers. In a delivery network, the reliability is observed at the network level as opposed to a component level, i.e., a drone. The drone failure probability can be estimated based on existing approaches for evaluating drone health, such as component-based fault diagnosis studies or entire drone health evaluation studies. Unlike typical drone routing problems, the decision to assign each drone to a subset of customers affects the delivery network reliability introduced in this paper. For example, one may wish to assign a more reliable drone to customers with high or sensitive demands. Other factors affecting the delivery network reliability include the delivery sequence, the amount

[1]Department of Industrial Engineering, University of Houston, TX, United States, maryam.torabbeigi@gmail.com

[2]Department of Industrial Engineering, University of Houston, TX, United States, ginolim@uh.edu, Corresponding Author

[3]Department of Industrial Engineering, University of Houston, TX, United States, nahmadian@uh.edu

[4]Republic of Korea Army, South Korea, sonjin64@gmail.com

of customers' demand, and the total travel distance [20].

In ground transportation network problems, such as the Vehicle Routing Problem (VRP) [32, 33], two common definitions of network reliability are [34] (1) *connectivity reliability*: the probability of at least one path existing between a pair of locations without disruption or heavy congestion [35], and (2) *travel time reliability*: the probability of traveling between a pair of locations within a specified time period [36–38].

Road connectivity reliability considers unexpected events, such as traffic congestion or bad weather, that makes roads inaccessible. According to the Federal Aviation Administration (FAA) regulations, drones should not fly above 400 feet [39]. Currently, the possibility of air traffic within 400 feet from the ground is negligible. Therefore, the road connectivity reliability definition may not be applied to the drone network. The travel time reliability in VRP is usually shown by including time window constraints and uncertainty in travel time. This can be applicable to the drone delivery network but does not reflect the impact of drone failures.

This study explores a new concept based on reliability calculation [40] to evaluate the reliability of a drone delivery network. A mathematical model is developed to determine more reliable paths for drone-based delivery networks. Although Sawadsitang et al. [16] considered failure probabilities for drone take-off and flight, their work was limited to a drone making one round trip between the depot and the customer. Another work by Torabbeigi et al. [20] adopted the idea of drone delivery network reliability. However, the drawback of their approach was that the drone delivery reliability was calculated after the routing was determined. Motivated by the drawbacks mentioned above, this paper introduces an optimization approach selecting reliable routes for drones to carry multiple packages on each trip. As such, the sequence of visiting customers and the assignment of drones to customers becomes an important factor. Contributions of this paper are as follows:

1) To propose a new method to schedule flight paths for a group of drones considering the probability of drone failure.
2) To develop a mathematical model to obtain the optimal assignment of customers to drones and the flight sequence of each drone to maximize the reliability of drone-based delivery.
3) To propose a computationally efficient simulated annealing (SA) method coupled with the Sweep [41] and Petal algorithm [42].

The rest of this paper is organized as follows. Section II explains the method to calculate the expected loss of demand (ELOD). Section III formally describes the problem, followed by the optimization model formulation. Section IV presents the proposed heuristic algorithm. In Section V, several case studies and an analysis of the results are discussed. Finally, Section VI concludes the paper with a potential extension of this work.

II. Expected Loss of Demand (ELOD) Calculation

In this paper, a drone is allowed to visit multiple customers to unload packages along the flight path. Therefore, each flight path consists of several segments (the flight between two consecutive stops). If a drone fails during transportation, the remaining customers along the flight path will not receive what was ordered. The amount of lost demand depends on the location of the failure and the amount of payload that the drone carries. The amount of lost demand will be more considerable if the drone fails closer to the depot as opposed to if it fails near the final flight segments. No demand will be lost if the drone fails in the final flight segment returning back to the depot after completing deliveries to all assigned customers. Therefore, the *expected loss of demand* is defined as the multiplication of the carried payload and its associated failure probability over all the flight segments in a flight path.

A. General ELOD Calculation

We begin this section by explaining how to calculate the ELOD for the flight path of one drone. This concept will be used in the mathematical model formulation in Section III to find the optimal routing strategy of drones to minimize the network ELOD. Suppose a drone leaving the depot is to visit n customers in sequence and return back to the depot. In the network setting, this flight path consists of $n+1$ nodes and a sequence of $n+1$ flight segments. The first flight segment is the flight from the depot to the first customer, and the last flight segment $n+1$ is for the drone to return back to the depot after visiting the last customer. All flights connecting two customers can be defined as flight segment $i \in \{2, 3, .., n\}$. The corresponding failure probability of flight segment i is denoted as $p_{i-1,i} = P(0 < t < t_{i-1,i})$, where variable t is the time of failure and $t_{i-1,i}$ is the flight time between location $i-1$ and i.

Considering a sequence of flight segments in a path, a failure could happen in a flight segment if it did not occur in all of the previous flight segments. To capture this, parameter f_i is defined as the probability of no failure in the previous flight segments up to flight segment i, and z_i is defined as the probability of a failure in flight segment i. For flight segment i, values of f_i and z_i depend on all of the previous flight segments, and they are calculated as:

$$f_i = f_{i-1} \cdot q_{i-1,i}, \qquad i \in \{1, 2, ..., n+1\} \tag{1}$$

$$z_i = f_{i-1} \cdot p_{i-1,i}, \qquad i \in \{1, 2, ..., n+1\}, \tag{2}$$

where $q_{i-1,i} = 1 - p_{i-1,i}$, $f_0 = 1$, and $z_0 = 0$.

The payload amount of a drone in a flight segment is the demand of remaining customers along the flight path. Parameter d_j is to show customer j's demand, and D_i represents the payload amount in flight segment i. Therefore, D_i is the summation of demand from customer i through customer n as in (3).

$$D_i = \sum_{j=i}^{n} d_j. \qquad i \in \{1, 2, ..., n\}, (D_{n+1} = 0) \tag{3}$$

Both $z_i, i \in \{1, 2, ..., n+1\}$ and f_{n+1} form a probability mass function (see Appendix A), and they are used to calculate the

ELOD as defined below:

$$ELOD = \sum_{i=1}^{n+1} z_i \ D_i = \sum_{i=1}^{n} z_i \ D_i + z_{n+1} \cdot 0 = \sum_{i=1}^{n} z_i \ D_i \tag{4}$$

In Formula (4), variable z_i depends on the flight segments prior to segment i, while D_i depends on the flight segments after segment i. An equivalent form of ELOD is provided by Equation (6), which only requires the information of the previous flight segments up to the current segment. Equation (6) is based on the customers' demand, and it does not depend on the carried payload within the flight segments.

$$\begin{aligned}
&\sum_{i=1}^{j} z_i = \sum_{i=1}^{j} f_{i-1} \ p_{i-1,i} = \sum_{i=1}^{j} f_{i-1} \ (1 - q_{i-1,i}) = \sum_{i=1}^{j} f_{i-1} \\
&- \sum_{i=1}^{j} f_{i-1} \ q_{i-1,i} = \sum_{i=1}^{j} f_{i-1} - \sum_{i=1}^{j} f_i = f_0 - f_j \\
&\rightarrow \sum_{i=1}^{j} z_i = 1 - f_j
\end{aligned} \tag{5}$$

$$\begin{aligned}
&ELOD = \sum_{i=1}^{n} z_i \ D_i = \sum_{i=1}^{n} z_i (\sum_{j=i}^{n} d_j) = \sum_{i=1}^{n} \sum_{j=i}^{n} z_i \ d_j = \\
&= \sum_{j=1}^{n} d_j (\sum_{i=1}^{j} z_i) = \sum_{j=1}^{n} d_j \ (1 - f_j)
\end{aligned} \tag{6}$$

B. An Illustration of ELOD Calculation

We illustrate the procedure of ELOD calculation using a small example in Figure 1, which consists of one depot and two customers being served by one drone. The amount of carried payload and the probability of failure in each flight segment are shown above and below the arcs, respectively.

