Novel Applications of

Optimization Models in Drone Routing and Scheduling

by
Hyungjin Park

A Thesis submitted to the Department of Industrial Engineering,

Cullen College of Engineering

in partial fulfillment of the requirements for the degree of

Master of Science
in Industrial Engineering

Chair of Committee: Gino Lim

Committee Member: Jiming Peng

Committee Member: Cumaraswami Vipulanandan

University of Houston
May 2021

Dedicated to

my almighty God

my beloved Family, Sungmi, Haneung and Haron

and my dear parents

with all my love

Acknowledgements

It is a pleasure to thank those who made this thesis 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 writing. I would like to thank my respected seniors of Korea Military Academy, Dr. Jeayoung Cho, Dr. Seonjin Kim and Dr. Donghwan Kim for his gracious support, invaluable ideas, and inspiring suggestions. I am also grateful to my friend Reza Mirjalili, who has always supported me in my research activities. Many thanks to the thesis committee members, Dr. Jiming Peng, Dr. Cumaraswamy Vipulanandan, for their insightful comments that improved the work's contents.

I am extremely grateful to the Korean Army for generously supporting me so that I can study in the US for two years. Although what I have learned here is insignificant compared to the many and deep knowledge of the world, even that little knowledge will contribute to the development of the Republic of Korea and the Army.

I would like to express my sincere gratitude to my beloved son Joshua Haron, who was born into the world safely despite the Covid-19 pandemic, my beloved daughter Haneung who always gives me only happiness, and my wife, Sungmi Ryu. Also, I express my gratitude and love to my beloved parents who always keep me through sincere prayer. Finally, I give all this glory to my God, who is always my strength and my good way.

Abstract

Drone technologies can have a positive impact on surveillance, emergency response, and delivery. Many existing optimization models in drone routing and scheduling focus on minimizing the cost or time required to complete a mission. This study explores novel applications of drones for healthcare delivery and structural inspection considering the physics of battery consumption that are often ignored in the Operations Research community.

The COVID-19 pandemic has affected everyone in ways never imagined and various social distancing measures are in place to reduce the spread of viruses. If at-home testing kits are safely and quickly delivered to a patient, it can potentially reduce human contact and positively affect disease spread before, during, and after diagnosis. Hence, the first subject of this thesis proposes testing kit delivery schedules using drones based on the Mothership and Drone Routing Problem (MDRP). Optimization models and a decomposition-based solution methodology are developed to solve the complex model. The performance on virus spread reduction rate was measured by the 'R' method. Computational results show that the proposed approach ($R_0 = 0.002$) resulted in considerably lower infection risk compared to the face-to-face testing practice ($R_0 = 0.0153$).

The second subject of this thesis introduces drone path planning for structural inspection considering the physics of battery consumption. The short battery duration of drones remains a major problem for small drones. Considering the shape of large structures, drones have a variety of flight dynamics during a mission, in which certain moves require a faster battery consumption than others. However, these factors have not been thoroughly considered in the existing routing models. Hence, this study examines different aspects of routing drones to cover multiple inspection points distributed on a three-dimensional structure. Both MIP models (labelled as SFD and MEC) are developed to obtain optimal routing strategies for both the shortest distance and the minimum battery consumption. Numerical results show that the optimal solutions form these two models produce different paths. Understanding that each decision maker may have different preference between those two objectives, a bi-objective optimization model has been developed to find an efficient frontier of solutions to satisfy the decision maker's preference.

Table of Contents

List of Tables

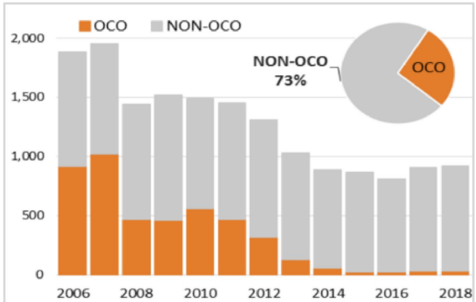
List of Figures

Chapter 1

Introduction

1.1 Background & Motivation

In the early days, drones were mainly used for military operations. Drones are being used in various fields in recent years under the defense policy of 1) Reduction of military power, 2) Prohibition of killing of non-combatants, and 3) Fusion of cutting-edge science and technologies with military technologies. Since drones have been used instead of soldiers for dangerous military missions, the mortality rate of combatants has decreased dramatically. In Figure 1.1, the orange-colored portion of the graph left-side represents the mortality rate of combatants on the battlefield. The U.S. military began using drones for military missions during the war in Afghanistan (2001$\sim$2014). At that time, drones mainly performed dangerous missions such as reconnaissance of enemy forces, assassination of factors, and strikes. We can see that the mortality rate of combatants has decreased by more than 90 *percent* compared with that of in 2006. However, it can also be observed that the mortality rate has not improved in non-combat mission areas where drones have been rarely used (See gray-colored portion of the left graph in Figure 1.1). With the gradual change in trend in civil sectors and military missions, it is predicted that machines will execute tasks that pose a threat to human.



Casualty Category	Non-OCO	OCO	Sum of Category	Category (as % of Total)
Accident	4,827	478	5,305	31.9%
Self-inflicted	3,579	284	3,863	23.2%
Illness / Injury	2,791	119	2,910	17.5%
Killed in Action	6	2,704	2,710	16.3%
Died of Wounds	1	885	886	5.3%
Homicide	479	41	520	3.1%
Undetermined	267	17	284	1.7%
Pending	148	2	150	Less than 1%
Terrorist	18	2	20	Less than 1%
While Captured	0	4	4	Less than 1%
Total	12,116	4,536	16,652	100%

Figure 1.1: U.S. army's death rate in 2006–2018 (*Sources*: U.S. Department of Defemse)

The consciousness and vigilance of people regarding 'social safety' are rapidly increasing worldwide. Various incidents threatening human life and safety, such as natural disasters and terrorism, are commonplace. In particular, the Covid-19 pandemic, which has continued since 2019, is increasing the human desire for safety. If the application of drones is extended to private sectors to aid human life (medical delivery) or to non-combat missions (landmine removal, chemical detection), it would be possible to reduce human casualties tremendously, similar to the case of combat drones. The drone routing and scheduling models studied in the field of Operation Research can be applied to effectively use drones in areas that directly pertain to social safety. For the past two years, we have been working on the application of drone routing models for 'social safety' in two fields.

The first is study of a medical testing kit delivery model that uses drones and a truck, designed to ensure 'social distancing' and lower the risk of infection. The threat of Covid-19 still persists. Additionally, there is an occurrence of a third-wave of infections due to mutated virus strains. The current face-to-face diagnosis method results in increased infections due to the exposure of infected patients to non-infected personnel. Therefore, we propose a system that will allow the patients to diagnose themselves while being in isolation at their home by delivering the recently developed at-home testing kits using drones. The application of this method can be extended for the diagnosis of several other infectious diseases as well.

The second is study on a model that inspects old and large structures using a drone. With the increase in incidents of building collapse due to natural disasters, safety management of these old and large structures has become a major concern. Even now, inspections using drones are partially being performed, but the short duration of drone batteries is a major problem. In particular, for this mission, the battery duration can be shorter because the drone is required to propagate within a three-dimensional space. Therefore, we propose a path planning model that comprehensively considers the various energy consumption patterns of drones. This can provide drone users with a practical solution.

Figure 1.2: Utilization of drones for delivery and inspection (*Sources*: Robotics Automation)

Many drone routing models and optimization models for drone scheduling have been developed mainly to reduce operating costs or total time required. The two optimization models proposed in this study are also designed based on objective functions to minimize total delivery time and battery consumption. However, these models are designed to ensure ‘social distancing’, or to perform tasks that are dangerous for humanbeings. Therefore, these models ensure human safety rather than just being a utility, as observed in previous optimization models.

1.2 Problem Description

For the delivery of testing kits, the key is to develop a routing model that minimizes human contact. Furthermore, for the inspection of large structures, the drone is required to propagate the horizontal and vertical space. Hence, it is crucial that the battery consumption rate is considered in the routing and scheduling model, which may vary accordingly.

1.2.1 Medical testing kit delivery for 'Social distancing'

Patients with suspected Covid-19 symptoms may be crowded in diagnostic places such as hospitals or 'Drive-thru' testing site. Despite the enhanced disinfection, the risk of infection persists in these places. In the case of 'MERS' (Middle East Respiratory Syndrome) and 'COVID-SARS', which are viruses of the same family as COVID-19 virus, there have been several cases of infection in hospitals [2]. For example, 186 ecases of 'MERS' in Korea in 2015, of which 99 % (185 cases) were infected in hospitals [3]. The data from the International Council of Nurses (ICN) estimated that more than 4-million healthcare workers worldwide are infected by the Covid-19 [4]. In other words, testing sites where a huge number of suspected patients are presented may trigger another case of infection.

If the testing kits are delivered to the patient's home using drones, it can prevent people with suspicious symptoms from crowding in a certain location. In addition, if the drone can take off and land from a mobile base such as a vehicle to overcome the drone's short battery duration, it will increase the delivery coverage and shorten the delivery time. Fortunately, various delivery models have been proposed that operate drones based on trucks. Large companies like Amazon have already introduced this system. However, when developing a medical testing kit delivery model, one of our particular considerations is minimizing the likelihood of infection. For example, in the flying-side kick delivery model of Murray *et al*. (2015), both drones and trucks participate in delivery missions [5]. In this case, the truck driver and the patient may come in contact at the point where the truck is serviced. Accordingly, social distancing is not guaranteed. To address this flaw, A ‘drone + vehicle’ scheduling model focused on social distancing should be developed to maintain the physical distance between the patient and the non-patient. With this system, if patients with suspected symptoms are quickly diagnosed and quarantined, it can prevent the spread of the coronavirus.

1.2.2 Drone path planning for structure inspection

As awareness regarding the safety of structures is increasing, countries around the world are reinforcing policies to manage old structures. Many countries, including Japan, have mandated safety inspections for large structures at least once a year in preparation for natural disasters such as earthquakes [6]. In addition, many experts in the field of ‘civil Engineering’ predict that the number of large structures that have been built for over 40 years will gradually increase. Therefore, the necessity for safety inspection of old and large structures will continue to increase. Until recently, these safety checks were mainly performed by technicians, who sometimes had to climb to the top of the structure or had to hang on a steep railing to inspect the structure closely, relying only on poor safety devices. According to the U.S. Department of Labor, 4,179 accidents occurred during structural inspection in the country from 2017 to July 2020. This means that at least 1,000 technicians are killed or injured every year during structural inspection in the U.S. [7]. Owing to such job risks, the number of technicians in this field is rapidly decreasing. Accordingly, in recent years, drones have been recognized as a substitute to perform their role.

In particular, as ‘hyper-spectral lenses’ and ‘Lidar’ (light detection and ranging), equipment are gradually being miniaturized and their performance is being improved, it is now possible to mount them on drones. Considering the typical payload (approximately 1 *kg*) of commercial small drones, the weight of these devices (infrared camera: 250–400 *grams*,

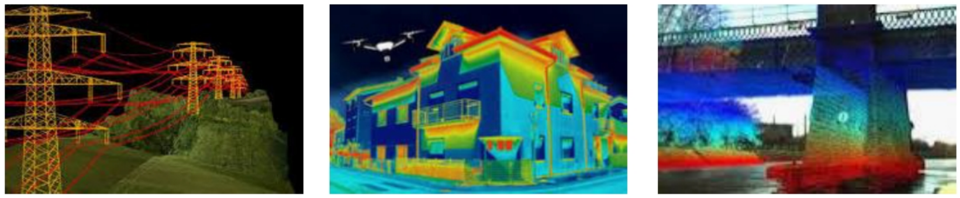


Figure 1.3: Structure inspection images using hyper-spectral lens/infrared cameras/Lidar

Lidar: 680–800 *grams*) is within the range that can be handled by drones. As shown in Figure 1.3, when a structure is inspected using a hyper-spectral lens or Lidar, it is possible to detect corrosion inside a structure, which is impossible to detect with the naked eye. In other words, drones have the ability to inspect structures with greater accuracy and speed compared with that of manual inspection performed by humans.

