Drone Scheduling Optimization Considering Capacity and Reliability of Batteries

by
Maryam Torabbeigi

A Dissertation submitted to the Department of Industrial Engineering,

Cullen College of Engineering

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Industrial Engineering

University of Houston
December 2020

Dedicated to

my dear parents Fatemeh and Morad,

and my beloved siblings,

with all my love

Acknowledgements

It is a pleasure to thank those who made this dissertation possible. In the beginning, I offer my deepest gratitude to my advisor, Dr. Gino Lim, for his immense knowledge, motivation, and patience that have given me more power and spirit to excel in the research. Many thanks to the dissertation committee members, Dr. Qianmei Feng, Dr. Taewoo Lee, Dr. Cumaraswamy Vipulanandan, and Dr. Jagannatha Rao, for their insightful comments that improved the work's contents.

I would like to express my sincere gratitude to Dr. Sitton. I have been very fortunate to had the opportunity get to know Dr. Sitton and work with him as his TA for several years. I learned a lot from him during most of my time as a Ph.D. student.

I am extremely grateful to my wonderful friends, Reza and Mohammad, who made the past five years very enjoyable with their kindness, support, and beautiful memories. I am thankful to my dear friends Amir, Navid, Saba, Anahita, SeonJin, and Saeedeh for their warm hearts and supports.

Last and most importantly, my warmest gratitude goes to my beautiful mother and my kind father for all of their sacrifices, caring, encouragement, unconditional love, and support. I am thankful to my dear siblings, Nahid, Moe, Mori, Marjan, Mari, and Ali for all the support and love they have given me during my life.

Abstract

Drones, or Unmanned Aerial Vehicles (UAVs), are aircrafts without a human pilot that are typically controlled and programmed by ground base centers. Although drone utilization is becoming more commonplace in recent years, there are a few challenges and drawbacks that should be addressed in drone scheduling such as: 1) limited weight capacity, 2) accurate estimation of battery consumption, 3) drone failures during the flight, 4) long and continuous flight missions, and 5) battery capacity degradation over time. This research aims to address these challenges in three separate studies.

The first contribution of my dissertation addresses the first and second challenges. Experimental data shows that the battery consumption rate (BCR) is a linear function of the payload amount. Any design of a parcel delivery system using drones then needs to consider the BCR, which includes strategic planning of the delivery system and operational planning for a given region. We developed a minimum set covering model for the strategic planning and a mixed integer linear programming problem (MILP) for the operational planning. In order to improve the computational time of the operational planning model, a variable preprocessing algorithm and several upper and lower bounds on the objective function are proposed. The results indicate the sequence of visiting customers impacts the remaining charge and that 3 out of 5 (60 %) flight paths are not feasible if the BCR is not considered. The proposed computational techniques enabled us to solve all the tested problems, which was not possible without them. Among the three proposed lower bound generation methods, the network configuration method computationally outperformed the other two methods.

The second contribution of my dissertation addresses the third challenge. Drone-based delivery network is considered to ship parcels to customers. In the event of failures, the demand of the corresponding customers would not be satisfied which is how we account for the lost demand. Therefore, drone failures in scheduling can be considered to minimize the expected loss of demand (ELOD). An optimization model (drone delivery schedule with drone failures (DDS-F)) is developed to determine the assignment of drones to customers along with the corresponding delivery sequence. A Simulated Annealing (SA) heuristic algorithm is proposed to solve the model 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 DDS-F model outperformed the well-known Makespan problem in reducing the ELOD by 23.6% on a test case. The proposed SA algorithm was able to reduce the computational time by 44.35%, on average, compared with the exact algorithm.

The third contribution of my dissertation addresses the last two challenges. A new approach based on a concept of autonomous battery swap stations (ABSSs) is proposed to reduce the time for battery swap in a drone-aided surveillance mission. A MILP model is proposed to determine the ABSS location based on the limitation on the revisiting gap between two consecutive visits at surveillance waypoints. A battery management algorithm is also proposed to optimize the number of batteries to be secured at a station for each drone with the goal of minimizing the battery acquisition and replacement cost over a planning horizon. The impact of average and standard deviation of battery State of Charge (SOC) on capacity degradation is considered in this study. Numerical results show that delay in charging the battery can improve the battery end-of-life by 15.6% when charging is delayed 25% more for a test case. The results also show that the battery end-of-life is a linear function of number of required batteries.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background & Motivation

Drones, or Unmanned Aerial Vehicles (UAVs), are aircrafts without a human pilot that are typically controlled and programmed by ground base centers. There are broad categories of unmanned aircrafts based on their size from small to big ones. They can be equipped with sensors, cameras, and other advanced scientific equipment [9]. Because of the attractive features of drones, there has been an increasing demand for civilian applications in recent years in areas such as: product delivery [10, 11], surveillance missions [12, 13], health-care [14, 15], humanitarian logistics [16], traffic management [17], scientific measurements [18], pollutant studies [19] and mapping [20].

Unlike ground vehicles that require road infrastructure, drones are able to fly directly from a starting location to a destination. Especially in rural areas, road networks are not evenly spread out. In the event of disaster, roads, bridges and railways might be damaged and become inaccessible for several days. For instance, one month after hurricane Maria in Puerto Rico, some towns continued to be isolated and delivery of emergency items was difficult [21] due to limited access to the area. Figure 1.1 shows water standing in Ponce, Puerto Rico, one week after hurricane Maria. In such cases, drones are able to fly and serve victims in areas that had been struck by disaster.

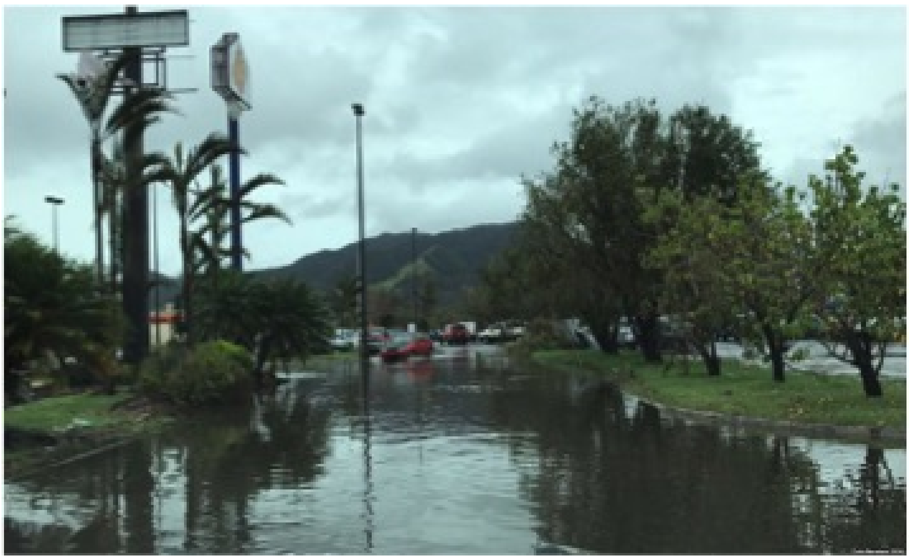


Figure 1.1: Standing water in Ponce, Puerto Rico, after hurricane Maria, 2017 [3]

Drone speeds can reach speeds up to 60 miles/hour, even for small-sized drones, which is much faster than the speed of other means of transportation when accounting for the congestion or obstacles on the way. Due to the limitation of road infrastructure and traffic congestion, extended travel time is very often seen especially in the case of ground vehicles [22]. Drones have the ability to travel to locations where it might be dangerous for ground vehicles and humans, which makes them ideal for remote surveillance, emergency response, delivery to remote locations, and disaster response. For example, drones were used to fly over the Arctic Ocean for scientific observations and research, one of the most difficult places on Earth because of the cold sea, ice dynamics, and aircraft icing. They are able to fly low enough to take detailed measurements that could not be taken with manned aircrafts or satellites alone [18]. Another example shows a drone's ability to fly inside buildings and reach the victims who are stuck there or immobilized [15]. Drones can be a cheaper alternative to manned vehicles such as trucks because they need fewer labor and typically have lower prices than manned vehicles. Drones are capable of carrying payloads for delivery. They can help improve the efficiency of parcel delivery or in coordination with ground vehicles as well [14].

In general, drones can either ship parcels in last-mile deliveries [23–25] or they can survey the environment to collect data [26]. Shipping cost and delivery time are the two most important factors for online shoppers to decide whether to continue to checkout or abandon their cart. A survey conducted in April 2017 by Statista [4] indicates that the major reasons for canceling online shopping include high shipping cost (40%) and long delivery time (35%) (Figure 1.2). There are many companies dealing with shipping services whose cost of shipping and services vary greatly. Hence, it is crucial for companies to reduce shipping costs and decrease delivery times. Delivery methods have evolved over time, and robots and unmanned systems have emerged as a new way to help deliver parcels in the near future. Figure 1.3 provides a comparison regarding shipping cost and delivery time between current delivery options offered by courier provider companies such as USPS or FedEx, and Amazon Prime Air delivery by drones [27]. Prices are calculated for delivery of a 5 lb. parcel within 10 miles.

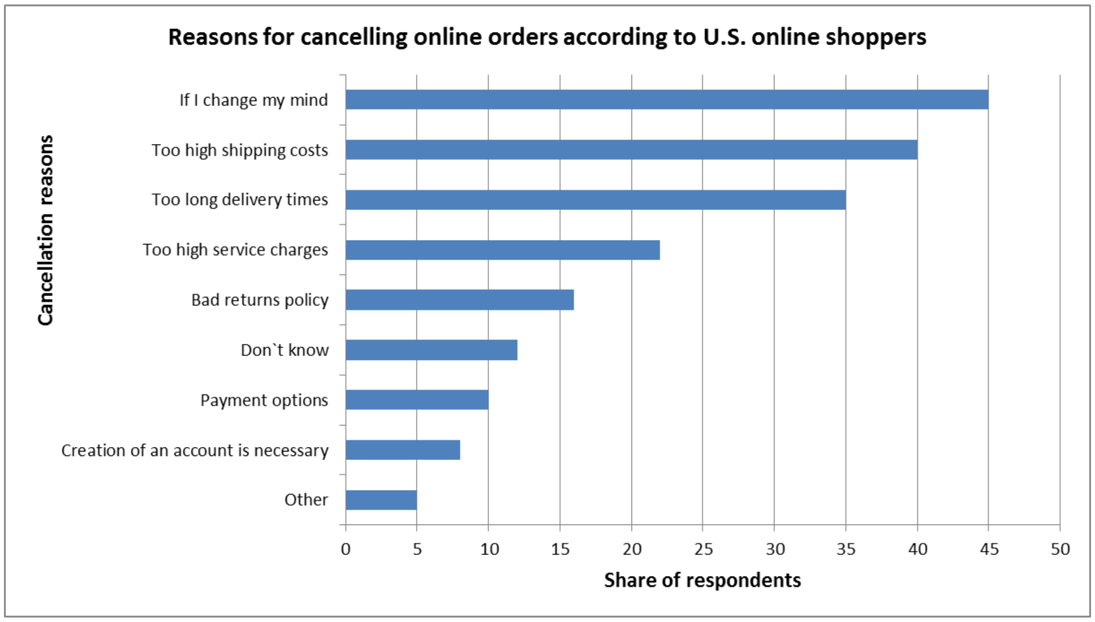


Figure 1.2: Reasons for canceling online shopping [4]

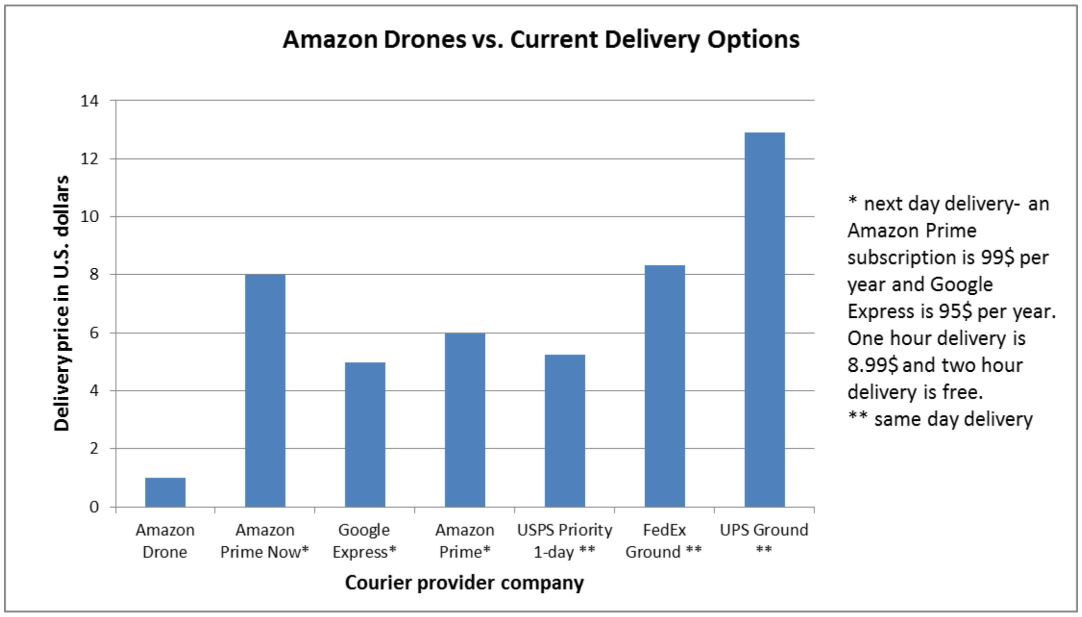


Figure 1.3: Delivery options costs and waiting time [5]

Another popular application of drones is to survey the environment and collect data. A group of equipped drones can fly over an area and monitor ground or aerial objects in a surveillance mission. The main challenge of using battery-powered drones in surveillance and monitoring missions is their limited flight time which prevents them from performing a long and continuous flight. They only can serve locations within their flight range restricted by the drone's battery life. This can be problematic specially in the case of a continuous or near-continuous flight. For example, a surveillance mission is in order to maintain the environment security or to collect necessary data from large areas.

Although there are numerous benefits of using drones for various applications, drawbacks of drones must be considered in designing a drone-based system, which may include limited weight capacity, estimation of battery endurance, drone failure during the flight, long and continuous flight missions, and battery capacity degradation over time. Therefore, this dissertation research focuses on drone scheduling for such applications as parcel delivery and surveillance, and provides methods to overcome those drawbacks.

1.2 Problem Description

To perform a delivery or surveillance mission using a group of drones it is essential to: 1) estimate the battery endurance to schedule a safe return of drones to the base station, 2) consider the failures of drones along the flight, and 3) to overcome the limited battery flight time for continuous flight missions.

1.2.1 Estimation of Battery Endurance

One of the biggest challenges that restricts the performance of drones is their limited flight time. It is necessary to estimate the energy consumption and energy requirement for any aircraft, and then schedule a safe flight based on that information [28]. Based on their requirements, drones may be powered by a piston-gasoline engine or by an electric motor. The endurance estimation of piston-gasoline engines are studied well. However, the range and endurance of electric engines is less studied [29]. Drones with civilian applications are usually battery-powered that can fly far based on the capability of their battery to supply electrical power. The maximum flight time is affected by different factors including payload amount, flight modes [30] and environmental conditions such as temperature [31] and wind [32, 33]. In delivery application of drones, the impact of payload amount on the battery endurance and range of flight is vital and is considered in this study.

Battery-powered drones operating in urban and rural areas are usually small in size and are able to carry a limited payload. The payload on a drone can, for example, be fire retardant or crop spraying chemicals. In military applications, the payload usually refers to different armament. The capability of commercial and military drones to carry payload can vary from less than one pound to thousands of pounds based on their specifications [34, 35]. Hence, one drone may be able to handle payloads for multiple customers. But, to ensure the flight safety, it is important to limit the total amount carried by the drone based on their respective weight capacity. This dissertation considers the limited weight capacity and addresses its impact on the estimation of battery endurance in a delivery of items by drones.

1.2.2 Drone Failures During Flight

Unmanned systems consist of different components, of which each component has a limited life span. Drones might fail during the flight due to various reasons such as human error, mechanical or even environmental issues. The types of vehicle failures can be major, where the drone is unable to return to the base location, or minor, where the drone can return to the base but is unable to complete the mission. The mishap rate of unmanned vehicles is significantly higher than manned vehicles and acts as a barrier to being operated widely in military and civilian applications. According to King *et al.* [36], by 2005, the UAV mishap rate was 100 times higher than that of manned aircraft and approximately half of them were attributed to aircraft failure. Unreliable UAVs are prohibited in crowded areas because there are risks for civilians. Although human error is the primary reason of the mishaps for manned aircrafts (approximately 85%), it is not the main reason for UAV mishaps (the related percentage is 17%) [37]. In practice, the reliability of drones is an important factor that is oftentimes missed in most optimization models for drone scheduling. In delivery applications of drones, a failure can result in losing payload and dissatisfaction of customers. We propose an optimization model for drone scheduling by considering the failure rate of drones. The proposed model minimizes the expected amount of demand that is lost due to drone failures.

1.2.3 Long and Continuous Flight Missions

One important challenge of using battery-powered drones is their limited flight time due to their battery capacity. Using dual or multiple batteries will increase the total flight time but also has drawbacks as it is not cheap and as the payload (battery weight) increases, the flight time will decrease. In order to serve long-distance locations, a drone battery can be charged along the flight or the drone can be initialized from a location near the destination instead of the base control center. By charging drone along the flight, it is able to fly for longer and can reach long-distance locations. There are different methods to charge drones during flight which include manned charging stations, autonomous charging stations, and wireless charging infrastructures along the flight. This research proposes to establish autonomous battery swap stations (ABSSs) in the surveillance environment to swap depleted batteries after each drone flight with fully-charged ones. By using ABSSs and providing sufficient number of batteries in each station, drones will be able to spend much more time in flight. Therefore, it is also important to determine the number of batteries required to enable a continuous flight mission that will allow drones to spend less time in stations waiting for the battery to be charged.

Lithium-ion batteries are a key solution to power storage systems for small-sized battery-powered drones. One drawback of these batteries is their durability and lifetime. The Lithium-ion batteries energy and capacity decrease over time. The battery capacity degradation rate depends on parameters including temperature, battery state of charge (SOC), depth of discharge (DOD), and current magnitude [38]. Among these different parameters, two major factors that affect the battery state of health (SOH) and its degradation are the average state of charge (SOC) and the swing of the batteries [2]. Over time, the battery capacity will decrease and therefore, will need replacement. The impact of average and standard SOC on the lifetime performance of Lithium-ion batteries and decisions regarding battery replacement and charging policies are studied in this research. Also questions such as how many batteries are need, when the battery should be replaced, and how they should be charged to increase their lifetime are answered in this study as well.

1.3 Objectives & Contributions

This research aims to address the mentioned challenges in three separate studies. The first work studies the estimation of battery endurance and the limited weight capacity, the second work is related to the failure of drones, and finally, continuous flight missions and battery capacity degradation over time are addressed in the third work.

For the first work, a delivery application of drones is considered to examine the impact of payload amount on the battery consumption rate (BCR). Customers are spread in a given area and their demand is satisfied by a group of drones. The amount of payload that a drone can carry is limited which affects the BCR. This work contributes to the literature through the following:

- Experiments conducted on a Phantom 4 pro+ drone to collect real data to investigate the effect of carried payload on the BCR. Experimental data shows that the BCR is a linear function of the payload amount.
- The design of a parcel delivery system including strategic and operational planning is proposed. The strategic planning portion decides the number of depots and their locations, and the operational planning portion determines the number of drones should be used daily, assignment of customers to drones, and sequence of visiting assigned customers for each drone.
- Novel mathematical models is provided for strategic and operational planning by developing a minimum set covering and a MILP model, respectively.
- A variable preprocessing algorithm, a primal bound generation method and three

different dual bounds are implemented a on the objective function to improve the computational time of operational planning model.

The second work considers the limited weight capacity and failure occurrences of a drone. We considered a delivery network using drones to ship parcels to customers. During the flight it is possible that a drone fails and is unable to deliver packages to customers. Therefore, in such cases the customer's demand is not satisfied and is considered lost demand. Our findings indicate the more reliable the network is, the less amount of demand will be lost. This work contributes to the literature through the following:

- The concept of calculating expected loss of demand (ELOD) is proposed to evaluate the reliability of the delivery network.
- A novel drone delivery schedule model with drone failures (DDS-F model) is proposed which is a MILP model that incorporates the impact of drone failures in the assignment of drones to customers and the flight path planning.
- Implementing a Simulated Annealing (SA) algorithm to improve the computational time to get optimal or good solutions in a reasonable time. In proposed method, the seed sequences are created based on the Petal algorithm to generate feasible flight paths and a local neighborhood search algorithm to find better solutions over iterations.

The third work considers a surveillance application of drones to address the issue of continuous flight missions and impact of battery degradation over time on the flight schedule and battery management. The flight range of drones is limited and is assumed to cover locations within their maximum flight range. Using battery-powered drones has numerous technical advantages for surveillance missions, albeit, the limited flight time is a major drawback for a long and continuous mission. This work contributes to the literature through the following:

- Proposing a drone-based continuous surveillance missions where batteries are swapped at autonomous battery swap stations (ABSSs) to increase drones flight time.

- Developing a novel MILP model to determine the location of ABSSs based on the maximum revisiting gap between two consecutive visits at each surveillance way-points.

- Considering the impact of State of Charge (SOC) average and standard deviation on the battery capacity degradation. The batteries are replaced with new ones at the end of their life time due to battery capacity degradation over time.

- Proposing a heuristic solution method to determine the number of required batteries secured for each drone based on the acquisition and replacement cost of the batteries over the planning horizon.

1.4 List of Outcomes

Journal Publications

- Maryam Torabbeigi, Gino J. Lim, and Seon Jin Kim, "*Drone delivery scheduling optimization considering payload-induced battery consumption rates*," Journal of Intelligent & Robotic Systems, vol. 97, no.3, pp. 471-487, 2020.

- Maryam Torabbeigi, Gino J. Lim, Navid Ahmadian, and Seon Jin Kim, "*An optimization approach to minimize the expected loss of demand considering drone failures in drone delivery scheduling*," **under revision** to Journal of Intelligent & Robotic Systems.

- Maryam Torabbeigi, Gino J. Lim, and Navid Ahmadian, "*Using Autonomous Battery Swap Stations (ABSS) to Enable Continuous Drone-aided Surveillance Missions*

Considering Battery Capacity Degradation," **submitted** to Journal of Intelligent & Robotic Systems.

Conference Proceedings

- Torabbeigi, Maryam , Gino J. Lim, and Seon Jin Kim, "*Drone delivery schedule optimization considering the reliability of drones*," In Proceeding of International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2018, pp. 1048-1053.

Conference Presentations

- Torabbeigi, Maryam, Gino J. Lim, Navid Ahmadian, and Seon Jin Kim. " *Impact of Drone Failures on the Drone Flight Schedule in a Delivery Application of Drones*," INFORMS Annual Meeting, Seattle, WA, Oct 2019.

- Torabbeigi, Maryam , Gino J. Lim, and Seon Jin Kim, "*Drone delivery schedule optimization considering the reliability of drones*," **presented** in International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2018, pp. 1048-1053.

- Torabbeigi, Maryam , Gino J. Lim, Seon Jin Kim, and Navid Ahmadian, "*Impact of Payload Amount on Battery Consumption Rate In a Delivery Application of Drones*," **Selected as Best Student Presentation** in INFORMS Annual Meeting. Phoenix, AZ, November 2018.

- Torabbeigi, Maryam , Gino J. Lim, Seon Jin Kim "*Application of Drones for Delivery of Emergency Items after Disasters*," In Proceeding of Texas Hurricane Center, Houston, TX, August 2018.

1.5 Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we review the relevant literature on delivery and surveillance application of drones, battery endurance, drone failure and battery charging.

In Chapter 3, we develop our drone delivery optimization model by considering the impact of payload amount on the BCR. We measure the relationship between BCR and payload amount through the real data collected. Two mathematical planning models are presented to first determine the location of ground depots and then the flight path of the drones. Afterwards, a variable preprocessing algorithm and several primal and dual bound generation methods are proposed to reduce the computational time.

In Chapter 4, we develop our mathematical optimization model to consider the reliability of drones in scheduling. An optimization model (DDS-F) is proposed to determine the assignment drones to customers and flight schedule. Then a Simulated Annealing (SA) heuristic algorithm is developed to reduce the computational time.

In Chapter 5, we develop a continuous drone-aided surveillance system by establishing autonomous battery swap stations (ABSSs). The maximum revisiting gap between two consecutive visit for each waypoint is guaranteed by proposed a mixed integer linear programming (MILP) model to determine the location of ABSSs. The battery management is implemented through a proposed algorithm that optimizes the number of assigned batteries to each drone, determines the replacement of degraded batteries and shows the impact of delay in charging on battery life time. We conclude the dissertation and discuss the possible related future research in Chapter 6.

Chapter 2

Literature Review

2.1 Vehicle Routing Problem (VRP)

The Vehicle Routing Problem (VRP) is a generic name referring to a class of combinatorial optimization problems in which customers are to be served by a number of vehicles. VRP, first described by Dantzig and Ramser in early 1959, was dedicated to optimizing the path used by oil trucks for a refinery factory in Atlanta [39]. VRP is a generalizations of Traveling Salesman Problem (TSP) in which the shortest route for a traveler is determined to visit all of the nodes in a given network. Dantzig, Fulkerson, and Johnson proposed a basic formulation for TSP [40] which can also be formulated as an integer linear model [41, 42]. Both TSP and VRP have a set of subtour elimination constraints that avoids subtours in the optimal solution. A formulation for subtour elimination constraints was proposed by Miller, Tucker, and Zemlin (MTZ) [43], a similar formulation of which is used in most of the studies in the literature. A basic VRP can be represented as the following problem.

Assume there are a set of nodes $N = 0, 1, ..., n$ where node j shows customers j and node 0 designates the base depot. There exists a set of vehicles $K = 1, 2, ..., k$. A given cost, $c_{i,j,k}$, is associated with each travel made from node i to node j by vehicle k. The binary variable

x_{ijk} is set to 1 if the vehicle k travels directly from node i to node j. Each customer is visited by one and only one of the vehicles. Each vehicle serves a set of customers starting and terminating at the depot with the goal of minimizing the total travel cost [44].

Several recent studies in the literature shown multi-UAV scheduling problems can be presented in the form of a mixed integer linear programming (MILP) that are similar to VRP models [45]. Similar to the VRP model, drones have to fly over some set of locations starting and terminating at a base location in the surveillance or delivery applications. The resulting operational planning problems can be modeled as a TSP [46, 47], a multiple TSP [48, 49], or even a VRP [14, 50].

2.2 Solution Approaches

2.2.1 Exact Solution Approaches

There are different exact solution methods to solve a TSP or VRP model and their variants. This can be categorized into: (1) direct tree search methods, (2) dynamic programming, and (3) set partitioning [51]. Direct tree search methods consist of sequentially building vehicle routes by using a branch and bound tree. One of the first results using this method was obtained by Christofides and Ellison [52]. Their results indicates the computational efficiency of the proposed branch-and-bound algorithm for the VRRP is significantly reduced compared with its efficiency in solving an equivalent TSP. Two well-known relaxations of the TSP are the assignment problem (AP) and the shortest spanning tree problem (SSTP). The solution or lower bound for either of these two problems is a valid bound on the TSP [53]. The branch-and-bound method is a well-known approach in the TSP and VRP literature [54]. Multiple studies have suggested better bounds and enhance its computational performance [55–58].

Eilon *et al.* [59] proposed one of the first dynamic programming approaches for the

VRP which was able to solve problems with up to 25 customers. Unlike the branching algorithms (branch and cut, branch and bound, and branch and price) that treat the VRP as an integer linear programming (ILP) or mixed ILP (MILP), dynamic programming approach breaks the problem into a number of simpler sub-problems [60]. The efficiency of the dynamic programming approaches depends on the number of states that can be eliminated through feasibility considerations. A well-known dynamic programming algorithm for the TSP is the Bellman-Held-Karp algorithm [61, 62]. Bellman-Held-Karp dynamic programming algorithm is recently implemented for a delivery system with combining a truck and a group of drones by Bouman *et al.* [63]. The authors considered a delivery system using a truck and a group of drones named the TSP with Drone (TSP-D). They implemented an approach to skip searching subproblems that are not relevant to the optimal solution. Their proposed method solved problems that were larger than the problems solved by the mathematical programming approaches presented in the literature at that time.