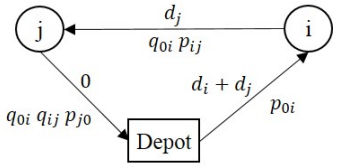


Figure. 1: An example flight path consisting of one depot, two customers, and one drone

Let us examine each flight segment for path "$Depot \rightarrow i \rightarrow j \rightarrow Depot$" in Figure 1 to calculate the failure probability and ELOD associated with each of the three segments. The final results are summarized in Table I.

1) Flight Segment 1: The drone departing from the depot carries the total demand for customers i and j. Hence, the demand of both customers will be lost if the drone fails in this flight segment. The probability of failure is $p_{0i} = P(0 < t < t_{0i})$, where index 0 is the depot.

2) Flight Segment 2: The drone continues the delivery for the next customer j after successfully completing the task for customer i. Hence, the probability of failure for this segment is the multiplication of (1) probability of no failure in the previous flight segment ($q_{0i} = 1 - p_{0i}$) and (2) probability of failure in the current segment ($p_{ij} = P(0 < t < t_{ij})$).

3) Flight Segment 3: The delivery mission is completed without a failure for both customers, and the drone returns back to the depot. The corresponding probability of failure in this segment is $p_{j0} = P(0 < t < t_{j0})$.

Therefore, the network ELOD for this single path problem is calculated as:

$$\begin{aligned}
&Network\ ELOD \\
&= ELOD_{Depot \rightarrow i} + ELOD_{i \rightarrow j} + ELOD_{j \rightarrow Depot} \\
&= p_{0i} \cdot (d_i + d_j) + q_{0i} \ p_{ij} \ d_j + q_{0i} \ q_{ij} \ p_{j0} \cdot 0 \\
&= p_{0i} \ (d_i + d_j) + q_{0i} \ p_{ij} \ d_j
\end{aligned} \tag{7}$$

TABLE I: A summary of ELOD calculations for Figure 1

Segment	f	D	Segment ELOD
depot → i	p_{0i}	$d_i + d_j$	$p_{0i}\ (d_i + d_j)$
i → j	$q_{0i}\ p_{ij}$	d_j	$q_{0i}\ p_{ij}\ (d_j)$
j → depot	$q_{0i}\ q_{ij}\ p_{j0}$	0	$q_{0i}\ q_{ij}\ p_{j0} \cdot 0$

III. Problem Description and Formulation

This work considers one depot, multiple customers, and a group of drones to deliver packages. Each customer has a certain demand and is served by exactly one drone. Our aim is to find the optimal drone flight schedule for a delivery network to minimize the network ELOD for a given drone failure probability distribution. A drone starts its flight path from the depot, visits the assigned customers in sequence, and returns to the depot. The maximum operation time of drone is the maximum flight time of a drone with one full charge of battery. The flight time and the total amount of payload depend on the type of drones. A mixed integer linear programming (MILP) model for drone flight schedules is presented in this section. The following notation is used to develop the drone delivery schedule model with drone failures (DDS-F model):

Sets

L	set of drones ($l \in L$)
N	set of nodes, node 0 is the depot ($i, j \in N, c \in N - \{0\}$)

Parameters

α_{ijl}	probability of no failure in flight segment (i, j) by drone l
d_c	customer c demand
M	sufficiently large number
n	number of customers
ot_l	maximum operation time of drone l
t_{ijl}	travel time from node i to node j by drone l
w_l	maximum weight capacity of drone l

Variables

f_c	probability of no failure arriving at customer c
x_{ijl}	1 if drone l goes directly from node i to node j, 0 otherwise
y_c	the order of visiting customer c in the path

The DDS-F model is similar to VRP models with the addition of constraints regarding the limitations of flight time and carried payload. For flight segment (i, j), the probability of a drone arriving at customer j without a prior failure can be calculated as:

$$\text{If } x_{ijl} = 1, \text{then } f_j = f_i \cdot P_l(t > t_{ij}), \quad \forall i, j \in N, l \in L, \tag{8}$$

where $f_0 = 1$. This "if-then" statement is formulated as a constraint stated in Inequality (9) by adding parameter "M", which is a large number.

$$\alpha_{ijl} f_i - M \cdot (1 - x_{ijl}) \leq f_j \leq \quad \alpha_{ijl} f_i + M \cdot (1 - x_{ijl}), \quad \forall i, j \in N, l \in L \tag{9}$$

where $\alpha_{ijl} = P_l(t > t_{ij})$. When variable x_{ijl} is equal to 1, Inequality (9) is of the form $\alpha_{ijl}\ f_i \leq f_j \leq \alpha_{ijl}\ f_i$, which means $f_j = \alpha_{ijl}\ f_i$. When variable x_{ijl} is equal to 0, variable f_j can be any value between $-M$ and M. As the variable f_j represents the probability of no failure, it only receives a value between [0, 1]. Hence, parameter M can be set to 1 without loss of generality. Inequality (9) is used as Constraints (19) and (20) in the DDS-F model. The resulting MILP formulation for the DDS-F model is provided as follows:

$$\text{Min} \sum_{c \in N - \{0\}} (1 - f_c)\ d_c \tag{10}$$

$$\sum_{i \in N \setminus \{c\}} \sum_{l \in L} x_{cil} = 1, \qquad \forall c \in N \setminus \{0\} \tag{11}$$

$$\sum_{i \in N \setminus \{c\}} \sum_{l \in L} x_{icl} = 1, \qquad \forall c \in N \setminus \{0\} \tag{12}$$

$$\sum_{i \in N \setminus \{c\}} x_{icl} = \sum_{j \in N \setminus \{c\}} x_{cjl}, \qquad \forall c \in N \setminus \{0\}, l \in L \tag{13}$$

$$\sum_{c \in N \setminus \{0\}} x_{0cl} = 1, \qquad \forall l \in L \tag{14}$$

$$\sum_{c \in N \setminus \{0\}} x_{c0l} = 1, \qquad \forall l \in L \tag{15}$$

$$\sum_{c \in N \setminus \{0\}} \sum_{i \in N} d_c\ x_{cil} \leq w_l, \qquad \forall l \in L \tag{16}$$

$$\sum_{i \in N} \sum_{j \in N} t_{ijl}\ x_{ijl} \leq ot_l, \qquad \forall l \in L \tag{17}$$

$$y_u - y_v + n \sum_{l \in L} x_{uvl} \leq n - 1, \quad \forall u, v \in N \setminus \{0\} \tag{18}$$

$$\alpha_{icl}\ f_i - M(1 - x_{icl}) \leq f_c, \qquad \forall i \in N, c \in N \setminus \{0\}, l \in L \tag{19}$$

$$f_c \leq \alpha_{icl}\ f_i + M(1 - x_{icl}), \qquad \forall i \in N, c \in N \setminus \{0\}, l \in L \tag{20}$$

$$x_{ijl} \in \{0, 1\}, f_i \geq 0, f_0 = 1, \qquad \forall i, j \in N, l \in L$$

The objective function is the minimization of network ELOD for all flight paths. Constraints (11) and (12) state that each customer should be served once and by exactly one drone. Constraint (13) is the flow balance equation. Constraints (14) and (15) show that all drones should start and finish their flight at the depot. The amount of commodity that each drone can carry is limited, to its weight capacity via Constraint (16). The drone total flight time is also limited as stated in Constraint (17). Constraint (18) prevents the sub-tours in the network according to the Miller-Tucker-Zemlin formulation [43]. Constraints (19) and (20) are related to the ELOD calculation and are obtained from Inequality (9).