However, despite these advantages, a few experts in the field still believe that the small battery capacity of drones is a major obstacle while performing this task [8]. For example, the average time required for a drone to scan a blade of a wind turbine is approximately 15 *minutes*. Considering the average duration of a small drone's lithium battery (approximately 20$\sim$25 *minutes*), this inspection time can be covered as long as the drone battery is sufficiently charged. Nevertheless, battery issues are constantly being raised in this field. To inspect a structure, drones have various flight statuses. They fly horizontally for several meters, then ascend and descend vertically. At certain instances, they have to turn frequently to reach the checkpoints distributed throughout the structure and hover for a few seconds to closely scan a specific area. In addition, because this task is mainly performed outdoors, the drone may face strong winds or unstable atmospheric conditions. Due to the above-mentioned factors, even if a drone flies at the same time or at the same distance, its battery consumption may increase exponentially depending on the flight status or atmospheric conditions. Therefore, we need to understand the factors that affect the drone's battery consumption rate. Therefore, we have developed a practical energy-aware routing model which considers the factors that affect the battery consumption of a drone.

1.3 Contributions

In this thesis, we aim to address the challenges that are explained in the previous sections. First, we propose a Covid-19 diagnostic system to ensure maximum social distancing using the 'drone + vehicle' delivery scheduling model. Second, we discuss a path planning considering drone's energy consumption patterns for inspecting large structures using drones.

As for the first contribution, we aim to provide a new method for delivering testing kits using drones and vehicles to suppress the Covid-19 pandemic. The proposed method is a Mothership and multi-Drone Routing Problem (MDRP) that can be applied for the delivery and collection of medical testing kits (samples). In this context, we contribute to the literature by the following

- To propose a concept of a new way to test COVID-19; Drone & truck-aided testing kit delivery blocks the possibility of human contact that may occur during the diagnosis process. This suppresses the spread of the Coronavirus
- To provide decision makers with an epidemic analysis tool ('R_0'method) [9] for using drone and truck in the testing kit delivery: It proves that this method has a significantly lower risk of infection (Effective Reproduction Number) than other testing methods.

As for the second contribution, we provide an approach to inspect old-and-large structures using drones with low energy consumption and shorter flight distance. To provide a bi-objective route that satisfies the various preferences of drone users, we ultimately develop a bi-objective routing model that gives various weights to two criteria (total flight distance and total battery consumption). In this context, we contribute to the literature by the followings:

- To propose a energy-aware routing model that considers various factors affecting battery consumption. Hence, the solution is more realistic for practical use.

- To explore the difference in optimizing drone path between the goals of 'minimum flight distance' and 'minimum battery consumption'. We demonstrate that minimizing drone flight distance can lead to infeasible solutions due to dynamic battery consumption rates under various circumstances mentioned in this paper.

- To provide a bi-objective optimization model that considers different decision maker's preference on battery consumption and the flight distance.

1.4 Organization

The remainder of this thesis is organized as follows. In Chapter 2, we review the relevant literature on 'drone + vehicle' scheduling problems for delivery purposes. Among these models, we explore the most suitable method or model to implement a delivery routing model that guarantees social distancing (Section 2.1). In Section 2.2, we review existing energy consumption models to estimate total amount of energy that a drone consumes for a particular distance or flight duration and energy-aware routing problems in the OR field.

In Chapter 3, we propose a drone+vehicle aided delivery routing model based on CVP-D (Carrier-Vehicle Problem with drones). In this model, multiple drones will be used to shorten the delivery time duration, and the vehicle will act as a mothership to prevent human contact. In a case study based on real road networks and patient information in parts of South Dakota, we explore a solution that covers patients in minimal time (Section 3.5). In addition, to analyze the risk of infection of this system, we predict the 'basic reproduction number' during the delivery time duration through the 'R-number(R_0)' method. A comparison of the proposed method with that of face-to-face diagnostics proves that this system is superior in terms of social distancing (Section 3.4).

In Chapter 4, we propose an energy-aware routing model for inspecting large structures. First, we review various factors that affect the battery consumption rate of drones, and develop a battery consumption model for each flight condition that comprehensively considers these factors (Section 4.3). Based on these models, the MIP model that minimizes the total flight distance and the total battery consumption are independently formulated (Section 4.4). If the optimal solutions of these two models are different, and these two models do not have a proportional relationship in the feasible area. Hence, we perform a bi-objective optimization to find a route that compromises these two criteria (Section 4.5). Finally, we conclude the thesis and discuss related future researches in Chapter 5.

Chapter 2

Literature Review

2.1 Medical testing kits delivery model for 'Social distancing'

2.1.1 A review on existing drone routing problems for delivery

There have been recent studies on mathematical models for using drones to deliver medical kits and packages to customers. The drone delivery routing problems can be classified into two categories (see Figure 2.1): (1) delivery by both trucks and drones and (2) delivery by drones only.

The main idea of this problem is to utilize both trucks and drones in direct delivery. Optimization models are often a variant of travelling salesman problem (TSP) or vehicle routing problem (VRP) depending on the number of trucks to be used in the problem. In TSP with drones (TSP-D), the delivery schedule is optimized for a combination of one truck and multiple drones. Similarly, the VRP with drones (VRP-D) is to optimize the delivery schedule utilizing multiple trucks and multiple drones. Common objectives in the literature include minimizing the duration of operation (i.e., makespan) routing and operation cost, and customers waiting time. Because this class of problem is identified as *NP-hard*, exact solution techniques are only used to solve small problems. But, most solution approaches are largely heuristic methods to handle larger scale problems providing good solutions in a

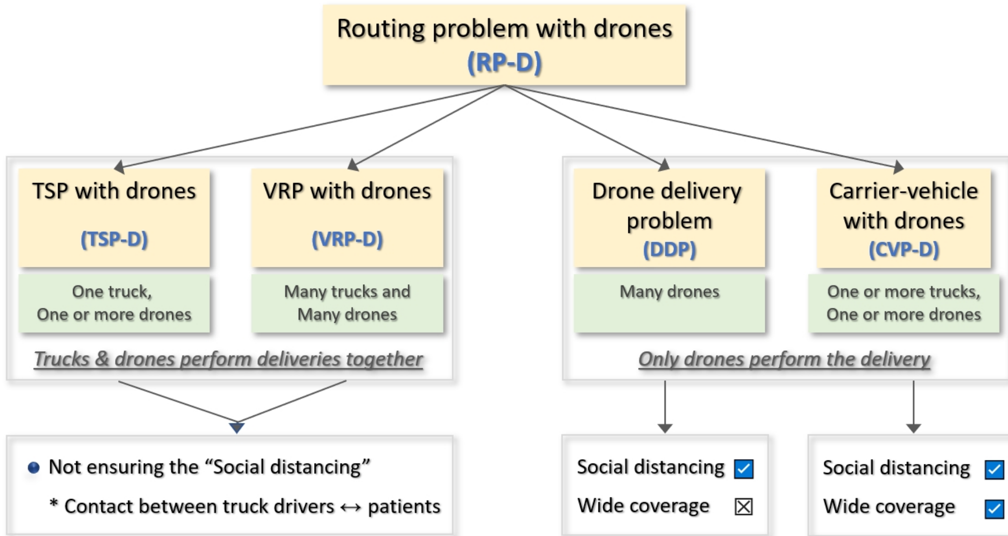


Figure 2.1: The categorization of drone routing problem by [1].

timely manner [1].

Delivery by drones only: This category of problems use drones only for parcel delivery, and there are two versions of models in the literature. The first version schedules drone flights from-and-to a depot. Kim *et al.* (2018) propose a model in which drones are utilized to deliver medical kits from fixed locations of ground control center (GCC) for patients with chronic diseases in rural area [10]. Due to the limited battery capacity of drones, their model suggests using a larger number of drones from multiple GCCs. Alternatively, the short flight range limitation from a GCC can be addressed by operating drones from a mobile control center such as a truck. It can also help ensure social distancing during a pandemic because people can stay at home to get the testing done. The main idea of the Carrier-Vehicle Problem with Drones (CVP-D) is that the mobile base (e.g. truck) serves as a 'mothership' that collects and dispatches drones, and multiple drones perform delivery. Hence, it requires a fleet of cooperating vehicles with complementary functions that perform autonomous delivery [1]. Table 2.1 presents main features of CVP-D found in the literature. In the majority of the works the fleet is composed of one truck and one or

Table 2.1: Review on existing CVP-D delivery models

References	Trucks	Drones	Objcetive function (To minimize..)	Time-Windows	Drop & Pickup	Multiple visits
Savuran *et al*. (2016)[11]	1	1	targets number	No	No	Yes
Mathew *et al*. (2015) [12]	1	1	total distance	No	No	No
Othman *et al*. (2017) [13]	1	1	total distance	No	No	No
Gambella *et al*. (2018) [14]	1	1	completion time	No	No	No
Boysen *et al*. (2018) [15]	1	multiple	make span	No	No	No
Dukkanci *et al*. (2019) [16]	multiple	multiple	operating cost	Yes	No	No
Karak *et al*. (2019) [17]	1	multiple	operating cost	No	Yes	Yes
Wikarek *et al*. (2019) [18]	multiple	multiple	total distance	No	Yes	Yes
Bai *et al*. (2019) [19]	1	1	completion time	No	No	No
Poikonen *et al*. (2020)(a) [20]	1	1	completion time	No	No	No
Moeini *et al*. (2019) [21]	1	multiple	total distance	No	No	No

several drones. Some papers consider multiple trucks carrying multiple drones, some deals with delivery time window, and others consider pick-up operations.

Variants of the CVP-D have been proposed for pickup and delivery. For example, Karak and Abdelghany (2019) propose a model in which a drone is mounted on a single vehicle to visit more than one customer to deliver and pickup parcels. Drones can be dispatched and collected multiple times, And each customer can only be visited once. The truck returns to the depot after collecting all drones which completed the delivery missions. In another CVP-D variant by Wikarek *et al.*(2019), drones can only be launched or retrieved from trucks at predefined mobile distribution centers where the launch and retrive locations are allowed to be different. Their model focused on optimizing drone routing and the location of mobile distribution centers. But, truck routing was not discussed. Mathew *et al.* (2017) presented a delivery system in which trucks carry drones and the delivery is carried entirely by the on-board drones. A drone performs a single delivery mission to a specific point.

Some researchers proposed drone routing approaches based on a predetermined route of the carrying truck (Savuran and Karakaya (2020), Bin Othman *et al.* (2017) and Boysen *et al.* (2018)). Typically, a large vehicle is used as mobile depot for a single drone onboard. Unlike Mathew et al (2017), drones are allowed to visit multiple locations. With the goal of visiting as many locations as possible, Savuran and Karakaya (2020) proposed a genetic algorithm to solve the model. Bin Othman *et al.* (2017) allowed drones to take off from an intermediate node and get picked up by a truck at another location. The problem was modelled as a graph problem and a polynomial-time approximation algorithm was developed as a solution method. Boysen *et al.* (2018) focuses on optimizing the schedule of drones. In this problem, since the path of the truck is known in advance, the author searches for an optimal schedule based on the number of drones available. The authors examined three truck-drone operation policies: same take-off and pick-up location, different locations for takeoff and pickup of drones, and no restriction on the locations.

Poikonen and Golden (2019) consider the problem of optimizing both the truck routes and the drone paths. In particular, two scenarios are considered in their study. In the first scenario, a drone carries one package at a time, and the second scenario considers a drone making multiple deliveries per each flight. The authors proposed a branch-and-bound algorithm to solve solve small-size instances, and heuristic methods such as a greedy algorithm and local search strategies for larger instances. As an extension (Poikonen and Golden (2020)), drones are allowed to visit multiple customers once taking off. Moreover, in this model, the energy consumption rate is considered to account for the limited capacity of a drone battery. And heuristic of "path, transform, shortest path" was proposed to solve the problem.

Bai *et al.* (2019) introduced precedence constraints to keep track of the order of the customers to be served with one truck equipped with one drone. They proposed several algorithms and discussed computational performance of their approaches compared to genetic algorithms. As a variant to the CVP-D, Moeini and Salewski (2020) introduced a problem in which the carrier is equipped with two different types of delivery vehicles: drones and autonomous vehicle (ATV). The carrier visits a designated point only once to dispatch and collect drones and ATVs. Each drone and each ATV perform a single delivery at a time. A genetic algorithm was used to solve the problem.