Balinski and Quandt [64] were one of the first authors suggest a solution method for the VRP based on the set partitioning approach. The basic set partitioning algorithm was extended using the column generation method by some studies in the literature and produced some of the best exact results [65–67]. The column generation method divides the given problem into two problems: the master problem and the sub-problem. The master problem is the original problem with a subset of variables. The sub-problem is a new problem that determines a new variable [68].

Variable preprocessing is a technique that sets the value of some variables before solving the model which reduces the computational time by shrinking the solution space. Several studies in the literature used the preprocessing technique based on capacity and travel time constraints to eliminate easily identifiable and infeasible situations from the search space in a routing problem [14, 69, 70].

Recently, Carlsson and Song [71] used the continuous approximation paradigm for a delivery truck that collaborates with a drone to make deliveries. The authors reduced the problem to a small set of parameters, and then determined how these parameters affect the outcome of the problem. In continuous approximation paradigm, the difficult to solve combinatorial quantities are replaced with simpler mathematical formulas that can provide accurate estimations of the original quantity [72]. Such an approach can be implemented for TSP and VRP problems and their variants [73, 74].

Pessoa *et al.* [75] developed a generic exact solver for the VRP and its variants to be solved by a branch-cut-and-price (BCP) algorithm. The developed solver incorporates relaxation, rank-1 cuts with limited memory, path enumeration, and rounded capacity cuts. The authors showed the efficiency of the solver in finding either comparable or better solutions than the specific algorithms for all VRP variants tested.

2.2.2 Meta-Heuristic Approaches

The complexity of the VRP model increases for larger cases. Therefore, the model requires methods to be implemented to find satisfactory solutions faster [76]. Meta-heuristic algorithms make few or no assumptions about the type of problem. They start with an initial solution and then improve it over iterations. The meta-heuristic algorithms do not guarantee to find the optimal solution. The theoretical underpinnings of what makes one meta-heuristic more effective than another are still poorly understood [77]. The followings are well-known meta-heuristic algorithms to solve VRP models:

- Genetic Algorithm (GA): GA was first proposed by Holand [78]. It is commonly used to generate high-quality solutions by using mutation, crossover and selection operators [79]. GA starts with a set of initial solutions and a fitness value is associated with each solution. In each iteration, the solution with better fitness value are selected from the current population which generates the next generation [80].

It is common to use a direct representation and to use specially designed operators (such as crossover) for the VRP. The reason is that the classic operations, which works on binary strings, would not work well on a VRP tour [77]. Two commonly used crossover operators for the routes are OrderCrossover (OX) [81] and the Edge Assembly Crossover (EAX) [82]. In general GA is not as competitive as other meta-heuristics at solving the VRP. However, Nagata [82] and Berger *et al.* [83] reached the best known solution by using GA and are competitive with Tabu Search methods.

- Simulated Annealing (SA): Inspired by the annealing in metallurgy, SA provides an approximate solution for the VRP based on a randomized local search algorithm [84]. It is usually used for the problems with a discrete search space such as TSP. The neighbor solution generated in each iteration is accepted with a probability dependent on the quality of the solution and a global parameter named *temperature*, T. The parameter T is reduced through the algorithm and controls the probability of accepting worst solutions. The final value of T can be used as a stopping criteria as well. SA can be preferable to exact algorithms such as Branch and Bound for finding the approximation of the global optimum in a limited amount of time. Robuste *et al.* were one of the first authors to implement the SA for the VRP [85]. Osman used the SA algorithm for the VRP by expanding upon many areas of the basic SA [86].

- Tabu Search (TS): TS is a neighborhood-search algorithm which iteratively searches the better solution in the current solution neighbourhood [87]. To overcome cycling between solutions, TS keeps a list of solutions that have already been investigated. These solutions are called *tabu* and are forbidden for investigation in the next moves. The first implementation of TS for the VRP was proposed by Willard [88]. Osman proposed a SA algorithm for the VRP that produced competitive results in compared with previous heuristic methods [86]. He developed Best-Improvement (BI) and

First-Improvement (FI) methods to search the neighbourhood in each move. Toth and Vigo proposed a process to remove direction of search from the neighbourhood that are unlikely to produce a better solution [89].

- Nearest Neighbor: In this approach, the closest node to the current node is added repeatedly. This heuristic has the same procedure as a spanning tree [87]. It was one of the first algorithms used to find the approximation of the TSP which yields a short, fast tour, but this is usually not optimal.

2.2.3 Swarm Intelligence (SI) Approaches

In the last decade, diverse efforts have been made to develop swarm intelligence methods. SI is the discipline that deals with systems composed of many individuals that coordinate using decentralized control and self-organization. The individuals have interactions with other individuals and with their environment [90, 91]. Here we mention most commonly used SI algorithms:

- Particle swarm optimization (PSO): PSO is inspired by fish and bird schooling in nature and was developed by Kennedy and Eberhart in 1995 [92]. The PSO searches the objective function space by adjusting the trajectories of individual agents, called particle. Each particle moves based on the position of the current global best solution, its own best solution collected in its history, and randomly as well [91]. This algorithm has been successfully applied for different version of the VRP, such as VRP with time windows (VRPTW) [93, 94], capacitated VRP (CVRP) [95], and multi-depot VRP (MDVRP) [96].

- Ant Colony Optimization (ACO): The ACO is developed based on the foraging behavior of social ants. Ants use pheromone as a chemical messenger, and find the shortest path from a food source to the nest [97]. Ants-based algorithms are particularly suitable for discrete optimization problems [98]. In the ant colony optimization algorithms, an artificial ant is a simple computational agent that searches for good solutions to a given optimization problem [99]. The movement of an ant is controlled by pheromone, which shows the quality of the solution and evaporates overtime. Without such a time-dependent evaporation, the algorithm convergences to the local optimal solutions. The ACO results in good solutions with a proper pheromone evaporation mechanism [98]. The VRP is very similar to food-seeking behaviors of ants in nature by considering the central depot as the nest and customers as the food [100]. This algorithm was first used for the TSP and then has been successfully applied to the VRP [101–103].

- Bees Algorithm: After the success of the ACO algorithm, there have been a number of algorithms proposed to exploit the behaviour of bees. The Bees Algorithm was first proposed in 1977 by Or to solve the TSP [104]. Different variants of these algorithms have slightly different characteristics. For example, forager bees are allocated to different food sources to maximize the total nectar in the honey bee-based algorithms [105]. The virtual bee algorithm (VBA) was developed by Yang in 2005, in which the pheromone concentrations is linked with the objective functions more directly [106]. These algorithms are very flexible in dealing with discrete optimization problems such as routing problems [107, 108].

2.3 Application of Drones

Drones are able to perform tasks that were traditionally conducted by manned systems. Drones can perform both outdoor and indoor missions and can be equipped with various equipment and tools such as night vision cameras and thermal sensors depending on the task. The applications of drones cover a wide range of civil and military applications such as security enhancement [109], search and rescue missions [110], damage assessment [111, 112], health-care services [14, 113], border patrol operations [114, 115], exploration of other planets, such as Mars [116], observations of marine organisms, identification of oil spill locations [117, 118], and communication relays [14]. In this dissertation, the delivery and surveillance application of drones are specifically considered.

2.3.1 Delivery Application of Drones

Ground vehicles such as trucks are typically used to deliver parcels across the logistic networks. 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 [14, 119–122] or in collaboration with other ground transportation vehicles [123–125]. The scheduling model for the drone delivery problem is similar to the traveling salesman problem [126] when each destination location is served by only one vehicle, and vehicles start and finish their path at the same location [14, 121, 122].

More recently, large companies such as Amazon, Mercedes-Benz, United Parcel Service (UPS), and DHL have planned to utilize drones for delivery purposes [27, 127–130]. Amazon aims to provide a service named Prime Air to deliver packages up to five pounds to customers within 30 minutes [127] with an estimation of 1$ delivery cost [27]. German automaker Mercedes-Benz in partnership with drone startup Matternet proposed a delivery system with a combination of van and drone to deliver and pick up packages [128]. Logistics company Matternet had previously operated a medical supply delivery project by drones in Switzerland [131] in 2017. UPS also tested a residential delivery by a combination of drone and truck in February 2017 [129]. Logistics firm DHL launched a Parcelcopter delivery project in 2013 and they were able to complete 130 autonomous flight cycles [130] by June 2016. There are also many academic studies that consider the application of drones for delivery purposes [31, 123, 132–134]. Some companies like Microsoft created a UAV simulator software to improve vehicle navigation [135].

2.3.2 Surveillance Application of Drones

In surveillance missions, a group of drones equipped with specific sensors and cameras monitor ground or aerial objects. High-quality cameras, sensors, and other information gathering equipment make drones a suitable option for the surveillance missions. Traditional surveillance methods are typically limited by the stationary nature of the camera and in the case of wired cameras, can only cover a limited range. Therefore, for surveying large areas, an abundance of cameras may be needed. in addition, the human resources needed to observe and analyze the camera recordings can be very expensive [136, 137]. Aerial surveillance can be performed using a helicopter to achieve desired result, but it is very costly. Drones provide the ideal solution to the problems by addressing the limitations faced by other surveillance methods. Furthermore, drones have features that make them highly attractive for use in surveillance missions. They can enter narrow and confined spaces, produce minimal noise, and can be equipped equipment such as night vision cameras and thermal sensors which allows them to provide images that the human eye is unable to detect [138].

SkyX Systems Corporation, Canadian based company, claims that by using drones the traditional monitoring costs in pipeline monitoring can be reduced by nearly 90% [139]. Eurecat company and its partners developed a micro aerial robot for sewer inspection (ARSI) in Barcelona, Spain. A flying sewer drone inspects the state of the city's sewers, measures air and water quality and keeps track of the state of the walls and blockages. They expect this project reduces the time that workers are exposed to harsh conditions by at least 50% and the cost of inspections by 30%, and increases the number and the frequency of inspections [140]. Drones can be used in traffic monitoring as well where they can read plate numbers, and detect the vehicle speed. The southwest China police successfully used drones to monitor traffic [141]. In addition, drones can be used in homeland security application and monitor the border lines. They can be equipped with high-quality cameras, sensors, and other information gathering equipment which primes them for their specific missions [142]. The Predator B for example, a military drone used by U.S. Customs and Border Protection (CBP), monitored about 5,000 hours per year from 2013 to 2016 [143, 144].

2.4 Drone Battery Endurance

Some drone studies have considered a fixed value for the maximum flight time that is not affected by other factors. Kim *et al.* [14] proposed drones to be utilized in medicine delivery to patients as well as test sample pickups for blood and urine in rural areas by drones with a constant value flight time. Deng *et al.* [145] proposed a drone system for efficient power line inspection in China while vehicles have a fixed maximum flight time. Scott and Scott [146] provided two mathematical model for healthcare delivery by drones under limited flight time and a limited budget.

Many researches conducted works on characterizing the energy consumption of UAVs but do not consider the scheduling problem. Therefore, their focus is on more accurately estimating the mathematical model by describing drone energy consumption [28, 147–151]. The maximum flight time is impacted by different factors including payload amount, flight modes [30] and environmental conditions such as temperature [31] and wind [32, 33]. Any rotating-wing cycle will be in different modes including hover, climb, descend, or forward flight or even the combination of these modes. Due to different flow pattern through the rotor, the moment and blade element analysis have specific calculations in each flight routine [30]. In [31], the authors introduced a robust optimization model to address the uncertainties in the flight time due to variation in temperature within a day. In cold weather, battery performance was decreased until a warm-up period has passed. Wind, on the other hand, is an element that either decreases the flight time depending on if a drone flies against the wind or increases the flight time due to the Effective Translational Lift (ETL) phenomenon. Terning [32] and Al-Sabban [33] included the effect of wind in their model.

Another factor impacting the drone flight time is the weight of payload it carries. According to [28, 148, 152], there is almost a linear relationship between the payload carried and maximum flight time for lightweight carried products. By increasing the transported weight, the power consumption rate increases too. In [148], experimental data indicates a very slight difference between different flight modes (hovering, forward flight, landing, and translational flight maneuvers). Also in general, due to ETL, the power consumption rate is larger in hovering mode than the other modes.

2.5 Reliability and Failure Rate

2.5.1 Reliability

Reliability is the probability that a product will operate or a service will be provided properly for a specified period of time under the design operating conditions without failure [153]. For each component, it is assumed that the time-dependent failure rate has a bathtub shape according to Figure 2.1 [154]. During the burn-in period, failure occurs because of different reasons such as incorrect use procedures, poor test specifications, incorrect installation or setup. During the useful life period, the hazard rate is almost constant and the failures occur randomly or unpredictably. After the useful life period, there will be wear-out period, in which the failure rates increases. Effective replacement and preventive maintenance policies can reduce the rate of failures in wear-out period [155]. Reliability assessment is a feasible measurement to tackle the uncertainty of systems, and in some industrial region as in power industry it has been a critical index for ensuring the safety of systems [156].

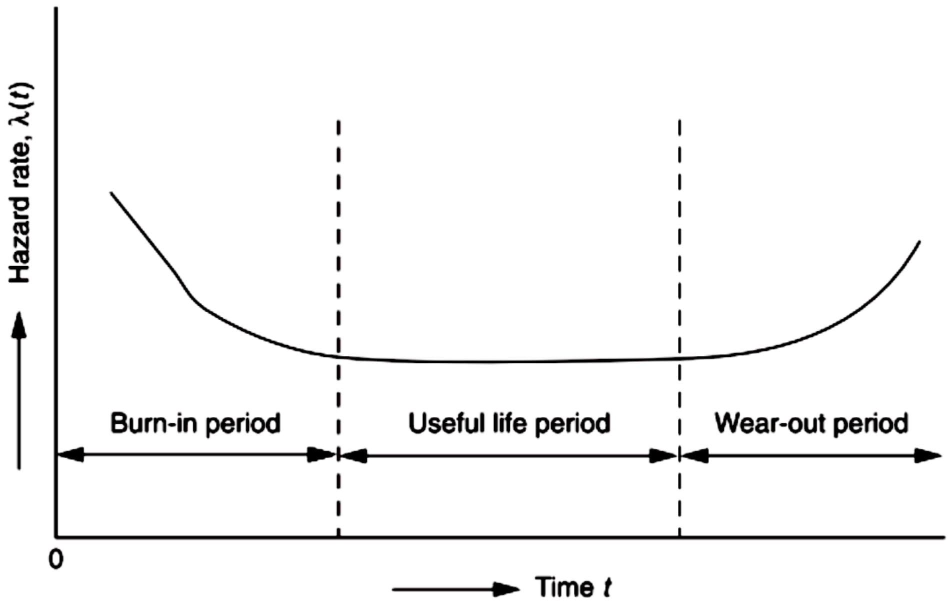


Figure 2.1: General Bathtub hazard curve

2.5.2 Reliability in Vehicle Routing Problem (VRP)

Road network reliability is a subject of interest in VRP reliability problems. There are two common definitions of road network reliability: connectivity reliability and travel time reliability [157–162]. Connectivity reliability is defined as a probability that there exists at least one path without disruption or heavy delay to a specific destination within a given time period. Travel time reliability is defined as the probability that traffic can reach a given destination within a stated time. There are currently 3 main areas of research in road network reliability analysis: (1) developing a modeling framework to investigate the reliability, (2) establishing a traffic management system by using hardware and software, and (3) developing a new optimization planning by incorporating road network reliability [159].

Lam *et al.* considered alternative paths for each pair of origin-destination locations [162]. They use the weighted sum of path travel time reliability index (RI) and path travel time index (TI) to quantify the attractiveness of each alternative path. Ando and Taniguchi considered the uncertainty of travel times in vehicle routing and scheduling problems with time windows [161]. The authors have considered the probabilistic travel time and an objective function which calculates the penalty cost based on the arrival time at each customer location. The Genetic Algorithm (GA) was implemented to solve the model. Gao studied the travel time reliability in a VRP problem is obtained by using the Monte-Carlo simulation method [163]. Each route travel time reliability is considered to be the multiplication of edge reliabilities. He proposed a multi-objective model to minimize the expected total travel time while satisfying travel time reliability constraint. Tang *et al.* used connectivity reliability, travel time reliability and road network capacity reliability to measure the capacity of the network [160]. They established an integer programming model and the an state transition probability formula which could adapt to the connection reliability and travel time reliability. The dynamic reliability analysis of a network by a heuristic genetic algorithm with Monte Carlo approach is applied in Zhang *et al.* study [158]. They defined the travel time reliability as the probability that the uncertain travel time does not exceed a given threshold. They also have implemented chance constraints to substitute the conventional deterministic time window constraints in their proposed model.

2.5.3 Failure Rate of Drones

There are five different type of failures for an aircraft or a drone [164]:

- *Soft failures* cause only light degradation of the drone's functions without interruption in the mission.
- *Moderate failures* cause a moderate degradation of the drone's functions that might interrupt the mission without severe damage.
- *Severe failures* cause heavy damages with low chance of repairing the drone.
- *Catastrophic failures* cause a crash of the drone with possible injuries or even death of persons on the ground.

The probability of a failure in a system is the sum of the failure probabilities of its components. Table 2.1 shows the percentage of failures in each main system of drone for six different drones [1]. The last shows the average percentage of failures for all of them. It can be seen that most of the failures are due to breakdowns in the power plant system that can be a result of the fatigue or the overheating of the engine [165].

Table 2.1: Percentage of drone failures in its main systems [1]

Drone Name	Power plant (%)	Flight controls (%)	Communications (%)	Human errors (%)	Miscellaneous (%)
Predator A	23	39	11	16	11
Predator B	53	23	10	2	12
Pioneer 2A	29	29	19	18	5
Pioneer 2B	51	15	13	19	2
Hunter 5A	38	5	31	7	19
Shadow	38	0	0	38	24
Average	38	19	14	17	12

The acceptable probability of a crash causing death or serious injury for the civilian airliners is 10^{-9} per flying hour. The initial reliability goal for the Predator drone [166] was a failure rate of less than 50 per 100,000 hours according to the US National Defense Magazine. Austin suggested an acceptable rate of one critical defect, that causes the aircraft to crash, in every 1000 flying hours for small-to-medium-size drones [1].

The available failure rate data in aviation is very limited because manufacturers tend to not release the information. The failure rate in aviation is measured by the number of occurrences per 100,000 flying hours [6]. Figure 2.2 shows failure rate over time for four different common types of drones (Pioneer, PR-3, Global hawk, and predator) and the F-16 jet fighter aircraft [1, 6].

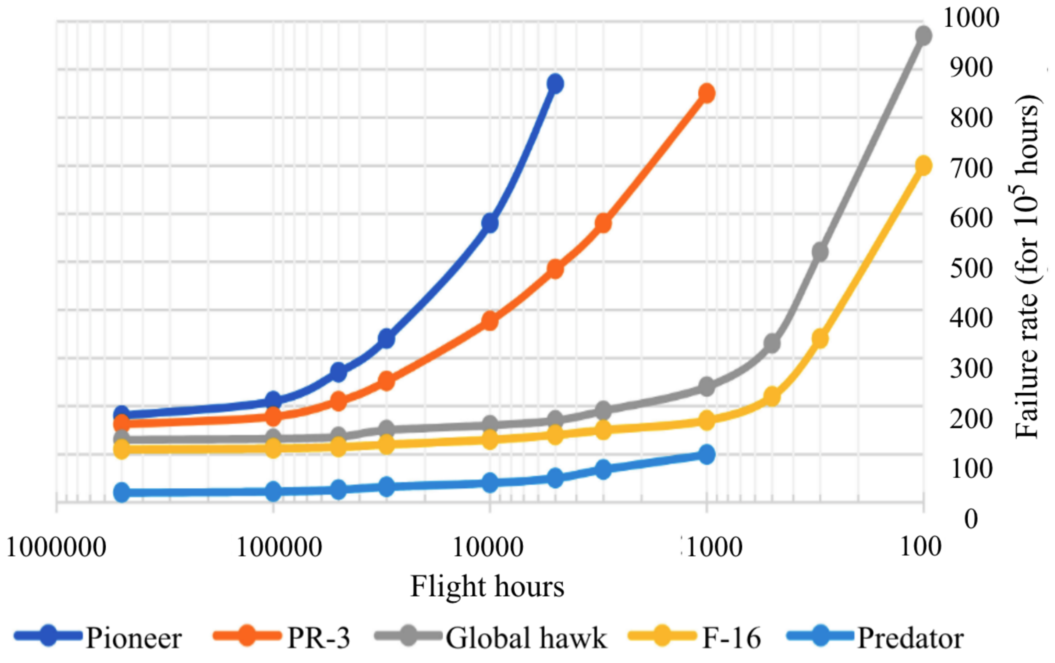


Figure 2.2: Failure rate vs. flight hours for the F-16 and some common drones [1, 6]

The historical reliability data depicted in Figure 2.2 shows the improvement over time. AS can be seen the failure rate trend for the Global Hawk almost matches that of the F-16 aircraft. Additionally, the smaller drones have a greater failure rate than the larger ones. Besides the technical and design aspects, the justification for this might be that they operate in a more hostile environment than the others [1].

2.6 Drone Battery Charging

The limited flight time of battery-powered drones is one of their most significant challenges in surveillance or delivery applications [10, 167, 168]. In order to serve long-distance locations, the battery has to be charged and the drone may need to load new packages (in delivery applications). There are different ways to increase drones flight range including: (1) loading drones with multiple batteries, (2) installing solar panels on the drone, (3) combining drones with other vehicles, (4) using wireless charging infrastructures, (5) charging batteries at the stations, and (6) incorporating autonomous battery charging and swapping systems.

Li-ion batteries are a common energy source for battery-powered drones. Using dual or multiple batteries may increase the flight duration at the downside of steep battery prices [2] and increased drone payloads. The battery consumption rate increases almost linearly by the amount of payload [10, 152, 169]. Therefore, using multiple batteries may not increase the flight time as much as one might expect.

Furthermore, fixed-wing drones are a type of drone that can carry solar cells and utilize solar energy as a primary source for a continuous long flights. However, an alternative power source is necessary in the absence of sunlight which can be a battery but still encounters limited flight time issues [170].

Another alternative is for drones to be used in combination with trucks or other ground vehicles to bring them to a location near the final destination. The ground vehicle carries drones to a location that is closer to final destinations meaning drones are able to meet their demand locations within their flight time. Drones can then return to the vehicle multiple times to be charged and to pick up new packages or transfer collected data. In such a case, both drone and the vehicle are able to perform missions and serve some locations. There are multiple studies that considered variations of drone combination with other vehicles

[123, 132–134].

Wireless charging infrastructures can be installed along the flight path [114, 115] or at stations [171] to increase flight time. In dynamic charging, the drone battery is wirelessly charged while flying over the charging infrastructure at a relatively slower speed than the normal flight speed [172]. Kim *et al.* [114] proposed installation of electrification line (E-line) battery charging systems on border walls. Using this approach, the drones do not return to the depot or stay on at the station for the batteries to get charged. A major challenge in using wireless charging infrastructures during flights or at stations is the low charging efficiency as well as long charge time [171, 173].

Another way to increase flight time is using the charging stations which can be stationary or mobile [174, 175]. Based on the size of the delivery or surveillance area, multiple recharging stations might be required in large urban areas to complete missions without fully depleting their battery [174]. Song *et al.* proposed using multiple charging stations for recharging and product-reloading purposes in a delivery application of drones [175]. Song *et al.* also considered the effect of payload on energy consumption and showed the impact of charging batteries in providing better service for the customers [175]. One drawback of using manned charging stations is that it requires to have a large amount of land and labor to swap or charge batteries when drones return to the station. Moreover, drones have to incur wait time for their batteries to get charged or swapped after landing at the station location.

An autonomous charging and swapping station is equipped with devices to remove the depleted battery and place it in a charger to get charged without any human interaction. Different design options for the components of an ABSS can be found in the literature [7, 176, 177]. Figure 2.3 shows a sample mechanism for an autonomous swap station. In Figure a, a drone is landing at the station. In Figure b, the depleted battery is extracted from the drone. And in Figure c, a new battery is placed inside the drone.

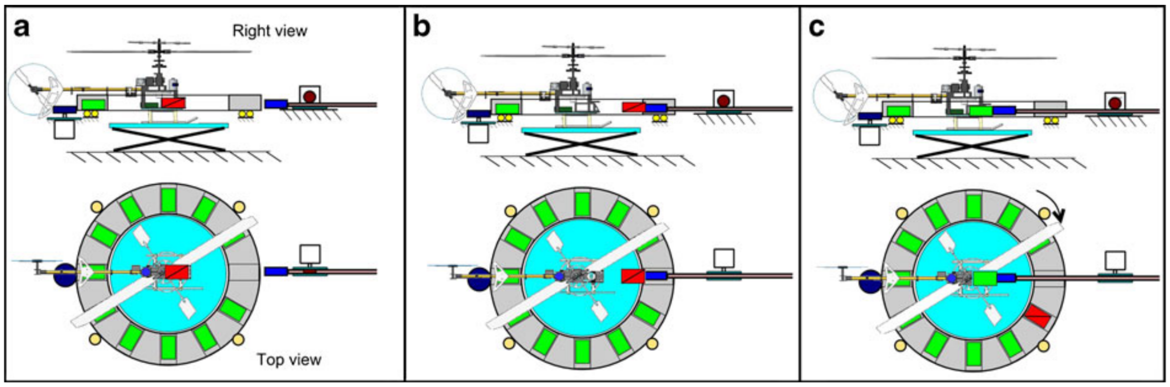


Figure 2.3: An example of an autonomous charging and swapping station[7]

The swapping time for most of the autonomous systems is around one minute [171]. This is a short period of time comparing to average time batteries needed to fully charge, which is between 45-60 minutes. These autonomous swapping stations can be used for both ground [178, 179] and aerial battery-powered vehicles [7, 167].

One application of ABSS can be seen with Tesla who aims to use battery swap technology to replace the model S battery in less than three minutes. Furthermore, they claim they can reduce this time to one minute at a cost slightly less than a full tank of gasoline [178]. Additionally, a cost-based multi-objective function is proposed in [179] to address monthly operating benefits of a taxi fleet with battery swapping stations. Fujii *et al.* [167] develop an automatic battery swap system for the drones. Their proposed method eliminates the need for charging drone's battery after a mission.

Chapter 3

Drone delivery scheduling optimization considering payload-induced battery consumption rates

3.1 Introduction

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 [180–182]. An accurate estimate of battery endurance during the planning stage is crucial in optimizing drone delivery schedule. For example, a delivery plan based on underestimated 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 amount of payload is one of key factors affecting the flight duration. Therefore, it should be considered in drone scheduling as it impacts on the battery endurance.

Some existing studies have considered the limitation on the total flight time or the payload amount in drone scheduling [14, 31, 146] and others considered the impact of carried payload on the total flight time calculation [152, 169, 183]. 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, 148, 151], but not in the context of drone scheduling. In order to avoid running out of charge and utilize drones efficiently, the battery consumption rate is included in drone routing in this research. A Phantom 4 Pro+ [184] 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 modelled using linear regression.

A group of drones is considered in this research 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, contributions of this research 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 chapter is organized as follows: Section 3.2 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 3.3, and the solution methodology for OP model is provided in Section 3.4. The numerical results and conclusion are discussed in Section 3.5 and Section 3.6, respectively.

3.2 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 [184] with specifications stated in Table 3.1.

Table 3.1: Phantom 4 Pro+ specifications

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

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 [148]. Furthermore, the power consumption in hovering mode is greater than the other flight modes due to effective translational lift [185]. Therefore, the data collected in the hovering mode was used in this research.