IV. Solution Method

The DDS-F model is a variant of the Vehicle Routing Problem, which is known to be NP-hard [44]. Exact algorithms for solving VRP models only work well for small-scale problems. Hence, meta-heuristic methods are often utilized to find near-optimal solutions to save time for medium- and large-scale problems. Simulated Annealing (SA) [45] is a commonly used meta-heuristic algorithm to solve drone scheduling problems [46, 47], and it is adopted in this paper. To develop a solution approach using SA, Section IV-A explains the solution representation and the procedure to calculate the minimum ELOD value corresponding to the solution. Section IV-B introduces the proposed SA-based solution approach coupled with (1) the Petal algorithm for generating various feasible flight paths, (2) a binary integer programming (BIP) model to select the best feasible solution among the candidates, and (3) a local neighborhood search to search for a better solution as the final clean-up. Therefore, the solution will be a local optimal, which may not be a global optimal.

A. Solution Representation and ELOD Calculation

The solution of the optimization model consists of flight paths for each of the drones starting from and terminating at the depot (location 0). Each drone visits a subset of customers in sequence to complete the delivery task. The proposed algorithm to solve this problem starts by sorting the customers in an ordered list in each iteration. The Petal algorithm [42] consisting of a three step procedure is used to determine a feasible path with the minimum network ELOD for the given list of customers (S) in each iteration: 1) generate feasible contiguous subsets from the given list of customers (Section IV-A1), 2) determine the flight path and the ELOD for each of the subsets (Section IV-A2), and 3) choose a set of flight paths with the minimum total ELOD to cover all customers (Section IV-A3). Figure 2 shows these three steps.

1) Create a Set of Contiguous Subsets: For a given list of customers (S) to serve, a contiguous subset (CS) is a subset of the customers to be visited by a drone. Because there can be numerous different possible contiguous subsets, Algorithm 1 is developed to efficiently create the subsets while satisfying the feasibility requirements on the weight capacity and the total operation time of drones (Constraints (16) and (17)).

The algorithm is initialized, which includes the number of customers (n), the list of customers to serve (SS), and empty sets of CS and a temporary set F. The contiguous subsets are generated on a cyclic order of customers to ensure the full inclusion of combinations for creating the CS. This is done by repeating the list of customers except for the last one, i.e.,

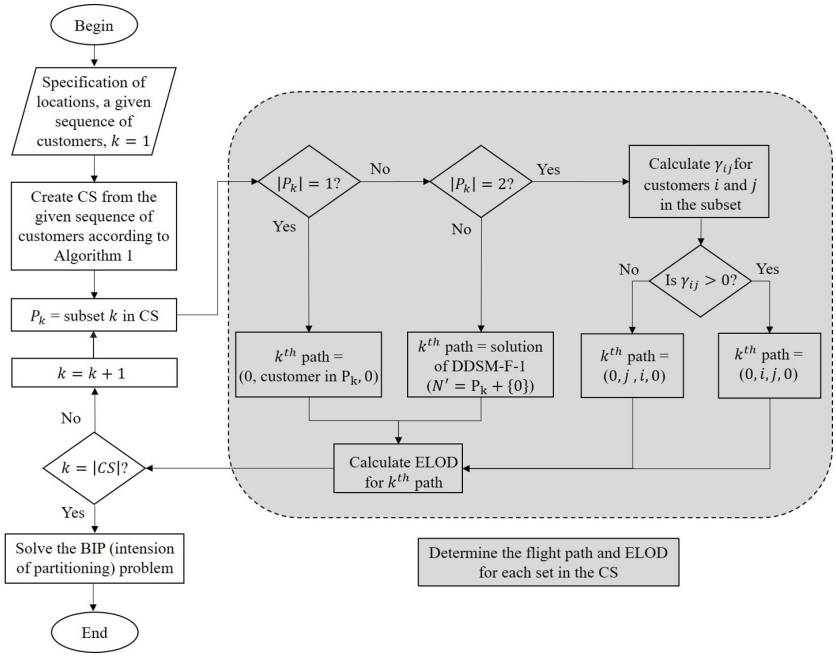


Figure. 2: Process of optimal solution calculation for a given sequence of customers

Algorithm 1 Create contiguous subsets from a given ordered list of customers

Define:
S[i:j]= An ordered list from element i to j
Input:
n= number of customers.
SS=[S[1:n] : S[1:n-1]]
c=1, CS={}, F={};
While ($c \leqslant n$)
k = c;
Add k^{th} customer in SS to F;
While (F feasible regarding Weight capacity and Flight time)
Add set F to CS;
k = k+1;
Add k^{th} customer in SS to F;
End While
F={}; c = c+1;
End While
Output:
CS: A contiguous subset of customers

$SS = [1, 2, \cdots, n, 1, 2, \cdots, n-1]$. First, the first customer in SS is added to F, i.e., $F = SS[1]$. Second, the inner **While-loop** is executed to construct a feasible CS. The loops are continued until all the customers in S are checked, i.e., $c = n$. Then, the algorithm stops and returns the CS.

2) Flight Path and ELOD Calculation for Contiguous Subsets: The order of visiting customers in each subset should be optimized to result in the minimum ELOD value. The optimal flight path for each of the contiguous subsets can be found by solving the DDS-F model with one drone (see the DDS-F-1 problem in Appendix B). Because a contiguous set contains a much smaller subset of customers to be served by a drone, the computational burden for solving the optimization model is substantially reduced. Since the capacity and the operational time of drones are limited, most of the drones are able to serve only a few customers. A contiguous subset with one customer corresponds to a flight path starting from the depot to visit the customer and then to return back to the depot. As shown in Figure 3, a contiguous subset with two customers i and j have two possible flight paths (Path I and Path II), and the feasibility of each path is checked in Step 3 of Algorithm 1. A

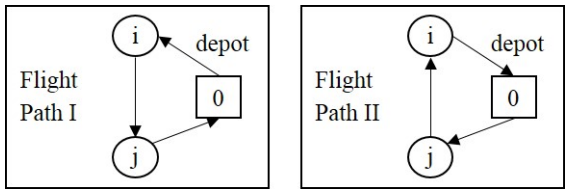


Figure. 3: Two possible flight paths for two customers and one drone

path with a lower ELOD value will be preferred for the drone delivery. The ELOD value for flight path I ($ELOD_I$) is lower than the ELOD for flight path II ($ELOD_{II}$) if Inequality (22) holds true:

$$\langle ELOD_I < ELOD_{II} \rangle \tag{21}$$

$$\rightarrow (1 - \alpha_{0il})\ d_i + (1 - \alpha_{0il}\alpha_{ijl})\ d_j < (1 - \alpha_{0jl})\ d_j + (1 - \alpha_{0jl}\alpha_{ijl})\ d_i \tag{22}$$

The optimization model is modified to solve the path finding problem for each contiguous subset more efficiently (see DDF-F-1 in Appendix B). Equations (6) and (8) will determine the ELOD value for the contiguous subsets after the determination of the flight paths.

3) Final Flight Paths Section: We now have multiple feasible flight paths to consider and their corresponding objective function values. The final selection of contiguous subsets to serve customers for the limited number of drones should be made such that: (1) all the customers are included in exactly one subset, which means each customer is served by one

drone, (2) the number of selected subsets is equal to the number of drones, and (3) the flight schedule for all drones must result in the minimum ELOD. This selection problem can be viewed as partitioning the customers into m groups (i.e., m drones). Therefore, we propose a Binary Integer Programming (BIP) model to solve this problem using the following notation.

Sets
P Set of flight paths ($p \in P$)
Parameters
m Number of drones,
σ_p ELOD for the flight path p,
θ_{pj} 1 if flight path p includes customer j, 0 otherwise, ($j \in N - \{0\}$).
Variables
f_p 1 if flight path p is selected, 0 otherwise.

$$\text{Min} \sum_{p \in P} \sigma_p f_p \tag{23}$$

$$\sum_{p \in P} \theta_{pj} f_p = 1 \qquad \forall j \in N - \{0\} \tag{24}$$

$$\sum_{p \in P} f_p = m \tag{25}$$

$$f_p \in \{0, 1\} \qquad \forall p \in P$$

The objective function (23) minimizes the summation of ELOD for the selected flight paths. Constraint (24) states that each customer must be included on only one of the selected flight paths. Constraint (25) is to limit the number of selected flight paths to the number of available drones.