Although the existing approaches work well for their respective problems, they do not consider social distancing as we discuss in this paper. Hence, this paper builds on the existing literature to solve a problem of delivering medical kits to patients to minimize human contacts while coping with a pandemic.

2.2 Drone path planning for structure inspection considering energy consumption patterns

2.2.1 Factors affecting battery consumption of drones

In this section, we review related literature to identify the factors that influence drone's battery consumption. Before reviewing the various factors influencing the drone's battery consumption, it is important to understand what kind of physical forces act on the drone and how the drone controls its rotors for each flight dynamics (see Figure 2.2). In order for a quadrotor-type drone to hover without moving in any direction, the four rotors rotate at the same speed [22]. If the four rotors increase the speed at the same rate, it rises to a higher altitude. Conversely, it descends when the rotational speed of the rotors is lowered equally. Also, it must generate thrust in order to move in any direction in the air. In this case, the two rotors located in the direction of movement of the drone reduce the rotational speed, and the two rotors located in the opposite direction maintain or increase the speed. In this way, the drone can move in any horizontal direction. Drag is also one of the forces included in predicting the battery consumption of a drone. When a drone is flying in the atmosphere, the drone's flight is hindered by the density or flow of air such as wind. This force that hinders the flight of drones is the drag force.

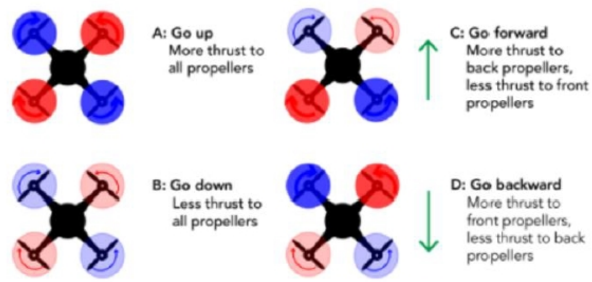


Figure 2.2: Principle of drone flight (*sources*: Pixabay)

According to Demir *et al.* (2014), the main factors influencing drone's battery consumption can be categorized into four categories: drone design, environment, drone mechanics, and delivery operations [23]. Since this study specifies the situation in which drones inspect structures, we exclude the 'delivery operation' item of these four factors from consideration. The main elements of the rest three categories are shown in Figure 2.3. Drone design elements include the weight and size of the drone, number and size of rotors, weight, size and energy capacity of batteries, power transfer efficiency, maximum speed and payload, lift-to-drag ratio, transmission and avionics. Environmental factors include air density, gravity, wind conditions, weather (snow, rain, etc.), and air temperature. Drone dynamics factors include drone movement speed, drone motion (e.g., takeoff/landing, hovering, horizontal flight), acceleration/deceleration, attack angle and flight altitude.

2.2.2 Literature review on battery consumption models

In this section, we review several papers that studied the correlation between the battery consumption rate of drones previously reviewed and the factors that influence it. We searched articles in journal papers, proceedings, book chapters, etc. using 6 keywords such as 'drone(UAV) energy', 'drone(UAV) battery' and 'drone(UAV) aerodynamics' in google scholar. As a result, we were able to find the following 11 studies directly related to our

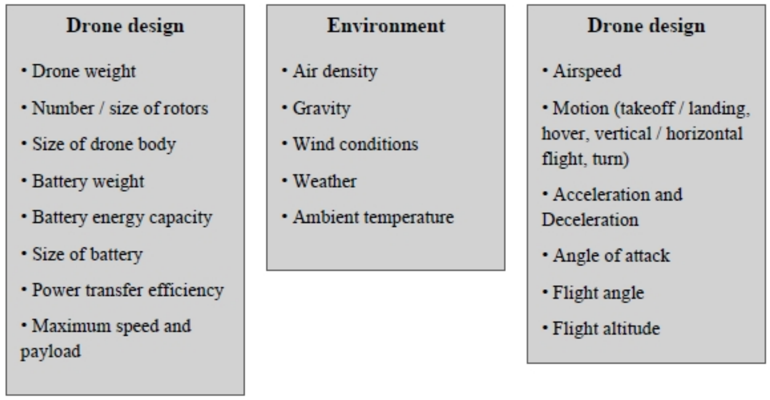


Figure 2.3: Factors that affect energy consumption of drones (adapted from Demir *et al*. (2014))

study, which measured or predicted the battery consumption rate by drone dynamics and design, or environmental factors.

As shown in Table 2.2, some studies have identified the correlation between the battery consumption rate by selecting one of the drone's dynamics, environmental factors, and design [24][25][26][27][28][29][30]. Other studies have developed a battery (energy) consumption model that comprehensively considers these factors[31][32]. In particular, since the mid-2010s, large companies such as 'Amazon' have been promoting the delivery business using drones, and the public's interest has increased. Naturally, studies have also been conducted on other battery consumption factors that should be considered when a drone performs delivery. In Table 2.2, 7 out of 11 studies developed battery consumption models for delivery drones. For example, the battery consumption rate when a drone loads a parcel and moves to the destination and the battery consumption rate when returning after unloading a parcel may differ due to differences in payload. These are defined as concepts such as 'Empty return' or embodied as 'm_1 (aircraft weight) + m_2 (payload weight)', and based on this, battery consumption models were established [29][30][31][32].

As explained in Section 2.2.1, drones can change their battery consumption rate by 'drag force' in addition to 'thrust forces' or 'lift force'. There are 4 main factors that cause

Table 2.2: Studies on Energy(battery) consumption model of drones

height	Purpose	dynamics						Environmental Factors		Design				Field test	Drone type	Energy(Battery) Consumption model	
		Speed	Motion					wind	Air density	Drone size	Payload	Empty return	Drag coefficient			Theoretical model	Regression model
			Horizon	Vertical	diagonal	hover	turn										
D'Andrea (2014)	delivery	x	x					x							Not specified	x	
Franco et al (2016) -	Surveillance	x	x	x		x	x							x	Quad-rotor		x
Dietrich (2017)	Energy Profiling			x	x						x			x	Quad-rotor		x
Figliozzi (2017)	Monitoring air pollution	x	x								x	x			Quad-rotor	x	
Lohn et al.(2017)	delivery	x	x		x	x		x			x	x			Quad-rotor	x	
Xu (2017)	delivery	x	x		x	x		x			x	x			Quad-rotor	x	
Felismina et al (2017)	agricultural Application	x	x					x	x	x	x		x	x	Octo-copter		x
Chen et al.(2018)	delivery	x	x			x		x	x		x				Quad-rotor	x	
Stolaroff et al (2018)	delivery	x	x					x			x	x		x	Octo-copter	x	
Torabbeigi et al (2019)	Delivery	x	x								x			x	Quad-rotor		x
Kirschstein (2020)	delivery	x	x		x	x		x	x		x	x		x	Octo-copter		x

drag. Among environmental factors, 'wind' and 'air density' are related to drag force. And among the drone design elements, drone's area (which is in direct contact with the atmosphere when flying) and the drag coefficient are related to drag force. There were some papers that studied the relationship between environmental factors such as wind and battery consumption. On the other hand, few studies have investigated the relationship between drone design and battery consumption. Felismina *et al.* (2018) directly measured the 'drone's area' in direct contact with the atmosphere and the 'drag coefficient' by scanning the shape of the drone [33]. And, they found that the air contact area and drag coefficient may vary depending on the drone's inclination (see Section 4.3.2), which can occur depending on the horizontal flight speed.

Meanwhile, studies focused on finding the correlation between drone dynamics and battery consumption rate directly measured battery consumption after establishing a controlled experimental environment [24][25][30][33]. A regression analysis model was also created based on directly measured battery consumption data. On the other hand, it is quite difficult to artificially create environmental factors. For example, there may be a slight difference in air density depending on the flight altitude, and even if an artificial wind is created, the wind cannot be blown evenly across the entire flight altitude of the drone. Accordingly, when developing a correlation model between environmental factors and battery consumption, a theoretical model was mainly developed [28][31][32].

2.2.3 Review on existing 'Energy-aware' routing models

In this section, we review the energy-aware routing models developed so far. Early drones were primarily used for military purposes, and because these drones were large in size and powered by fuel rather than batteries, drones did not need to be severely limited by flight duration [34]. However, as commercial small drones became popular, and especially as big companies such as Amazon began to use commercial small drones for their product delivery, studies on energy aware-routing models gradually began to be published [5]. We searched related literature on Google scholar by combining a total of 6 keywords follows; 1) ‘drone’, 2) ‘UAV’, 3) ‘energy-aware’, 4) ‘battery’, 5) ‘routing’, and 6) ‘path planning’. And we found 8 journal papers, proceedings that were directly related to this field. And we summarized the models proposed in their study as shown in Table 2.3.

Like the previous studies on energy (battery) consumption models, energy-aware routing models in the OR field were mainly designed for delivery[24][35][36][37]. Some of the routing models for delivery even considered battery consumption due to vertical movement during takeoff and landing[35]. However, most of the models were for drones to cover a 2-dimensional horizontal space with a minimum distance (time or cost), and accordingly, dynamic factors other than the horizontal speed of the drone were not thoroughly considered [24][25][28][36][37][38][39]. There was also an energy-aware routing model for structure inspection, but this study rather focused on finding a battery saving route considering weather factors rather than various dynamics of drones[40]. In addition, although a existing study in Section 2.2.2 showed that the drone area and drag coefficient change depending on the design of the drone, especially inclination [33], this could cause a difference in the battery consumption rate, but no studies have considered these in the routing model. Instead, the most frequently considered drone design element for these energy-aware routing models was the payload, which is related to ‘lift’ [24][37]. Therefore, those routing models designed to focus on payloads, ‘drag force’ is not thoroughly considered.

The classification according to the objective function is as follows. 3 out of 8 studies have established routing models that minimize ‘total battery consumption’ or ‘total operating cost’ based on data on battery consumption rates by dynamics or environmental factors [24][25][38]. The other three models have an objective function that minimizes the total flight time (distance) covering all points by roughly converting the battery consumption rate data into time or distance [37][40][39]. Some models have an objective function that

Table 2.3: Studies on Energy-aware routing models for drones

height Authors	Purpose	Battery(Energy) consumption factors considered				Area for Flight mission		Objective Function (Z = Min)		Optimization Model	
		Dynamics	Weather Factors	Design	Operational Characteristics	2D horizontal	3D space	Total battery consumption (or Cost)	Total Flight distance (or time)	Robust Optimization	Data-Driven Optimization
Guerrero et al.(2013)	Structure inspection		x				x		x	x	
Franco et al.(2016)	for Surveillance	x			x	x		x			x
Dorling et al.(2017)	for Delivery	x					x	x	x	x	
Tseng et al.(2017)	Energy Profiling	x	x			x			x		x
Troudi et al.(2018)	for Delivery	x			x	x		x	x	x	
Kim et al.(2018)	for Surveillance				x	x		x		x	
Torabbeigi et al.(2019)	for Delivery	x		x		x		x			x
Jeong et al.(2019)	for Delivery	x		x		x			x	x	

minimizes total flight distance (hours) and battery consumption at the same time [35][36]. However, these differ from our research in two respects: 1) These are models that are implemented to cover a 2-dimensional horizontal space, and focus on finding a route that consumes minimum energy by drag force during horizontal flight without considering ‘vertical flight’ and ‘hovering’. 2) These studies include a function that minimizes the total flight distance (time) and a function that minimizes the total battery consumption by drag in a single objective function from the start. Therefore, these studies neither compare routes with minimum total flight distance (hours) to routes with minimum total battery consumption, nor do bi-objective optimization to find a route that compromises these two criteria. Kim *et al.* (2018) proposed an energy-aware routing model based on a new concept called ‘E-line’ that allows drones to charge batteries while flying. However, strictly speaking, this routing model focuses on ‘where to recharge the battery’ rather than ‘taking the route to further reduce battery consumption’. In this context, this model takes a different approach from our research direction[38].

In summary, existing energy-aware routing models, including routing models for inspection of structures that must cover 3-dimensional space, have not comprehensively considered these battery consumption factors. Hence, more practical routing model for structural inspection needs to be developed. Furthermore, a bi-objective solution may also need to be provided to allow users the flexibility to choose routes that focus on reducing total flight distance, or focusing on total energy consumption, depending on their preferences.