Table 3.2 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.). For example, it took 2.35 minutes for the battery charge to drop from 95% to 85% when the payload was 0.22 lb. The data are also plotted in Figure 3.1, 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 fully charged battery until it reaches the hovering mode. Therefore, the initial battery charge was set to 95% to be consistent in all experiments conducted in this research. Another parameter is the minimum remaining battery charge to ensure 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 3.2: Flight time data (in minutes) collected with Phantom 4 Pro+

Remaining	Payload amount				
battery level (%)	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

Table 3.3 lists the linear regression lines for the data shown in Figure 3.1 and the corresponding R-squared values. Since the R-squared values are 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

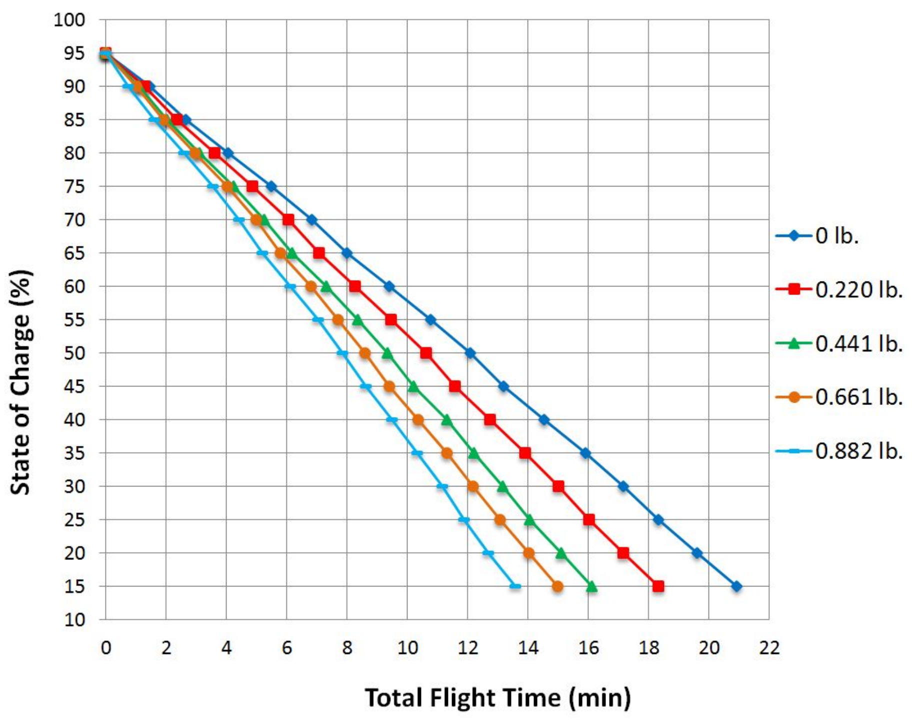


Figure 3.1: Total travel time vs. SOC for different amount of payloads

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 3.2.

Table 3.3: 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

In Figure 3.2, 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 3.3.2). We further analyze the resulting linear regression model:

$$BCR = \alpha \times payload + \beta, \tag{3.1}$$

where α is the slope and β is intercept. According to Table 3.4, the estimated value of α is 2.297 ($\frac{\%}{min \times lb.}$) and β is 3.879 ($\frac{\%}{min}$). 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.

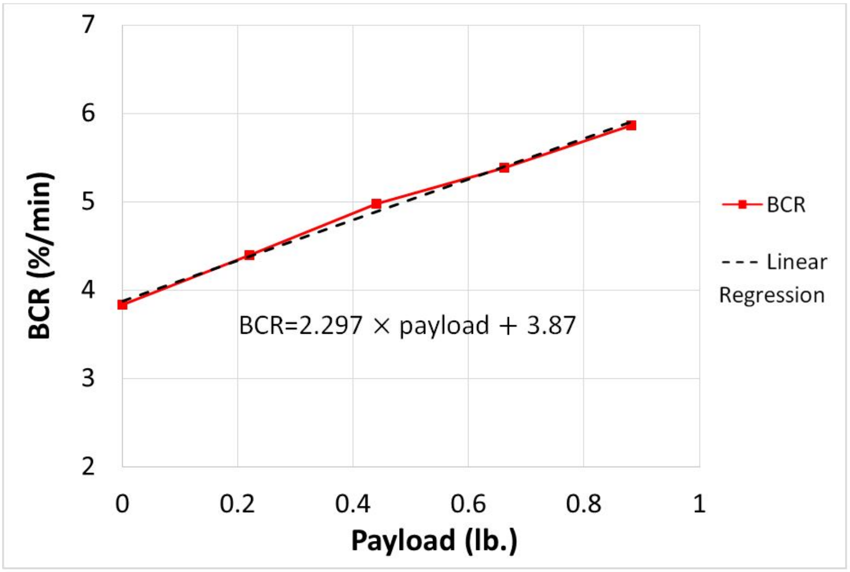


Figure 3.2: The relationship between the BCR and the payload amount

Table 3.4: 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

3.3 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 carrying 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 base. If the demand of a customer is greater than the carrying 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.

3.3.1 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. Figure 3.3 shows a flight path consisting of one depot and one customer. 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 (3.1),
$MinCh_k$	minimum required battery charge for safe landing of drone k, (k=1, 2, ...),
$MaxP_k$	maximum payload capacity of drone k, (k=1, 2, ...).

The battery charge at each stop (depot, customer's location, and again depot) is calculated through equations (3.2.1)-(3.2.3) by using Formula (3.1):

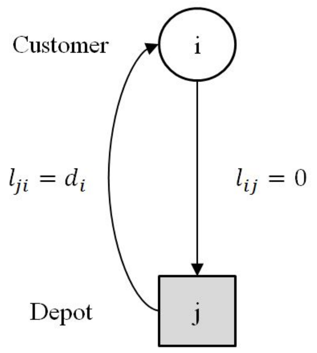


Figure 3.3: Covering customer i by depot j

$$SOC_j = 100, \tag{3.2.1}$$

$$SOC_i = SOC_j - t_{ji}(\alpha_k d_i + \beta_k) = 100 - t_{ji}(\alpha_k d_i + \beta_k), \tag{3.2.2}$$

$$\text{and} \quad RC_k = SOC_i - t_{ij}(\alpha_k \times 0 + \beta_k) = 100 - t_{ji}(\alpha_k d_i + \beta_k) - t_{ij}\beta_k. \tag{3.2.3}$$

The drone battery is assumed to be fully charged at the beginning of the flight as stated in equation (3.2.1). Equation (3.2.2) 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 (3.2.3).

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:

$$
\begin{aligned}
& RC_k > MinCh_k \quad \text{and} \quad t_{ij} = t_{ji}, \\
\rightarrow \quad & t_{ij}(\alpha_k d_i + 2\beta_k) < 100 - MinCh_k, \\
\rightarrow \quad & t_{ij} < \frac{100 - MinCh_k}{\alpha_k d_i + 2\beta_k}.
\end{aligned} \tag{3.3}
$$

The demand of customers changes daily. Finally, inequality

$$
t_{ij} < \frac{100 - MinCh_k}{\alpha_k MaxP_k + 2\beta_k} \tag{3.4}
$$

is obtained by replacement of d_i with the drone weight capacity. If inequality (3.4) holds, then inequality (3.3) holds for different amounts of load.

The right hand side of inequality (3.4) 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:

$$\text{Min} \quad \sum_{j \in S} f_j u_j, \tag{3.5}$$

$$\text{s.t.:} \quad \sum_{j \in S} \theta_{ij} u_j \geq 1, \qquad \forall i \in C \tag{3.6}$$

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

The objective function (3.5) minimizes the initial cost of opening a depot, while all the customers are covered by at least one open depot (Constraint (3.6)). The parameter θ_{ij} is calculated based on drone maximum coverage range.

3.3.2 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 3.3.1, 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 3.2. The following mathematical notation is defined to formulate the optimization model:

Sets:

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

Parameters:

M	A large number,
d_i	Customer i's demand,
$MaxP_k$	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 model for drone k, $(k \in K)$.

Variables:

x_{ijk}	1 if drone k goes directly 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 carried from node i to node j $(i, j \in \{C \cup D\})$,
SOC_i	State of charge (remaining battery level) at customer location $i(i \in C)$,
RC_k	Remaining charge of drone k at returning to depot $(k \in K)$,
y_c	The order of sequence of visiting customer c in a path $(c \in C)$.

The routing planning mathematical model is given below:

$$\text{Min} \quad \sum_{k \in K} h_k, \tag{3.7}$$

$$\text{s.t.:} \quad \sum_{j \in \{C \cup D\}} \sum_{k \in K} x_{ijk} = 1, \qquad \forall i \in C \tag{3.8}$$

$$\sum_{i \in \{C \cup D\}} \sum_{k \in K} x_{ijk} = 1, \qquad \forall j \in C \tag{3.9}$$

$$\sum_{i \in C \cup D} x_{ijk} = \sum_{i \in \{C \cup D\}} x_{jik}, \qquad \forall j \in C, \forall k \in K \tag{3.10}$$

$$\sum_{i \in C} x_{ijk} = \sum_{i \in \{C \cup D\}} x_{jik}, \qquad \forall j \in D, \forall k \in K \tag{3.11}$$

$$\sum_{i\in D}\sum_{j\in C} x_{ijk} = h_k, \qquad \forall k \in K \tag{3.12}$$

$$\sum_{i\in C}\sum_{j\in D} x_{ijk} = h_k, \qquad \forall k \in K \tag{3.13}$$

$$\sum_{i\in\{C\cup D\}} l_{ij} - \sum_{i\in\{C\cup D\}} l_{ji} = d_j, \qquad \forall j \in C \tag{3.14}$$

$$\sum_{i\in C}\sum_{j\in\{C\cup D\}} d_i\, x_{ijk} \leq MaxP_k\, h_k, \qquad \forall k \in K \tag{3.15}$$

$$SOC_j \leq SOC_i - t_{ijk}(\alpha_k l_{ij} + \beta_k) + M(1 - x_{ijk}), \qquad \forall i \in \{C \cup D\}, \forall j \in C, \; i \neq j, \forall k \in K \tag{3.16}$$

$$RC_k \leq SOC_i - t_{ijk}(\beta_k) + M(1 - x_{ijk}), \qquad \forall i \in C, \forall j \in D, \forall k \in K \tag{3.17}$$

$$RC_k \geq MinCh_k, \qquad \forall k \in K \tag{3.18}$$

$$y_i - y_j + n\sum_{k\in K} x_{ijk} \leq n - 1, \qquad \forall i, j \in C \tag{3.19}$$

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

and $$SOC_m = 100\%. \qquad \forall m \in D.$$

The objective function (3.7) is to minimize the number of drones used in the network. Constraints (3.8) and (3.9) ensure that each customer is served only once by exactly one drone. Flow conservation is guaranteed through constraints (3.10) and (3.11) by which when a drone enters a node, it must leave the node and visit another one until it completes its delivery tour. Constraints (3.12) and (3.13) show the utilization of drones. Constraint (3.14) is to satisfy customer demand. Constraint (3.15) 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 (3.16) and (3.17). 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 (3.16)). The remaining level at returning to the depot is also stated in Constraint (3.17). Parameter M

is a large positive number. On one hand, if the value of M is too large, it may 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 research, the value of M is determined by the following formula:

$$M = 100 + \max_{k \in K}(\alpha_k MaxP_k + \beta_k) \times \max_{i,j \in \{C \cup D\}} t_{ijk}. \tag{3.20}$$

A threshold value for the battery level is considered in Constraint (3.18) to ensure a safe return to the depot from a flight without running out of battery. Constraint (3.19) is to eliminate any sub-tours in the network [43].

3.4 Solution Approach

The OP model in Section 3.3.2 is an extended version of Vehicle Routing Problem (VRP), which is known to be hard to solve [186]. Therefore, this section introduces methods to solve the OP model faster. By preprocessing in Section 3.4.1 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 time to solve the model. Section 3.4.2 introduces a primal bound generation method and Section 3.4.3 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.

3.4.1 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 $(|C|+|D|)\times(|C|+|D|)\times(|K|)$. 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 (1), (2), and (3.24).

1) The model is an extension of VRP and Constraint (3.19) 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 as stated in the following:

$$\textit{Rule 1:} \quad \forall i,j \in \{C \cup D\}, \forall k \in K, \quad if \quad i=j \quad then \quad x_{ijk}=0. \tag{3.21}$$

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 as stated in the following:

$$\textit{Rule 2:} \quad \forall i,j \in C,\ \forall k \in K, \quad if \quad d_i+d_j > MaxP_k \quad then \quad x_{ijk}=0. \tag{3.22}$$

Note that if path ($depot \rightarrow customer\ i \rightarrow customer\ j \rightarrow depot$) is infeasible due to load capacity, then path ($depot \rightarrow customer\ j \rightarrow customer\ i \rightarrow depot$) is also infeasible because the summation of demand in both paths is d_i+d_j, which is greater than drone capacity.

3) A feasible path should satisfy Constraint (3.18). 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 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 3.4 shows a path consisting of two customers. According to Equation (3.23.4), the remaining charge depends on time to travel between locations, payload, and drone specifications. The battery level in each step of the path can be determined by (3.23.1)-(3.23.5) as follows:

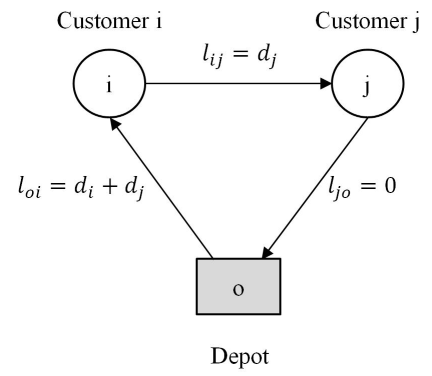


Figure 3.4: Example of a path consisting of two customers

$$SOC_o = 100, \tag{3.23.1}$$

$$SOC_i = SOC_o - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] = 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k], \tag{3.23.2}$$

$$SOC_j = SOC_i - t_{ijk}(\alpha_k d_j + \beta_k) = 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] - t_{ijk}(\alpha_k d_j + \beta_k), \tag{3.23.3}$$

$$RC_k = SOC_j - t_{jok}\beta_k = 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k] - t_{ijk}(\alpha_k d_j + \beta_k) - t_{jok}\beta_k, \tag{3.23.4}$$

and $RC_k \geq MinCh_k \rightarrow 100 - t_{oik}[\alpha_k(d_i + d_j) + \beta_k)] - t_{ijk}(\alpha_k d_j + \beta_k) - t_{jok}\beta_k \geq MinCh_k.$

(3.23.5)

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 (3.24). 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 as follow:

Rule 3: $\forall i, j \in C, \quad \forall k \in K, \quad if \quad \nexists o \in D:$

$$t_{oik}[\alpha_k(d_i + d_j) + \beta_k] + t_{ijk}(\alpha_k d_j + \beta_k) + t_{ojk}\beta_k \leq 100 - MinCh_k, \quad then \quad x_{ijk} = 0. \tag{3.24}$$

Note that even if path ($depot \rightarrow customer\ i \rightarrow customer\ j \rightarrow depot$) is infeasible as a result of insufficient remaining battery level, then path ($depot \rightarrow custome\ j \rightarrow customer\ i \rightarrow depot$) 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.

3.4.2 Primal Bound Generation

The objective function of 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 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)

```
Inputs:
  The number and location of base depots (result of SP model with objective function =
$\omega$
  Parameters in OP problem
Step 1:
  Assign each customer to the nearest depot.
Step 2:
  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 3.3.1 to find the location of centers; at least one customer is assigned to each depot. In Step 2, OP problem is solved for each depot to find the number of required drones and their flight paths. If for depot r, the optimal number of drones is z_r, then $\sum_{r=1}^{\omega} |D|$ is a primal bound for the OP problem because it represents a feasible solution.

3.4.3 Dual Bound Generations

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 3.4.3.1), network configuration (Section 3.4.3.2) based bound, and weight capacity based bound (Section 3.4.3.3). 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.

3.4.3.1 Lagrangian Relaxation

Some constraints of an optimization model are harder to satisfy than others. The general idea of a Lagrangian relaxation [187] 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 (3.14) 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 $\mu \in \{\mu_1, ..., \mu_{|C|}\} \geq 0$. The resulting model is stated as:

$$L(\mu) = \text{Min} \sum_{k \in K} h_k + \sum_{j \in C} \mu_j (d_j - \sum_{i \in C \cup D} l_{ij} + \sum_{i \in C \cup D} l_{ji}), \qquad (3.25)$$

$$\text{S.t:} \qquad (3.8) - (3.13), (3.15) - (3.19).$$

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.

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,
  $\mu_0, \theta$, max_iteration, threshold-value.
While(k < max_iteration or $\Delta\mu$ > threshold-value)
Do {
  Step 1: solve the Lagrangian dual problem (3.25) with Constraints (3.8)-(3.13), (3.15)-(3.19) 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 =$\theta \times \frac{\text{upper\_bound} - L(\mu_k)}{||\text{direction}||}$;
  Step 4: update $\mu$: $\mu_{k+1} = max\{0, \quad \mu_k + \text{direction} \times \text{step size}\}$;
  Step 5: converged or not? $\Delta\mu = |\mu_k - \mu_{k+1}|$ ;
  Step 6: k = k +1;
  } End While
```

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.

3.4.3.2 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 4.4 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 $E_C = (i, j) \in C \times C : i||j$ 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 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 clique and a maximum clique is a clique with the largest possible number of vertices [187]. 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 [188].

3.4.3.3 Drone Weight Capacity

The assignment of customers to drones has to meet drone weight capacity. Constraint (3.15) 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 $\rho_{ik} = \sum_{j \in \{C \cup D\}} x_{ijk}$ as a binary variable that gets value 1 if drone k serves customer i, and 0 otherwise. Constraint (3.8) and Constraint (3.15) in the OP model are simplified and rewritten by this new variable:

$$Constriant(3.8): \quad \sum_{j \in \{C \cup D\}} \sum_{k \in K} x_{ijk} = \sum_{k \in K} (\sum_{j \in \{C \cup D\}} x_{ijk}) = \sum_{k \in K} \rho_{ik} = 1. \tag{3.26}$$

$$Constriant(3.15): \quad \sum_{i \in C} \sum_{j \in \{C \cup D\}} d_i \, x_{ijk} \leq MaxP_k \, h_k,$$

$$\rightarrow \sum_{i \in C} d_i \, (\sum_{j \in \{C \cup D\}} x_{ijk}) \leq MaxP_k \, h_k,$$

and $$\rightarrow \sum_{i\in C} d_i\, \rho_{ik} \leq MaxP_k\, h_k. \tag{3.27}$$

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) [189] 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:

BPP-weight $$\text{Min} \sum_{k\in K} h_k, \tag{3.28}$$

s.t: $$\sum_{k\in K} \rho_{ik} = 1, \qquad \forall i \in C, \tag{3.29}$$

$$\sum_{i\in C} d_i\, \rho_{ik} \leq MaxP_k\, h_k, \qquad \forall k \in K, \tag{3.30}$$

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

The objective function (3.28) minimizes the number of drones used for delivery. Constraint (3.29) guarantees each customer is served by one drone while the weight capacity of a drone is satisfied through Constraint (3.30). 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.

3.5 Numerical Results

This section begins with a case study to demonstrate how the proposed methods work (Section 3.5.1). Further experiments are conducted to understand the impact of considering BCR with respect to payload in drone scheduling (Section 3.5.2), and investigate the computational efficiency of the proposed solution approach (Section 3.5.3). The Bron-Kerbosch Maximal clique finding algorithm [188] is implemented in MATLAB [190]. The other algorithms and the proposed SP and OP models are implemented in GAMS [191] and solved by CPLEX 12.6.3. [192]. All computational experiments were conducted using a Linux server with 24 cores and 384GB RAM.

3.5.1 A Case Study

A random test network having 20 customers and 5 depot candidate locations is shown in Figure 3.5, where square nodes are candidate depots and circle nodes represent the customers. A homogeneous fleet of drones similar to the drone used in Section 3.2 is used to serve the customers. According to the results of Section 3.2, the linear relationship between BCR and the payload amount is $BCR = 2.297 \times payload + 3.879$.

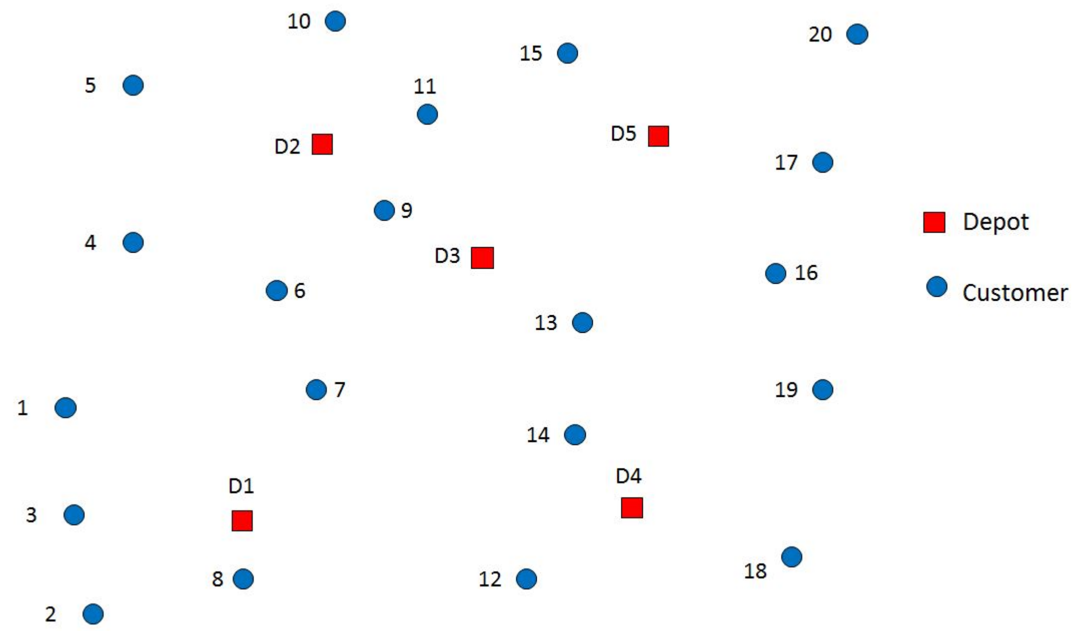


Figure 3.5: A random test network with 20 customers and 5 candidate locations

First, the covering range of each depot candidate location is identified as shown in Figure 3.6 and Table 3.5. This is determined by the drone battery specifications, the minimum required charge of 15%, and the maximum payload capacity of 1 lb.

Table 3.5: Covering customers by depot candidate locations for the case study data

Customer	Depot				
	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

Second, a subset of candidate locations is determined by the proposed SP model in Section 3.3.1. As can be seen from Figure 3.6, 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 3.3.1, two (D1 and D3) out of five candidate locations are selected to establish depots there. Third, the optimal assignment of customers and drone paths are determined by the OP problem as discussed in Section 3.3.2. We apply both the primal bound and dual bound generation

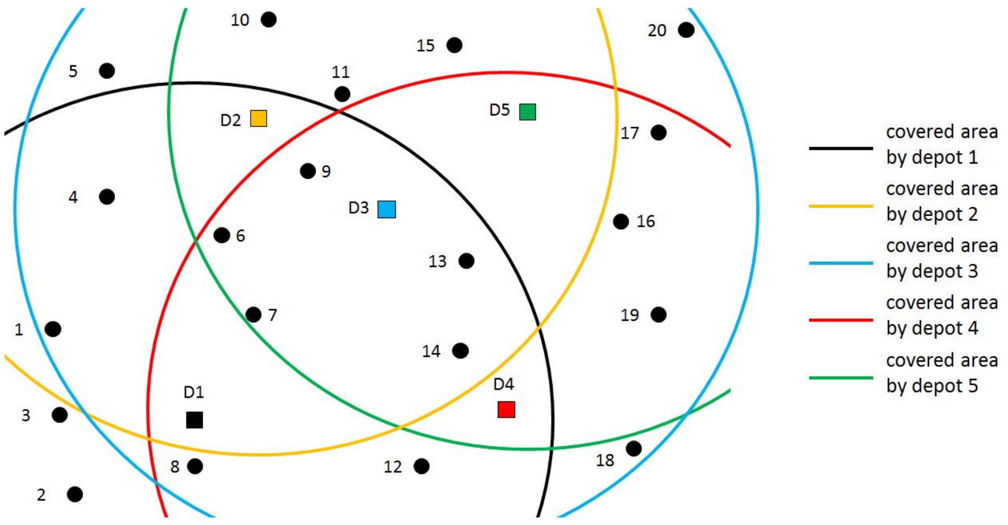


Figure 3.6: Covering customers by depot candidate locations

methods on the objective function (Section 4.4). 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 = $\frac{PrimalBound - DualBound}{PrimalBound} \times 100$.

- 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 3.6, in which four drones are assigned to Depot 1 and six drones are needed in D3.

- Dual Bound Generation

Different dual bounds are calculated according to Section 3.4.3 and the final dual bound on the objective function of OP is the maximum value of them. Table 3.7

Table 3.6: 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

shows the values of 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 $max\{9,\ 7,\ 8\} = 9$.

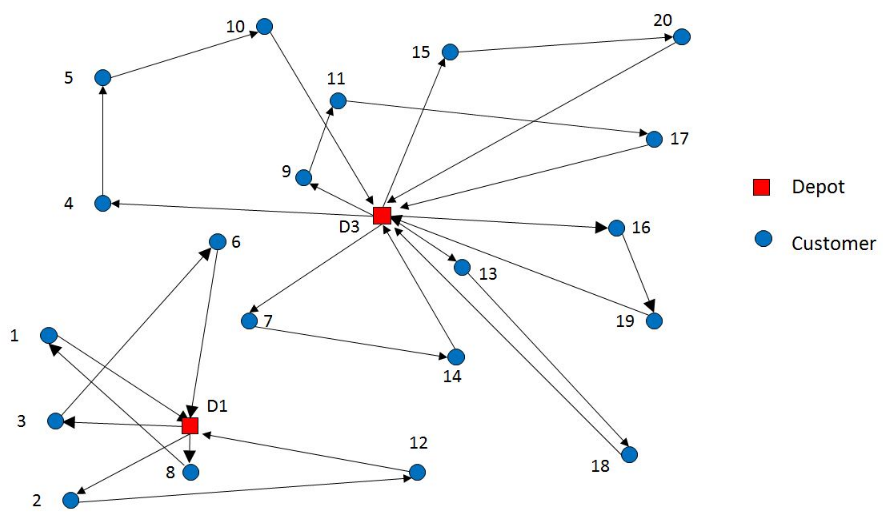
Table 3.7: dual bound calculation for the case study

Dual bound generation problem	Value
Lagrangian relaxation	$\lceil 8.54 \rceil = 9$
Network configuration	7
BPP-weight	8
Dual Bound	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 3.7, 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 3.8 shows the details of optimal flight paths, which includes the total demand, total travel time, and remaining battery level at the end of each flight path.



]

Figure 3.7: Results of OP model for the random test network

Table 3.8: Optimal drone flight path for the case study

Drone	Path	Total demand (lb.)	Total travel time (min)	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

3.5.2 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 backwards. For example, the flight path of drone 1 in Table 3.8, 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 3.9 shows the optimal flight paths in the case study (Figure 3.7) 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.

- 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 3.2) 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 3.1, 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

Table 3.9: Path reverse of optimal solution for the case study

Path		Total demand (lb.)	Total flight time (min)	Remaining charge (%)
Drone 1:	$D1 \rightarrow 8 \rightarrow 1 \rightarrow D1$	1	11.92	42.03
Reverse:	$D1 \rightarrow 1 \rightarrow 8 \rightarrow D1$			38.14
Drone 2:	$D1 \rightarrow 2 \rightarrow 12 \rightarrow D1$	0.9	16.88	19.93
Reverse:	$D1 \rightarrow 12 \rightarrow 2 \rightarrow D1$			14.22 *
Drone 3:	$D1 \rightarrow 3 \rightarrow 6 \rightarrow D1$	1	12.23	42.21
Reverse:	$D1 \rightarrow 6 \rightarrow 3 \rightarrow D1$			35.03
Drone 4:	$D3 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow D3$	0.7	18.07	17.25
Reverse:	$D3 \rightarrow 10 \rightarrow 5 \rightarrow 4 \rightarrow D3$			13.52 *
Drone 5:	$D3 \rightarrow 7 \rightarrow 14 \rightarrow D3$	1	12.53	39.05
Reverse:	$D3 \rightarrow 14 \rightarrow 7 \rightarrow D3$			34.92
Drone 6:	$D3 \rightarrow 9 \rightarrow 11 \rightarrow 17 \rightarrow D3$	1	15.81	26.84
Reverse:	$D3 \rightarrow 17 \rightarrow 11 \rightarrow 9 \rightarrow D3$			14.2 *
Drone 7:	$D3 \rightarrow 13 \rightarrow 18 \rightarrow D3$	0.6	18.35	19.38
Reverse:	$D3 \rightarrow 18 \rightarrow 13 \rightarrow D3$			12.93 *
Drone 8:	$D3 \rightarrow 15 \rightarrow 20 \rightarrow D3$	1	17.28	18.93
Reverse:	$D3 \rightarrow 20 \rightarrow 15 \rightarrow D3$			7.32 *
Drone 9:	$D3 \rightarrow 16 \rightarrow 19 \rightarrow D3$	0.8	14.11	33.34
Reverse:	$D3 \rightarrow 19 \rightarrow 16 \rightarrow D3$			31.29

* Infeasible flight path

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 highest value and it can fly longer when it has no payload. On the other hand, the BCR has the lowest value if it is fully loaded. For the drone used in Section 3.2, 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 3.7 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 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 3.10. 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 3.10: Optimal paths with the minimum BCR

Drone	Path	Total demand (lb.)	Total travel time (min)	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

3.5.3 Computational Efficiency of Proposed Solution Method

- Dual Bound Generation Methods

In this section, the performance of proposed dual bounds (Section 3.4.3) is tested by six different problems including two depots and 11 customers and homogeneous fleet of drones stated in Table 3.11. 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 demand to the test case 1 and the ratio of each test case travel time to the test case 1 are presented in first section of Table 3.11. The second section of the table shows the number of variable x_{ijk} reductions by using three rules of the preprocessing algorithm introduced in Section 3.4.1. Rule 2 and Rule 3 show the incompatibility among customers regarding the drone weight capacity and the remaining charge, respectively (incompatible customers cannot be assigned to one drone). 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.