B. The Simulated Annealing Algorithm

The proposed SA in this paper starts with an initial solution and searches for better solutions using a neighborhood search method to improve the objective function value. The SA algorithm controls the probability of accepting the new solution using the temperature parameter T, which is set to a high value initially and is gradually decreasing during the iterations. The SA algorithm has two loops. In the inner loop, new solutions are generated in the neighborhood of the current solution at the current temperature. The temperature is decreased at the rate of $\alpha \in (0, 1)$ in the outer loop until the stopping criteria are met [48]. The proposed SA stops if either of two conditions is met: (1) no improvement in the objective function after a certain number of iterations, or (2) a minimum temperature value is reached. The following two subsections explain two specific steps of SA in sequence: the *initial solution generation* method and the *neighborhood search* method to find a better solution through iterations.

1) Initial Solution: The initial solution (i.e., a sequence of customers to visit) for the SA algorithm is generated following the steps outlined in Figure 4 [41, 42]. The input data includes the location information for both the depot and the customers. The polar coordinate angles of the customers are used to calculate the proximity of each customer to the depot. Then, the initial solution is generated in ascending order of distance from the depot to the customers. This initial solution yield an optimum or near-optimum solution to the single-depot vehicle-dispatch problem citegillett1974heuristic

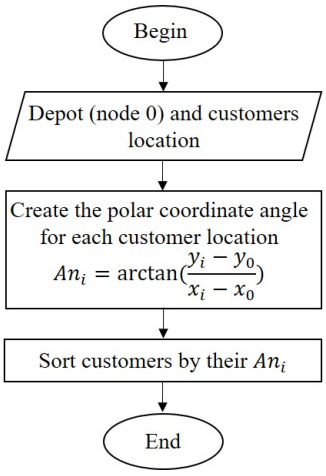


Figure. 4: Initial visit sequence of customers

2) Neighborhood Search Methods: In each iteration of the algorithm, a new solution is generated from the neighborhood of the current solution. In the neighborhood search, a solution can be accepted or rejected based on the temperature (T) and the corresponding objective function value. The probability of accepting a better or unchanged solution is always 1, while a worse solution may be accepted with a low probability to avoid the local entrapment [49]. This paper explores three different ways to generate neighborhood solutions in each SA iteration: shift, reverse, and exchange. We explain these methods using an example. Consider a list of customers to be visited by a drone, $S = [1, 2, 3, 4, 5, 6]$.

1) Shift method: shift k customers after customer i to the first location after customer j (i, j and k are randomly generated). For example, if $i = 1$, $k = 2$ and $j = 5$, a new solution will be $S_{new} = [1, 4, 5, 2, 3, 6]$.
2) Reverse method: reverse the sequence of customers between customer i and j (i and j are randomly generated). For example, if $i = 1$ and $j = 5$, a new solution will be $S_{new} = [1, 4, 3, 2, 5, 6]$
3) Exchange method: change the locations of customers i and j (i and j are randomly generated). For example, if $i = 1$ and $j = 5$, a new solution will be $S_{new} = [5, 2, 3, 4, 1, 6]$.

At each iteration of the SA algorithm, one of the three methods is randomly selected to generate a new solution.

V. Numerical Experiments

Numerical results are conducted to test the proposed solution method using small and large-size problems. A deterministic model without drone failures is used as a benchmark to test the performance of our approach. Since the deterministic model does not have failure probabilities, and requires to have a new objective function, the well-known Makespan (MS) problem (Appendix D) is chosen as it is the closes deterministic model to the proposed DDS-F model. The sensitivity

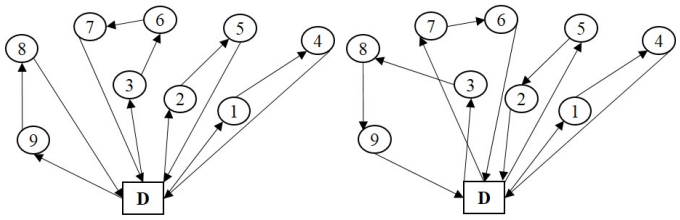
analysis on the failure distribution is presented in Section V-C, and the performance of the proposed SA algorithm is discussed in Section V-E. All experiments are done on a Linux server with 24 cores and 384GB RAM, and they are implemented in the Python environment [50]. Gurobi solver 8.1 [51] was used to solve the optimization model. The random drone parameter settings in Table II are used for all test cases. We considered 32 minutes of flight capacity in this research [1].

TABLE II: The parameter setting for the test cases

Parameter	Value
t_{ij}	Euclidean distance between each pair of nodes (min)
w_l	1 lb.
ot_l	32 minutes [4].
λ_l	0.005 (failures per minute)

A. Case Study

A sample network with 9 customers and 4 homogeneous drones is used in this section. The drone failures are assumed to follow a Weibull distribution with a constant failure rate according to Table II. Note that the Weibull distribution is one of the most commonly used failure distributions in reliability analysis [52]. Hence, the DDS-F model uses parameter $\alpha_{ijl} = P_l(t > t_{ij}) = e^{-(t_{ij}/\eta_l)}$, where η is the scale parameter and $\frac{1}{\eta_l} = \lambda_l$ is the failure rate. Figure 5 shows the resulting optimal flight paths for the DDS-F problem in Figure 5(a) and the makespan model in Figure 5(b).



(a) DDS-F problem **(b)** Makespan problem

Figure. 5: Optimal solutions for (a) DDS-F and (b) Makespan problem

In VRP and transportation problems, a common objective function is to minimize the makespan, i.e. the maximum time that a drone spends for any of the flight paths [53]. Therefore, a comparison is made to investigate the impact of the DDS-F model against the makespan problem. Figure 5(b) shows that the optimal flight paths derived from the makespan problem differ from the solution outputted by the DDS-F model.

Furthermore, Table III gives the optimal path details for DDS-F and Makespan regarding the payload amount, ELOD, and flight time. Although DDS-F slightly underperformed (1.1%) when compared to Makespan, the resulting flight schedule is more reliable, as it reduced the ELOD value by 33.4%.

The second path of Makespan ($0 \rightarrow 5 \rightarrow 2 \rightarrow 0$) was revealed to be the reverse of the second path in the DDS-F model. As expected, the minimization of the makespan does not differentiate between a path and its reversed path because a path and its reversed path have the same flight time. This is the drawback of the makespan approach for the problem discussed in this paper. It is trivial to see that the ELOD values can be different between a path and its reverse path as different customers often request different amounts of payload.

B. Comparison between the DDS-F and the Makespan

Different test cases are used to compare the DDS-F model and the MS model in terms of *ELOD* and *makespan*. The results are summarized in Table IV, where n and m are the number of customers and the number of drones, respectively. The first part shows the randomly generated test cases where customers are located randomly around the depot. We used the set of benchmark instances of Augerat et al. test cases [54] in the second part of the table as it is one of the most used benchmarks in the literature. For these test cases, the vehicle capacity is multiplied by 10^{-2} to be adjusted for drone capacity. As expected, DDS-F performed better in minimizing ELOD, while the MS model worked better in reducing makespan for all test cases. The percentage increase in ELOD and the percentage decrease in makespan were calculated according to Formula (26) and (27), respectively.

$$\text{ELOD decrease } (\%) = \frac{\text{ELOD}_{\text{MS}} - \text{ELOD}_{\text{DDSF}}}{\text{ELOD}_{\text{DDSF}}} \times 100, \quad (26)$$

$$\text{makespan increase } (\%) = \frac{\text{makespan}_{\text{DDSF}} - \text{makespan}_{\text{MS}}}{\text{makespan}_{\text{MS}}} \times 100, \quad (27)$$

where subscripts MS and DDSF are to indicate the results obtained by the Makespan and DDS-F problems, respectively. In all test cases, the reduction of ELOD by DDS-F ranged from 1.5% up to 33.4%, while the increase of makespan was negligible, ranging from 0.1% to 10%. Compared to the traditional makespan model for delivery, the DDS-F can generate much more reliable drone flight paths by minimizing the expected loss of demand due to a drone failure.