Chapter 3

Medical testing kit delivery strategy using a truck and drones under pandemic situation

3.1 Introduction

Although the world's largest pharmaceutical companies are developing and distributing COVID-19 vaccines and treatments the spread of pandemic still persists due to personal contacts. In a situation where it is difficult to expect immediate results from vaccines or treatments, countries are establishing various policies to slow the spread of infection.

As part of those policies, health authorities in many countries have set up drive-thru testing sites to cover the enormous demand for diagnostics. And the principle diagnostic method to confirm COVID-19 cases at drive-thru testing sites is by a molecular diagnostic kit such as Polymerase Chain Reaction (PCR). However, there are two considerations with this approach. The first is the slow diagnostic speed. It takes about 6 hours to determine its negative/positive after collecting the sample [41]. Furthermore, it normally takes 2 to 5 days to receive the results. If the patients who have COVID-19 symptom are not quarantined and contact other people during this waiting time, the spread of the virus will not be prevented. Secondly, the PCR testing kit requires a sterile plastic swab to be inserted deep into the nose to collect sputum as a sample. Since the sample must be taken by someone else, this method can cause personal contacts.

In recent, the U.S FDA approved a testing kit that allows patients to self-test for the coronavirus at home. In particular, this testing kits allow patients to take samples by themselves and to check the results directly within 30 *minutes*. If suspicious patients do not go directly to purchase testing kits and a system is built in which they are delivered to their homes, it can dramatically reduce the chance of infecting others.

In this study, we propose a strategy to deliver COVID-19 self-testing kits to patients' homes using a truck and drones. Hence, potential patients can use at-home testing kits to determine whether they are infected with Coronavirus while social distancing is assured as much as possible. The primary focus of this study is to lower the infection rate by avoiding unnecessary personal contacts to receive the COVID-19 diagnosis.

The rest of this study is organized as follows. Section 3.2 describes the problem in detail and presents the mathematical models applied to find a solution. In Section 3.3, solution techniques to solve the MDRP (Mothership and Drones Routing Problem). Section 3.4 describes an analysis tool for numerically measuring the risk of infection in the process of each diagnostic method. In Section 3.5, numerical experiments are performed to illustrate the effectiveness of the proposed model. And numerical results for infection risk will also be presented in this section. Finally, Section 3.6 concludes with a discussion of the highlights in this paper and the potential extension of this work.

3.2 Problem Description and Formulation

To achieve social-distancing, and address patients' demand in the fastest possible way, we need to develop a delivery system, in which delivery only happens by drones (not the truck) and no direct human to human interactions are assured. Due to limited battery capacity of drones, the flight range of a drone is limited; therefore to reach some of patients, truck must carry the drones at some points, until the farthest patient(s) fall into drone's flight range. Moreover, the number of drones are limited and therefore the maximum number of parallel flights are bounded to the available drones. The truck, as a mobile launch and return station for drones, must be scheduled in a way that to be present at start and end of all flights. By considering these goals and limitation, the delivery system turns into a 'Mothership and Drones Routing Problem' (MDRP).

We developed a model (see Model (3.1–3.26)) by making some assumptions using similar quadrotor type of drones that can fly up to 30 *mph* and pickup truck sized vehicles can drive normally 20 *mph* considering traffic. The assumptions are as follows:

- The flight duration is not affected by the amount of load to be carried by a drone. Since most testing kits are lightweight, the payload has minimal impact on the flight time. The driving time is not affected by the number of drones or testing kits loaded.
- Truck can stop and wait at intermediate nodes (I) only.
- The drones can be collected at intermediate nodes (I) only.
- Setup time for a drone to start a flight includes setting the itinerary and loading the test-kits. Retrieve time includes battery swap and sanitizing the drone. All the setup and retrieve actions are done by the operators accommodating in the truck.
- Any flights launch from depot and end with it will use drones from depot and they will not use the ones mounted in the truck.
- The drones are equipped with proper devices to land and pickup items as intended, and the truck also has proper devices to take-off and land the drones.
- The starting time of operation is at time 0.
- The parameters of the model are constant during the planning and operation timeline.

Table 3.1: Sets, parameters, and variables that are used in Mothership and drone routing problem

Sets	
I	Set of the intermediate nodes on the truck network. C as the depot (center) belongs to intermediate nodes. $C \in I$.
A_t	Set of arcs representing highways,streets, ...
N	Set of Patients
A_d	Set of all possible flight arcs.
Parameters	
$G_t(I, A_t)$	The directed network graph of truck
$G_d(\{N \cup I\}, A_d)$	The directed network graph of flights
$wt_{i,j}$(min)	Travel time of truck traversing arc $(i, j) \in A_t$
$wd_{i,j}$(min)	Flight time of drone traversing link $(i, j) \in A_d$
s_i(min)	Service time at location of patient i.
n	Total number of drones
m	Total number of operators performing launch and retrieve
$W_{\max}$(min)	Maximum waiting time
$H_{\max}$(min)	Maximum hovering time
δ(min)	Setup time for a drone(loading testing kits, etc.)
θ(min)	Retrieve time for a drone.(sanitizing, and changing the battery)
M	A very big number
$R_{\max}$(min)	A Maximum flight time of drone
Variables	
$y_{i,j} \in \{0, 1\}$	If truck chooses link $(i, j) \in A_t$ to traverse it gets 1 otherwise 0.
$x_{i,j} \in \{0, 1\}$	If drone chooses link $(i, j) \in A_d$ to flight it gets 1 otherwise 0.
$v_i \in \mathbb{Z}_+^0$	Order of visiting node $i \in I$ in truck path.
$u_i \in \mathbb{Z}_+^0$	Order of visiting node $i \in I \cup N$ in flight path.
$w_i \in \mathbb{R}_+^0$	Waiting time of truck at node $i \in I$.
$h_i \in \mathbb{R}_+^0$	Waiting time of drone (hovering) over node $i \in N$.
$T_i \in \mathbb{R}_+^0$	Time of entrance of truck at node $i \in I$.
$t_i \in \mathbb{R}_+^0$	Time of entrance of drone at node $i \in I \cup N$.
$b_i \in \mathbb{R}_+^0$	Remaining flight range, at the time of reaching node $i \in I \cup N$.
$D_i \in \mathbb{Z}_+^0$	The number of drones mounted in truck after leaving node $i \in I$.

Sets, parameters, and variables of MDRP are shown in Table 3.1. The objective of model, (3.1), is to minimize the total truck operation and flight time, and defined as:

$$\min_{x,y,w,h,t,T,b,u,v,D} \quad Z = \sum_{i,j \in A_t} wt_{i,j} y_{i,j} + \sum_{i,j \in A_d} wd_{i,j} x_{i,j} + \sum_{i \in I} w_i + \sum_{i \in N} (h_i + s_i). \tag{3.1}$$

And the constraints are divided into three categories: Constraints 1) for drones, 2) for a truck, and 3) for mutual operation. The constraints for a truck are as follows:

$$\sum_{(i,C) \in A_t} y_{i,C} = \sum_{(C,j) \in A_t} y_{C,j} = 1, \tag{3.2}$$

$$\sum_{(i,j) \in A_t} y_{i,j} = \sum_{(j,i) \in A_t} y_{j,i} \quad \forall j \in I \backslash \{C\}, \tag{3.3}$$

$$and \quad v_i - |I|(1 - y_{i,j}) + 1 \leq v_j \quad \forall (i,j) \in A_t, j \neq C. \tag{3.4}$$

Constraint (3.2) makes sure that truck starts and ends its journey from and to the depot. Constraint (3.3) conserves the flow of truck for the chosen nodes on the truck path. Constraint (3.4) prevents the appearance of sub-tours as the truck's solution:

$$T_i + w_i + wt_{i,j} \leq T_j + M(1 - y_{i,j}) \quad \forall (i,j) \in A_t, \tag{3.5}$$

$$T_i + w_i + wt_{i,j} \geq T_j - M(1 - y_{i,j}) \quad \forall (i,j) \in A_t, \tag{3.6}$$

$$and \quad w_i \leq W_{\max} \quad \forall i \in I. \tag{3.7}$$

To make truck's timeline, (3.5) and (3.6) will calculate the arrival time of truck at the visiting nodes. (3.7) does not allow truck to wait more than maximum waiting time at any intermediate node. Next, the constraints for drones are as follows:

$$h_i \leq H_{\max} \quad \forall i \in N, \tag{3.8}$$

$$\sum_{(i,j)\in A_d} x_{i,j} = \sum_{(j,i)\in A_d} x_{j,i} = 1 \quad \forall j \in N, \tag{3.9}$$

$$1 - |N|(1 - x_{i,j}) \leq u_j \quad \forall (i,j) \in A_d,\ i \in I, \tag{3.10}$$

$$\mathit{and} \quad 1 + u_i - |N|(1 - x_{i,j}) \leq u_j \quad \forall (i,j) \in A_d,\ j \notin I. \tag{3.11}$$

Constraint (3.8) enforces any drones to limit the hovering time over the patient's location to be less than maximum waiting time. The reason of adding maximum hovering time for drones and waiting time for truck is to create a synchronization between truck and drones, so truck can collect the drones at the intermediate nodes. (3.9) guarantees that only one drone can visit each of the patients. (3.10) and (3.11) act as sub-tour eliminator for flights.

$$b_j + s_i + h_i + wd_{i,j} - M(1 - x_{i,j}) \leq b_i \quad \forall (i,j) \in A_d,\ i \notin I, \tag{3.12}$$

$$b_j + s_i + h_i + wd_{i,j} - M(1 - x_{i,j}) \leq R_{\max} \quad \forall (i,j) \in A_d,\ i \in I, \tag{3.13}$$

$$t_i + s_i + h_i + wd_{i,j} \leq M(1 - x_{i,j}) + t_j \quad \forall (i,j) \in A_d,\ i \notin I, \tag{3.14}$$

$$\mathit{and} \quad t_i + s_i + h_i + wd_{i,j} \geq t_j - M(1 - x_{i,j}) \quad \forall (i,j) \in A_d,\ i \notin I. \tag{3.15}$$

Constraints (3.12) and (3.13) will not allow drones to fly more than their maximum flight range ($R_{\max}$). Constraints (3.14) and (3.15) create the timeline of drones flights. Next, the constraints for mutual operation are as follows:

$$T_i - M(1 - x_{i,j}) \leq t_i \leq T_i + w_i + M(1 - x_{i,j}) \quad \forall (i,j) \in A_d,\ i \in I, \tag{3.16}$$

$$\mathit{and} \quad T_j - M(1 - x_{i,j}) \leq t_j \leq T_j + w_j + M(1 - x_{i,j}) \quad \forall (i,j) \in A_d,\ j \in I \backslash \{C\}. \tag{3.17}$$

To synchronize the movement of truck and drones in launch and return nodes, (3.16) and (3.17) assure the presence of truck for the starting and ending flight's nodes. Constraints (3.18) to (3.23) conserve the flow of the drones in the G_t (#drones in-going = #drones out-going in each intermediate node):

$$D_i - D_j - M(1-y_{i,j}) \leq \sum_{(j,ii)\in A_d} x_{j,ii} - \sum_{(ii,j)\in A_d} x_{ii,j}, \forall((i,j)\neq C)\in A_t, \tag{3.18}$$

$$D_i - D_j + M(1-y_{i,j}) \geq \sum_{(j,ii)\in A_d} x_{j,ii} - \sum_{(ii,j)\in A_d} x_{ii,j}, \forall((i,j)\neq C)\in A_t, \tag{3.19}$$

$$D_i + \sum_{(ii,C)\in A_d} x_{ii,C} \leq n + M(1-y_{i,C}), \quad \forall(i,C)\in A_t, \tag{3.20}$$

$$D_i + \sum_{(ii,C)\in A_d} x_{ii,C} \geq n - M(1-y_{i,C}), \quad \forall(i,C)\in A_t, \tag{3.21}$$

$$D_C + \sum_{(C,jj)\in A_d} x_{C,jj} \geq n - M(1-y_{C,j}), \quad \forall(C,j)\in A_t, \tag{3.22}$$

$$and \quad D_C + \sum_{(C,jj)\in A_d} x_{C,jj} \leq n + M(1-y_{C,j}), \quad \forall(C,j)\in A_t. \tag{3.23}$$

As the final set of constraints, (3.24) and (3.25) prevent any launch and return from unvisited nodes by truck. (3.26) assures that the setup time for and retrieve time for arrivals do not exceeds the allotted waiting time in any intermediate node:

$$\sum_{(i,k)\in A_d} x_{i,k} \leq n \sum_{(i,j)\in A_t} y_{i,j} \quad \forall i \in I, \tag{3.24}$$

$$\sum_{(k,j)\in A_d} x_{k,j} \leq n \sum_{(i,j)\in A_t} y_{i,j} \quad \forall j \in I, \tag{3.25}$$

$$and \quad \frac{\delta}{m} \sum_{(i,j)\in A_d} x_{i,j} + \frac{\theta}{m} \sum_{(j,i)\in A_d} x_{j,i} \leq w_j \quad \forall j \in I. \tag{3.26}$$

3.3 Solution Technique

The MDRP is an *NP*-hard problem because a simpler version of the problem, a modified Location Routing Problem, is known to be *NP*-hard [17]. Therefore, most exiting literature on such problems often resort to developing computationally efficient algorithms to find good feasible solutions. In this paper, we propose a decomposition method coupled with fast heuristic algorithms to expedite the solution process. In the decomposition scheme, the original problem is decomposed into a truck schedule problem and a flight schedule problem so that each problem can be solved much faster.