Table 3.11: 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
# 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

The obtained dual bounds by using three different dual bound generation methods are presented in Table 3.12. 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 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 5. The network configuration and the BPP-weight method give the largest dual bound in 66.7% and 50% of these cases, respectively.

Table 3.12: 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 3.13 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. 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 almost 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 3.13, have the lowest number of variables fixed by the preprocessing algorithm in Table 3.12 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 run time 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 weigh capacity of drones. In test case 1 and 4, no variable is fixed by Rule 2 in Table 3.13 and these cases have the highest computational time by BPP-weight problem in Table 3.12. 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 the 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

Table 3.13: Computational time (seconds) with different dual bound generation methods

Method	Solved problem	Test case					
		1	2	3	4	5	6
Lagrangian relaxation	Dual bound	6,901.4	36.1	0.4	25,210.7	15,906.1	30.6
	Test case	880.8	0.3	0.2	18.0	12,722.8	0.2
	Total	7,782.3	36.5	0.6	25,228.7	28,628.9	30.8
Network configuration	Dual bound	0.02	0.01	0.02	0.01	0.01	0.01
	Test case	880.8	0.3	0.2	5.4	12,722.8	0.2
	Total	**880.8**	**0.3**	**0.2**	**5.4**	12,722.8	**0.2**
BPP-weight	Dual bound	0.2	0.1	0.1	0.3	0.2	0.1
	Test case	1280.9	0.3	0.3	18.0	0.8	0.2
	Total	1281.2	0.4	0.4	18.3	**1.0**	0.3

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 4.4. 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 3.14, 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 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 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 get 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 3.14, 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.

Table 3.14: Effect of preprocessing and bound generation methods on the computational time

Problem	Bounds	Prep.	time (s)	MIP objective function value	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

3.6 Conclusion

A delivery application of drones was studied in this research, 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 system. Among factors affecting the drone battery consumption, the payload amount and flight time were two factors studied in this research. 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 pre-processing algorithm and several bound generation methods were proposed to improve the computational time. The proposed models and the solution method were implemented on 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 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 configuration 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.

Chapter 4

Drone Delivery Schedule Optimization Considering the Reliability of Drones

4.1 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 [109], damage assessment [111, 112], health-care services [14, 113], border patrol operations [114, 115], and communication relay [14]. Among different applications, the delivery of lightweight parcels is one of the most rapidly growing civilian applications observed in recent years [10]. 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 [26, 27, 193]. Therefore, parcel delivery by drones is gaining more attention among courier companies, such as Amazon, DHL, and UPS [23–25].

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 [194]. Drone failures are inevitable, and they occur mainly due to mechanical issues, environmental conditions, cyber-physical attack, collision, or human errors [13, 195]. The mishap rate of unmanned vehicles is much higher than manned vehicles [36, 194], 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 [11]. 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 [196]. 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 [197]. 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 [198, 199], actuator [200–202], accelerometers and inclinometers [203], communication system [204] and battery [205]. However, these studies not only focus solely on drone health evaluation, but use a uniform health indicator for determining faults in all drone components [197, 206].

Studies have typically focused on one drone to capture faults and failures [197–206]. However, the impact of drone failures in the delivery network has not been well investigated. To fill this gap in the literature, this research 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 research. 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 of customers' demand, and the total travel distance [11].

In ground transportation network problems, such as the Vehicle Routing Problem (VRP) [207, 208], two common definitions of network reliability are [159] (1) *connectivity reliability*: the probability of at least one path existing between a pair of locations without disruption or heavy congestion [160], and (2) *travel time reliability*: the probability of traveling between a pair of locations within a specified time period [157, 158, 161].

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 [209]. 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 [156] 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.* [194] 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 Torabbeagi *et al.* [11] 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 research 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 research 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 [210] and Petal algorithm [211].

The rest of this chapter is organized as follows. Section 4.2 explains the method to calculate the expected loss of demand (ELOD). Section 4.3 formally describes the problem, followed by the optimization model formulation. Section 4.4 presents the proposed heuristic algorithm. In Section 4.5, several case studies and an analysis of the results are discussed. Finally, Section 4.6 concludes the research with a potential extension of this work.

4.2 Expected Loss of Demand (ELOD) Calculation

In this research, 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.

4.2.1 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 4.3 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:

$$f_i = f_{i-1} \cdot q_{i-1,i}, \qquad i \in \{1,2,...,n+1\}, \tag{4.1}$$

$$\text{and } z_i = f_{i-1} \cdot p_{i-1,i}, \qquad i \in \{1,2,...,n+1\}, \tag{4.2}$$

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:

$$D_i = \sum_{j=i}^{n} d_j, \qquad i \in \{1,2,...,n\}, (D_{n+1} = 0). \tag{4.3}$$

Proposition 1. *Both $z_i, i \in \{1,2,...,n+1\}$ and f_{n+1} form a probability mass function.*

Proof. 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.

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: $\sum_{i=1}^{k} z_i = 1 - f_k$.

$\rightarrow 1 - f_k + z_{k+1} + f_{k+1} = 1 - f_k + f_k\ p_{k,k+1} + f_k\ q_{k,k+1} = 1 - f_k\ (-1 + p_{k,k+1} + q_{k,k+1}) = 1 - f_k(-1+1) = 1.$

□

Therefore, $z_i, i \in \{1,2,...,n+1\}$ and f_{n+1} are used to calculate the ELOD as defined below:

$$ELOD = \sum_{i=1}^{n+1} z_i\, D_i = \sum_{i=1}^{n} z_i\, D_i + z_{n+1} \cdot 0 = \sum_{i=1}^{n} z_i\, D_i. \tag{4.4}$$

In Formula (4.4), 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:

$$\sum_{i=1}^{j} z_i = \sum_{i=1}^{j} f_{i-1}\, p_{i-1,i} = \sum_{i=1}^{j} f_{i-1}\,(1-q_{i-1,i}) = \sum_{i=1}^{j} f_{i-1} - \sum_{i=1}^{j} f_{i-1}\, q_{i-1,i} = \sum_{i=1}^{j} f_{i-1} - \sum_{i=1}^{j} f_i,$$

$$= f_0 - f_j \quad \rightarrow \quad \sum_{i=1}^{j} z_i = 1 - f_j, \tag{4.5}$$

$$\text{and } ELOD = \sum_{i=1}^{n} z_i\, D_i = \sum_{i=1}^{n} z_i (\sum_{j=i}^{n} d_j) = \sum_{i=1}^{n}\sum_{j=i}^{n} z_i\, d_j = \sum_{j=1}^{n} d_j (\sum_{i=1}^{j} z_i) = \sum_{j=1}^{n} d_j\,(1-f_j), \tag{4.6}$$

which only requires the information of the previous flight segments up to the current segment. Equation (4.6) is based on the customers' demand, and it does not depend on the carried payload within the flight segments.

4.2.2 An Illustration of ELOD Calculation

We illustrate the procedure of ELOD calculation using a small example in Figure 4.1, 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.

Let us examine each flight segment for path "*Depot* $\rightarrow i \rightarrow j \rightarrow$ *Depot*" in Figure 4.1 to calculate the failure probability and ELOD associated with each of the three segments.

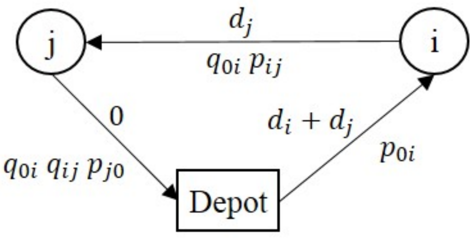


Figure 4.1: An example flight path consisting of one depot, two customers, and one drone

Table 4.1: A summary of ELOD calculations for Figure 4.1

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

The final results are summarized in Table 4.1.

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:

$$Network\ ELOD = ELOD_{Depot \rightarrow i} + ELOD_{i \rightarrow j} + ELOD_{j \rightarrow Depot} = p_{0i} \cdot (d_i + d_j)$$
$$+ q_{0i}\ p_{ij}\ d_j + q_{0i}\ q_{ij}\ p_{j0} \cdot 0 = p_{0i}\ (d_i + d_j) + q_{0i}\ p_{ij}\ d_j. \tag{4.7}$$

4.3 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 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:

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

Parameters:

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

Variables:

x_{ijl}	1 if drone l goes directly from node i to node j, 0 otherwise ,
f_c	probability of no failure arriving at customer c,
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:

$$\text{If } x_{ijl} = 1, \text{then } f_j = f_i \cdot P_l(t > t_{ij}), \quad \forall i, j \in N, l \in L, \tag{4.8}$$

where $f_0 = 1$. This "if-then" statement is formulated by adding parameter "M" (a large number) as the following inequality:

$$\alpha_{ijl} f_i - M \cdot (1 - x_{ijl}) \leq f_j \leq \alpha_{ijl} f_i + M \cdot (1 - x_{ijl}) \forall i, j \in N, l \in L, \tag{4.9}$$

where $\alpha_{ijl} = P_l(t > t_{ij})$. When variable x_{ijl} is equal to 1, Inequality (4.9) 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 (4.9) is used as Constraints (4.19) and (4.20) in the DDS-F model. The resulting MILP formulation for the DDS-F model is provided as follows:

$$\text{Min} \quad \sum_{c \in N - \{0\}} (1 - f_c)\ d_c, \tag{4.10}$$

$$\text{s.t:} \quad \sum_{i \in N \setminus \{c\}} \sum_{l \in L} x_{cil} = 1, \qquad \forall c \in N \setminus \{0\}, \tag{4.11}$$

$$\sum_{i \in N \setminus \{c\}} \sum_{l \in L} x_{icl} = 1, \qquad \forall c \in N \setminus \{0\}, \tag{4.12}$$

$$\sum_{i \in N \setminus \{c\}} x_{icl} = \sum_{j \in N \setminus \{c\}} x_{cjl}, \qquad \forall c \in N \setminus \{0\}, l \in L, \tag{4.13}$$

$$\sum_{c \in N \setminus \{0\}} x_{0cl} = 1, \qquad \forall l \in L, \tag{4.14}$$

$$\sum_{c \in N \setminus \{0\}} x_{c0l} = 1, \qquad \forall l \in L, \tag{4.15}$$

$$\sum_{c \in N \setminus \{0\}} \sum_{i \in N} d_c\ x_{cil} \leq w_l, \qquad \forall l \in L, \tag{4.16}$$

$$\sum_{i \in N} \sum_{j \in N} t_{ijl}\ x_{ijl} \leq ot_l, \qquad \forall l \in L, \tag{4.17}$$

$$y_u - y_v + n \sum_{l \in L} x_{uvl} \leq n - 1, \qquad \forall u, v \in N \setminus \{0\}, \tag{4.18}$$

$$\alpha_{icl}\, f_i - M(1 - x_{icl}) \leq f_c, \qquad \forall i \in N, c \in N \setminus \{0\}, l \in L, \tag{4.19}$$

$$f_c \leq \alpha_{icl}\, f_i + M(1 - x_{icl}), \qquad \forall i \in N, c \in N \setminus \{0\}, l \in L, \tag{4.20}$$

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

The objective function is the minimization of network ELOD for all flight paths. Constraints (4.11) and (4.12) state that each customer should be served once and by exactly one drone. Constraint (4.13) is the flow balance equation. Constraints (4.14) and (4.15) 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 (4.16). The drone total flight time is also limited as stated in Constraint (4.17). Constraint (4.18) prevents the sub-tours in the network according to the Miller-Tucker-Zemlin formulation [43]. Constraints (4.19) and (4.20) are related to the ELOD calculation and are obtained from Inequality (4.9).

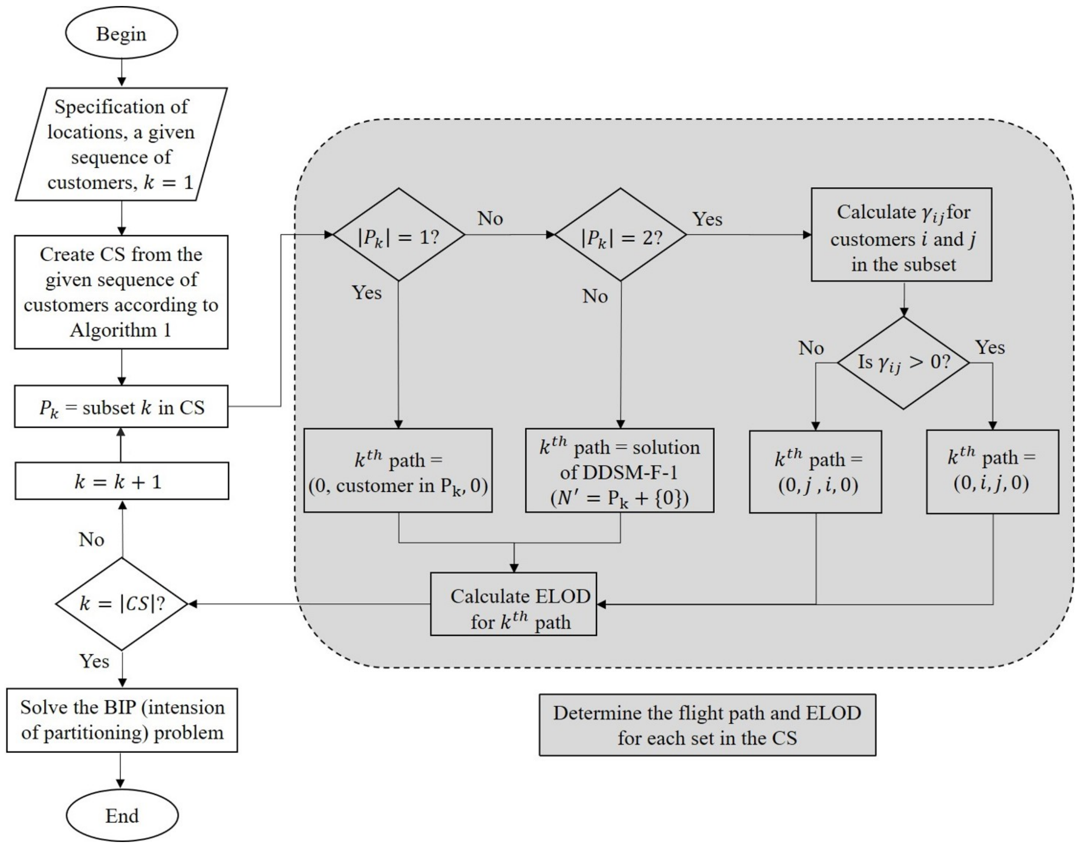
4.4 Solution Method

The DDS-F model is a variant of the Vehicle Routing Problem, which is known to be NP-hard [186]. 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) is a commonly used meta-heuristic algorithm for finding the global optimum of a given function specially in discrete search spaces such as the TSP [212]. The SA may be preferable to exact algorithms (such as gradient descent, Branch and Bound) when the approximation of the global optimal solution is more important than finding a precise local optimum in a fixed amount of time. The main characteristic of the SA that differs from the other meta-heuristic algorithms is the exploration of worse solutions. In each iteration, a worse solution might be chosen with a probability (that decreases over iterations) in order to escape from the local optimal [213]. A recent comparison between Nearest Neighbour, Tabu Search, and SA methods for the VRP shows the SA provides better solutions [214]. Vincent *et al.* verified with benchmark data of the capcacitated vehicle routing problem (CVRP) that SA performs well and efficiently salves the CVRP [215]. Following the successes of SA in solving the vehicle routing problems (VRP) and its variants [213, 215–217], and drone scheduling problems [152, 218], this dissertation proposes SA to solve the DDS-F model.

To develop a solution approach using SA, Section 4.4.1 explains the solution representation and the procedure to calculate the minimum ELOD value corresponding to the solution. Section 4.4.2 introduces the proposed SA procedure along with our methods to generate an initial solution and the neighborhood search.

4.4.1 Solution Representation and ELOD Calculation

The solution for the DDS-F 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. A three step procedure, known as the Petal algorithm [211], is used to determine a feasible path with the minimum network ELOD for the ordered list of customers (S) in each iteration: 1) generate feasible contiguous subsets from the given list of customers (Section 4.4.1.1), 2) determine the flight path and the ELOD for each of the subsets (Section 4.4.1.2), and 3) choose a set of flight paths with the minimum total ELOD to cover all customers (Section 4.4.1.3). Figure 4.2 shows these three steps.



Begin
Specification of locations, a given sequence of customers, $k = 1$
Create CS from the given sequence of customers according to Algorithm 1
P_k = subset k in CS
$k = k + 1$
No
$k = |CS|$?
Yes
Solve the BIP (intension of partitioning) problem
End
$|P_k| = 1$?
No
Yes
$|P_k| = 2$?
Yes
No
Calculate γ_{ij} for customers i and j in the subset
Is $\gamma_{ij} > 0$?
No
Yes
k^{th} path = (0, customer in P_k, 0)
k^{th} path = solution of DDSM-F-1 ($N' = P_k + \{0\}$)
k^{th} path = $(0, j, i, 0)$
k^{th} path = $(0, i, j, 0)$
Calculate ELOD for k^{th} path
Determine the flight path and ELOD for each set in the CS

Figure 4.2: Process of optimal solution calculation for a given sequence of customers

4.4.1.1 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 3 is developed to efficiently create the subsets while satisfying the feasibility requirements on the weight capacity and the total operation time of drones (Constraints (4.16) and (4.17)).