C. Sensitivity Analysis on the Failure Rate

The failure rate in Section V-A and Section V-B was assumed to be constant. However, it might be subject to fluctuations over time. Note that the Weibull distribution was considered here because it is a commonly used probability distribution function involving a failure function. The cumulative distribution function for the Weibull distribution is $1 - e^{-(t/\eta)^\beta}, \quad t \geq 0$ where parameters η and β are the scale and shape parameters, respectively. Therefore, it results in $\alpha_{ijl} = P_l(t > t_{ij}) = e^{-(t_{ij}/\eta)^\beta}$ in the DDS-F model. The Weibull distribution represents a variety of shapes based on the shape parameter: $\beta < 1$ means the failure rate decreases over time, $\beta = 1$ shows a constant failure rate over time, and $\beta > 1$ exhibits an increase in the failure rate with time. The impact of parameter β on the Network ELOD for the case study shown in Figure 5(a) is investigated here. The optimal flight paths were obtained by changing the value of β between 0.8 and 2.5 with 3, 4, and 5 drones. Note that β is dimensionless.

Figure 6 shows that the network ELOD decreased as β was increased. This trend can be explained by looking at the ratio (t_{ij}/η_l) of the exponent in $\alpha_{ijl} = e^{-(t_{ij}/\eta_l)}$. The results were

TABLE III: DDS-F and Makespan results for the test case problem

Problem	Path	Payload (10^{-2} lb.)	ELOD	Flight time (min)	Network ELOD (10^{-2} lb.)	makespan (min)
DDS-F	$0 \to 1 \to 4 \to 0$	40	1.722	26.86	9.511	28.67
	$0 \to 2 \to 5 \to 0$	35	2.499	27.24		
	$0 \to 3 \to 6 \to 7 \to 0$	45	2.442	28.67		
	$0 \to 9 \to 8 \to 0$	60	2.848	25.44		
Makespan	$0 \to 1 \to 4 \to 0$	40	2.499	27.24	12.688	28.36
	$0 \to 5 \to 2 \to 0$	35	2.808	26.86		
	$0 \to 3 \to 8 \to 9 \to 0$	80	5.583	26.44		
	$0 \to 7 \to 6 \to 0$	25	1.798	28.36		

TABLE IV: Comparison between DDS-F and MS Problems

n	m	MS Problem ELOD (10^{-2} lb.)	MS Problem makespan (min)	DDS-F Problem ELOD (10^{-2} lb.)	DDS-F Problem makespan (min)	ELOD decrease (%)	makespan increase (%)
Randomly generated test cases:							
9	4	12.69	28.37	9.51	28.67	33.4	1.1
9	5	11.77	27.20	9.37	27.24	25.6	0.1
10	4	13.57	28.43	10.73	28.67	26.4	0.9
10	5	11.81	27.20	10.59	27.24	11.5	0.1
11	4	15.31	29.25	12.59	29.29	21.6	0.1
11	5	15.07	28.29	12.11	28.67	24.4	1.3
12	4	16.25	29.25	13.47	29.67	20.7	1.4
12	5	16.16	28.37	12.93	28.67	25.0	1.1
Augerat et al. set P test cases [54]:							
10	6	21.35	68.61	21.02	75.53	1.5	10.0
12	7	26.41	68.61	23.46	75.53	12.5	10.0
12	8	28.68	65.11	22.75	65.57	26.0	0.7

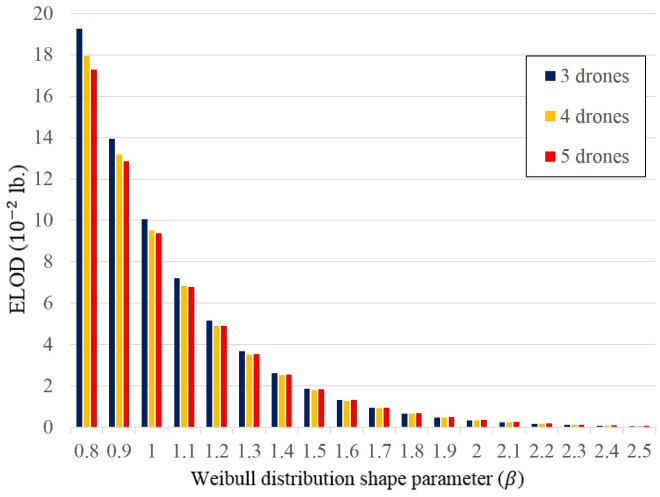


Figure. 6: Network ELOD (10^{-2} lb.) for different shape parameter values with the fixed scale parameter

generated based on a fixed $\eta = 200$. However, the maximum travel time between two nodes in Figure 5(a) is 13.6 minutes, which lead to $\frac{t_{ij}}{\eta} << 1$. Thus, increasing the value of β will result in the increase in both the α_{ijl} parameter value and the variable f_c, and it reduces the ELOD accordingly. Considering the limited flight time of drones, the network ELOD value will decrease as the Weibull distribution shape parameter increases.

Furthermore, as seen in Figure 7, increasing the value of β resulted in the reduction of the optimal number of required drones. This is because a higher value of β means the drone's

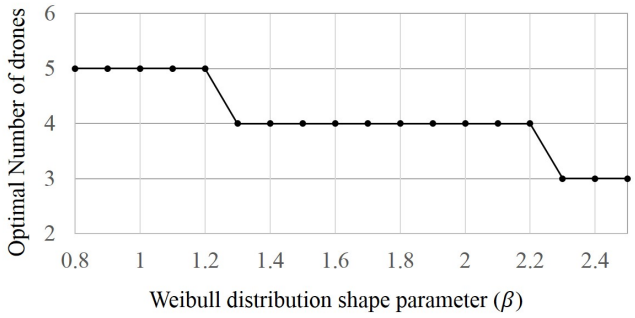


Figure. 7: Optimal number of drones (between 3, 4, and 5) by changing Weibull distribution shape parameter

reliability (parameter α_{ijl}) increases. As a result, a lesser number of drones are needed as their reliability is higher. For example, the optimal number of drones reduced from 5 drones to 3 drones when β was increased from 0.8 to 2.5. Note that further experiments showed that parameter β influences on the optimal flight path selection, while η does not. However, parameter η affects the ELOD value.

D. The SA parameter tuning

Prior to solving the test instances using the SA, the SA parameter values were carefully tuned. The design of experiments (DOE) is an effective approach to investigate the cause and effect relationship between an algorithm parameters and the output [55]. Five SA parameters with two levels (high and low) for each of the, according to Table V are considered for the tuning. A statistical two-level full factorial analysis, the most popular designs among experimenters [56], is performed to examine the impact of these parameters on SA performance.

TABLE V: The SA parameter tuning

Parameter	Level 1 value	Level 2 value	Selected value
Initial temperature (x_1)	1	5	1
Final temperature (x_2)	0.005	0.05	0.005
Temperature reduction ratio (x_3) (α)	0.95	0.98	0.95
Number of iterations per temperature (x_4)	1	5	1
Consecutive iterations without improvement (x_5)	10	20	20

The SA performance with each parameter setting is evaluated by the computational time (y_1) and the ELOD value (y_2)

for a test case with 12 customers and 5 drones. The results are presented in Table VI and depicted in Figure V-D. Based on these results, parameter x_1 should be on its lower level to reduce both y_1 and y_2. Parameter x_5 does not have significant impact on both dependent variables, so we chose its higher level. Parameters x_3 and x_4 have low impact on y_2 but their lower level can significantly decrease y_2, therefore their lower level value is chosen. Parameter x_2 has high impact on y_2, so its lower level is chosen. The selected values are shown in Table V and are used in all SA implementations in this study.