3.3.1 Decomposition scheme

The MDRP constraints can be categorized into three groups:

- Truck related: Constraints (3.2) to (3.7) are only related to design trucks schedule, independent from drones.
- Drone related: Constraints (3.8) to (3.15) manage the flight schedule of drones, independent from truck.
- Mutual: The group of constraints, (3.16) to (3.26), coordinates the truck and drone schedule and enables the truck and drones to collaborate.

Based on the categories, we can transform the model into a bi-level model structure:

TS: The upper level problem is to optimize the 'truck schedule'. Based on the sequence of serving patients, which is determined at the lower level, the goal is to find an optimal schedule for the truck and assign launch-return locations for drones. Thus, the optimization model (see Section 3.3.1.1) contains 'Truck related' constraints as well as 'Mutual' constraints of MDRP. Note that the 'Drone related' constraints are already satisfied for at this stage.

FS: For a given truck schedule, the lower level problem is optimize the drones' 'flight schedule'. The model (see Section 3.3.1.2) contains 'Drone related' and the 'Mutual' constraints only.

Our approach recursively solves TS and FS, in which a solution from one helps improve the solution for the other. This iterative process continues until no better solution is found.

3.3.1.1 Optimize Truck Schedule for a given sequence of flight sequence

Sets, parameters, and variables are presented in Table 3.2.

Table 3.2: Sets, parameters, and variables that are used in Truck schedule optimization

Sets	
SP	Set of sequences of serving the patients. $SP = \{SP^i \vert SP^i \subset N,\ \forall i,j\ SP^i \cap SP^j = \phi,\ \bigcup_{i=1}^{\vert SP\vert} SP^i = N\ \ i = 1,\ldots,\vert SP\vert\}$
LFR	Set of possible flights within the maximum flight range. $LFR = \{(l,i,r) \vert F^i_{l,r} = \{l\} \cup SP^i \cup \{r\},\ l,r \in I,\ t^i_{l,r} \leq R_{\max}\}$
Parameters	
$t^i_{l,r}$(min)	The flight duration of $F^i_{l,r} \in LFR$ including service, flight, and hovering time.
Variables	
$f^i_{l,r} \in \{0,1\}$	If a drone assigned to serve SP^i, chooses node $l,r \in I$ as its launch and return nodes, then it gets 1 otherwise 0.

For a given sequence of patients to visit, the objective of TS is to find a shortest truck path with complete drone flights along the path. A complete drone flight schedule includes the sequence of patients and the launch ($l \in I$) and return ($r \in I$) locations to join the truck along the truck path. The sequences already determined the values of $x_{i,j}$ where $(i,j) \in \{(i,j) \vert (i,j) \in A_d,\ i,j \in N\}$ [1]. However, the variables regarding to links that completes the

[1] if i,j are consecutive therefore $x_{i,j} = 1$ otherwise 0.

flights are still unknown and via transforming them to variable f, TS model will finalize the flights for all drone flights along the truck path (See Equation (3.27)).

$$SP^i = c^i_1 \rightarrow c^i_2 \rightarrow ... \rightarrow c^i_{|SP^i|} \tag{3.27.1}$$

$$f^i_{l,r} = (x_{l,c^i_1})(x_{c^i_{|SP^i|},r}) \tag{3.27.2}$$

$$t^i_{l,r} = wd_{l,c^i_1} + \sum_{j=1}^{|SP^i|-1} wd_{c^i j, c^i j+1} + wd_{c^i_{|SP^i|},r} + \sum_{j \in SP^i} (h_j + s_j) \tag{3.27.3}$$

The sequences of patients to be visited already determined by MDRP model. However, the variables regarding to links that completes the flights from a intermediate node are still unknown. By converting the links to variable f, we can find feasible intermediate nodes that link with a patient, and this process is implmented by following Equations:

$$SP^i = c^i_1 \rightarrow c^i_2 \rightarrow ... \rightarrow c^i_{|SP^i|}, \tag{3.28}$$

$$f^i_{l,r} = (x_{l,c^i_1})(x_{c^i_{|SP^i|},r}), \tag{3.29}$$

$$and \quad t^i_{l,r} = wd_{l,c^i_1} + \sum_{j=1}^{|SP^i|-1} wd_{c^i j, c^i j+1} + wd_{c^i_{|SP^i|},r} + \sum_{j \in SP^i} (h_j + s_j). \tag{3.30}$$

In this way, TS model will finalize all flight path along the truck path. For the TS model, new parameter $t^i_{l,r}$ is introduced to determine $f^i_{l,r}$. This parameter denotes the drone's total flight duration which includes the travel time(w_d), service time(s), and the waiting time for a truck at a node (h) and Equation (3.30) is used to calculate the value of $t^i_{l,r}$. By integrating those parameter and variables, which were independently denoted in the MDRP model, into a parameter called $t^i_{l,r}$, it helps reduce a considerable number of variables and constraints in MDRP. therefore, TS is much easier to solve than the original MDRP model. The goal of TS is to build the truck path and drone flight schedule to minimize the total flight time for drones and travel time for trucks (3.31). The objective function is defined as follows:

$$\min_{y,f,w,T,D} \quad TS = \sum_{i,j \in A_t} wt_{i,j} y_{i,j} + \sum_{(l,i,r)\in LFR} t^i_{l,r} f^i_{l,r} + \sum_{i \in I} w_i. \tag{3.31}$$

And, following constraints are for drone's flight path:

$$\text{(3.2)-(3.8)}, \tag{3.32}$$

$$T_r - T_l - w_l \leq t^i_{l,r} + M(1 - f^i_{l,r}) \quad \forall (l,i,r \neq C) \in FLR, \tag{3.33}$$

$$t^i_{l,r} - M(1 - f^i_{l,r}) \leq T_r + w_r - T_l \quad \forall (l,i,r \neq C) \in FLR, \tag{3.34}$$

$$and \sum_{\substack{(l,ii,r)\in FLR, \\ ii=i}} f^i_{l,r} = 1, \; i = 1, ..., |SP|. \tag{3.35}$$

Constraints (3.32) are adopted from Section 3.2 and they are creating flight paths that flow at each path is conserved and they are subtour free. The constraints (3.33) and (3.34) are to synchronize the drone's departure and return to join the truck based on the truck schedule. If the return location is the depot, the delivery mission is complete. Constraint (3.35) to ensure that there will be one launch node (location) and one return node for the sequence.At last, (3.36) to (3.43) are the updated version of (3.18)-(3.26) due to variable f as follows:

$$D_l - D_r \leq \sum_{(r,i,j)\in LFR} f^i_{r,j} - \sum_{(k,i,r)\in LFR} f^i_{k,r} + M(1 - y_{l,r}) \; , \forall (l,r) \neq C \in A_t, \tag{3.36}$$

$$D_l - D_r \geq \sum_{(r,i,j)\in LFR} f^i_{r,j} - \sum_{(k,i,r)\in LFR} f^i_{k,r} - M(1 - y_{l,r}) \; , \forall (l,r) \neq C \in A_t, \tag{3.37}$$

$$D_l + \sum_{(l,i,C)\in FLR} f^i_{l,C} \leq n + M(1 - y_{l,C}), \quad \forall (l,C) \in A_t, \tag{3.38}$$

$$D_l + \sum_{(l,i,C)\in FLR} f^i_{l,C} \geq n - M(1 - y_{l,C}), \quad \forall (l,C) \in A_t, \tag{3.39}$$

$$D_C + \sum_{(C,i,r)\in FLR} f^i_{C,r} = n, \tag{3.40}$$

$$\sum_{(l,i,r)\in FLR} f^i_{l,r} \leq n \sum_{(l,j)\in A_t} y_{l,j} \quad \forall l \in I, \tag{3.41}$$

$$\sum_{(l,i,r)\in FLR} f^i_{l,r} \leq n \sum_{(k,r)\in A_t} y_{k,r} \quad \forall r \in I, \tag{3.42}$$

$$and \quad \frac{\delta}{m} \sum_{(l,i,j)\in FLR} f^i_{l,j} + \frac{\theta}{m} \sum_{(j,i,r)\in FLR} f^i_{j,r} \leq w_j \quad \forall j \in I. \tag{3.43}$$

3.3.1.2 Optimize Flight Schedule based on a given truck schedule

For a given truck schedule, the lower level problem is to optimize drone flight schedule following the tuck path. Using sets, parameters, and variables are defined in Table 3.3.

Table 3.3: New or modified sets, parameters, and variables for Flight schedule optimization

Sets	
P	Itinerary of truck. $P: C \rightarrow p_2 \rightarrow \ldots \rightarrow p_{\|P\|-1} \rightarrow C$
A'_d	$\{(i,j) \mid i \neq j \in P \cup N,\ i \& j \notin P\}$
Parameters	
$G'_d(\{P \cup N\}, A'_d)$	The new network graph of drones based on new truck path (P). G'_d is a subgraph of G_d.
w_{p_i}(min)	The allotted waiting for truck in node $p_i \in P$
T_{p_i}(min)	Entrance time of truck at node $p_i \in P$. This value includes all the travel and wait times till the node p_i. $T_{p_{i-1}} + w_{p_{i-1}} + wt_{p_{i-1},p_i} = T_{p_i}, i = 2, \ldots, \|P\| - 1$

In the optimization model, the objective is to minimize the total flight time of drones to complete all visits along the truck path and it is defined as follows:

$$\min_{x,h,t,v,D,b} \quad FS = \sum_{i,j \in A'_d} wd_{i,j} x_{i,j} + \sum_{i \in N} h_i. \tag{3.44}$$

Constraints (3.44) are adopted from Section 3.2 and they are related to truck's scheduling. In particular, Constraint (3.26) assures that the setup time for take-off and retrieve time for arrivals do not exceeds the allotted waiting time in any intermediate node. Constraints (3.46) and (3.47), secure the presence of truck, when the drone(s) launches and returns:

$$(3.8)-(3.15),(3.26), \tag{3.45}$$

$$T_{p_i} - M(1 - x_{p_i,j}) \leq t_{p_i} \leq T_{p_i} + w_{p_i} + M(1 - x_{p_i,j}) \quad \forall p_i \in P, \forall (p_i, j) \in A'_d, \tag{3.46}$$

$$T_{p_j} - M(1 - x_{i,p_j}) \leq t_{p_j} \leq T_{p_j} + w_{p_j} + M(1 - x_{i,p_j}) \quad \forall p_j \in P, \forall (i, p_j \neq C) \in A'_d. \tag{3.47}$$

Constraints (3.48) and (3.49) based on the available drones (idle ones in truck + returned ones) determine the number of launching drones. These are the equivalent of (3.18) to (3.23), when the path of truck has already been determined.

$$D_{p_{i-1}} - D_{p_i} = \sum_{(j,p_i) \in A_d} x_{j,p_i} - \sum_{(p_i,j) \in A'_d} x_{p_i,j} \ , i = 2, ..., |P|, \tag{3.48}$$

$$and \quad D_C + \sum_{(C,i) \in A'_d} x_{C,i} = D_{p_{|P|-1}} + \sum_{(j,C) \in A'_d} x_{j,C} = n. \tag{3.49}$$

3.3.2 The Proposed Solution approach

The general procedure to solve MDRP using the bi-level framework discussed in Section 3.3.1 is shown in Algorithm 1, which contains three stages:

1. **Initialization:** The main purpose of this stage, is to create a very good and fast warm-up solutions for both TS and FS, and therefore save a considerable amount of time for solver to come up with a feasible solution. As the initial solution, we assembled one patient flights and their launch and return point are depot (F_{-1}). The set of flights may be infeasible due to violation of flight range and availability of drones. To avoid infeasibility, the 'TSP_{BFS}' (see Section 3.3.2.1) will design a truck path (P_0) that guarantees the feasibility of one patient flights. In the next step, "Exchange" algorithm takes action. This heuristic algorithm is deliberately designed for producing shorter and feasible flights (F_0). The idea is to consolidate and exchange patients between any two pair of flights so the new set of feasible flights has shorter flight length. Since in the next stages, the MIP models of TS and FS will be used, we utilized two common termination criteria embedded in solver, which are time and optimality gap. The initial values for them are determined at the end of this stage.