Algorithm 3 : 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
```

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., $SS = [1,2,\cdots,n,1,2,\cdots,n-1]$. 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.

4.4.1.2 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 (the DDS-F-1 problem). 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. The following showcases the notation used to provide the mathematical model for the DDS-F-1 problem:

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 $(i, j \in N')$.

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

$$\text{Min} \quad \sum_{c \in N' - \{0\}} (1 - f_c)\, d_c. \tag{4.21}$$

$$\text{s.t:} \quad \sum_{j \in N' - \{c\}} x_{cj} = 1, \qquad \forall c \in N', \tag{4.22}$$

$$\sum_{j \in N' - \{c\}} x_{jc} = 1, \qquad \forall c \in N', \tag{4.23}$$

$$\sum_{i \in N'} \sum_{j \in N'} t_{ij}\, x_{ij} \le ot, \tag{4.24}$$

$$y_i - y_j + n'\, x_{ij} \le n' - 1, \qquad \forall i,j \in N' - \{0\}, i \ne j, \tag{4.25}$$

$$\alpha_{ic}\, f_i - M(1 - x_{ic}) \le f_c, \qquad \forall i \in N', c \in N' - \{0\}, \tag{4.26}$$

$$f_c \le \alpha_{ic}\, f_i + M(1 - x_{ic}), \qquad \forall i \in N', c \in N' - \{0\}, \tag{4.27}$$

and $\quad x_{ij} \in \{0,1\}, y_i \ge 0, f_i \ge 0, f_0 = 1, \qquad \forall i,j \in N'.$

The DDF-F-1 optimization model is modified to solve the path finding problem for each contiguous subset more efficiently. Equations (4.6) and (4.8) will determine the ELOD value for the contiguous subsets after the determination of the flight paths.

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 4.3, 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 3. A path with

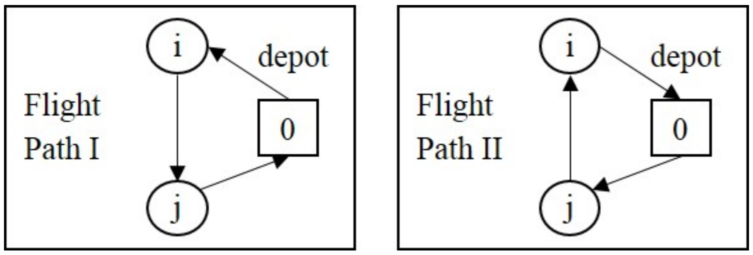


Figure 4.3: Two possible flight paths for two customers and one drone

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 following inequality holds true:

$$\langle ELOD_I < ELOD_{II} \rangle \rightarrow (1 - \alpha_{0il})\, d_i + (1 - \alpha_{0il}\alpha_{ijl})\, d_j < (1 - \alpha_{0jl})\, d_j + (1 - \alpha_{0jl}\alpha_{ijl})\, d_i. \tag{4.28}$$

Two special cases of Figure 4.3 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 $d_i > d_j$ ($\Delta d = d_i - d_j$). Therefore, the followings hold:

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

$$0 < \alpha_{0il}, \alpha_{ijl} < 1 \rightarrow 1 - \alpha_{0il} < 1 - \alpha_{0il}\alpha_{ijl}, \rightarrow ELOD_{II} - ELOD_I > 1. \tag{4.29}$$

As a result, 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 $t_{0i} > t_{0j} \rightarrow \alpha_{0il} < \alpha_{ojl}$. Therefore, the following holds:

$$\text{If } d_i = d_j \rightarrow \frac{ELOD_I}{ELOD_{II}} = \frac{1-\alpha_{0il}+1-\alpha_{0il}\alpha_{ijl}}{1-\alpha_{0jl}+1-\alpha_{0jl}\alpha_{ijl}} = \frac{2-\alpha_{0il}(1-\alpha_{ijl})}{2-\alpha_{0jl}(1-\alpha_{ijl})} \rightarrow \frac{ELOD_I}{ELOD_{II}} > 1. \tag{4.30}$$

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

4.4.1.3 Final Flight Paths

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.

$$\text{Min} \quad \sum_{p \in P} \sigma_p f_p. \tag{4.31}$$

$$\text{s.t:} \quad \sum_{p \in P} \theta_{pj} f_p = 1, \qquad \forall j \in N - \{0\}, \tag{4.32}$$

$$\sum_{p \in P} f_p = m, \tag{4.33}$$

$$\text{and} \quad f_p \in \{0,1\}, \qquad \forall p \in P.$$

The objective function (4.31) minimizes the summation of ELOD for the selected flight paths. Constraint (4.32) states that each customer must be included on only one of the selected flight paths. Constraint (4.33) is to limit the number of selected flight paths to the number of available drones.

4.4.2 The Simulated Annealing Algorithm

The proposed SA in this research shown in Figure 4.4 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 [219]. 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.

4.4.2.1 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 4.5 [210, 211]. 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.

4.4.2.2 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 [220]. This research 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

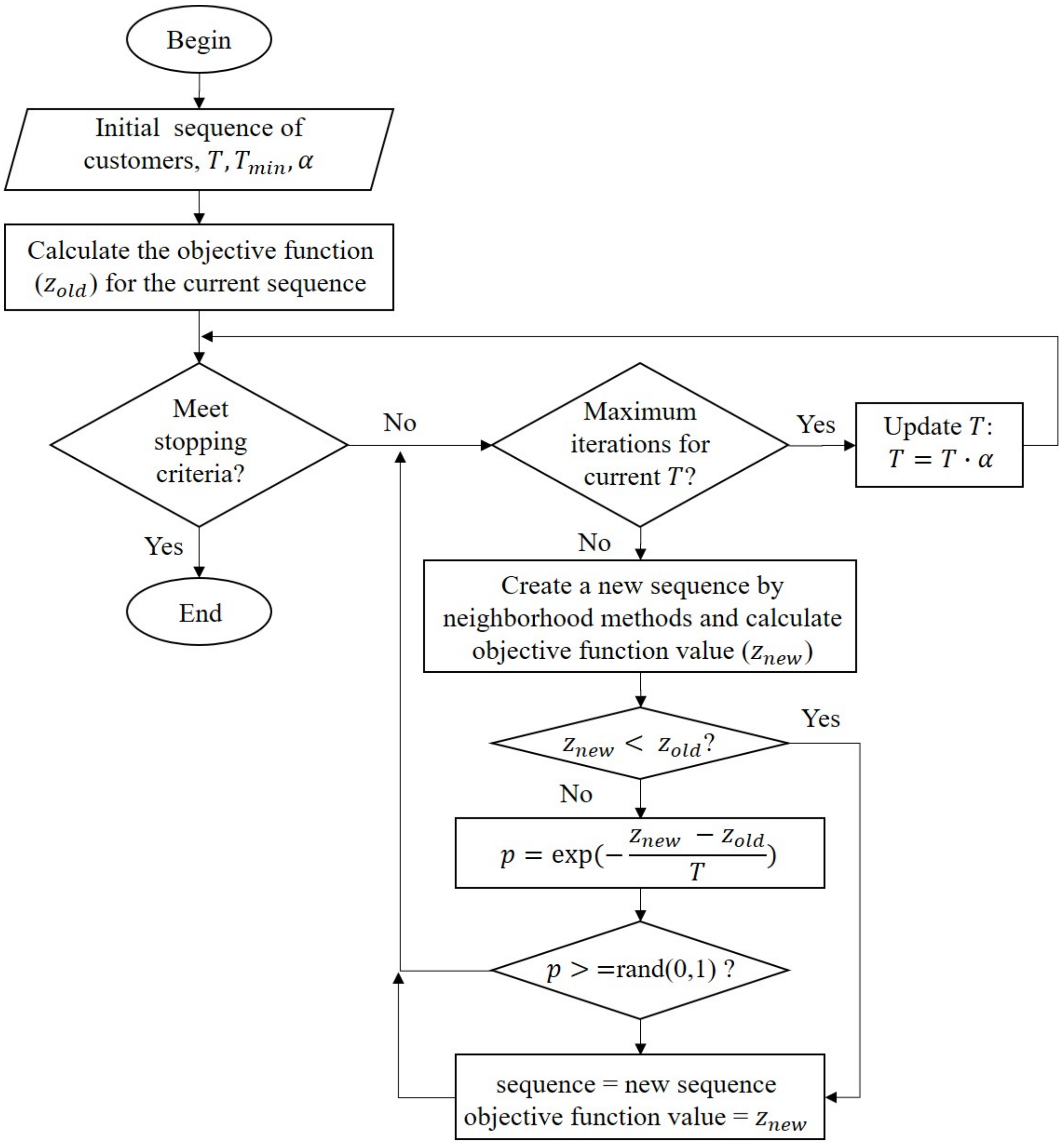

Begin
Initial sequence of customers, T, T_{min}, α
Calculate the objective function (z_{old}) for the current sequence
Meet stopping criteria?
No
Yes
End
Maximum iterations for current T?
Yes
Update T: $T = T \cdot \alpha$
No
Create a new sequence by neighborhood methods and calculate objective function value (z_{new})
$z_{new} < z_{old}$?
Yes
No
$p = \exp(-\frac{z_{new} - z_{old}}{T})$
$p >=$rand(0,1) ?
sequence = new sequence
objective function value = z_{new}

Figure 4.4: SA Algorithm

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 $S_{new} = [1,4,3,2,5,6]$

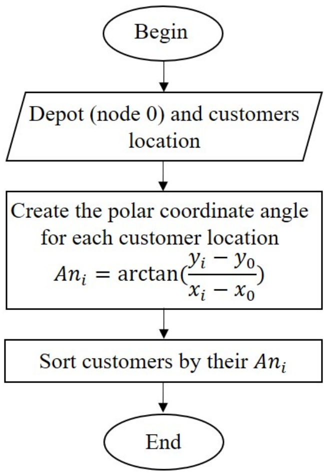

Begin
Depot (node 0) and customers location
Create the polar coordinate angle for each customer location
$An_i = \arctan(\frac{y_i - y_0}{x_i - x_0})$
Sort customers by their An_i
End

Figure 4.5: Initial visit sequence of customers

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.

4.5 Numerical Experiments

Numerical results are conducted to test the proposed solution method using small and large-size problems. The well-known makespan problem is used as a benchmark to test the performance of our approach. In the Makespan problem, the objective function is to minimize the required time needed to complete all flight paths. The objective function of minimizing the maximum drone flight time is $Min\ Max\ \{\sum_{i \in N} \sum_{j \in N} t_{ijl}\ x_{ijl}\}$. Constraints (4.11)-(4.18) 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:

$$\begin{aligned} &\text{Min} && u. \\ &\text{s.t:} && u \geq \sum_{i \in N} \sum_{j \in N} t_{ijl}\, x_{ijl}, \qquad \forall l \in L, \\ &\text{and} && \text{Constraints (4.11)-(4.18).} \end{aligned}$$

The sensitivity analysis on the failure distribution is presented in Section 4.5.3, and the performance of the proposed SA algorithm is discussed in Section 4.5.4. All experiments are done on a Linux server with 24 cores and 384GB RAM, and they are implemented in the Python environment [221]. Gurobi solver 8.1 [222] was used to solve the optimization model. The random drone parameter settings in Table 4.2 are used for all test cases.

Table 4.2: 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 min.
λ_l	0.005 (failures per minute)

4.5.1 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 4.2. Note that the Weibull distribution is one of the most commonly used failure distributions in reliability analysis [223]. 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. The resulting optimal flight paths for the DDS-F problem and the makespan model are shown in Figure 4.6 and Figure 4.7, respectively.

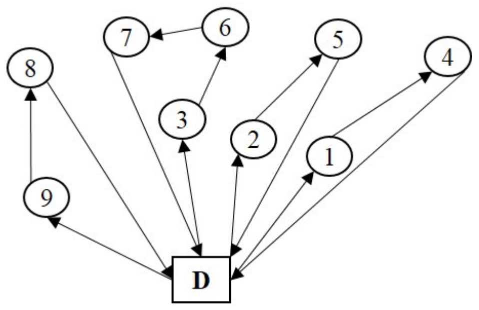


Figure 4.6: Optimal solutions for DDS-F problem

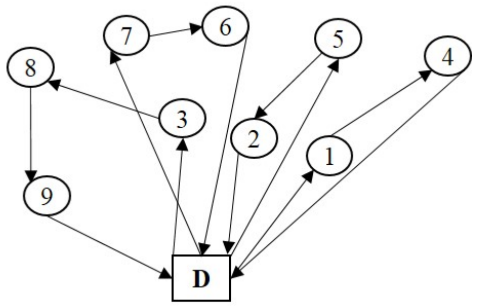


Figure 4.7: Optimal solutions for Makespan

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 [224]. Therefore, a comparison is made to investigate the impact of the DDS-F model against the makespan problem. Figure 4.7 shows that the optimal flight paths derived from the makespan problem differ from the solution outputted by the DDS-F model.

Furthermore, Table 4.3 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

Table 4.3: DDS-F and Makespan results for the test case problem

Problem	Path	Payload amount (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		

discussed in this research. 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.

4.5.2 Comparison between the DDS-F and the Makespan

Different random test cases are used to compare the DDS-F model and the Makespan (MS) model in terms of *ELOD* and *makespan*. The results are summarized in Table 4.4, where n and m are the number of customers and the number of drones, respectively. 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 the following formulas:

$$\text{ELOD decrease } (\%) = \frac{\text{ELOD}_{\text{MS}} - \text{ELOD}_{\text{DDSF}}}{\text{ELOD}_{\text{DDSF}}} \times 100, \tag{4.34}$$

$$\text{and makespan increase } (\%) = \frac{\text{makespan}_{\text{DDSF}} - \text{makespan}_{\text{MS}}}{\text{makespan}_{\text{MS}}} \times 100, \tag{4.35}$$

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 11.5% up to 33.4%, while the increase of makespan was negligible, ranging from

Table 4.4: Comparison between DDS-F and MS Problems

		MS Problem		DDS-F Problem		ELOD	makespan
n	m	ELOD	makespan	ELOD	makespan	decrease	increase
		(10^{-2} lb.)	(min)	(10^{-2} lb.)	(min)	(%)	(%)
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%

0.1% to 1.4%. 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.

4.5.3 Sensitivity Analysis on the Failure Rate

The failure rate in Section 4.5.1 and Section 4.5.2 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 4.6 is investigated here. As mentioned in Section 2.5.3 and depicted in Figure 2.2, the historical reliability data shows an

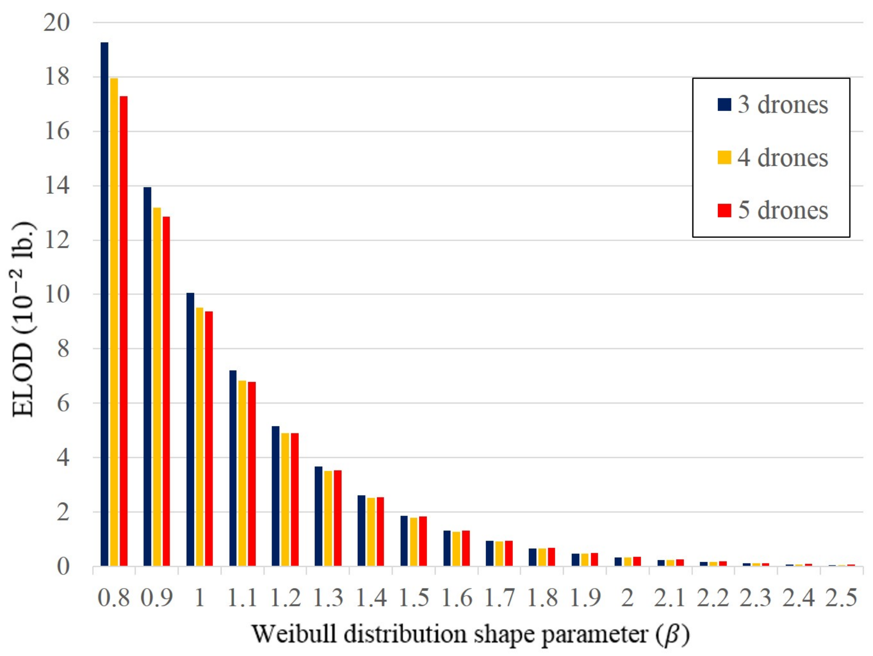


Figure 4.8: Network ELOD for different shape parameter values with the fixed scale parameter

improvement in drone failure rate over time for four different drone types. This implies the parameter β in Weibull distribution to be less than 1. Since the available data for the drone failure is very limited, not only the decreasing failure rate but also constant and increasing rates are investigated in this section. 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 4.8 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 generated based on a fixed $\eta = 200$. However, the maximum travel time between two nodes in Figure 4.6 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 4.9, 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

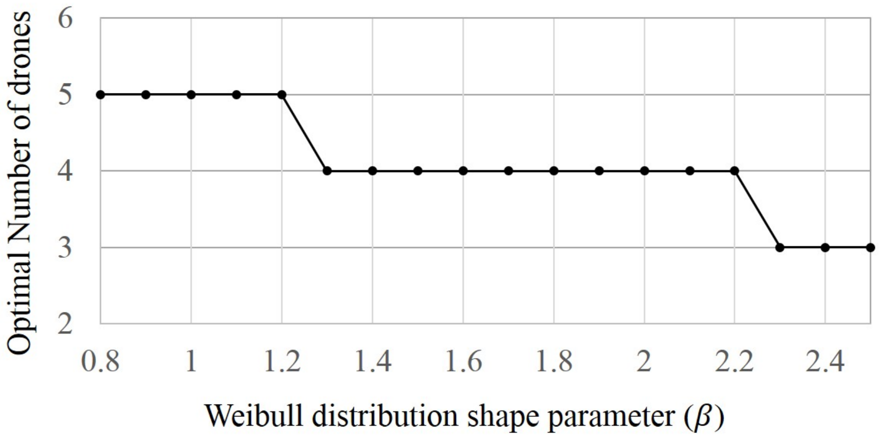


Figure 4.9: Optimal number of drones by changing Weibull distribution shape parameter

drone's 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.

4.5.4 The SA Performance

This section evaluates the computational performance of the proposed SA algorithm (see Section 4.4) compared to the exact method (Gurobi solver implemented in Python [222]). 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.

Prior to solving the test instances using the SA, the SA parameter values were carefully tuned. The initial temperature (T) is set to 5 for all of the test cases. The temperature reduction ratio (α) ranged between 0.93 and 0.98, depending on the size of the problem. The algorithm will stop if there is no improvement in 10 consecutive iterations or if it

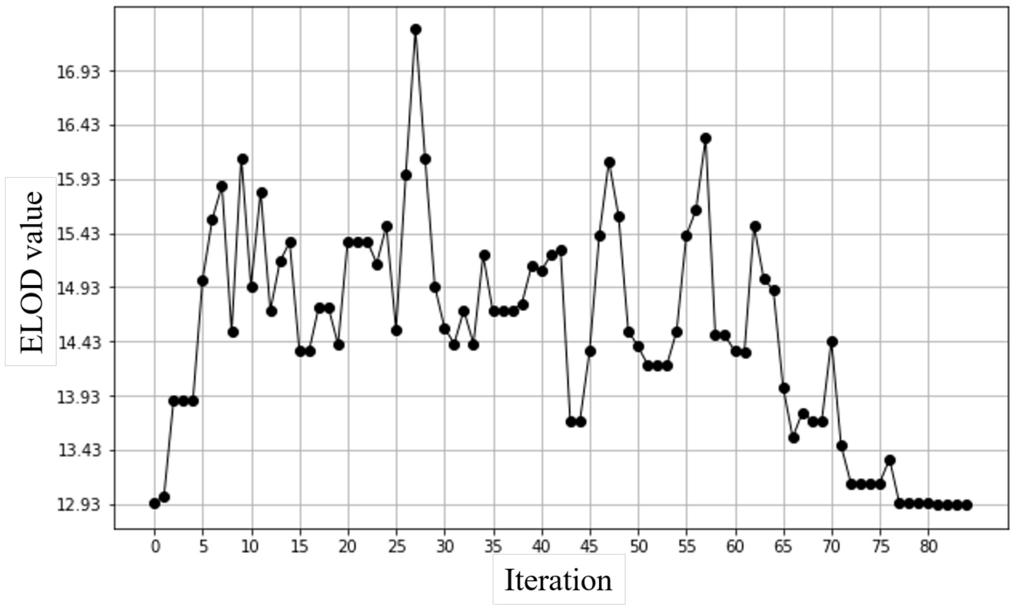


Figure 4.10: The ELOD value over iterations: a case study with 12 customers and 5 drones

reaches the temperature of 0.05. Figure 4.10 shows the progression of the SA algorithm until it converged. As can be seen from Figure 4.10, 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 4.5 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:

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

Table 4.5: Optimal solution obtained for DDS-F model by the exact and SA methods

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	78.9	0.0%	24.9%
10	5	10.58	68.4	10.58	65.7	0.0%	-4.0%
11	4	12.59	296.0	12.59	154.4	0.0%	-47.8%
11	5	12.11	388.1	12.11	116.5	0.0%	-69.9%
12	4	13.46	763.3	13.46	206.2	0.0%	-72.9%
12	5	12.92	4460.7	12.92	123.6	0.0%	-96.2%
20	7	NA	NA	21.60	318.6	-	-
20	8	NA	NA	21.35	297.0	-	-
40	16	NA	NA	39.90	1052.2	-	-
40	17	NA	NA	39.75	948.5	-	-
60	19	NA	NA	60.64	2152.7	-	-
60	20	NA	NA	59.50	1677.5	-	-
100	31	NA	NA	88.52	6128.6	-	-
100	32	NA	NA	87.87	2450.9	-	-

For the cases with 10, 11, and 12 customers, both methods found optimal solutions. As expected, the SA outperformed the exact method in CPU time up to 96% 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. Although the CPU time of the SA increased as the problem size increased, the SA found solutions for all problem instances within 40 minutes. 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.

4.6 Conclusion

This chapter 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 23.6% more reliable on average at a minor increase in makespan by 0.78%. 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 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 optimal solutions to smaller problem instances at a small fraction of CPU time for solving the DDS-F using the exact method. Overall, the SA solved all test problem instances within 40 minutes.

Chapter 5

Using Autonomous Battery Swap Stations (ABSS) to Enable Continuous Drone-aided Surveillance Missions Considering Battery Capacity Degradation

5.1 Introduction

Despite vast improvements in battery-powered drone technologies, limited flight time remains a major drawback for drones. A common approach to increase the travel time of battery-powered vehicles is to charge their battery during the trip [208, 225]. Several methods that involve charging a drone battery during its flight time exists, such as installing solar panels on the drone [170], using wireless charging infrastructures along the flight path [114, 115] or at the stations [171], and charging and swapping batteries at the stations [226, 227]. However, each of these methods have flaws. First, as seen in fixed-wing drones, drones can carry solar cells and use solar energy as a primary source for a continuous, long flight [170]. However, in the absence of the sun, an alternative power source would be necessary for the drones. Secondly, a major challenge in using wireless charging infrastructures during the flight or at the stations is the low charging efficiency and long charge time in comparison to wired charging at the station [171, 173]. Finally, to address the third aforementioned solution, it is possible that charging stations swap the depleted batteries with fully charged ones and recharge them for their next use. Given the nature of these stations, such charging stations with manual operations require a large space and enough labor to swap or charge batteries when drones return to the station.

In this study, we propose using Autonomous Battery Swap Stations (ABSSs) as an alternative for the manual charging stations during drone group surveillance missions. ABSSs are equipped with an automated battery replacement system to swap the depleted batteries with recharged batteries without human interventions [7]. ABSSs can perform all the tasks that manned stations can do. Each station is equipped with a sufficient number of batteries so that drones will spend more time surveying rather than waiting at the station for their battery to be charged.

The swapping time for most ABSSs takes approximately one minute, which is much shorter than the standard battery charging time [171]. Figure 5.1 shows the swapping mechanism in an ABSS. A drone takes off from an ABSS and surveys the area to collect data. The drone lands in the station when the battery charge is less than a preset threshold to ensure the safe return for a battery swap before continuing the surveillance mission. In the ABSS, the depleted battery is replaced with a fully-charged one, and the removed battery is placed in the charger to be charged for future use. Different design options for the components of an ABSS can be found in the literature [7, 176, 177].

The installation location for ABSSs can be on rooftops, street lights, cell towers, inside the sewer system, or along borders [228]. By overcoming the drones' limited flight time, it is possible to use drones for long-duration flights. The short swapping time achieved by ABSSs can even facilitate the use of drones in semi-continuous surveillance missions.

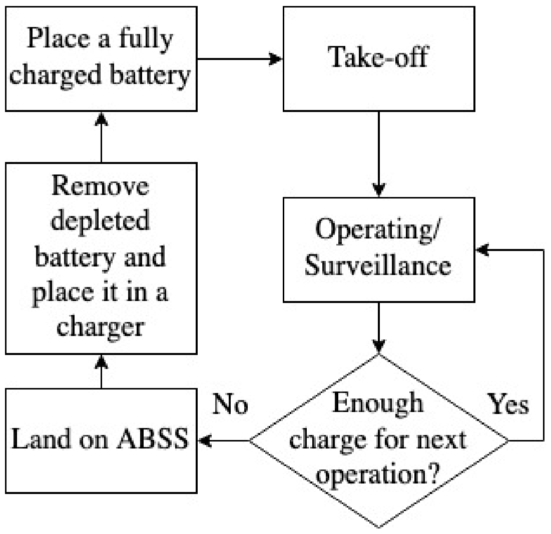

Place a fully charged battery
Take-off
Remove depleted battery and place it in a charger
Operating/ Surveillance
Land on ABSS
No
Enough charge for next operation?
Yes

Figure 5.1: Autonomous battery swapping system algorithm [8]

The application of installing ABSSs to perform semi-continuous drone flights includes: monitoring smart grid infrastructure [229], border line control [114], security operations in the oil and gas industry [109], surveillance in tunnel-like environments (such as sewers, road tunnels and mines) [230, 231], and data collection for the marine ecology, such as aerial shark spotting [232, 233].

In this study, the application of battery-powered drones to survey predetermined paths is considered, and ABSSs are utilized to perform long and semi-continuous flights. One important factor in ABSS application for continuous flights is the approach of managing the drones' battery charges. In continuous surveillance by battery-powered drones, a fully-charged battery should always be prepared for each drone. Lithium-ion batteries are a common power source for the battery-powered drones. The batteries have the disadvantage of degradation, which means their capacity decreases over time [38]. Two major factors that affect the battery state of health (SOH) and its degradation are the average state of charge (SOC) and its swing of the batteries [2], both of which are studied in this research. A consistent battery management system has to be implemented in all of the stations to prevent surveillance mission interruptions. The battery management system determines the battery acquisition, charging, and replacement policies by answering the following questions: 1) How many batteries are needed to perform the non-interrupted mission?; 2) How does the battery capacity change and degrade over time?; 3) What should the battery charging policy be in order to decrease battery degradation?; and 4) When should the battery be replaced with a new one?

Overall, the contributions of this research to the literature include: 1) proposing a strategic level framework to use autonomous battery swap stations for a continuous surveillance performed by drones; 2) proposing a novel mathematical model to determine the optimal location and the number of swap stations while taking into account the criticality level of the surveillance area; 3) providing an algorithm to determine the required number of batteries in each charging station while taking into account the SOH degradation of the batteries over time.

The remainder of this chapter is organized as follows: Section 5.2 explains the proposed framework to use swap stations for continuous surveillance. A mathematical optimization problem is proposed in Section 5.2.1 to determine the optimal number and location of swap stations among candidate locations. Section 5.2.2 studies the battery capacity degradation over time. Section 5.3 provides a heuristic algorithm to determine the required number of batteries in each station. The numerical results are discussed in Section 5.4 and and the conclusion is presented in Section 5.5.

5.2 Problem Description and Formulations

To increase drones flight time, a set of candidate locations are considered in this study to establish battery swap stations. The goal is to continuously survey the area with the minimum number of stations in order to reduce the overall cost. As the location of ABSSs is fixed, the considered surveillance environment can be an area that requires monitoring waypoints along a predetermined path, such as pipelines, borders, and sewer systems.

The required time to swap the battery is negligible [7]. There is one charger per battery in each station to charge the depleted batteries automatically. Upon arrival at the station, if the remaining battery charge percentage is enough for another flight, the drone will perform another survey. Otherwise, the drone battery will be swapped automatically with a fully charged battery. The swapped battery is placed on one of the chargers to be charged for a future flight. The batteries in the chargers can be used for a flight when they are fully charged. Due to the battery capacity degradation over time, batteries should be replaced with new batteries when the capacity is less than a preset threshold. The battery capacity degradation affects the required number of the batteries assigned to each drone, which is considered in Section 5.2.2.

As depicted in Figure 5.2, there are two drones in each station s: One of these drones covers the area on the right side of the station ($drone_{s,r}$) and the other covers the area on the left side ($drone_{s,l}$). Drones, their batteries, and chargers are assumed to be homogeneous.

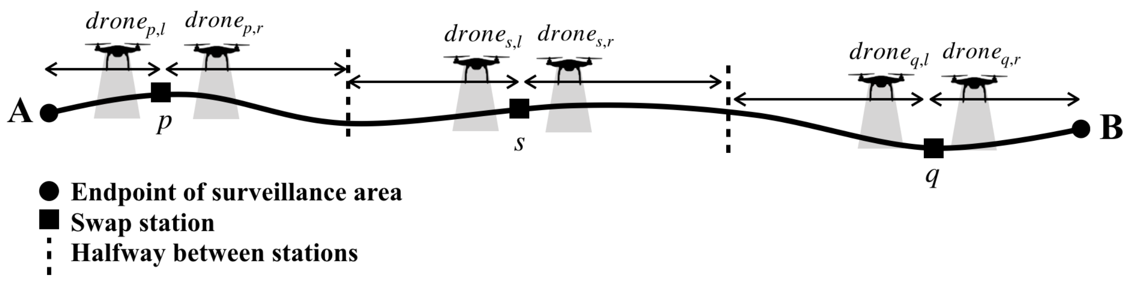


Figure 5.2: An example of the surveillance environment

The monitoring of all locations in the given area is essential; however, the importance level of the locations are different. Some locations in the surveillance area may require more frequent visits depending on the priority with respect to security, safety, and criticality. To accommodate this need, Ahmadian *et al.* [234] defined the revisiting gap as the largest time gap between two consecutive visits of a drone in a particular location. This concept is used to evaluate the frequency of visiting a particular location in the surveillance area.

To parameterize the importance level of surveillance locations, a threshold for the revisiting gap ($\tilde{g_j}$) is defined for location j, which shows the maximum allowable time gap between two consecutive visits of a drone at location j. Meeting the maximum revisiting gap in the drone flight schedule ensures that at no time in the planning horizon, location j is unseen for more than $\tilde{g}$ time unit. The parameter $\tilde{g_j}$ is an input parameter placed into the model, which is defined by the decision-makers based on the time interval they prefer to have data from location j. Locations with higher priority should be visited more frequently, so their $\tilde{g}$ is lower.