TABLE VI: SA parameter tuning regression results

Dependent variable: computational time (y_1)				
Parameter	coefficient	std error	t	P > |t|
constant	362.20	23.30	15.54	0.000
x_1	54.16	23.30	2.32	0.028
x_2	-59.99	23.30	-0.574	0.016
x_3	137.81	23.30	5.913	0.000
x_4	229.44	23.30	9.845	0.000
x_5	11.31	23.30	0.486	0.631
Dependent variable: ELOD value (y_2)				
Parameter	coefficient	std error	t	P > |t|
constant	13.49	0.063	214.53	0.000
x_1	0.11	0.063	1.81	0.081
x_2	0.31	0.063	5.07	0.000
x_3	-0.09	0.063	-1.50	0.144
x_4	-0.16	0.063	-2.57	0.016
x_5	-0.0003	0.063	-0.005	0.996

E. The SA Performance

This section evaluates the computational performance of the proposed SA algorithm (see Section IV) compared to the exact method (Gurobi solver implemented in Python [51]). Test cases are randomly generated with each case having a different number of customers (n) and drones (m). The problem size of the test instances varies from 10 customers with 4 drones to 100 customers with 32 drones. The stopping criteria for solving the exact method are (1) 2 hours of CPU run time, and (2) a 1% optimality gap.

Figure 9 shows the progression of the SA algorithm until it converged. As can be seen from Figure 9, the initial solution to the SA algorithm is close to the final solution. The SA began with an initial solution (ELOD=12.94) and searched for a better solution. However, the ELOD fluctuated, as the algorithm is allowed to accept a worse solution with a low probability to avoid local entrapment. However, it eventually converged to a solution whose ELOD value (12.92) was lower than the initial solution. This figure shows that worse solutions were frequently accepted at the initial stage, but the rate dwindled as it approached the final iterations.

Table VII summarizes the results obtained by both the exact method and the SA. *obj* is the ELOD value and *time* shows the computation time in seconds. The last two columns show the performance comparison of the SA against the exact method. Δ*obj* shows the percentage increase of ELOD in SA compared to the exact method, while Δ*time* is the percentage decrease in computation time. These values are calculated as:

$$\Delta \text{ obj } (\%) = \frac{\text{obj}_{\text{SA}} - \text{obj}_{\text{Exact}}}{\text{obj}_{\text{Exact}}} \times 100$$

$$\Delta \text{ time } (\%) = \frac{\text{time}_{\text{SA}} - \text{time}_{\text{Exact}}}{\text{time}_{\text{Exact}}} \times 100$$

TABLE VII: Optimal solution obtained for DDS-F model by the exact method and the proposed SA algorithm

n	m	Exact Method		SA Method		Δobj (%)	Δ time (%)
		obj (10^{-2}lb.)	time(s)	obj (10^{-2}lb.)	time(s)		
10	4	10.72	63.1	10.72	74.6	0.0	18.2
10	5	10.58	68.4	10.58	72.4	0.0	7.3
11	4	12.59	296.0	12.60	107.8	0.3	-63.5
11	5	12.11	388.1	12.13	69.0	0.1	-82.2
12	4	13.46	763.3	13.50	157.6	0.5	-79.3
12	5	12.92	4460.7	12.90	70.4	0.1	-98.4
20	7	NA	NA	21.60	310.4	-	-
20	8	NA	NA	21.35	178.6	-	-
40	16	NA	NA	39.90	686.4	-	-
40	17	NA	NA	39.75	603.5	-	-
60	19	NA	NA	60.69	3206.2	-	-
60	20	NA	NA	60.85	2404.8	-	-
100	31	NA	NA	88.88	6484.9	-	-
100	32	NA	NA	88.67	5053.8	-	-

For the test cases with 10 customers, both methods found optimal solutions. For the cases with 11 and 12 customers, the solution provided by SA on average was 0.25% more than the optimal solution but it reduced the computational time significantly, on average by 80.85%. As expected, the SA outperformed the exact method in CPU time up to 98.4% with an exception for the small case. Because the exact method failed to find a solution for cases with more than 12 customers, it was not possible to compare the performance of the SA for larger instances. Overall, the SA is capable of finding good solutions at a fraction of the time the exact method took to solve the problems. The computational performance of SA was more pronounced when the problem size was increased because the exact method could not handle larger problem instances.

VI. Conclusion

This paper presented a reliable way to deliver light-weight parcels to customers using drones. Because drone failure during flights can result in unmet customer demand, the concept of minimizing the expected loss of demand was introduced to determine a more reliable flight path considering the probability of drone failure following a Weibull distribution. A mathematical optimization model (DDS-F) was developed to determine a flight schedule with minimum network ELOD subject to drone specific constraints: maximum flight time and payload capacity. The performance of the DDS-F was analyzed and compared to the traditional makespan model using various sizes of test problems. The numerical results showed that the DDS-F provided solutions about 20.78% more reliable on average at a minor increase in makespan by 2.43%. Furthermore, the impact of the failure distribution on the network ELOD value was investigated, and it was found that the network ELOD had a reverse correlation to the shape and scale parameters of the Weibull distribution. The numerical results showed that the impact of the shape parameter on the flight schedule was higher than the scale parameter. The shape parameter had an influence on the optimal flight paths and the

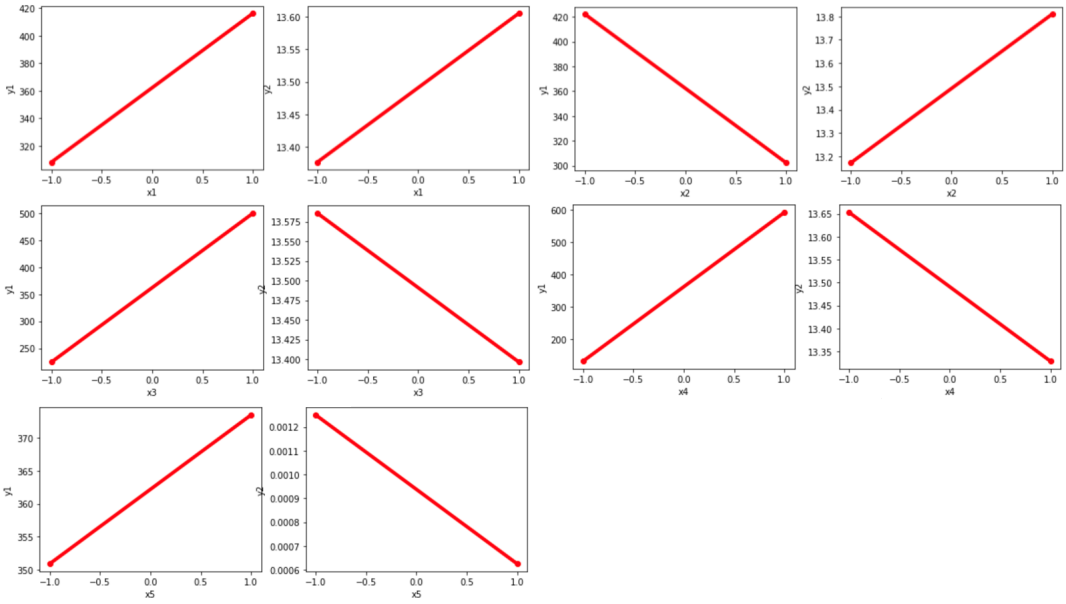


Figure. 8: Impact of SA parameters $(x_1, ..., x_5)$ on ELOD (y_1) and computational time (y_2)

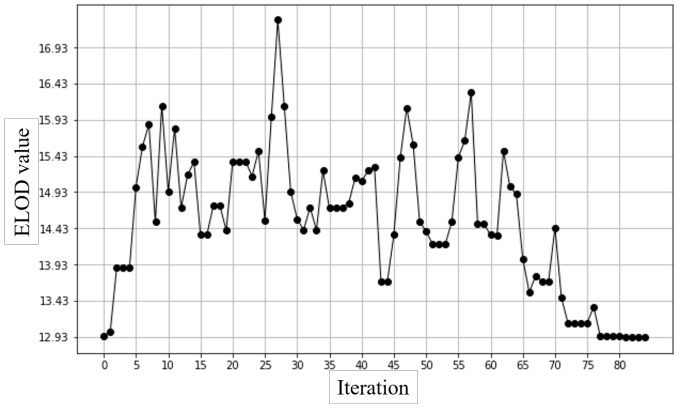


Figure. 9: The ELOD value over iterations: a case study with 12 customers and 5 drones

optimal number of required drones, but the scale parameter did not. The computational performance of SA was analyzed as the DDS-F was not scalable to a larger size of problems. The results showed that the SA was able to find good solutions to smaller problem instances at a small fraction of CPU time for solving the DDS-F using the exact method.