2. **Heuristic-Exact Loop:** The TS replaces the 'TSP_{BFS}' to find a better truck path. The 'Exchange' algorithm is used for scheduling drone flights. The idea is to find an optimal truck path using the solutions provided by Exchange algorithm.

3. **Final Loop:** As the final attempt to reach optimality, the TS and FS will optimize recursively until no further improvement are possible. Any former feasible solution will be used as the 'warm-up' solution to expedite the solution time.

The proposed algorithm contains two termination criteria:

- **Optimality gap percentage:** All the solver has optimality gap criterion, which at any stage of solving are achieved by upper-bound (UB) and lower-bound (LB) (see Equation(3.50)). For a typical MIP minimization, the UB is the best primal objective value and LB is the maximum objective value of dual of relaxed MIP in the Branch&Bound algorithm. The optimality happens when the gap is zero.

$$gap = \frac{UB - LB}{UB} * 100\% \tag{3.50}$$

  In the ideal way, 'TS' and 'FS' must communicate with each other simultaneously, and any slight improvement at each of the levels must be reflected into other one, however, to find practical solution time, we need to compromise with bigger gap reduction such as 5%. At any steps, since the solutions are not final; therefore letting each level to reach its optimality gap while the other side is not optimal may disturb the optimality of the other level.

- **Time:** Due to magnitude of each level, especially 'FS'[2], the solver may take a lot of time to reach the termination gap. In order to enhance solving time, we add time criterion, so the heuristics enhance the UB and therefore the gap.

3.3.2.1 Heuristic

In order to overcome the hardship of solving TS and FS, and creating the closest upper-bound the the optimal solution, we designed three heuristic algorithms as follows.

[2]The G_d is almost a complete graph and for medium set of patients, it gets days to reach 10% gap.

Data: $G_t(I, A_t), G_d(I \cup N, A_d), wd, wt, s, N, I, n, M$
Result: Optimal Truck path (P^*), and Flight paths (F^*)
\# **Initialization**;
$F_{-1} = \{\{C, i, C\} | i \in N\}$;
$P_0, F_{-1} = TSP_{BFS}(F_{-1})$;
$F_0 = Exchange(P_0, F_{-1})$;
$gap_0 = gap_{max}\%$;
$tm = tm_{max}$";
\# **Heuristic-Exact Loop**;
while $gap >= 0$ **do**
    $P_i,\ F_i = TS(P_{i-1},\ F_{i-1})$;
    if *TS does not reach the* gap_i *in* tm *unit of time* **then**
        $P_i = TSP_{BFS}(F_i)$
    end
    $F_i = Exchange(P_i, F_i)$;
    $i+ = 1$;
    $gap_i = gap_{i-1} - 10\%$;
end
\# **Final Loop**;
while *TS or FS provides new solution* **do**
    $P_i,\ F_i = TS(P_{i-1},\ F_{i-1})$;
    $P_i,\ F_i = FS(P_i,\ F_i)$;
    $i+ = 1$;
end

Algorithm 1 The general procedure of Solving MDRP

Launch-Return Optimizer (LRO): For a given truck path (P) and sequence of patients (SP)[3], the goal of LRO is to find drone flight schedule to minimize the total flight time. The LRO is a simplified MIP model of "TS", where the truck path is known. The output of this optimizer will be F (Complete flights) and δ (the indicator of optimality/feasibility status). The following needs a revision for clarification: The LRO is to ensure that the output of the heuristics are feasible to TS and FS. If the LRO in anyway can not find a feasible solution then it will output δ as 0 otherwise it will output 1. The LRO model is presented in Appendix A.

TSP with Breadth First approach (TSP_{BFS}): In this heuristic, we developed an iterative algorithm that at each iteration solves a modified travelling salesman problem(mTSP). The objective of this algorithm is to build a feasible and shortest truck path (cycle) that contains a subset of set of I and LRO results feasible flights (see Algorithm 2). The TSP is modified

[3]It also accepts F as input. LRO transforms the F to SP by dropping the launch and return nodes from the flights.

since a subset of I must be visited (MV) by truck and visiting the rest is optional. On the other hand, we combined the idea of the ranking, offered by Breadth First Search tree (BFS), to get the shortest truck path. BFS ranks the nodes based on their closeness [4] to the root. By running BFS on G_t and considering the depot as root, we will get a list of nodes that are sorted by their closeness to the depot. The list will give us the order of adding nodes to "Must Visited"(MV) nodes for mTSP. The algorithm expands the MV until the LRO gives a feasible solution.

Data: $G_t(I, A_t), SP, wd, wt, s, N, I, n, M$
Result: feasible truck path (P), and feasible flight paths (F)
\# initialization;
$R = BFS(G_t, C)$# Run BFS and R will the sorted set of nodes based on their closeness to depot(node C);
$MV = [C]$;
R.remove(C);
$P = mTSP(MV)$;
$F, \delta = LRO(P, SP)$;
\# Main Loop ;
while $\delta = 0$ **do**
    MV.append($R[1]$);
    R.remove($R[1]$);
    $P = mTSP(MV)$;
    $F, \delta = LRO(P, SP)$;
end

Algorithm 2 The procedure of modified TSP with respect to ranking achieve by BFS search tree

Exchange between Flights One of the common heuristic approaches in routing problem is the local search method. The goal of local search is to find the best savings by exchanging nodes to enhance the length of routes gradually. However, if the size of the routes are not too big, then the optimal reassignment of the nodes to paths can happen very fast.

In our situation, we want to use the optimal reassignment for any two pair of flight paths to lower the total duration of two flights. Algorithm 3 shows the procedure of the Exchange

[4] Unit of distance is the number of nodes

algorithm. Each flights, due to limited flight range, contains four patients in average; therefore the reassignment will solve very fast and performing it many times will not create any computational burden.

Data: $P, F_{\text{old}}, Sq, wd, s, N, I, n, M$
Result: F_{new}
\# Initialization;
\# Main Loop ;
while $\exists f^o, f^k \subset F \ni mFS(f^k, f^o)$ *lowers FS objective value* **do**
    $(f^k_{new}, f^o_{new}) = \text{mFS}(f^k_{old}, f^o_{old})$;
    update F;
end

Algorithm 3 The Exchange algorithm to improve flight time

Now, for the reassignment of two flight paths $f^k,\ f^o$, we are using a modified version of FS (mFS). The difference between FS and mFS, is the variables regarding to any flights belong to $F' = \{f^q | f^q \subset F, q \neq k, o\}$ will be predetermined[5]. Therefore, a huge reduction in solution space of FS will happen and due total number of patients in $f_k,\ f_o$, the mFS will solve very fast. The mFS($f^k,\ f^o$) are shown in Model (3.51-3.53).

$$\min_{x,h,t,v,D,b} \quad mFS(f^k, f^o) = \sum_{i,j \in A'_d} wd_{i,j} x_{i,j} + \sum_{i \in N} h_i, \tag{3.51}$$

$$\text{s.t.} \quad (3.45) - (3.49), \tag{3.52}$$

$$\textit{and} \quad x_{f^q_i, f^q_{i+1}} = 1, \quad i = 1, ..., |f^q| - 1, f^q \subset F'. \tag{3.53}$$

[5]The values of $x_{i,j}$ - as the main variables of flights- will be determined in the way that constructs all the flights in F'

3.4 Methods for estimating the risk of infection of proposed model

The testing kits delivery and at-home diagnosis system significantly reduces the chances of human contact compared to the existing face-to-face diagnosis. Therefore, we expect this method to contribute significantly to preventing the spread of Covid-19. We use the 'R-naught(R_0)' method to show the proposed method lowers the risk of Covid-19 infection rate compared to face-to-face diagnostic method. The 'R-naught(R_0)' method estimates the 'Basic reproductive number' which refers to the number of total infection case by one infected patient during the average recovery period of a virus. In the case of Covid-19, it has not yet been completely ended, and since many mutations of the virus are occurring, the exact 'R' number cannot be predicted, but according to studies to date, it has a value between about from 1.4 to 5.5. In other words, one virus carrier is spreading the virus to about 1.4 to 5.5 others during the average recovery period. This virus has a significantly stronger contagion than the 'R' number of MERS (<1) or Ebola ($1.5 \sim 2.5$) [42].

$$R = \beta S/\gamma, \tag{3.54}$$

$$and \quad \gamma = \frac{\text{Average recovery period of a virus } (in\ days)}{\text{total time spent for an activity } (in\ days)} \tag{3.55}$$

The Equation (3.54) is the most traditional and basic equation for calculating 'R' number. 'R' number is determined by the transmission rate of a virus (β), the number of contacts with the infected person (S), and the period which the infected person can spread the virus (γ). However, this formula cannot be used to selectively calculate the risk of infection in a specific region or at a specific time [43][44]. And more recently, the basic formula has been transformed into various forms so that we can calculate the 'R' number for a specific activity, or a specific time period [45][46]. Among them, to predict the 'R' number over a specific time period during an activity, some researchers have modified the 'γ' as shown in equation (3.55). In this study, we use this modified termination(γ) to calculate the 'R' number for the time duration that drone and truck complete delivery for 'n' patients.

3.5 Numerical Results

This section is divided into two parts. The first part tests the proposed models on a sample network. The algorithms and mathematical models are implemented in Python 3.7 [47] and Gurobi 9.1.0 (Python API)[48]. Based on the optimal solution derived from the first part, 'R' number analysis is presented to prove that the testing kit delivery method has a significantly lowers level of infection risk than other diagnostic methods as the second part. All experiments are made on a server running RedHat Linux 64-bit with 15 core Intel Xeon processors and 16GB RAM.

3.5.1 Numerical Results for Drone and Truck scheduling

Numerical example used to illustrate the models proposed in Section 3 is a real-world example based on the city Sioux Falls, South Dakota, U.S (see Figure 3.1). There are a total of 40 patients (Yellow circles with number 1 to 40) to be served by 3 drones in the same service schedule. Actual patient locations can be slightly different due to patient information confidentiality. And the routes that a truck can use are based on the existing road network (Black lines).