Each station is equipped with two drones that fly in opposite directions and return to the same station after the surveillance. The assigned surveillance area to each drone is either the area between the station and halfway to the another station ($drone_{p,r}$, $drone_{s,l}$, $drone_{s,r}$, and $drone_{q,l}$ in Figure 5.2) or the area between one of the endpoints of the area and the station ($drone_{p,l}$ and $drone_{q,r}$ in Figure 5.2). Note that the halfway point between two stations is a location that has equal distance from both of these stations. The following proposition [234] is used to determine the maximum revisiting gap for surveillance waypoints.

Proposition 2. *Assume a and b are the endpoints of the assigned area to a drone where waypoint j is located between them. Let λ_j and γ_j be the travel time between waypoint j and the assigned endpoints a and b, respectively, and g_j^* be the minimum revisiting gap for waypoint j. Then:*

i) the minimum revisiting gap for waypoint j is achieved by flying directly between a and b and changing the direction only at the endpoints of the assigned surveillance area.

ii) $g_j^* = 2 \cdot max\{\lambda_j, \gamma_j\}$.

Proof. See Ahmadian *et al.* [234].

□

Proposition 2 determines the drones' flight path as directly flying between the assigned endpoints and changing direction only at these endpoints. This information is used in the remainder of this section to calculate the revisiting gap for the waypoints of the surveillance area, which are located either between two swap stations or between one swap station and one end point.

According to Figure 5.3, if candidate locations i and i' are two consecutive swap stations, then halfway point between them is location h, which has an equal distance from both i and i' locations. Each drone starts the flight from a swap station, monitors the locations between the station and the halfway point, and then returns to the same station. This flight pattern is proven to result in the lowest revisiting gap for surveillance waypoints [234].

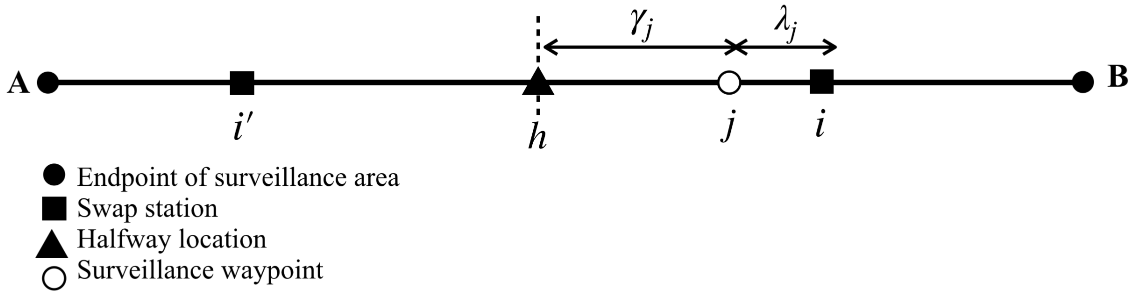


Figure 5.3: Revisiting gap for waypoints located between two swap stations

The revisiting gap for location j in Figure 5.3 (g_j) is calculated by [234]:

$$g_j = 2 \cdot max\{\lambda_j, \gamma_j\}, \tag{5.1}$$

where γ_j is the flight time between location j and halfway point h, and λ_j is the flight time between location j and the nearest station ($\lambda_j = min\{time_{j,i}, time_{j,i'}\}$). In Figure 5.3, $\lambda_j = time_{j,i'}$ if location j is located before the halfway point h, and $\lambda_j = time_{j,i}$ if location j is located after the halfway point h.

Some waypoints may be located between one of the swap stations and one of the surveillance area endpoints. They are located either before the first station or after the last station, such as waypoints j' and j'' in Figure 5.4, respectively.

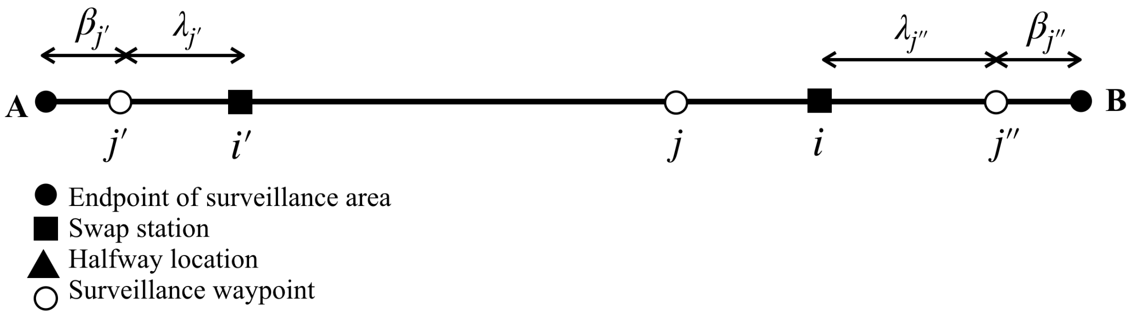


Figure 5.4: Revisiting gap for waypoints located between one station and one endpoint

For waypoints j' and j'', the flight time to the endpoint is taken into account rather than the flight time to the halfway point in order to calculate the revisiting gap as follows:

$$g_{j'} = 2 \cdot max\{\lambda_{j'}, \beta_{j'}\}, \tag{5.2}$$

$$\text{and } g_{j''} = 2 \cdot max\{\lambda_{j''}, \beta_{j''}\}, \tag{5.3}$$

where $\beta_{j'}$ is the flight time between waypoint j' and endpoint A, and $\beta_{j''}$ is the flight time between location j'' and endpoint B.

A two-stage procedure is proposed in this research. The first stage (see Section 5.2.1) determines the optimal location for swap stations while minimizing the number of stations and meeting the maximum allowable revisiting gap for locations in the surveillance environment. The second stage (see Section 5.2.2) determines the required number of batteries for each drone needed to perform continuous surveillance.

5.2.1 Swap Station Location Problem

The aim of this section is to propose a mathematical optimization model to determine the number and location of battery swap stations. There is a given set of candidate locations (set I), a subset of which will be chosen to establish swap stations. The goal is to cover surveillance waypoints (set SW) and meet their maximum revisiting gap while maintaining a minimum number of swap stations. If the surveillance area does not have any specific waypoint, adding dummy waypoints can be considered in the mathematical model. The parameter e_1 and e_2 are the left and right endpoints of the surveillance area, respectively. As depicted in Figure 5.3 and Figure 5.4, each waypoint in the surveillance area is either located between two stations or between one endpoint and one station. Therefore, it can be assumed that each waypoint is assigned to either two stations or one station and one endpoint. By considering set $I' = I \cup \{e_1, e_2\}$ as the set of candidate locations and endpoints, each surveillance waypoint is assigned to exactly two waypoints in set I'. The variable $\tau_{i'j}$ equals 1 if surveillance waypoint j is assigned to waypoint $i' \in I'$.

The surveillance waypoints located between two stations are located either before or after the halfway point of those stations. The binary variable x_j equals 1 if surveillance waypoint j is located after the halfway point of its assigned stations, and it is 0 if j is located after the halfway point. The constraints related to the calculation of the variable x_j would not be binding for the surveillance waypoints not located between two stations. The binary variables $\tau_{i'j}$ and x_j are used to determine the waypoint j revisiting gap based on its location relative to station and endpoint locations. The notations used in this section are as follows:

The objective function is minimizing the number of selected candidate locations to establish swap stations:

$$\text{Min} \sum_{i \in I} u_i. \tag{5.4}$$

Constraints (5.5) and (5.6) limit the gap to the maximum allowable gap for waypoints and

Sets:

e_1, e_2	the left and right endpoints of the surveillance area, respectively,
I	set of candidate locations for swap stations ($I' = I \cup \{e_1, e_2\}, i \in I, i' \in I'$),
SW	set of surveillance waypoints ($k \in SW, SW \cup I = \emptyset$),
W	set of all waypoints ($j \in W, W = I \cup SW$).

Parameters:

ν	average drone speed,
$\tilde{g}_j$	revisiting gap threshold for waypoint j in time unit,
d_j	distance of waypoint j from the endpoint e_1,
M	a big number,
TL	total length of area to cover, distance of the endpoint e_2 from the endpoint e_1.

Variables:

$\tau_{i'j}$	1 if surveillance waypoint j is assigned to waypoint i', 0 otherwise,
g_j	revisiting gap for waypoint j in time unit,
u_i	1 if candidate location i is chosen for swap station, 0 otherwise,
x_j	1 if waypoint j is located after halfway point of its assigned stations, 0 otherwise.

also candidate locations that are not selected as swap stations according to the followings:

$$g_k \leqslant \tilde{g}_k, \qquad \forall k \in SW, \tag{5.5}$$

and

$$g_i \leqslant \tilde{g}_i + Mu_i, \qquad \forall i \in I. \tag{5.6}$$

Each waypoint $j \in W$ is either located between two consecutive stations or between a station and one of the endpoints as shown in Figure 5.3 and Figure 5.4. Therefore, each waypoint $j \in W$ is located exactly between two locations in the set I' as stated by:

$$\sum_{i' \in I'} \tau_{i'j} = 2, \qquad \forall j \in W. \tag{5.7}$$

Following constraint ensures that parameter $\tau_{ij}, i \in I$, can get the value of 1 only if the candidate location i is selected as a swap station:

$$\tau_{ij} \leqslant u_i, \qquad \forall i \in I, j \in W. \tag{5.8}$$

The remaining constraints are to calculate the revisiting gap for waypoints and unopened candidate stations according to Equations (5.1)-(5.3). For the modeling purposes, the maximization function in (5.1)-(5.3) is converted to a greater than or equality constraint. Constraints (5.9) and (5.10) calculate the gap for each waypoint located between the first endpoint e_1 and the station i as:

$$g_j \geqslant \frac{2}{\nu}(d_i - d_j) - M(2 - \tau_{e_1 j} - \tau_{ij}), \qquad \forall i \in I, j \in W, \tag{5.9}$$

and
$$g_j \geqslant \frac{2}{\nu}(d_j)\tau_{e_1 j}, \qquad \forall j \in W. \tag{5.10}$$

These constraints are binding only when waypoint j is located between both station i and endpoint e_1 (such as waypoint j' in Figure 5.4), otherwise the right hand side would be a very large negative value. A constant drone speed is considered to convert the distance to the flight time.

Constraints (5.11) and (5.12) are similar to Constraints (5.9) and (5.10), except that they calculate revisiting gap for waypoints between the last swap station i and the second endpoint e_2 as:

$$g_j \geqslant \frac{2}{\nu}(d_j - d_i) - M(2 - \tau_{e_2 j} - \tau_{ij}), \qquad \forall i \in I, j \in W, \tag{5.11}$$

and
$$g_j \geqslant \frac{2}{\nu}(TL - d_j)\tau_{e_2 j}, \qquad \forall j \in W. \tag{5.12}$$

These constraints are binding only when waypoint j is located between both station i and endpoint e_2 (such as waypoint j'' in Figure 5.4).

Constraints (5.13)-(5.18) are to calculate the revisiting gap for the rest of the waypoints, which are located between two stations (such as waypoint j in Figure 5.3). First, it should be determined whether waypoint j is located before or after the halfway point. the following constraints determine which one of the swap stations is closer to waypoint j:

$$\frac{1}{2}\sum_{i\in I} d_i \tau_{ij} \geqslant d_j - Mx_j, \qquad \forall i \in I, j \in W, \tag{5.13}$$

and
$$\frac{1}{2}\sum_{i\in I} d_i \tau_{ij} \leqslant d_j + M(1-x_j), \qquad \forall i \in I, j \in W. \tag{5.14}$$

The halfway point for waypoint j is located at location $\frac{1}{2}\sum_{i\in I} d_i \tau_{ij}$. Note that according to constraint (5.7), there are two variables τ_{ij} with the value of 1 for each $j \in W$. Variable x_j is a binary variable so only one of the Constraints (5.13) and (5.14) are tight for each waypoint $j \in W$. For waypoints located after their halfway point, variable x_j gets the value of 1 and Constraint (5.14) is binding. Constraint (5.13) is binding for waypoints located before their halfway point.

Constraints (5.15) and (5.16) calculate the revisiting gap for the waypoints that are located between two stations and are also located after the halfway points of these stations:

$$g_j \geqslant \frac{2}{\nu}(d_i - d_j) - M(1-x_j) - M(1-\tau_{ij}), \qquad \forall i \in I, j \in W, \tag{5.15}$$

and
$$g_j \geqslant \frac{2}{\nu}(d_j - \frac{1}{2}\sum_{i\in I} d_i \tau_{ij}) - M(1-x_j) - M(2-\sum_{i\in I}\tau_{ij}), \qquad \forall j \in W. \tag{5.16}$$

If the waypoint j is not located between two stations or is located before the halfway, Constraints (5.15) and (5.16) are not binding.

Constraints (5.17) and (5.18) are similar to Constraints (5.15) and (5.16), except that they consider waypoints located before their halfway point:

$$g_j \geqslant \frac{2}{\nu}(d_j - d_i) - Mx_j - M(1-\tau_{ij}), \qquad \forall i \in I, j \in W, \tag{5.17}$$

$$g_j \geqslant \frac{2}{\nu}(\frac{1}{2}\sum_{i\in I} d_i\tau_{ij} - d_j) - Mx_j - M(2-\sum_{i\in I}\tau_{ij}), \quad \forall j \in W, \tag{5.18}$$

and $$g_j \geqslant 0, \tau_{i'j}, u_i, x_j = \{0,1\}, \quad \forall i' \in I', i \in I, j \in W. \tag{5.19}$$

5.2.2 Number of Required Batteries

This section determines the number of required batteries in each station by considering the battery capacity degradation over time. The battery capacity is the maximum possible flight time with one charge of the battery. The battery capacity decreases over time and eventually they have to be replaced with a new one.

The battery state of charge (SOC) shows the remaining battery percentage. The average and standard deviation of battery SOC (μ_{SOC}, σ_{SOC}), and the operating temperature (T_B) are the most important factors impacting the battery capacity degradation [235]. The model proposed in [235], calculates capacity fading ($L(m)$) each time the battery is used through Equation (5.22). The intermediate parameters L_1 and L_2 capture the impact of μ_{SOC} and σ_{SOC} on the capacity degradation according to the following equations:

$$L1 = K_{co} \cdot N \cdot exp((\sigma_{SOC} - 1)\frac{T_{ref}}{K_{ex}(T_B)}) + 0.2\frac{t_{cycle}}{t_{life}}, \tag{5.20}$$

$$L2 = L1 \cdot exp(4K_{SOC}(\mu_{SOC} - 0.5))(1-L), \tag{5.21}$$

and $$L_m = L2 \cdot exp(K_t \cdot (T_B - T_{ref})\frac{T_{ref}}{T_B}), \tag{5.22}$$

where $K_{co}, K_{ex}, K_{SOC}, K_t$ are battery-specific parameters, $T_{ref}, t_{cycle}, t_{life}$ are reference battery temperature at $25°C$, duration of one cycle, and shelf life at $25°C$ at 50% SOC until 80% of the initial capacity, respectively [2, 235]. The capacity degradation over T flights or cycles is given by parameter L according to:

$$L = \sum_{m=1}^{T} L_m. \tag{5.23}$$

Variable L shows the remaining battery capacity as a fraction of the initial battery capacity. For example, the remaining battery capacity after T cycle is $cap = cap_{max}(1-L)$, where cap_{max} is the maximum battery capacity. In this research, the impact of μ_{SOC} and σ_{SOC} on the battery capacity degradation and determination of the required number of batteries are considered. The parameter T_B is assumed to be constant so its impact on the battery capacity degradation is not studied here. In Section 5.3, we explain how SOC and its average and standard deviation in each cycle are calculated to determine the capacity degradation and find the required number of batteries for each drone.

As mentioned in Section 5.2, there are two homogeneous drones in each station s, $drone_{s,k}, k \in \{r,l\}$. Drone $drone_{s,r}$ covers the waypoints on the right side of the station and drone $drone_{s,l}$ covers the waypoints on the left side of the station. The required number of batteries for each drone to maintain a continuous flight depends on the length of the area that they survey. The goal is to minimize the total battery acquisition and replacement cost. The initial cost of purchasing batteries is $c \cdot n_{s,k}$, where c is the purchasing cost of one battery (\$) and $n_{s,k}$ is the number of assigned batteries to drone k at station s. $\Delta t_{s,k}$ is the total time of using one battery assigned to drone k at station s before replacing it, therefore, the fraction of $\frac{TH}{\Delta t_{s,k}}$ is the average number of times that batteries are replaced over the planning horizon TH. The objective function for drone k in station s is:

$$Min \quad c \cdot n_{s,k} \cdot (\frac{TH}{\Delta t_{s,k}} + 1). \tag{5.24}$$

where, the variable $\Delta t_{s,k}$ is the cumulative cycle time from the first cycle of using the battery up to the replacement cycle. The procedure to calculate $\Delta t_{s,k}$ is explained in Section 5.3.

5.3 Solution Method

One cycle of using a battery includes both discharging and charging of the battery as depicted in Figure 5.5. Each cycle starts with the battery discharging while it is used in a flight by a drone. Batteries are homogeneous and they are used one after each other, so it is assumed that they have the same discharging-charging curve in the same cycle. The parameters used in this section are as follows:

Sets:	
SS	set of swap stations, $SS = \{i \in I \mid u_i^* = 1\}, s \in SS$,
K	set of drones at each station, $K = \{r, l\}, k \in K$.
Parameters:	
cap_{max}	maximum battery capacity (min),
SOC_{max}	maximum battery percentage (%),
ϕ^c	battery charging rate (%/min),
ϕ^d	battery discharging rate (%/min),
TFC	required time to fully charge a battery to SOC_{max} in a cycle,
ω	battery total idle time at station in a cycle (min),
ω_1, ω_2	battery idle time before and after charging the battery, respectively,
ρ	percentage of total idle time that happens before charging a battery ($\omega_1 = \rho \cdot \omega$),
$\alpha_{s,k}$	required surveillance flight time for drone k in station s (min),
t_{cycle}	duration of one cycle (min).

The parameter $\alpha_{s,k}$ is drone k flight time from station s to the halfway or the endpoint and then return to the same station. It is obtained based on the optimal solution of the swap station location problem in Section 5.2.1. Based on the battery capacity and remaining battery charge, a drone can perform multiple trips in one flight. Parameter f shows the number of drone trips in a cycle and is calculated by:

$$f = \lfloor \frac{cap}{\alpha} \rfloor, \tag{5.25}$$

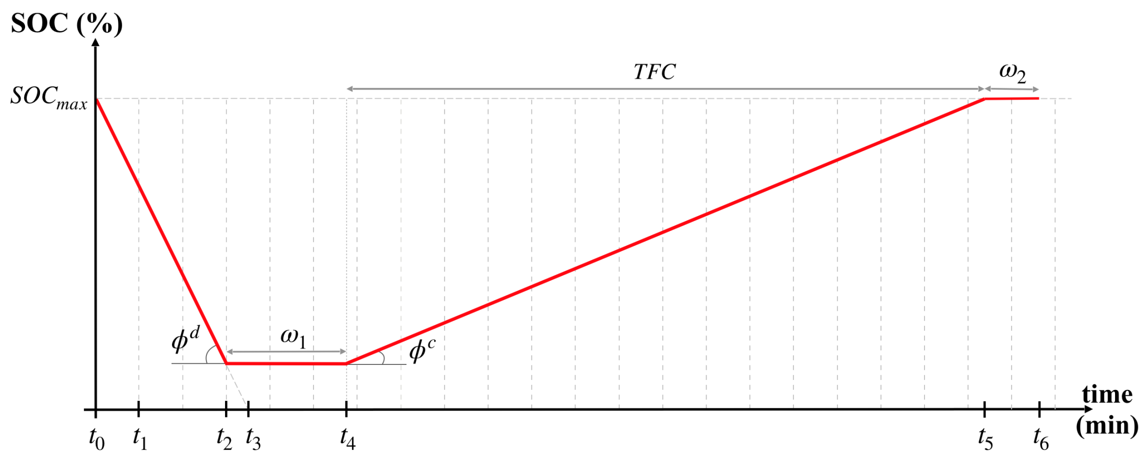


Figure 5.5: Battery state of charge (SOC) during one cycle

where *cap* is the remaining battery capacity. For example, in Figure 5.5 the time distance between dashed lines is $\alpha_{s,k}$, therefore, drone is able to perform 3 trips in one flight. In Figure 5.5, t_0 shows the start time of a cycle and without loss of generality, we assume $t_0 = 0$, then we have the followings:

$$t_1 = \alpha_{s,k}, \tag{5.26}$$

$$t_2 = f \cdot \alpha_{s,k}, \tag{5.27}$$

$$t_3 = cap, \tag{5.28}$$

$$t_4 = f \cdot \alpha_{s,k} + \omega_1, \tag{5.29}$$

$$t_5 = f \cdot \alpha_{s,k} + \omega_1 + TFC, \tag{5.30}$$

and

$$t_6 = f \cdot \alpha_{s,k} + \omega_1 + TFC + \omega_2. \tag{5.31}$$

The $n_{s,k}$ batteries that are assigned to drone k are used one after each other, so the length of a cycle which includes both discharging and charging on a battery is as follow:

$$t_{cycle} = n_{s,k} \cdot f \cdot \alpha_{s,k}. \tag{5.32}$$

The drone might be idle when it returns to the station after a flight because other batteries should be used in the drone. The idle time (ω) can be before ω_1 or after ω_2 charging the battery. The total waiting time in each cycle is calculated by:

$$\omega = t_{cycle} - \mathit{flight\ time} - \mathit{charging\ time},$$

$$\rightarrow \omega = t_{cycle} - f \cdot \alpha_{s,k} - TFC = (n-1) \cdot f \cdot \alpha_{s,k} - TFC, \tag{5.33}$$

$$\text{and } \omega_1 = \rho \cdot \omega, \quad \omega_2 = (1-\rho) \cdot \omega. \tag{5.34}$$

The discharging rate (ϕ^d) in each cycle depends on the battery capacity while the charging rate (ϕ^c) is a constant value in all the cycles. The discharging rate in a cycle depends on the remaining battery capacity and is calculated by:

$$\phi^d = \frac{SOC_{max}}{cap}. \tag{5.35}$$

As parameter ϕ^c is constant, the required time to charge the battery (TFC) depends on the battery *SOC* when the battery is back at the station and is calculated as follows:

$$\phi^c = \frac{SOC_{max} - SOC_{t_4}}{TFC}, \tag{5.36}$$

$$\text{and } SOC_{t_4} = SOC_{max} - \phi^d \cdot t_2 = SOC_{max} - \frac{SOC_{max}}{cap} \cdot f \cdot \alpha_{s,k}, \tag{5.37}$$

$$\rightarrow TFC = \frac{\alpha_{s,k} \cdot f \cdot SOC_{max}}{\phi^c \cdot cap}. \tag{5.38}$$

The battery SOC for the cycle in Figure 5.5 is according to:

$$SOC(t) = \begin{cases} SOC_{max} - \frac{SOC_{max} \cdot t}{cap}, & t_0 \leqslant t < t_2, \\ SOC_{max} - \frac{SOC_{max} \cdot f \cdot \alpha_{s,k}}{cap}, & t_2 \leqslant t < t_4, \\ SOC_{max} - \frac{SOC_{max} \cdot f \cdot \alpha_{s,k}}{cap} + \phi^c \cdot (t - \omega_1 - f \cdot \alpha_{s,k}), & t_4 \leqslant t < t_5, \\ SOC_{max}, & t_5 \leqslant t \leqslant t_6. \end{cases} \tag{5.39}$$

The *SOC* average and standard deviation, μ_{SOC} and σ_{SOC}, are calculated by the following equations, respectively:

$$\mu_{SOC} = \frac{1}{t_{cycle}} \int_{cycle\ time} SOC(t)\ dt, \tag{5.40}$$

and

$$\sigma_{SOC} = \frac{3}{t_{cycle}} \int_{cycle\ time} (SOC(t) - \mu_{SOC})^2. \tag{5.41}$$

Therefore, for Figure 5.5, μ_{SOC} and σ_{SOC} are:

$$\begin{aligned} \mu_{SOC} \cdot t_{cycle} &= SOC_{max} \cdot \omega + \frac{SOC_{max} \cdot f \cdot (\omega_1 + TFC)}{cap} + SOC_{max}(f + TFC), \\ &- \frac{SOC_{max}}{2cap} \cdot f^2 \cdot \alpha_{s,k}^2 - \phi^c(\omega_1 + f \cdot \alpha_{s,k}) + \frac{\phi^c}{2}(f \cdot \alpha_{s,k} + \omega_1 + TFC)^2 - \frac{\phi^c}{2}(f \cdot \alpha_{s,k} + \omega_1)^2. \end{aligned} \tag{5.42}$$

$$\begin{aligned} \sigma_{SOC}^2 \cdot \frac{t_{cycle}}{3} &= (SOC_{max} - \mu_{SOC})^2 \cdot f \cdot \alpha_{s,l} + \frac{SOC_{max}}{3cap} \cdot (f \cdot \alpha_{s,k})^3, \\ &- \frac{SOC_{max}}{cap} \cdot (SOC_{max} - \mu_{SOC}) + (SOC_{max} - \frac{SOC_{max} \cdot f \cdot \alpha_{s,k}}{cap}) \cdot \omega_1, \\ &+ (A - \mu_{SOC})^2 \cdot TFC + \frac{(\phi^c)^2}{3} \cdot ((f \cdot \alpha_{s,k} + \omega_1 + TFC)^3 - (f \cdot \alpha_{s,k} + \omega_1)^3), \\ &+ \phi^c(A - \mu_{SOC}) \cdot ((f \cdot \alpha_{s,k} + \omega_1 + TFC)^2 - (f \cdot \alpha_{s,k} + \omega_1)^2) + (SOC_{max} - \mu_{SOC}) \cdot \omega_2, \end{aligned} \tag{5.43}$$

where $A = SOC_{max} - \frac{SOC_{max} \cdot f \cdot \alpha_{s,k}}{cap} - \phi^c(f \cdot \alpha_{s,k} + \omega_1)$. Parameters μ_{SOC} and σ_{SOC} are then used to calculate the battery degradation according to Equation (5.22).

In each swap station s, $drone_{s,l}$ and $drone_{s,r}$ require $n_{s,l}$ and $n_{s,r}$ batteries to operate a semi-continuous flight, respectively. Therefore, a total of $n_s = n_{s,l} + n_{s,r}$ batteries should be provided at station s. Algorithm 4 shows the procedure to calculate the required number of batteries.

Algorithm 4 : Determine the number of required batteries