There are several ways to extend the work presented in this paper. First, one can study more on selecting the appropriate failure distribution function based on physical or virtual experiments. Second, the ELOD concept can be used for a delivery network combining both trucks and drones. Third, a more efficient solution algorithm could be developed, such as a two-stage approach, in which the first stage determines a flight schedule at minimum makespan, and the second stage aims to minimize the ELOD.

REFERENCES

[1] J. Cho, G. Lim, T. Biobaku, S. Kim, and H. Parsaei, "Safety and security management with unmanned aerial vehicle (UAV) in oil and gas industry," *Procedia Manufacturing*, vol. 3, pp. 1343–1349, 2015.

[2] G. J. Lim, S. Kim, J. Cho, Y. Gong, and A. Khodaei, "Multi-UAV pre-positioning and routing for power network damage assessment," *IEEE Transactions on Smart Grid*, 2016.

[3] A. Varghese, J. Gubbi, H. Sharma, and P. Balamuralidhar, "Power infrastructure monitoring and damage detection using drone captured images," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1681–1687.

[4] 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.

[5] E. Ackerman and E. Strickland, "Medical delivery drones take flight in east africa," *IEEE Spectrum*, vol. 55, no. 1, pp. 34–35, 2018.

[6] S. J. Kim and G. J. Lim, "Drone-aided border surveillance with an electrification line battery charging system," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3-4, pp. 657–670, 2018.

[7] S. Kim and G. Lim, "A hybrid battery charging approach for drone-aided border surveillance scheduling," *Drones*, vol. 2, no. 4, 2018.

[8] S. J. Kim, G. J. Lim, and J. Cho, "Drone relay stations for supporting wireless communication in military operations," *Advances in Intelligent Systems and Computing Advances in Human Factors in Robots and Unmanned Systems*, pp. 123–130, 2017.

[9] M. Torabbeigi, G. J. Lim, and S. J. Kim, "Drone delivery scheduling optimization considering payload-induced battery consumption rates," *Journal of Intelligent & Robotic Systems*, pp. 1–17, 2019.

[10] 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.

[11] T. Keeney, "Drone delivery: How can amazon charge $1 for drone delivery?" May 2016. [Online]. Available: https://ark-invest.com/research/drone-delivery-amazon

[12] Z. Shi and W. K. Ng, "A collision-free path planning algorithm for unmanned aerial vehicle delivery," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 358–362.

[13] Amazon.com Inc., "Amazon prime air." [Online]. Available: http://www.amazon.com/primeair

[14] V. Bryan, "Drone delivery: DHL 'parcelcopter' flies to german isle," Sep 2014. [Online]. Available: http://www.reuters.com/article/us-deutsche-post-drones-idUSKCN0HJ1ED20140924

[15] J. Stern, "Like amazon, ups also considering using unmanned flying vehicles," Dec 2013. [Online]. Available: http://abcnews.go.com/Technology/amazon-ups-drone-delivery-options/story?id=21086160

[16] S. Sawadsitang, D. Niyato, P.-S. Tan, and P. Wang, "Joint ground and aerial package delivery services: A stochastic optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2241–2254, 2018.

[17] N. Ahmadian, G. J. Lim, M. Torabbeigi, and S. J. Kim, "Collision-free multi-UAV flight scheduling for power network damage assessment," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 794–798.

[18] V. Kharchenko and V. Torianyk, "Cybersecurity of the internet of drones: Vulnerabilities analysis and imeca based assessment," in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. IEEE, 2018, pp. 364–369.

[19] D. W. King, A. Bertapelle, and C. Moses, "UAV failure rate criteria for equivalent level of safety," in *International Helicopter Safety Symposium, Montreal*, vol. 9, 2005.

[20] 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.

[21] F. Schenkelberg, "How reliable does a delivery drone have to be?" in *2016 Annual Reliability and Maintainability Symposium (RAMS)*, Jan 2016, pp. 1–5.

[22] Z. Zhao, Q. Quan, and K.-Y. Cai, "A health evaluation method of multicopters modeled by stochastic hybrid system," *Aerospace Science and Technology*, vol. 68, pp. 149–162, 2017.

[23] P. Lu, E.-J. van Kampen, C. de Visser, and Q. Chu, "Nonlinear aircraft sensor fault reconstruction in the presence of disturbances validated by real flight data," *Control Engineering Practice*, vol. 49, pp. 112–128, 2016.

[24] H. Rafaralahy, E. Richard, M. Boutayeb, and M. Zasadzinski, "Simultaneous observer based sensor diagnosis and speed estimation of unmanned aerial vehicle," *2008 47th IEEE Conference on Decision and Control*, 2008.

[25] A. Ansari and D. S. Bernstein, "Aircraft sensor fault detection using state and input estimation," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 5951–5956.

[26] G. Heredia, A. Ollero, M. Bejar, and R. Mahtani, "Sensor and actuator fault detection in small autonomous helicopters," *Mechatronics*, vol. 18, no. 2, pp. 90–99, 2008.

[27] E. C. Larson, B. E. Parker, and B. R. Clark, "Model-based sensor and actuator fault detection and isolation," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 5. IEEE, 2002, pp. 4215–4219.

[28] A. Freddi, S. Longhi, and A. Monteriu, "A model-based fault diagnosis system for a mini-quadrotor," in *7th workshop on Advanced Control and Diagnosis*, 2009, pp. 19–20.

[29] S. Bhattacharya and T. Basar, "Game-theoretic analysis of an aerial jamming attack on a UAV communication network," *Proceedings of the 2010 American Control Conference*, 2010.

[30] B. Saha, E. Koshimoto, C. C. Quach, E. F. Hogge, T. H. Strom, B. L. Hill, S. L. Vazquez, and K. Goebel, "Battery health management system for electric UAVs," *2011 Aerospace Conference*, 2011.

[31] Z. Zhao, X. Wang, J. Xu, and J. Yu, "A performance evaluation algorithm of stochastic hybrid systems based on fuzzy health degree and its application to quadrotors," *IEEE Access*, vol. 6, pp. 37 581–37 594, 2018.

[32] P. Toth and D. Vigo, *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 2002.

[33] S. S. Fazeli, S. Venkatachalam, and J. M. Smereka, "Efficient algorithms for autonomous electric vehicles' min-max routing problem," *arXiv preprint arXiv:2008.03333*, 2020.

[34] Y. Iida, "Basic concepts and future directions of road network reliability analysis," *Journal of Advanced Transportation*, vol. 33, no. 2, pp. 125–134, 1999.

[35] L. Tang, W. Cheng, and J. Liang, "Vehicle routing problem research based on road network reliability," *International Conference on Transportation Engineering*, 2009.

[36] J. Gao, "Optimization of distribution routing problem based on travel time reliability," *2009 International Conference on Information Management, Innovation Management and Industrial Engineering*, 2009.

[37] X. Zhang, Y. Pu, H. Liu, and D. Yang, "Vehicle routing optimization considering dynamic reliability during lasting period of traffic incidents," *International Conference of Logistics Engineering and Management (ICLEM), American Society of Civil Engineers.*, Sep 2010.

[38] N. Ando and E. Taniguchi, "Travel time reliability in vehicle routing and scheduling with time windows," *Networks and Spatial Economics*, vol. 6, no. 3-4, p. 293–311, 2006.