The truck must stop to take off the drone loaded with the testing kits or to pick up the drones that has finished the delivery mission. However, due to road conditions, the number of points that the truck can stop is limited to a total of 24 (Red circles with alphabet 'A' to 'X'). The drone can fly for a maximum of 26 *minutes* at an average flight speed of 20 *mph*, assuming that the atmospheric conditions are stable during delivery. The truck can run for

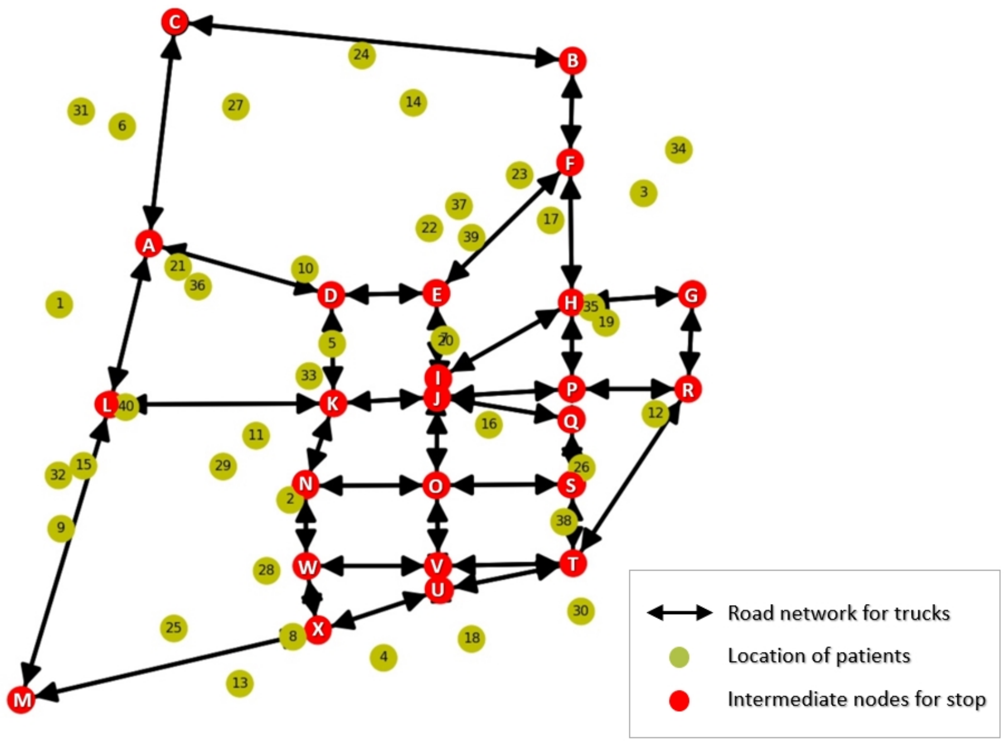


Figure 3.1: Numerical Example with 40 patients and the road network(G_t)

200 *minutes* at an average speed of 20 *mph* considering the traffic conditions. We assumed that the service time per patient is up to 3 *minutes* taking into account the time required for unloading the testing kit, etc. And the waiting time for drones landing at each truck stop is up to 5 *minutes*.

The optimal routes of a truck and drones for serving 40 patients are shown in Figure 3.2 (Black arrows : Truck's path, Green arrows : Drones' Path). The three drones take off from the truck at one intermediate node and land on the truck at the other intermediate node. The number of patients being served in a path varies with the drones and truck's specification(i.e., maximum flight time and truck's stopping point). Then, the three drones return to the depot(the red circle with 'c') rather than the truck after performing their final delivery mission from intermediate nodes 'A' and 'L' respectively.

The patients (stop points) assigned to each drone and a truck, and the traveling order

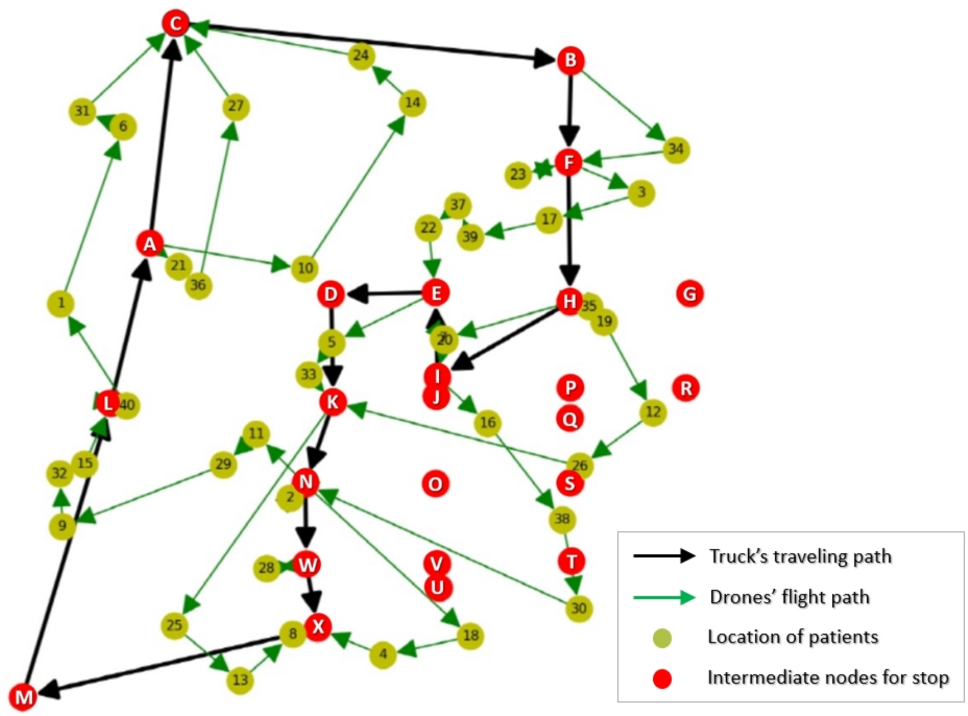


Figure 3.2: The assignment of drones to patients and optimal flight schedule

are described more clearly in Figure 3.3. The truck stops at a total of 13 points using the existing road network. At these stopping points, the truck either picks up the drone that has finished the delivery mission or takes off the drone waiting for the mission. In summary, it takes a total of 98 *minutes* to deliver the testing kit to 40 patients with a total of 3 drones. Each drone has a flight schedule of 1 to 3 times.

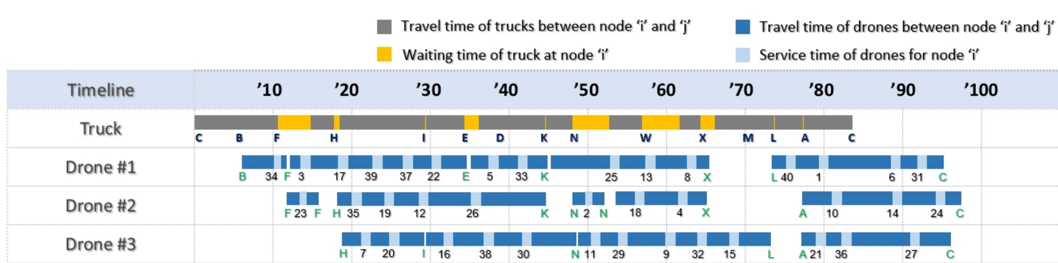


Figure 3.3: Optimal Schedule for Truck and Drones

3.5.2 Computational results

One of the important aspects of a heuristic method is the quality of the solution that it provides. For a small number of patients (for e.g., 20) due to magnitude of G_d, FS will solve in non-realistic time and as we discussed in Section 3.3.2.1, and we use *Exchange* as its substitution. Figure 3.4 shows the process of solving MDRP via Algorithm 2 and 3 for a sample problem with 20 patients and 3 drones. At time zero ($t = 0$), TSP_{BFS} has already proposed its solution, which is compatible with the 1-patient flights. As the next step, *Exchange* ($Exchange^1$) merges the flights and therefore the number of flights will be reduced to 25 *percent*. This reduction results in smaller size of TS and as appeared in Figure 3.4, TS reaches gap of 20*percent* in less than 180 *seconds*.

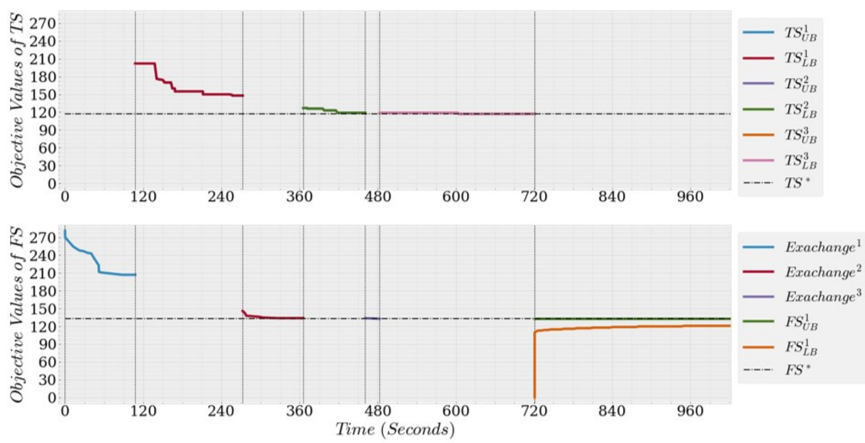


Figure 3.4: The convergence of TS and FS to optimal solution

The second round of $Exchange^2$, shows a good improvement and allegedly optimality in FS. This shows the impact of the solution of TS on FS (compare the final value of $Exchange^1$ and the beginning value of $Exchange^2$). This improvement demonstrates the value of information awarded to FS, by reassigning a new set of launch and return nodes to each *SP*. On the other side, FS also affects the TS; due to improvement by $Exchange^2$, the reduction in upperbound of TS is not negligible (see the final value of TS^1_{UB} and initial value of TS_{UB}^2 in Figure 3.4). This indirect improvements, is the privilege of using the decomposition and the recursive optimization, in which by improving in one level, the other level without any extra effort will be improved automatically, and prevents unnecessary computational efforts. For this example, the TS and FS reach optimality, in less than 8 minutes. The rest of computation timeline ($0 \sim 3600$ *seconds*) is served to improve the lower-bound of the models, and since none of the models provides the new information, therefore the algorithm stops.

3.5.3 Numerical Results for 'R_0' analysis

We calculated the 'R_0' of the 'drive-thru' test method together and used it as a comparison object to prove that the risk of infection of our testing kit delivery system is significantly lower. According to Chu *et al.*(2020) if two people wearing masks talk to each other within 6 feet, the transmission rate(β) was 3.1 *percent*, and when the both people did not wear masks, the transmission rate(β) was increased to 17.4 *percent*[49]. Actually, during the test for COVID-19 with face-to-face method, patients should take off the mask for collecting the sample, but sampling time is very short and they do not have any conversation during sampling. Therefore, we assumed that the transmission rate of the face-to-face test is 0.031. On the other hand, when a testing kit is delivered with a drone, human contact does not occur. If the surface of a drone or testing kit is contaminated with the virus, it is possible to spread the virus by contact with it, but it is extremely low compared to the transmission rate by droplets. Although no research has been conducted on this, we conservatively assume the transmission rate of 0.01 due to contact with contaminated drones or testing kits.

And the 'S' (the number of people at risk of infection) of both testing methods is assumed as follows. In the drive-thru testing site, 40 patients and 10 medical staff (5 staffs per testing booth for interviewing, collecting samples, etc.) included in the 'S'. In the delivery method, a total of 42 person are included in 'S', considering 40 patients, drone operator (1

Table 3.4: 'R_0'(Basic Reproduction Number) for each testing system

	β **(Transmission rate)**	S **(no of person)**	γ **(Time period)**	R_0
Face-to-face test (Drive-thou sysem)	0.031	50	101.82 (14 / 0.137)	0.0153
Testing kit delivery using drones&truck	0.010	42	205.73 (14 / 0.068)	0.0020

person), and truck driver (1 person). 'γ' is the average recovery period(in days) divided by total testing (or delivery) time (in days). In other studies, many researchers assumed that the average recovery period for COVID-19 was 14 days. And in the case of a face-to-face test, it takes about 10 *minutes* to collect a sample per patient. If there are about 2 test booths at the testing site, the total time required for 40 patients is 3.3 *hours*, this is the same as 0.137 in days. And dividing this value by 14 days, the average recovery period of COVID-19, the 'γ' value will be 101.82. For testing kit delivery, the minimum required time we found is 1.63 hrs (0.0681 days). Accordingly, 'γ' for this method will be 205.73.