```
1:  For each $s \in SS$:
2:     $\alpha_{s,l} = \min\{2d_s \ , \min\{d_s - d_{s'} | d_s > d_{s'}, s' \in SS\}\} \ /\nu$
3:     $\alpha_{s,r} = \min\{2(TL - d_s) \ , \min\{d_{s'} - d_s | d_s < d_{s'}, s' \in SS\}\} \ /\nu$
4:     For each $k \in \{r, l\}$
5:        $obj^*_{s,k} = BigM, n^*_{s,k} = n^{min}_{s,k}$
6:        While(true)
7:           $cap = cap_{max}, L = 0, \Delta t_{s,k,n} = 0$
8:           While $cap > cap_{final}$
9:              $f = \lfloor \frac{cap}{\alpha_{s,k}} \rfloor$
10:             $TFC = \frac{\alpha_{s,k} \cdot f \cdot SOC_{max}}{\phi^c \cdot cap}$
11:             $\omega = (n-1)\alpha_{s,k} \cdot f - TFC$
12:             $\omega_1 = \rho \cdot \omega, \ \omega_2 = (1-\rho) \cdot \omega$
13:             $t_{cycle} = \alpha_{s,k} \cdot f \cdot n$
14:             $\Delta t_{s,k,n} = \Delta t_{s,k,n} + t_{cycle}$
15:             calculate $L_m$ according to Formula (5.22)
16:             $L = L + L_m$
17:             $cap = (1 - L)cap_{max}$
18:          End While
19:          $obj_{s,k,n} = n \cdot (\frac{TH}{\Delta t_{s,k,n}} + 1)$
20:          If $obj_{s,k,n} \leqslant obj^*_{s,k}$
21:             $obj^*_{s,k} = obj_{s,k,n}, n^*_{s,k} = n, n = n + 1$
22:          Else
23:             Break
24:       End While
25:    End For
26: End For
```

Lines 2 and 3 in Algorithm 4 calculate the surveillance flight time ($\alpha_{s,k}$) of drone k

for one round trip departing station s and then returning to station s again. Index set k shows the drones at each station, drone r and l. The outer *While* loop in lines 6-24 of Algorithm 4 calculates the required number of batteries assigned to each drone. It starts from the minimum required number of batteries, $n_{s,k}^{min}$, and increases it one by one, until the objective function value worsens. The objective function value is calculated according to (5.24). The minimum number of batteries that should be assigned to drone k at station s is calculated based on the last cycle of using batteries as:

$$\text{Last cycle:} \quad cap = cap_{final}, \quad f = \lfloor \frac{cap_{final}}{\alpha_{s,k}} \rfloor, \quad TFC = (n_{s,k}^{min} - 1) \cdot \alpha_{s,k},$$
$$\text{and } TFC = \frac{\alpha_{s,k} \cdot f \cdot SOC_{max}}{\phi^c \cdot cap_{final}} = (n_{s,k}^{min} - 1) \cdot \alpha_{s,k} \rightarrow n_{s,k}^{min} = \lceil \frac{SOC_{max} \cdot \lfloor \frac{cap_{final}}{\alpha_{s,k}} \rfloor}{\phi^c \cdot cap_{final}} + 1 \rceil. \tag{5.44}$$

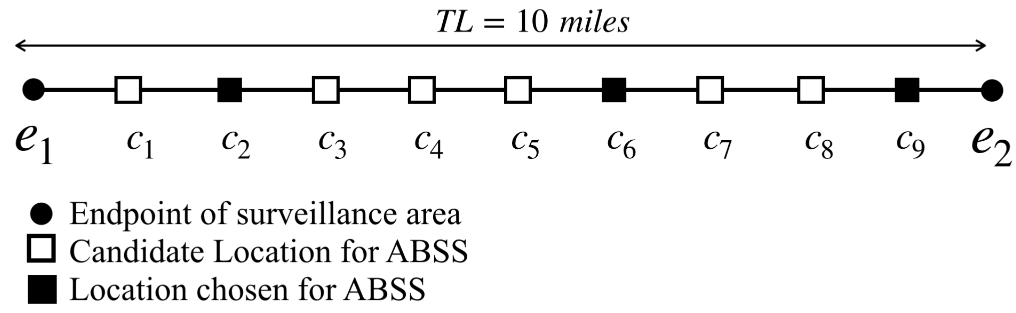
where, parameter $\alpha_{s,k}$ is the flight time of drone k at station s, which is determined by step 2 and 3 of Algorithm 4 for left-side and right-side drones, respectively. The inner *While* loop in lines 8-18 calculates parameters f, TFC, $\omega, \omega_1, \omega_2$, and t_{cycle} according to Equations (5.25), (5.38), (5.33), (5.34), and (5.32), respectively. The parameter $\Delta t_{s,k,n}$ is the total time of using a battery assigned to drone k at station s before replacing it while a total of n batteries are assigned to this drone. These parameters are used to calculate the percentage of capacity fading in one cycle of using the battery, L_m, according to Equation (5.22). The remaining battery capacity is $(1-L)cap_{max}$, where L is the cumulative percentage of capacity fading. A battery is used in drone flight missions until its capacity reached the final battery capacity, cap_{final}, when the battery should be replaced with a new one. The parameter cap_{final} is an input to the model based on the decision makers opinion and is usually set to 80% of the original battery capacity [235]. The output of Algorithm 4 is the required number of batteries assigned to drone k in station s, $n_{s,k}^*$ and its associated objective function value, $obj_{s,k}^*$, and the total time of using each battery, $\Delta t_{s,k,n}$.

5.4 Case Study and Numerical Results

Numerical results are conducted to test the proposed models and the solution method using randomly generated test cases. In all the experiments, equally distributed dummy (or virtual) waypoints are considered to survey the whole area. By increasing the number of dummy waypoints, the accuracy of the modeling environment increases, but the complexity of the model also increases accordingly. All experiments are implemented in the Python environment [221] on a Linux server with 24 cores and 384GB RAM. Gurobi solver 8.1 [222] was used to solve optimization problem in Section 5.2.1 and the solution algorithm proposed in Section 5.3.

5.4.1 A Case Study

The case study considers a surveillance area with the length of 10 miles ($TL = 10\,mile$), which is divided into 100 equally distributed dummy waypoints ($|SW| = 100$). There are 9 candidate locations ($|I| = 9$) for the ABSSs as shown in Figure 5.6. Figure 5.7 shows the randomly generated maximum allowable revisiting gap ($\tilde{g_j}$) for every waypoint of this area. The locations to be visited more frequently are assigned with a shorter permitted revisiting gap.



$TL = 10$ *miles*
e_1
c_1
c_2
c_3
c_4
c_5
c_6
c_7
c_8
c_9
e_2
● Endpoint of surveillance area
□ Candidate Location for ABSS
■ Location chosen for ABSS

Figure 5.6: Case study with 9 candidate locations for ABSS

The rest of the parameters are shown in Table 5.1. The battery charging rate is set to

0.55% per minute, which means an empty battery can be fully charged in 3 hours. The drone speed is assumed to be constant at 30 miles per hour, which is 0.5 miles per minute. The battery-specific parameter values in Equations (5.20)-(5.22) are based on a lithium ion cell battery (i.e., A123 ANR26650M1A) [2].

Table 5.1: Parameter setting for the test cases [2]

	Parameter	Value	Parameter	Value
Battery-specific parameters	N	1	T_{ref}	$25°C$
	K_{ex}	0.717	t_{life}	1 year
	K_{soc}	0.916	K_{co}	$3.66 \cdot 10^{-5}$
	L	0	K_t	0.0693
Common parameters	T_B	$25°C$	c	100 \$
	TH	2 years	cap_{max}	45 minutes
	ϕ^c	0.55 %/min	SOC_{max}	100%
	ρ	1	ν	30 mph

5.4.2 Numerical Results

This section begins by solving the mathematical model proposed in Section 5.2.1, which resulted in three candidate locations $C2$, $C6$ and $C9$ for the ABSS. These stations are selected considering the maximum revising gap requirements of the waypoints. Figure 5.7. shows both the $\tilde{g_j}$ levels and the actual revisiting gap achieved by selecting those three ABSSs locations, $C2$, $C6$ and $C9$.

As mentioned in Section 5.2, two identical drones are placed in each of the ABSS locations. One of them flies in the left direction and the other one in the right direction. Having enough replacement batteries available at each station for each drone can help seamless surveillance of the region with a minimum interruption. The following procedure (see Section 5.2.2) is followed to determine a sufficient number of batteries for seamless operation. First, the flight time ($\alpha_{s,k}$) of drone k for one round trip departing station s is calculated in

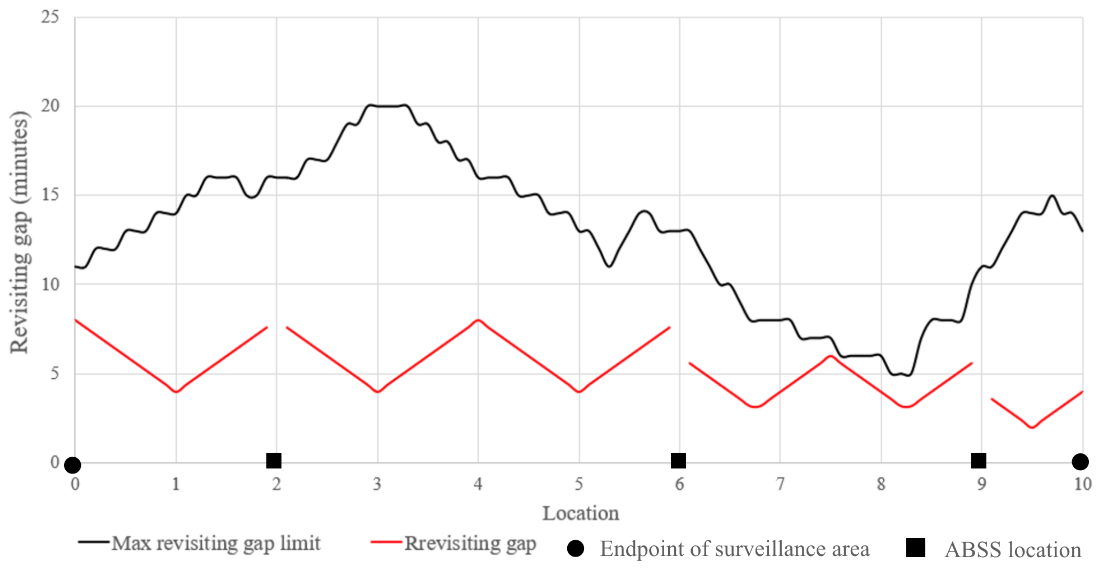


Figure 5.7: Maximum allowable revisiting gap for each waypoint of case study

lines 2 and 3 in Algorithm 4:

$$\alpha_{2,l} = (2 \cdot d_2)\ /0.5 = 8, \tag{5.45}$$

$$\alpha_{2,r} = \min\{2 \cdot (TL - d_2)\ , d_6 - d_2, d_9 - d_2\}\ /0.5 = \min\{16,4,7\}\ /0.5 = 8, \tag{5.46}$$

$$\alpha_{6,l} = \min\{2 \cdot d_6\ , d_6 - d_2\}\ /0.5 = \min\{12,4\}\ /0.5 = 8, \tag{5.47}$$

$$\alpha_{6,r} = \min\{2 \cdot (TL - d_6)\ , d_9 - d_6\}\ /0.5 = \min\{8,3\}\ /0.5 = 6, \tag{5.48}$$

$$\alpha_{9,l} = \min\{2 \cdot d_9\ , d_9 - d_2, d_9 - d_6\}\ /0.5 = \min\{18,7,3\}\ /0.5 = 6, \tag{5.49}$$

$$\text{and } \alpha_{9,r} = 2 \cdot (TL - d_9)\ /0.5 = 4. \tag{5.50}$$

Second, the cap_{final} in Algorithm 4 is set to 80% of the initial battery capacity which is $cap_{final} = 0.8 \cdot 45 = 36$. Third, Equation (5.44) is used to determine the minimum number of required batteries per drone (n_{min}) as: $n_{2,l}^{min} = 21$, $n_{2,r}^{min} = 21$, $n_{6,l}^{min} = 21$, $n_{6,r}^{min} = 31$, $n_{9,l}^{min} = 31$, and $n_{9,r}^{min} = 46$.

Fourth, Algorithm 4 is used to determine the required number of the batteries. Table 5.2 shows the results including the required number of batteries ($n_{s,k}^*$) obtained by Algorithm 4.

In these test cases, we observed that the minimum required number of batteries for each drone resulted in the best value. Overall, a total of 171 batteries are needed for this continuous surveillance mission and the total battery replacement cost would be \$27,928.8 over the planning horizon of two years. The value of $\Delta t_{s,k}$ in the table shows the number of days

Table 5.2: Required number of batteries secured for each drone

station (s)	drone (k)	$\alpha_{s,k}$ (min)	n_{min}	$n^*_{s,k}$	$\Delta t_{s,k}$ (days)	obj (\$)
2	left	8	21	21	910	3784
2	right	8	21	21	910	3784
6	left	8	21	21	910	3784
6	right	6	31	31	1268	4884
9	left	6	31	31	1268	4884
9	right	4	46	46	1522	6805
Total				171	-	27,928 \$

a battery can be used before it reaches the decision point cap_{final} to replace it with a new battery.

5.4.3 Sensitivity Analysis

In this section, the behavior of the proposed model and the algorithm are analyzed under different parameter value settings regarding the battery management system. Section 5.4.3.1 explains when to change a battery with a brand new one. Section 5.4.3.2 shows how the total battery replacement cost varies as relevant parameter values are changed. Finally, Section 5.4.3.3 discusses the time of battery charging and its impact on the battery capacity degradation over time.

5.4.3.1 Battery Replacement Policy

Electrical vehicle batteries are usually replaced with a new battery when the battery reaches 80% of its initial capacity [235, 236]. We investigate the impact of using drone batteries beyond their 80% of the initial capacity on long term battery capacity degradation.

The minimum required battery power for drone k to make a round trip is $\alpha_{s,k}$ when drone makes a single round trip departing from station s and returning to it again. Table 5.3 shows how the required number of batteries and the battery replacement cost will change by setting the end of life of drone $k's$ battery to $\alpha_{s,k}$ instead of replacing it when it reaches 80% of the initial capacity.

Table 5.3: Impact of f_{final} value on the required number of batteries and replacement cost

station (s)	drone (k)	$\alpha_{s,k}$ (min)	n_{min}	$n^*_{s,k}$	$\Delta t_{s,k}$ (days)	obj ($)
2	left	8	24	24	2504	3199
2	right	8	24	24	2504	3199
6	left	8	24	24	2504	3199
6	right	6	31	31	3364	3772
9	left	6	31	31	3364	3772
9	right	4	46	46	4683	5317
Total				180	-	22,161 $

Table 5.2 and Table 5.3 have the same parameter settings and data, except the parameter cap_{final} is decreased in Table 5.3. In Table 5.2, batteries are replaced when the capacity reaches 80% of the initial battery capacity; however, in Table 5.3, batteries are used for flight mission as long as it is possible to return to the station with the remaining battery capacity, which means $cap_{final} = \alpha_{s,k}, \forall s,k$. For this test case, $\alpha_{s,k}$ is 8, 6, or 4 minutes of fly, so cap_{final} is decreasing from 36 minutes in Table 5.2 to $\alpha_{s,k}$ in this section. Comparing Table 5.3 and Table 5.2, it can be seen that the number of required batteries is increased from a total of 171 to 180 (5.2%) due to the fact that the cap_{final} in Equation (5.44) is decreased.

The value of $\Delta t_{s,k}$ is also increased for all the drones. The reason for this increase is directly related to the increased number of available batteries in the station. Because more batteries are available, each battery is used less frequently. Hence, the total battery replacement cost is decreased from 27,928.8 $ to 22,161.0 $ (20.4%) in this case. By decreasing the value of cap_{final}, $\Delta t_{s,k}$ and the required number of batteries $n^*_{s,k}$ will always increase.

However, according to (5.24) since both nominator and denominator are increasing, the total cost may increase or decrease, depending on other parameters.

5.4.3.2 Battery Replacement Cost

The battery replacement cost for a drone depends on the number of batteries secured for the drone ($n_{s,k}$) and the battery replacement service time ($\Delta t_{s,k}$) as stated in Equation (5.24). It is easy to see that variable $\Delta t_{s,k}$ depends on $n_{s,k}$. As there are more batteries available to support a drone, each of those batteries will be used less frequently. This helps slow down the battery degradation, which will result in less frequent battery replacement. Different cases with the same parameter setting as Section 5.4.1 are examined here to show the relationship between Δt and α. The results of this section are based on one drone with the surveillance flight time of α minutes ranging from 19 to 22 minutes. Figure 5.8 shows the relationship between the number of available batteries (n) and the duration of service (Δt) in days for a battery until it gets replaced. The figure starts from having six batteries per drone, which is the minimum number of required batteries for the test problem. Figure 5.8 suggests that there is an approximately linear relationship between Δt and n, which is further detailed in Table 5.4. The adjusted R^2 value is more than 98% for all these cases.

Table 5.4: Linear relationship between Δt and n for different α values

α	n^{min}	linear regression line	R^2	n^*	average total cost
19	6	Δt = 46.8 n - 121.5	98.3%	10	3,017.7
20	6	Δt = 39.3 n - 96.3	98.3%	11	3,351.9
21	6	Δt = 30.0 n - 73.5	98.5%	11	4,051.5
22	6	Δt = 18.9 n - 52.9	98.9%	13	5,966.2

Table 5.4 also shows the required number of batteries (n^*) assigned to the drone and the average total cost associated with this number of batteries over a time horizon of 2 years ($TH = 2$ years). In lines 20-23 of Algorithm 4, the value of n increases from n^{min} until the

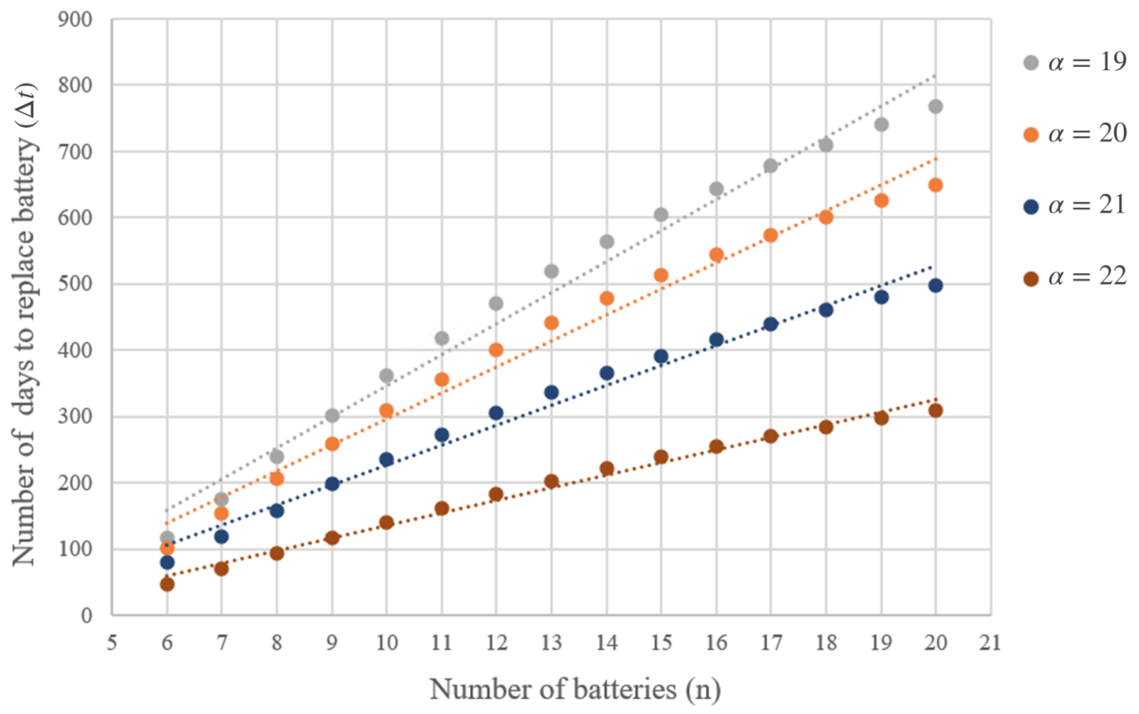


Figure 5.8: Impact of n and α on Δt

objective function value (battery cost) worsen. Figure 5.9 shows how the objective function value changes by increasing n for α values (the flight time of a single route trip) between 19 and 22. The n^{min} are shown with stars in this figure. As it can be seen, by increasing the number of batteries from n^{min} to n^*, the objective function value decreases and then it increases beyond the value of n^*.

5.4.3.3 Battery Charging Decision: Charge or Wait

Upon arrival of each drone to the station from a flight, a decision needs to be made regarding the drone charge: charge the battery or delay the charge to the next decision point. When the battery is fully charged, it will be stored idle at the station until it gets picked up to replace another. Figure 5.5 shows battery idle time before (ω_1) and after (ω_2) being charged. Hence, the total waiting time (ω) in each cycle of using battery is the summation of ω_1 and ω_2. For example, the percentage of the idle time (ρ) before being

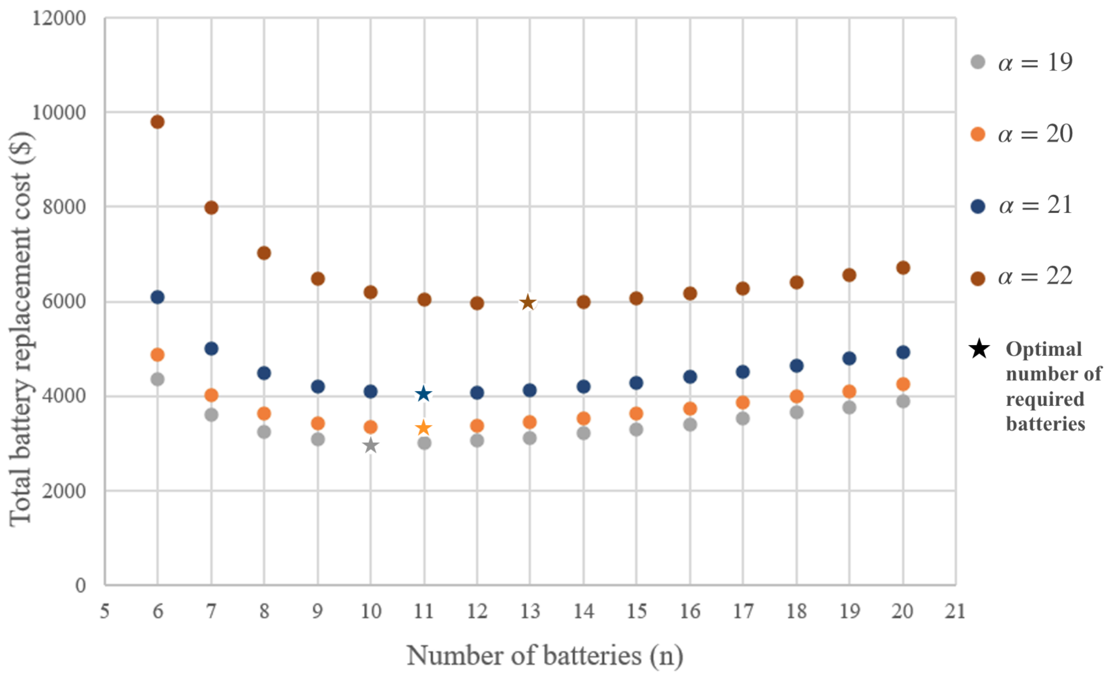


Figure 5.9: Impact of n on average total battery replacement cost

charged can be calculated as $\rho = \frac{\omega_1}{\omega}$. In all previous test cases, ρ was set to 1, which means battery charging is delayed until it is charged for an immediate use. Figure 5.10 shows a comparison between two scenarios $\rho = 0.75$ and $\rho = 1$ for charging a battery upon arrival at the station.

The same parameter setting as Table 5.1 is used for the scenarios in Figure 5.10. The value of α is set to 20 minutes in this section, which results in the minimum required number of batteries to be 6 ($n^{min}_{s,k} = 6$). The time span of using one battery assigned to a drone in a station (Δt) varies and it is 86 days for the case of $\rho = 0.75$ and 102 days for $\rho = 1.0$. For a given battery capacity of cap, a drone can make up to $f = \lfloor \frac{cap}{\alpha} \rfloor$ trips (Equation (5.25)). As an illustration, a drone with cap ranging from 40 and 45 can fly two rounds of trips without swapping the batteries. As the cap value decreases, the maximum trips on one fully charged battery decreases accordingly. The slope of graph in Figure 5.10 is steeper for $cap < 40$ minutes (i.e. $f = 1$) because the battery has a higher average remaining state of charge (μ_{SOC}) when it returns to the station and higher μ_{SOC} results in

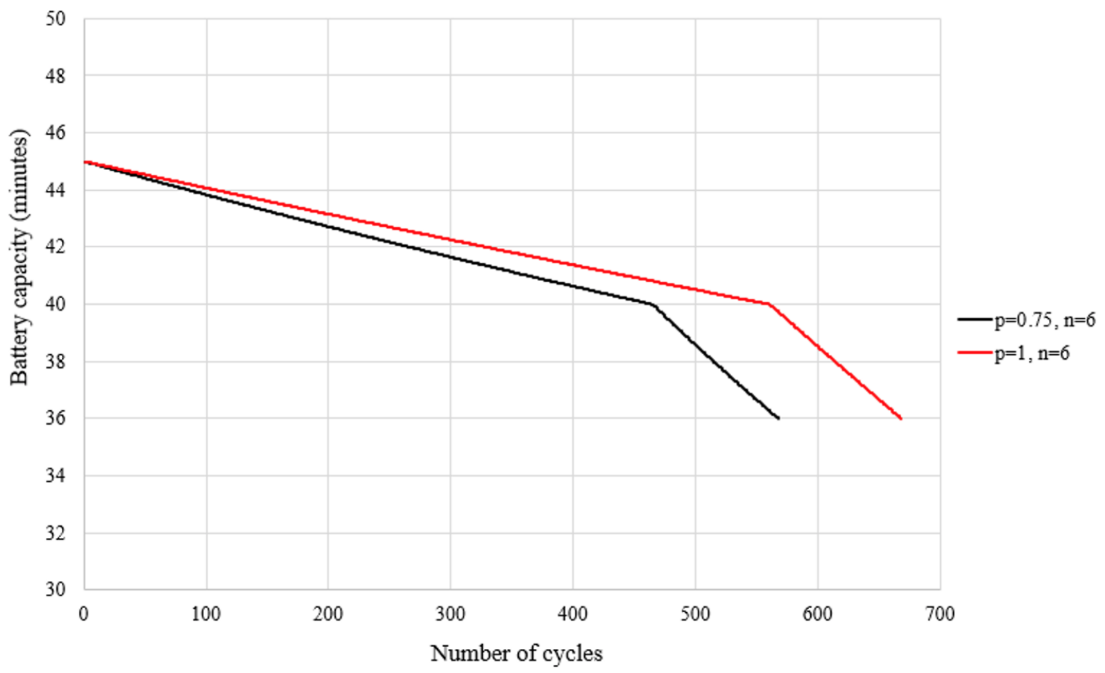


Figure 5.10: Two scenarios for battery charging upon arrival at the station

higher capacity degradation. Two major factors impacting the battery capacity degradation considered in this study are the mean (μ_{SOC}) and the standard deviation (σ_{SOC}) of SOC. As it can be observed from Figure 5.10, the battery capacity degradation was lower when the charging is delayed as much as possible. The reason is that higher values of μ_{SOC} and σ_{SOC} result in higher capacity degradation as Equations (5.20) and (5.21) state. By delaying the charging, the battery is remained idle at the station with empty charge which means μ_{SOC} and σ_{SOC} are lower and therefore, it degrades less.

The results indicate that the battery can be in service for a longer time period to make more flight missions by delaying the charge until it needs to be charged for the next use. However, the charging delay strategy has a potential risk of interrupting the continuous flight surveillance. A fully charged battery might not be available at the station due to postponing the charging when the drone returns to the station for battery swap, and the drone has to wait until a battery is ready.

5.5 Conclusion

This chapter presented a new concept, autonomous battery swap stations (ABSS), to expedite the battery swap operation so that drones can spend more time for surveillance missions of a critical region. Under the proposed approach, an ABSS was equipped with a sufficient number of batteries that were fully charged while the drone was out flying for the surveillance. Our approach considered battery capacity degradation over time. When a battery reached end-of-life, it was replaced with a new one. The impact of two main factors μ_{SOC} and σ_{SOC} on the battery capacity degradation were studied.

In order to visit important locations more frequently, the maximum revisiting gap concept was used to parameterize the criticality of surveillance locations. A MILP model was developed to determine the location of ABSSs to meet the maximum revisiting gap limitation of each waypoint along the surveillance region. An algorithm has been developed to find an appropriate number of batteries to be allocated to each drone by minimizing the battery replacement cost over a planning time horizon.

Several case study were conducted to study the impact of battery management decisions and parameter settings on the number of required batteries, their useful life span and the total battery replacement cost. The results showed that replacing a battery at a lower remaining capacity resulted in an increase in both $n^*_{s,k}$ and Δt. For the test case studied here, reducing the cap_{final} from 80% of the initial capacity to α resulted in 5.2% increase in total required batteries and an increase in Δt for all the batteries. We investigated the time to start recharging the returned battery. Numerical results indicate that charging the battery should be delayed until it is needed for the next flight. This strategy may help improve the battery end-of-life (Δt) so that it can serve longer at the station. For a test case, increasing the percentage of the idle time before being charged from $\rho = 075\%$ to $\rho = 100\%$ increases the Δt by 15.6%. The results also showed that there is an approximately linear relationship between Δt and n.

Chapter 6

Summary and Future Work

In summary, the primary objective of this dissertation is on the applications of drones in delivery and surveillance missions with the consideration to some factors impacting both path planning and flight schedules.

In Chapter 3, a delivery system design using drones was proposed that considered the impact of BCR on the fleet scheduling and flight path. A Phantom 4 Pro+ drone was tested to collect flight time and remaining battery charge data for different payload amount. Experimental data stated that the BCR was a linear function of carried payload, and therefore, a linear regression was used to model it. To design the drone-based delivery system, the SP and the OP models were proposed. The SP model determined the location of delivery bases by solving a set covering problem that took into account the feasible flight range from each candidate location. Furthermore, the OP model minimized the number of drones while considering the payload and flight time limitation, as well as the impact of payload on the flight time. The OP model determined optimal drone delivery assignments and their paths by solving a MILP model. To reduce the computational time of the OP model, a variable preprocessing algorithm, a primal and three different dual bound generation methods were developed. Several randomly generated delivery networks were used to test the proposed models and the efficiency of the proposed solution algorithms. The results indicated that the dual bound by network configuration method computationally outperformed the Lagrangian relaxation and the BPP-weight methods. The preprocessing algorithm and the bound generation methods were able to solve all tested cases, which was impossible without the inclusion of these methods. The contribution of this chapter can be extended by including other factors affecting the BCR such as flight speed and environmental conditions. By including the impact of flight speed on battery endurance, the flight speed itself can be optimized. Also, the battery capacity degradation and its impact on battery depth of charge and flight duration can be considered in the delivery network.

In Chapter 4, a reliable drone-based light-weight parcels delivery schedule considering drone failures was developed to minimize ELOD. A new concept based on how reliability is calculated, minimizing the expected loss of demand, was developed to evaluate the reliability of a drone delivery network and determine more reliable paths. A DDS-F optimization model was proposed to determine the flight schedule with minimum network ELOD subject to the drone's limited flight time and payload capacity. The numerical results stated that the DDS-F provided solutions more reliable on average compared to the traditional makespan model at a minor increase in makespan. Next, sensitivity analysis was performed on Weibull distribution (commonly used failure distributions in reliability analysis) parameters to elaborate on the model behavior which showed: 1) the network ELOD had a reverse correlation to the shape and scale parameters, 2) the impact of the shape parameter on the flight schedule was higher than the scale parameter, and 3) the shape parameter had an impact the optimal number of required drones and the optimal flight paths. A computationally efficient simulated annealing (SA) method coupled with the Sweep and Petal algorithms were proposed. The proposed SA was able to find optimal solutions to small-sized problems much faster than the exact method. There are several ways to extend more the contributions in Chapter 4. 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.

In Chapter 5, a novel utilization of ABSSs was proposed to extend the battery-powered drone flight time in a long and near-continuous surveillance mission. The ABSS had all of same functionalities as manned stations including swapping the depleted battery with a recharged one, charging the battery, and replacing it with a new one at the end of its lifetime. Typical drone behaviour included them departing from an ABSS to collect data and returning to the same station when the battery charge was less than the required level. Each station was equipped with a sufficient number of batteries so that drones would not spend a lot of time waiting at the station for the battery to be charged. In order to provide more frequent visits for more important or higher-risk locations, this chapter considered the importance associated with each location which was parameterized as the maximum revisiting gap. A MILP model was proposed to determine the ABSS location so that the minimum number of ABSS was established to cover the surveillance waypoints based on their maximum allowable revisiting gap. Furthermore, an algorithm was also proposed for the battery management mechanism in each station to prevent surveillance mission interruptions which determined the following: 1) required number of batteries secured for each drone to perform the non-interrupted mission; 2) the battery capacity degradation over time; 3) the battery charging policy; and 4) the replacement of the degraded battery with a new one. The impact of average SOC and the swing of the batteries (μ_{SOC} and σ_{SOC}) on the battery degradation was considered in the battery management. A case study was conducted to implement the proposed MILP model. Numerical results showed an almost linear relationship between Δt and n. Several other case studies were tested in sensitivity analysis. The results showed that replacing a battery at a lower remaining capacity resulted in an increase in both $n^*_{s,k}$ and Δt. Numerical results also indicated delay in charging the battery improved Δt so that it can serve longer at the station. As an extension of this chapter, one can develop a dynamic model in which the decision to land at the station or continuing the surveillance are made based on real-time information. The important level of surveillance locations might change even daily, which can be addressed in a dynamic model. The proposed MILP model can be extended by considering uncertainty in the battery charging and discharging rates, as well as the drone flight time, which can be included in the model. Another extension can be addressing the impact of environmental conditions such as temperature on the battery capacity degradation.

References

[1] R. Austin, *Unmanned aircraft systems: UAVS design, development and deployment*. John Wiley & Sons, 2011, vol. 54.

[2] S. Park, L. Zhang, and S. Chakraborty, "Battery assignment and scheduling for drone delivery businesses," in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2017, pp. 1–6.

[3] P. Heinlein, "Trump blasts mayor of hurricane-devastated San Juan, Puerto Rico," (access date: November, 2020). [Online]. Available: https://www.voanews.com/usa/trump-blasts-mayor-hurricane-devastated-san-juan-puerto-rico

[4] Statista Survey, "What were the reasons that made you/would make you cancel an order process?" (access date: November, 2020). [Online]. Available: https://www.statista.com/statistics/706565/reasons-for-cancelling-online-orders-in-the-us/

[5] T. Keeney ARK Analyst, "Amazon drones could deliver a package in under thirty minutes for one dollar," (access date: November, 2020). [Online]. Available: https://ark-invest.com/research/amazon-drone-delivery

[6] E. Petritoli, F. Leccese, and L. Ciani, "Reliability and maintenance analysis of unmanned aerial vehicles," *Sensors*, vol. 18, no. 9, p. 3171, 2018.

[7] K. A. Suzuki, P. Kemper Filho, and J. R. Morrison, "Automatic battery replacement

system for UAVs: Analysis and design," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 563–586, 2012.

[8] M. N. Boukoberine, Z. Zhou, and M. Benbouzid, "A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects," *Applied Energy*, vol. 255, p. 113823, 2019.

[9] S. A. Cambone, K. J. Krieg, P. Pace, and W. Linton, "Unmanned aircraft systems roadmap 2005-2030," *Office of the Secretary of Defense*, vol. 8, 2005.

[10] M. Torabbeigi, G. J. Lim, and S. J. Kim, "Drone delivery scheduling optimization considering payload-induced battery consumption rates," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 3, pp. 471–487, 2020.

[11] ——, "Drone delivery schedule optimization considering the reliability of drones," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 1048–1053.

[12] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous UAV surveillance in complex urban environments," in *2009 IEEE/ WIC/ ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2. IEEE, 2009, pp. 82–85.

[13] 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.

[14] 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.

[15] P. Anand, P. Arjun, N. B. Kumar, and K. Gowtham, "Drone ambulance support system," *International Journal of Engineering and Techniques*, vol. 4, 2018.

[16] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S. G. Nurre, and L. Bian, "Drones for disaster response and relief operations: a continuous approximation model," *International Journal of Production Economics*, vol. 188, pp. 167–184, 2017.

[17] A. Puri, "A survey of unmanned aerial vehicles (UAV) for traffic surveillance," *Department of computer science and engineering, University of South Florida*, pp. 1–29, 2005.

[18] J. Curry, J. Maslanik, G. Holland, and J. Pinto, "Applications of aerosondes in the arctic," *Bulletin of the American Meteorological Society*, vol. 85, no. 12, pp. 1855–1862, 2004.

[19] C. Corrigan, G. Roberts, M. Ramana, D. Kim, and V. Ramanathan, "Capturing vertical profiles of aerosols and black carbon over the indian ocean using autonomous unmanned aerial vehicles," *Atmospheric Chemistry and Physics*, vol. 8, no. 3, pp. 737–747, 2008.

[20] M. Car, D. Jurić Kaćunić, and M. S. Kovačević, "Application of unmanned aerial vehicle for landslide mapping," *Proceedings of the International Symposiun on Engineering Geodesy-SIG*, pp. 549–559, 2016.

[21] D. Giel, C. Brewer, and L. A. LaRocco, "Puerto Rico, short on fuel, cannot deliver food and medicine to the victims of Hurricane Maria," (access date: November, 2020). [Online]. Available: https://www.cnbc.com/2017/09/28/puerto-ricos-fuel-supply-breaks-down-in-the-wake-of-marias-devastation.html

[22] J. Gonzalez-Feliu, "Models and methods for the city logistics: The two-echelon capacitated vehicle routing problem," Ph.D. dissertation, Politecnico di Torino, 2008.

[23] Amazon.com Inc., "Amazon prime air," (access date: November, 2020). [Online]. Available: http://www.amazon.com/primeair

[24] V. Victoria Bryan, "Drone delivery: DHL 'parcelcopter' flies to german isle," Sep (access date: November, 2020). [Online]. Available: http://www.reuters.com/article/us-deutsche-post-drones-idUSKCN0HJ1ED20140924

[25] J. Stern, "Like Amazon, UPS also considering using unmanned flying vehicles," Dec (access date: November, 2020). [Online]. Available: http://abcnews.go.com/Technology/amazon-ups-drone-delivery-options/story?id=21086160