[39] "Fact sheet – small unmanned aircraft regulations (part

107).” [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615
[40] R. N. Allan *et al.*, *Reliability evaluation of power systems*. Springer Science & Business Media, 2013.
[41] B. E. Gillett and L. R. Miller, “A heuristic algorithm for the vehicle-dispatch problem,” *Operations Research*, vol. 22, no. 2, pp. 340–349, 1974.
[42] D. M. Ryan, C. Hjorring, and F. Glover, “Extensions of the petal method for vehicle routeing,” *Journal of the Operational Research Society*, vol. 44, no. 3, pp. 289–296, 1993.
[43] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.
[44] T. Caric and H. Gold, *Vehicle routing problem*. IntechOpen, 2008.
[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[46] 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.
[47] A. Ponza, “Optimization of drone-assisted parcel delivery,” Master’s thesis, University of Padua, 2016.
[48] D. Mu, C. Wang, F. Zhao, and J. W. Sutherland, “Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm,” *International Journal of Shipping and Transport Logistics*, vol. 8, no. 1, pp. 81–106, 2016.
[49] G. J. Lim, L. Kardar, and W. Cao, “A hybrid framework for optimizing beam angles in radiation therapy planning,” *Annals of Operations Research*, vol. 217, no. 1, pp. 357–383, 2014.
[50] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
[51] “Python interface,” access date: August, 2019. [Online]. Available: https://www.gurobi.com/documentation/8.1/quickstart_windows/py_python_interface.html#section: Python
[52] F. Bistouni and M. Jahanshahi, “Evaluating failure rate of fault-tolerant multistage interconnection networks using weibull life distribution,” *Reliability Engineering & System Safety*, vol. 144, pp. 128–146, 2015.
[53] N. Jozefowiez, F. Semet, and E.-G. Talbi, “Multi-objective vehicle routing problems,” *European Journal of Operational Research*, vol. 189, no. 2, pp. 293–309, 2008.
[54] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi, *Computational results with a branch and cut code for the capacitated vehicle routing problem*. IMAG, 1995, vol. 34.
[55] A. Jain and S. K. Jain, “Formulation and optimization of temozolomide nanoparticles by 3 factor 2 level factorial design,” *Biomatter*, vol. 3, no. 2, p. e25102, 2013.
[56] Q. Y. Kenny *et al.*, “Indicator function and its application in two-level factorial designs,” *The Annals of Statistics*, vol. 31, no. 3, pp. 984–994, 2003.

APPENDIX A

We prove that $\sum_{i=1}^{n} z_i + f_{n+1} = 1$, where n is the number of customers in a drone flight path.
Proof:
$For\ n = 1,\ \sum_{i=1}^{1} z_i + f_1 = z_1 + f_1 = p_{0,1} + q_{0,1} = 1$
We assume that for n=k, we have $\sum_{i=1}^{k} z_i + f_k = 1$, then we show it is true for $n = k + 1$.
$n = k + 1:\ \sum_{i=1}^{k+1} z_i + f_{k+1} = \sum_{i=1}^{k} z_i + z_{k+1} + f_{k+1}$
From the assumption: $\sum_{i=1}^{k} z_i = 1 - f_k$.
$\rightarrow 1 - f_k + z_{k+1} + f_{k+1} = 1 - f_k + f_k\ p_{k,k+1} + f_k\ q_{k,k+1} = 1 - f_k\ (-1 + p_{k,k+1} + q_{k,k+1}) = 1 - f_k(-1 + 1) = 1.$
End of proof.

APPENDIX B

The following showcases the notation and the mathematical model for the DDS-F-1 problem used to obtain the flight path and the ELOD value for one drone and a set of assigned customers.

Sets:
N' Set of assigned locations to the drone, 0 shows the depot.
Parameters:
n' Number of assigned customers,
t_{ij} Travel time from node i to node j, $(i, j \in N')$
ot Maximum operation time of drone.
Variables:
x_{ij} 1 if drone goes directly from node i to node j, 0 otherwise $(i, j \in N')$.

$$\text{Min} \sum_{c \in N' - \{0\}} (1 - f_c)\ d_c \tag{B.28}$$

$$\sum_{j \in N' - \{c\}} x_{cj} = 1 \qquad \forall c \in N' \tag{B.29}$$

$$\sum_{j \in N' - \{c\}} x_{jc} = 1 \qquad \forall c \in N' \tag{B.30}$$

$$\sum_{i \in N'} \sum_{j \in N'} t_{ij}\ x_{ij} \leq ot \tag{B.31}$$

$$y_i - y_j + n'\ x_{ij} \leq n' - 1 \qquad \forall i, j \in N' - \{0\}, i \neq j \tag{B.32}$$

$$\alpha_{ic}\ f_i - M(1 - x_{ic}) \leq f_c \qquad \forall i \in N', c \in N' - \{0\} \tag{B.33}$$

$$f_c \leq \alpha_{ic}\ f_i + M(1 - x_{ic}) \qquad \forall i \in N', c \in N' - \{0\} \tag{B.34}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i, j \in N'$$

$$y_i \geq 0, f_i \geq 0, f_0 = 1 \qquad \forall i, j \in N'$$

APPENDIX C

Two special cases of Figure 3 are considered here:

Case 1: customers are located within the same distance from the depot ($t_{0i} = t_{0j} \rightarrow \alpha_{0i} = \alpha_{0j}$). Without loss of generality, we assume that $d_i > d_j$ $(\Delta d = d_i - d_j)$.

$$0 < \alpha_{0il}, \alpha_{ijl} < 1 \rightarrow 1 - \alpha_{0il} < 1 - \alpha_{0il}\alpha_{ijl} \tag{C.35}$$

$$\frac{ELOD_I}{ELOD_{II}} = \frac{(1 - \alpha_{0il})\ d_i + (1 - \alpha_{0il}\alpha_{ijl})\ d_j}{(1 - \alpha_{0il})\ d_j + (1 - \alpha_{0il}\alpha_{ijl})\ d_i}$$

$$\rightarrow ELOD_{II} - ELOD_I = \Delta d(\alpha_{0il} - \alpha_{0il}\alpha_{ijl})$$

from (C.35): $ELOD_{II} - ELOD_I > 1$

Therefore, if two customers are located within the same distance from the depot, then the customer with higher demand should be served first.

Case 2: customers have the same amount of demand. Without loss of generality, we assume that $t_{0i} > t_{0j} \rightarrow \alpha_{0il} < \alpha_{ojl}$.

If $d_i = d_j \rightarrow \dfrac{ELOD_I}{ELOD_{II}} = \dfrac{1 - \alpha_{0il} + 1 - \alpha_{0il}\alpha_{ijl}}{1 - \alpha_{0jl} + 1 - \alpha_{0jl}\alpha_{ijl}}$

$= \dfrac{2 - \alpha_{0il}(1 - \alpha_{ijl})}{2 - \alpha_{0jl}(1 - \alpha_{ijl})} \rightarrow \dfrac{ELOD_I}{ELOD_{II}} > 1$

Therefore, if two customers have the same amount of demand, then the closer customer to the depot should be served first.

APPENDIX D

In the MS problem, the objective function is to minimize the required time needed to complete all flight paths. The MS model used here has the same constraints as the proposed DDS-F model except that is doesn't have the constraints to calculate failure probabilities and the objective function is the minimization of the maximum drone flight time ($Min\ Max\ \{\ \sum_{i \in N} \sum_{j \in N} t_{ijl}\ x_{ijl}\}$). Constraints (11)-(18) provide a feasible area of drone flight schedule without failure consideration. This MinMax problem can be changed to a linear model by using the new variable $u = max\ \{\ \sum_{i \in N} \sum_{j \in N} t_{ijl}\ x_{ijl}\}$ as follows:

$Min\ u$

$$u \geq \sum_{i \in N} \sum_{j \in N} t_{ijl}\ x_{ijl} \qquad \forall l \in L$$

Constraints (14)-(21)