[26] 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.

[27] T. Keeney, "Drone delivery: How can Amazon charge $1 for drone delivery?" (access date: November, 2020). [Online]. Available: https://ark-invest.com/research/drone-delivery-amazon

[28] 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.

[29] L. W. Traub, "Range and endurance estimates for battery-powered aircraft," *Journal of Aircraft*, vol. 48, no. 2, pp. 703–707, 2011.

[30] G. J. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge university press, 2006.

[31] 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.

[32] N. A. Terning, "Real-time flight optimization for tracking stop-and-go targets with micro air vehicles," Ph.D. dissertation, Air Force Institute of Technology Department of The Air Force, 2008.

[33] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for unmanned aerial vehicles using markov decision processes," in *IEEE International Conference on Rotics and Automation*, 2013, pp. 784–789.

[34] D. M. Marshall, R. K. Barnhart, S. B. Hottman, E. Shappee, and M. T. Most ,editors, *Introduction to unmanned aircraft systems*. CRC Press, 2016.

[35] AltiGator, "Drone comparison chart," (access date: November, 2020). [Online]. Available: https://drones.altigator.com/professional-drone-UAV-comparison-chart

[36] 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.

[37] R. Schaefer, "Unmanned aerial vehicle reliability study," *Office of the Secretary of Defense, Washington, DC*, 2003.

[38] I. Baghdadi, O. Briat, J.-Y. Delétage, P. Gyan, and J.-M. Vinassa, "Lithium battery aging model based on dakin's degradation approach," *Journal of Power Sources*, vol. 325, pp. 273–285, 2016.

[39] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80–91, 1959.

[40] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[41] A. Tucker, "On directed graphs and integer programs," in *Symposium on Combinatorial Problems, Princeton University*, 1960.

[42] G. B. Dantzig, *Linear programming and extensions.* Princeton University Press, 1998, vol. 48.

[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] M. Fisher, "Vehicle routing," *Handbooks in Operations Research and Management Science*, vol. 8, pp. 1–33, 1995.

[45] S. Karaman and E. Frazzoli, "Linear temporal logic vehicle routing with applications to multi-UAV mission planning," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1372–1395, 2011.

[46] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[47] S. Sancı and V. İşler, "A parallel algorithm for uav flight route planning on gpu," *International Journal of Parallel Programming*, vol. 39, no. 6, pp. 809–837, 2011.

[48] K. Sundar and S. Rathinam, "Algorithms for heterogeneous, multiple depot, multiple unmanned vehicle path planning problems," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2-4, pp. 513–526, 2017.

[49] S. G. Manyam, S. Rathinam, and S. Darbha, “Computation of lower bounds for a multiple depot, multiple vehicle routing problem with motion constraints,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, 2015.

[50] X. Jiang, Q. Zhou, and Y. Ye, “Method of task assignment for uav based on particle swarm optimization in logistics,” in *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, 2017, pp. 113–117.

[51] G. Laporte and Y. Nobert, “Exact algorithms for the vehicle routing problem,” in *North-Holland Mathematics Studies*. Elsevier, 1987, vol. 132, pp. 147–184.

[52] N. Christofides and S. Eilon, “An algorithm for the vehicle-dispatching problem,” *Journal of the Operational Research Society*, vol. 20, no. 3, pp. 309–318, 1969.

[53] N. Karmarkar and R. M. Karp, *The differencing method of set partitioning*. Computer Science Division (EECS), University of California Berkeley, 1982.

[54] E. L. Lawler and D. E. Wood, “Branch-and-bound methods: A survey,” *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[55] A. I. Ali and J. L. Kennington, “The asymmetric m-travelling salesmen problem: A duality based branch-and-bound algorithm,” *Discrete Applied Mathematics*, vol. 13, no. 2-3, pp. 259–276, 1986.

[56] M. Fischetti and P. Toth, “An additive bounding procedure for the asymmetric travelling salesman problem,” *Mathematical Programming*, vol. 53, no. 1-3, pp. 173–197, 1992.

[57] G. Carpaneto, M. Dell'Amico, and P. Toth, "Exact solution of large-scale, asymmetric traveling salesman problems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 21, no. 4, pp. 394–409, 1995.

[58] M. Fischetti, A. Lodi, and P. Toth, "Exact methods for the asymmetric traveling salesman problem," in *The Traveling Salesman Problem and its Variations*. Springer, 2007, pp. 169–205.

[59] S. Eilon, C. D. T. Watson-Gandy, N. Christofides, and R. de Neufville, "Distribution management-mathematical modelling and practical analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 6, pp. 589–589, 1974.

[60] R. Goel and R. Maini, "Vehicle routing problem and its solution methodologies: a survey," *International Journal of Logistics Systems and Management*, vol. 28, no. 4, pp. 419–435, 2017.

[61] R. Bellman, "Dynamic programming treatment of the travelling salesman problem," *Journal of the ACM (JACM)*, vol. 9, no. 1, pp. 61–63, 1962.

[62] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied mathematics*, vol. 10, no. 1, pp. 196–210, 1962.

[63] P. Bouman, N. Agatz, and M. Schmidt, "Dynamic programming approaches for the traveling salesman problem with drone," *Networks*, vol. 72, no. 4, pp. 528–542, 2018.

[64] M. L. Balinski and R. E. Quandt, "On an integer program for a delivery problem," *Operations research*, vol. 12, no. 2, pp. 300–304, 1964.

[65] M. Desrochers, J. Desrosiers, and M. Solomon, “A new optimization algorithm for the vehicle routing problem with time windows,” *Operations research*, vol. 40, no. 2, pp. 342–354, 1992.

[66] B. A. Foster and D. M. Ryan, “An integer programming approach to the vehicle scheduling problem,” *Journal of the Operational Research Society*, vol. 27, no. 2, pp. 367–384, 1976.

[67] M. Rao and S. Zionts, “Allocation of transportation units to alternative trips—a column generation scheme with out-of-kilter subproblems,” *Operations Research*, vol. 16, no. 1, pp. 52–63, 1968.

[68] I. Elhallaoui, D. Villeneuve, F. Soumis, and G. Desaulniers, “Dynamic aggregation of set-partitioning constraints in column generation,” *Operations Research*, vol. 53, no. 4, pp. 632–645, 2005.

[69] A. B. Kahng, L. Wang, and B. Xu, “Tritonroute: an initial detailed router for advanced vlsi technologies,” in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–8.

[70] S. M. Gonçalves, L. S. da Rosa, and F. d. S. Marques, “An improved heuristic function for A*-based path search in detailed routing,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[71] J. G. Carlsson and S. Song, “Coordinated logistics with a truck and a drone,” *Management Science*, vol. 64, no. 9, pp. 4052–4069, 2018.

[72] S. Ansari, M. Başdere, X. Li, Y. Ouyang, and K. Smilowitz, “Advancements in continuous approximation models for logistics and transportation systems: 1996–2016,” *Transportation Research Part B: Methodological*, vol. 107, pp. 229–252, 2018.

[73] M. Saberi and İ. Ö. Verbas, "Continuous approximation model for the vehicle routing problem for emissions minimization at the strategic level," *Journal of Transportation Engineering*, vol. 138, no. 11, pp. 1368–1376, 2012.

[74] B. A. Davis and M. A. Figliozzi, "A methodology to evaluate the competitiveness of electric delivery trucks," *Transportation Research Part E: Logistics and Transportation Review*, vol. 49, no. 1, pp. 8–23, 2013.

[75] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, "A generic exact solver for vehicle routing and related problems," in *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2019, pp. 354–369.

[76] R. Matai, S. P. Singh, and M. L. Mittal, "Traveling salesman problem: an overview of applications, formulations, and solution approaches," *Traveling Salesman Problem, Theory and Applications*, vol. 1, 2010.

[77] A. Fenton, "The bees algorithm for the vehicle routing problem," *arXiv preprint arXiv:1605.05448*, 2016.

[78] J. Holand, "Adaptation in natural and artificial systems, the university of michigan press," *Ann Arbour*, p. 3l, 1975.

[79] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[80] N. M. Razali, J. Geraghty *et al.*, "Genetic algorithm performance with different selection strategies in solving tsp," in *Proceedings of the world congress on engineering*, vol. 2, no. 1. International Association of Engineers Hong Kong, 2011, pp. 1–6.

[81] I. Oliver, D. Smith, and J. R. Holland, "Study of permutation crossover operators on the traveling salesman problem," in *Genetic algorithms and their applications:*

proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA. Hillsdale, NJ: L. Erlhaum Associates, 1987., 1987.

[82] Y. Nagata, "Edge assembly crossover for the capacitated vehicle routing problem," in *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 2007, pp. 142–153.

[83] J. Berger and M. Barkaoui, "A hybrid genetic algorithm for the capacitated vehicle routing problem," in *Genetic and evolutionary computation conference*. Springer, 2003, pp. 646–656.

[84] K. Meer, "Simulated annealing versus metropolis for a tsp instance," *Information Processing Letters*, vol. 104, no. 6, pp. 216–219, 2007.

[85] F. Robust, C. F. Daganzo, and R. R. Souleyrette II, "Implementing vehicle routing models," *Transportation Research Part B: Methodological*, vol. 24, no. 4, pp. 263–286, 1990.

[86] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of operations research*, vol. 41, no. 4, pp. 421–451, 1993.

[87] M. L. Fredman, D. S. Johnson, L. A. McGeoch, and G. Ostheimer, "Data structures for traveling salesmen," *Journal of Algorithms*, vol. 18, no. 3, pp. 432–479, 1995.

[88] J. Willard, "Vehicle routing using r-optimal tabu search. master's thesis," *The Management School, Imperial College, London*, 1989.

[89] P. Toth and D. Vigo, "The granular tabu search and its application to the vehicle-routing problem," *Informs Journal on computing*, vol. 15, no. 4, pp. 333–346, 2003.

[90] E. Bonabeau, M. Dorigo, D. d. R. D. F. Marco, G. Theraulaz, G. Théraulaz *et al.*, *Swarm intelligence: from natural to artificial systems.* Oxford university press, 1999, no. 1.

[91] X.-S. Yang and M. Karamanoglu, "Swarm intelligence and bio-inspired computation: an overview," in *Swarm intelligence and bio-inspired computation.* Elsevier, 2013, pp. 3–23.

[92] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[93] T. J. Ai and V. Kachitvichyanukul, "A particle swarm optimisation for vehicle routing problem with time windows," *International Journal of Operational Research*, vol. 6, no. 4, pp. 519–537, 2009.

[94] Q. Zhu, L. Qian, Y. Li, and S. Zhu, "An improved particle swarm optimization algorithm for vehicle routing problem with time windows," in *2006 IEEE International Conference on Evolutionary Computation.* IEEE, 2006, pp. 1386–1390.

[95] T. Ai and V. Kachitvichyanukul, "A particle swarm optimization for the capacitated vehicle routing problem," *International journal of logistics and SCM systems*, vol. 2, no. 1, pp. 50–55, 2007.

[96] S. Geetha, G. Poonthalir, and P. Vanathi, "Nested particle swarm optimisation for multi-depot vehicle routing problem," *International Journal of Operational Research*, vol. 16, no. 3, pp. 329–348, 2013.

[97] L. M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric tsps by ant colonies," in *Proceedings of IEEE International Conference on Evolutionary Computation.* IEEE, 1996, pp. 622–627.

[98] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[99] J. Yang, X. Shi, M. Marchese, and Y. Liang, “An ant colony optimization method for generalized tsp problem,” *Progress in Natural Science*, vol. 18, no. 11, pp. 1417–1422, 2008.

[100] B. Yu, Z.-Z. Yang, and B. Yao, “An improved ant colony optimization for vehicle routing problem,” *European journal of operational research*, vol. 196, no. 1, pp. 171–176, 2009.

[101] B. Bullnheimer, R. F. Hartl, and C. Strauss, “An improved ant system algorithm for thevehicle routing problem,” *Annals of operations research*, vol. 89, pp. 319–328, 1999.

[102] J. E. Bell and P. R. McMullen, “Ant colony optimization techniques for the vehicle routing problem,” *Advanced engineering informatics*, vol. 18, no. 1, pp. 41–48, 2004.

[103] C.-H. Chen and C.-J. Ting, “An improved ant colony system algorithm for the vehicle routing problem,” *Journal of the Chinese institute of industrial engineers*, vol. 23, no. 2, pp. 115–126, 2006.

[104] I. Or, “Traveling salesman type combinatorial problems and their relation to the logistics of regional blood banking.” 1977.

[105] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, “The bees algorithm—a novel tool for complex optimisation problems,” in *Intelligent production machines and systems*. Elsevier, 2006, pp. 454–459.

[106] X.-S. Yang, "Engineering optimizations via nature-inspired virtual bee algorithms," in *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer, 2005, pp. 317–323.

[107] M. Alzaqebah, S. Jawarneh, H. M. Sarim, and S. Abdullah, "Bees algorithm for vehicle routing problems with time windows," *International Journal of Machine Learning and Computing*, vol. 8, no. 3, pp. 234–240, 2018.

[108] K. Kantawong and S. Pravesjit, "An enhanced abc algorithm to solve the vehicle routing problem with time windows," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 14, no. 1, pp. 46–52, 2020.

[109] 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.

[110] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," in *2010 International Conference on Emerging Security Technologies*. IEEE, 2010, pp. 142–147.

[111] G. J. Lim, S. Kim, J. Cho, Y. Gong, and A. Khodaei, "Multi-UAV pre-positioning and routing for power network damage assessment," p. 1–1, 2016.

[112] 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.

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

[114] 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.

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

[116] D. Howarth, “NASA could use drones to explore Mars,” (access date: December, 2020). [Online]. Available: https://www.dezeen.com/2015/01/28/nasa-helicopter-drones-explore-mars-jet-propulsion-laboratory/

[117] W. R. Koski, T. Allen, D. Ireland, G. Buck, P. R. Smith, A. M. Macrander, M. A. Halick, C. Rushing, D. J. Sliwa, and T. L. McDonald, “Evaluation of an unmanned airborne system for monitoring marine mammals,” *Aquatic Mammals*, vol. 35, no. 3, p. 347, 2009.

[118] M. Fingas and C. Brown, “Review of oil spill remote sensing,” *Marine pollution bulletin*, vol. 83, no. 1, pp. 9–23, 2014.

[119] 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.

[120] 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.

[121] P. Oberlin, S. Rathinam, and S. Darbha, “Today’s traveling salesman problem,” *IEEE robotics & automation magazine*, vol. 17, no. 4, pp. 70–77, 2010.

[122] 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.

[123] 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.

[124] 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.

[125] J. G. Carlsson and S. Song, "Coordinated logistics with a truck and a drone," *Management Science*, 2017.

[126] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[127] Amazon Inc., "Amazon Prime Air," (access date: July, 2018). [Online]. Available: www.amazon.com/primeair

[128] Mercedes-Benz co., "Vans & drones in zurich," (access date: November, 2020). [Online]. Available: https://www.mercedes-benz.com/en/mercedes-benz/vehicles/transporter/vans-drones-in-zurich/

[129] M. McFarland, "UPS drivers may tag team deliveries with drones," (access date: July, 2018). [Online]. Available: http://money.cnn.com/2017/02/21/technology/ups-drone-delivery/index.html

[130] DHL, "Successful trial integration of DHL Parcelcopter into logistics chain,"

(access date: July, 2018). [Online]. Available: http://www.dhl.com/en/press/releases/releases_2016/all/parcel_ecommerce

[131] Swiss Post Co., "Swiss post is planning to use drones for the ticino hospital group in lugano," (access date: 2017). [Online]. Available: https://www.post.ch/en/aut-us/company/media/press-releases/2017/swiss-post-is-planning-to-use-drones

[132] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transportation Science*, 2018.

[133] H. H. Minh, Y. Deville, Q. D. Pham, and M. H. Quang, "Heuristic methods for the traveling salesman problem with drone," 2015.

[134] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

[135] Microsoft, "Welcome to AirSim," (access date: November, 2020). [Online]. Available: https://github.com/Microsoft/AirSim

[136] A. Kandhalu, A. Rowe, and R. Rajkumar, "Dspcam: A camera sensor system for surveillance networks," in *2009 Third ACM/ IEEE International Conference on Distributed Smart Cameras (ICDSC)*. IEEE, 2009, pp. 1–7.

[137] R. T. Collins, A. J. Lipton, and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 745–746, 2000.

[138] "Surveillance and security drone: Airborne drones," (access date: December, 2020). [Online]. Available: https://www.airbornedrones.co/surveillance-and-security/

[139] "Solution- an aerial monitoring solution of the future in operation today," (access date: December, 2020). [Online]. Available: https://skyx.com/solution/

[140] "EURECAT– flying robot makes sewer inspection safer and faster," (access date: December, 2020). [Online]. Available: https://www.earto.eu/rto-innovation/eurecat-flying-robot-makes-sewer-inspection-safer-and-faster/

[141] Xinhua, "SW China police use drones to monitor traffic," (access date: December, 2020). [Online]. Available: https://www.chinadailyasia.com/nation/2016-09/18/content_15496963.html

[142] C. C. Haddal and J. Gertler, "Homeland security: Unmanned aerial vehicles and border surveillance." LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE, 2010.

[143] K. J. Hayhurst, J. M. Maddalon, P. S. Miner, M. P. DeWalt, and G. F. McCormick, "Unmanned aircraft hazards and their implications for regulation," in *2006 IEEE/AIAA 25th Digital Avionics Systems Conference*. IEEE, 2006, pp. 1–12.

[144] D. J. Bier and M. Feeney, *Drones on the Border: Efficacy and Privacy Implications*. Cato Institute, 2018.

[145] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications," *J. Commun*, vol. 9, no. 9, pp. 687–692, 2014.

[146] J. Scott and C. Scott, "Drone delivery models for healthcare," in *Hawaii International Conference on System Sciences*, 2017, pp. 3297–3304.

[147] M. Nakayama, S. Wada, S. Kuroki, and M. Nogami, "Factors affecting cyclic durability of all-solid-state lithium polymer batteries using poly (ethylene oxide)-based

solid polymer electrolytes," *Energy & Environmental Science*, vol. 3, no. 12, pp. 1995–2002, 2010.

[148] 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.

[149] Y. Mulgaonkar, M. Whitzer, B. Morgan, C. M. Kroninger, A. M. Harrington, and V. Kumar, "Power and weight considerations in small, agile quadrotors," in *Micro-and Nanotechnology Sensors, Systems, and Applications VI*, vol. 9083. International Society for Optics and Photonics, 2014, p. 90831Q.

[150] D. W. Beekman, "Micro air vehicle endurance versus battery size," in *Micro-and Nanotechnology Sensors, Systems, and Applications II*, vol. 7679. International Society for Optics and Photonics, 2010, p. 767910.

[151] 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.

[152] 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.

[153] E. A. Elsayed, "Life cycle costs and reliability engineering," *Wiley StatsRef: Statistics Reference Online*, 2014.

[154] J. Lienig and H. Bruemmer, *Fundamentals of electronic systems design.* Springer, 2017.

[155] B. S. Dhillon, *Engineering maintenance: a modern approach.* CRC press, 2002.

[156] R. N. Allan *et al.*, *Reliability evaluation of power systems.* Springer Science & Business Media, 2013.

[157] J. Gao, "Optimization of distribution routing problem based on travel time reliability," *International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 1, pp. 19–22, 2009.

[158] X. Zhang, Y. Pu, H. Liu, and D. Yang, "Vehicle routing optimization considering dynamic reliability during lasting period of traffic incidents," in *International Conference of Logistics Engineering and Management (ICLEM), American Society of Civil Engineers.*, 2010, pp. 2871–2877.

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

[160] L. Tang, W. Cheng, and J. Liang, "Vehicle routing problem research based on road network reliability," in *International Conference on Transportation Engineering 2009*, 2009, pp. 663–668.

[161] 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, pp. 293–311, 2006.

[162] W. H. Lam, C. Chen, K. Chan, and A.-Z. Ren, "Optimizing vehicle fleet management with travel time reliability constraint," *Communication and Transportati0n Systems Engineering and Information*, vol. 5, p. 021, 2005.

[163] J. Gao, "Optimization of distribution routing problem based on travel time reliability," in *2009 international conference on information management, innovation management and industrial engineering*, vol. 1. IEEE, 2009, pp. 19–22.

[164] A. Hobbs and S. R. Herwitz, "Human challenges in the maintenance of unmanned aircraft systems," *FAA and NASA Report*, 2006.

[165] R. Khan, P. Williams, P. Riseborough, A. Rao, and R. Hill, "Active fault tolerant flight control system design," *arXiv preprint arXiv:1610.02139*, 2016.

[166] R. Whittle, *Predator: the secret origins of the drone revolution*. Macmillan, 2014.

[167] K. Fujii, K. Higuchi, and J. Rekimoto, "Endless flyer: a continuous flying drone with automatic battery replacement," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*. IEEE, 2013, pp. 216–223.

[168] M. Khonji, M. Alshehhi, C.-M. Tseng, and C.-K. Chau, "Autonomous inductive charging system for battery-operated electric drones," in *Proceedings of the Eighth International Conference on Future Energy Systems*. ACM, 2017, pp. 322–327.

[169] 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.

[170] M. N. Boukoberine, Z. Zhou, and M. Benbouzid, "Power supply architectures for drones-a review," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 5826–5831.

[171] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.

[172] Z. Bi, T. Kan, C. C. Mi, Y. Zhang, Z. Zhao, and G. A. Keoleian, "A review of wireless power transfer for electric vehicles: Prospects to enhance sustainable mobility," *Applied Energy*, vol. 179, pp. 413–425, 2016.

[173] "Is wireless charging going to be a future of electric vehicle charging?" (access date: November, 2020). [Online]. Available: http://www.murali.today/wireless-charging-future-of-electric-vehicle-charging

[174] I. Hong, M. Kuby, and A. Murray, "A deviation flow refueling location model for continuous space: A commercial drone delivery system for urban areas," in *Advances in Geocomputation*. Springer, 2017, pp. 125–132.

[175] B. D. Song, K. Park, and J. Kim, "Persistent UAV delivery logistics: Milp formulation and efficient heuristic," *Computers & Industrial Engineering*, vol. 120, pp. 418–428, 2018.

[176] M. Wang, "Systems and methods for UAV battery exchange," Jul. 30 2019, US Patent 10,363,826.

[177] ——, "Multi-zone battery exchange system," Jan. 24 2017, US Patent 9,550,582.

[178] The Tesla Team, "Battery swap pilot program," (access date: November, 2020). [Online]. Available: https://www.tesla.com/blog/battery-swap-pilot-program

[179] G. Ning, Z. Zhen, P. Wang, Y. Li, and H. Yin, "Economic analysis on value chain of taxi fleet with battery-swapping mode using multiobjective genetic algorithm," *Mathematical Problems in Engineering*, vol. 2012, 2012.

[180] 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.

[181] 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.

[182] 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.

[183] 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.

[184] SZ DJI Technology Co., Ltd., “Phantom 4 Pro,” (access date: November, 2020). [Online]. Available: https://www.dji.com/phantom-4-pro

[185] Handbook, Helicopter Flying, “FAA-H-8083-21A,” 2014.

[186] T. Caric and H. Gold, *Vehicle routing problem.* IntechOpen, 2008.

[187] L. A. Wolsey, “Integer programming,” *IIE Transactions*, vol. 32, no. 273-285, pp. 2–58, 2000.

[188] 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.

[189] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms.* Algorithms and Combinatorics, 2006.

[190] *MATLAB and Statistics Toolbox Release R2016a The MathWorks, Inc., Natick, MA, USA.*

[191] GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.7.3. Washington, DC, USA, 2016. [Online]. Available: http://www.gams.com/

[192] IBM ILOG, "CPLEX reference manual," vol. 12.6.3.0, Released: July 2016. [Online]. Available: http://www.ilog.com

[193] 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.

[194] 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.

[195] 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.

[196] F. Schenkelberg, "How reliable does a delivery drone have to be," in *2016 Annual Reliability and Maintainability Symposium (RAMS). IEEE. ISBN*, 2016, pp. 978–1.

[197] 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.

[198] 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.

[199] 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.

[200] 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.

[201] 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.

[202] 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.

[203] 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.

[204] 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.

[205] 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.

[206] 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.

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

[208] 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.

[209] L. Dorr and A. Duquette, "Fact sheet–small unmanned aircraft regulations (part 107)," *Federal Aviation Administration*, 2016.

[210] 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.

[211] 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.

[212] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[213] I. Ilhan, "A population based simulated annealing algorithm for capacitated vehicle routing problem," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 28, no. 3, pp. 1217–1235, 2020.

[214] P. A. Wicaksono, D. Puspitasari, S. Ariyandanu, and R. Hidayanti, "Comparison of simulated annealing, nearest neighbour, and tabu search methods to solve vehicle routing problems," 2020.

[215] F. Y. Vincent, A. P. Redi, Y. A. Hidayat, and O. J. Wibowo, "A simulated annealing heuristic for the hybrid vehicle routing problem," *Applied Soft Computing*, vol. 53, pp. 119–132, 2017.

[216] V. F. Yu, S. S. Purwanti, A. A. N. P. Redi, C.-C. Lu, S. Suprayogi, and P. Jewpanya, “Simulated annealing heuristic for the general share-a-ride problem,” *Engineering Optimization*, vol. 50, no. 7, pp. 1178–1197, 2018. [Online]. Available: https://doi.org/10.1080/0305215X.2018.1437153

[217] A. A. N. P. Redi, F. R. Maula, F. Kumari, N. U. Syaveyenda, N. Ruswandi, A. U. Khasanah, and A. C. Kurniawan, “Simulated annealing algorithm for solving the capacitated vehicle routing problem: A case study of pharmaceutical distribution,” *Jurnal Sistem dan Manajemen Industri*, vol. 4, no. 1, pp. 41–49, 2020.

[218] A. Ponza, “Optimization of drone-assisted parcel delivery,” Master’s thesis, University of Padua, 2016.

[219] 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.

[220] 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.

[221] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[222] “Python interface,” access date: November, 2020. [Online]. Available: https://www.gurobi.com/documentation/8.1/quickstart_windows/py_\python_interface.html

[223] 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.

[224] 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.

[225] S. S. Fazeli, S. Venkatachalam, R. B. Chinnam, and A. Murat, “Two-stage stochastic choice modeling approach for electric vehicle charging station network design in urban communities,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[226] V. Hassija, V. Saxena, and V. Chamola, “Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory,” *Computer Communications*, vol. 149, pp. 51–61, 2020.

[227] N. K. Ure, G. Chowdhary, T. Toksoz, J. P. How, M. A. Vavrina, and J. Vian, “An automated battery management system to enable persistent missions with multiple aerial vehicles,” *IEEE/ASME transactions on mechatronics*, vol. 20, no. 1, pp. 275–286, 2014.

[228] E. W. Saad, J. L. Vian, M. A. Vavrina, J. A. Nisbett, and D. C. Wunsch, “Vehicle base station,” Dec. 2 2014, uS Patent 8,899,903.

[229] R. Masum, “Continuous surveillance design for critical smart grid infrastructure using unmanned aerial vehicles,” Ph.D. dissertation, Tennessee Technological University, 2018.

[230] C. Rizzo, P. Cavestany, F. Chataigner, M. Soler, G. Moreno, D. Serrano, F. Lera, and J. L. Villarroel, “Wireless propagation characterization of underground sewers towards autonomous inspections with drones,” in *Iberian Robotics conference*. Springer, 2017, pp. 849–860.

[231] J. Jones, "Sewer drones set to take over one of the smelliest jobs in Barcelona," Dec (access date: November, 2020). [Online]. Available: https://www.thelocal.es/20151207/could-drones-soon-replace-workers-in-barcelonas-sewers

[232] A. P. Colefax, P. A. Butcher, and B. P. Kelaher, "The potential for unmanned aerial vehicles (UAVs) to conduct marine fauna surveys in place of manned aircraft," *ICES Journal of Marine Science*, vol. 75, no. 1, pp. 1–8, 2018.

[233] R. Gorkin, K. Adams, M. J. Berryman, S. Aubin, W. Li, A. R. Davis, and J. Barthelemy, "Sharkeye: Real-time autonomous personal shark alerting via aerial surveillance," *Drones*, vol. 4, no. 2, p. 18, 2020.

[234] N. Ahmadian, G. J. Lim, M. Torabbeigi, and S. J. Kim, "Smart border patrol using drones and wireless charging system under budget limitation." University of Houston, 2020 (access date: November, 2020). [Online]. Available: http://e2map.egr.uh.edu/publications

[235] A. Millner, "Modeling lithium ion battery degradation in electric vehicles," in *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*. IEEE, 2010, pp. 349–356.

[236] K. N. Genikomsakis, C. S. Ioakimidis, A. Murillo, A. Trifonova, and D. Simic, "A life cycle assessment of a li-ion urban electric vehicle battery," in *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*. IEEE, 2013, pp. 1–11.