Consequently, when 40 patients were diagnosed by the 'drive-thru' method, 'R_0' is 0.0153. This means that during the face-to-face test of 40 patients, about 0.0153 new infection cases could be occurred. In other words, this means that if a drive-through test is performed on approximately 2,614 patients, one another positive case of infection can occur. However, when the testing kit is delivered to the same number of patients, the 'R_0' is further lowered to 0.002. This means that if the test kit is delivered to about 20,000 patients, one case of indirect infection can occur. Of course, the effective reproduction number of face-to-face test at drive-thru testing site(R_0=0.0153) is very low in terms of infection management. However, considering the current pandemic of tens of millions of confirmed cases worldwide, the proposed testing kit delivery method with a significantly lower infection risk(R_0=0.002) is worth supplementing or replacing existing face-to-face diagnosis method.

3.6 Conclusion

A truck and drones-aided delivery system was studied in this paper, in which a truck was considered as a mothership of drones and the group of drones deliver testing kits for potential patients. The ultimate goal of this study is to prove that this hybrid delivery model is superior in terms of infection prevention compared to face to face diagnosis. In order to find the optimal delivery route, we decomposed the schedule of the drone (FS) and the truck (TS) and proposed a planning optimization model for each. After deriving the optimal traveling path of the truck, the optimal flight route for the drone was determined based on that. In addition, in the solution method, we specified the optimality gap percentage to improve the quality of the solution, and proposed heuristic algorithms to reduce computational time. The numerical results showed that 3 drones and 1 truck could deliver testing kits for 40 patients within 4-*miles* radius in 98 *minutes*.

In order to show the superiority of this method in terms of social distancing, we used 'R' number method. In this method, the total time to complete delivery, the number of people involved in this activity, and the transmission rate of the virus by contact type were considered, comprehensively. In addition, for a clear comparison, we also derived the 'R' number of the face-to-face diagnosis. As a result, 'R' number of the testing kit delivery system (R_0=0.002) was significantly lower than that of the face to face method (R_0=0.0153). This indicates that the proposed system reduces the probability of infection by more than 7.5 times compared to face-to-face diagnosis in a 'drive-thru' testing site.

Covid-19 has greatly increased people's awareness of infectious diseases. Therefore, this testing kit delivery model is not a temporary trend, but can become a 'New Normal' that can be continuously applied to the diagnosis of other infectious diseases such as Flu, MERS, and SARS. In addition, we hope that this research will expand not only testing kits, but also researches to deliver sc-type vaccines to be developed in the future.

Chapter 4

Drone path planning for structure inspection considering energy consumption patterns

4.1 Introduction

Commercial small drones are on the rise for various purposes in recent years, such as surveillance, rescue missions, package delivery, and many other applications [50]. Although features and flight performance of drones are rapidly improving, the short battery duration of small drones remains to be a major drawback for practical use [8]. The drone routing models typically focus on minimizing total flight distance or total flight time so as to perform their missions as efficiently as possible within a limited battery capacity [38]. When the drone is flying only horizontally at constant speed in a stable atmosphere, there is little difference in the energy consumption rate. Therefore, finding a route that minimizes total flight distance can be one way to find an energy efficient solution. However, this is no longer true for a three-dimensional structural inspection because the battery consumption rate (BCR) is different for each flight motion or each environmental condition. Literature shows that the vertical flight of a drone consumes about 20% more energy than when it flies horizontally [24, 25, 51]. The battery consumption rate is also sensitive to the weather condition [51]. The average battery duration can be reduced from $20-25min$ to less than $15min$ depending on the flight pattern and the weather condition. Therefore, it is necessary consider such factors in drone path planning to minimize not only the flight distance (time) but also battery consumption.

Three-dimensional (3D) structure inspection by drones involves frequent vertical and horizontal flights. With the miniaturization of 'hyper-spectral' lenses and 'Lidar'(Light Detection and Ranging), drones can be equipped with these equipment to inspect bridges or wind power generators [52–54]. For a large structure, several checkpoints are distributed on the surface of the structure. If an inspection path is designed without considering the battery consumption patterns by flight dynamics or by environmental factors, the actual battery consumption can be significantly shorter than the planned time to complete the inspection.

To address the shortcoming, this paper explores and compares two path plan optimization strategies to inspect 3D structures. The proposed models consider various factors affecting the battery consumption rate such as drone dynamics (motions, speed), drone design (shape, weight), and environmental conditions (wind, air density). Two mixed integer programming (MIP) models are presented to find a route to minimize the total flight distance (labelled as SFD) and the total battery consumption (labelled as MEC). Furthermore, a bi-objective optimization model is developed to satisfy the user preference between the distance minimization and the battery consumption minimization.

4.2 Research Methodologies

To develop a energy consumption model for estimating total amount of battery consumption of a drone in a certain distance(time), first of all, it is necessary to collect as much data as possible directly measuring or predicting the battery consumption rate of a drone according to dynamics, design, and environmental factors. We find largely consistent and reliable experimental results from vast amounts of data and build battery consumption models based on them.

And then, two-step of operational planning to find optimal inspection route will be conducted (see Figure 4.1). The first step of the OP includes two MIP models. One MIP model aims to minimize the total flight distance of drone during a structural inspection, and it is referred to as ‘SFD’ (the Shortest Flight Distance). The other MIP model seeks to find a route with the minimum battery consumption and it is referred to as ‘MEC’ (the Minimum Energy Consumption). The route with the shortest flight distance and the route with the minimum battery consumption derived by the two MIP models can be different. In particular, these two objective functions may have an ‘inverse’ relationship in the feasible area. In other words, as the total flight distance of the drone increases, the total battery consumption may decrease and vice versa. In this case, a bi-objective optimization model is proposed as the second step of OP to find a trade-off route that satisfies the two goals as much as possible.

There are many kind of bi-objective optimization methods [55]. In the structural inspection, the priority (importance) for ‘the shortest total flight distance’ and ‘the minimum total battery consumption’ may vary depending on the operating conditions and the preferences of the drone user. Therefore, weighted method is applied in the bi-objective optimization model. The weighted method gives various relative weights for each objective function. If we implement a bi-objective optimization model by choosing the more appropriate one of these two methods, we will find solutions on the efficient frontier. And those solutions will be options that drone users can choose to suit their situation.

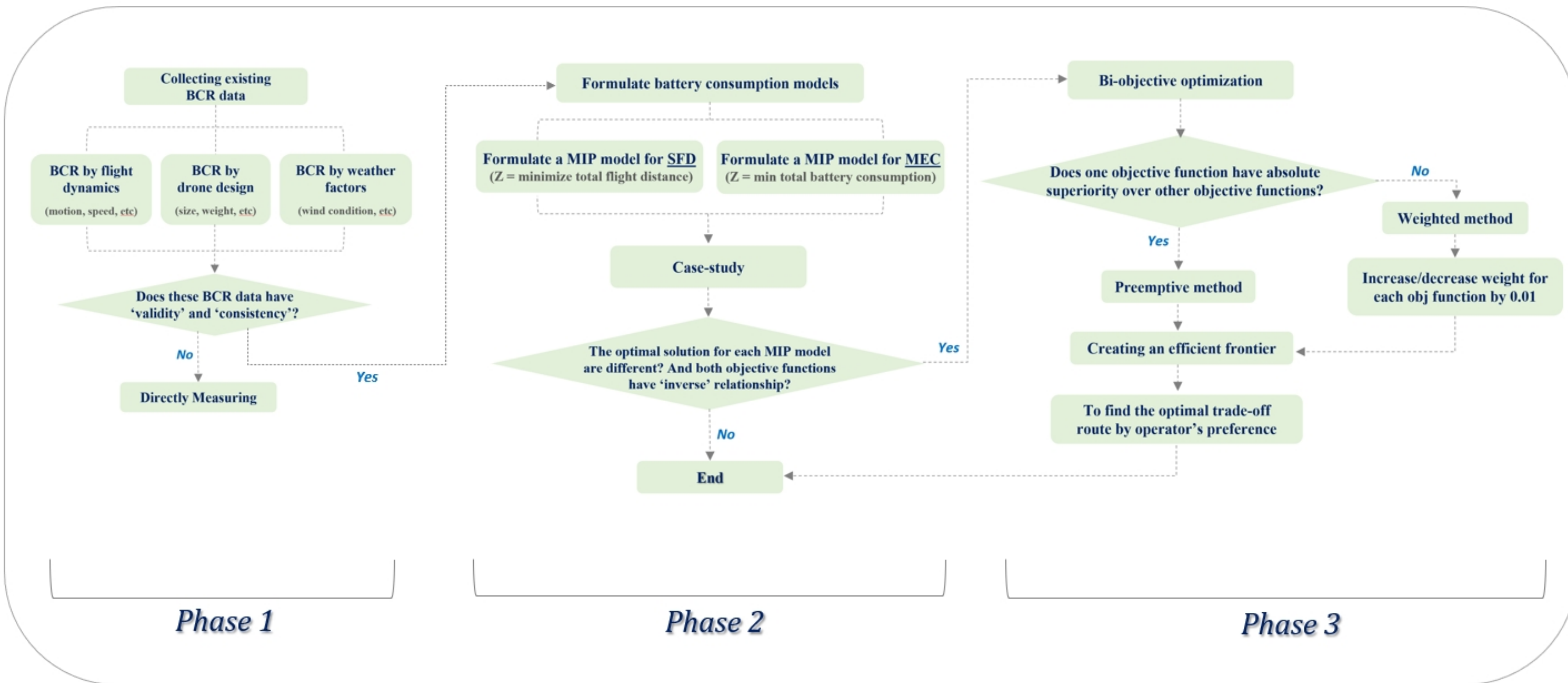


Figure 4.1: Flow chart of two phase Operational Planning

The rest of the chapter is organized as follows. In Section 4.3, we aim to obtain consistent and reliable battery consumption rate data by flight dynamics, drone's design and environmental condition. Based on the data obtained, an equation for calculating the amount of battery consumed for each flight condition under certain weather and flight conditions is formulated. In Section 4.4, two MIP models are developed to find a route with the shortest total flight distance (SFD) and the minimum total battery consumption (MEC), separately. In particular, case studies for inspecting several types of structures are carried out. Section 4.5 describes the process of finding a bi-objective optimal solution that compromises the two optimal solutions found. In addition, we will provide various optimal solutions by conducting computational experiments in different weighted cases. In Chapter 4.6, we conclude the thesis with a summary of our contributions and the direction of future research.

4.3 Battery consumption rate(BCR)

In this Section, we collect data that measures or predicts changes in the battery consumption rate of drones for each of the factors reviewed above, and builds 'Battery consumption models' by selecting data with reliability and consistency. Even in large structures, there are several spots that should be inspected by drones. Accordingly, the drone flies a short distance of 50 *meters* or less, rather than flying a long distance, and when it reaches the inspection point, it hovers there and scans the point for several seconds with mounted equipment. In particular, drones may need to make a large number of turns to move to each point distributed in front of, behind, or on the side of the structure. Therefore, in Section 4.3.1, we describe the battery consumption rate by drone dynamics. Section 4.3.2 reviews the battery consumption rate by drone design and environmental factors. In particular, 16 of the 19 factors which can affect battery consumption rate specified in Figure

2.3 are directly and indirectly included in these two Sections.

The key to this data collection is to provide clear evidence that there is a difference in battery consumption rate depending on the factors. This difference serves as the rationale for proving that the inspection route with the shortest total distance and the route with the minimum total battery consumption in Section 4.4, will be different.

4.3.1 Battery consumption rate by Drone Dynamics

Among the three factors that affect the battery consumption rate of drones, the field in which the most research has been conducted is drone dynamics. Numerous researchers have directly measured the amount of battery consumption while flying a drone to clarify the relationship between drone dynamics and battery consumption rate. Representatively, Tseng *et al.* (2017) and Franco *et al.* (2016) measured battery consumption rate by each drone dynamic [25][39]. When measuring the battery consumption of a drone depending on the drone dynamics, the key point to obtain reliable data is to control environmental conditions as much as possible. However, Tseng *et al.* (2017) found that there was a large deviation in the measured values as environmental factors were not controlled thoroughly during the experiments. By contrast, Franco *et al.* (2016) could obtain consistent results by measuring the battery consumption rate in a well controlled conditions. And they could describe interpolated BCR (Battery consumption rate) curves for each flight speed, flight direction by measuring it several times. Therefore, in this study, we develop 'Battery consumption models' by drone dynamics mainly based on the findings of Franco *et al.* (2016)

4.3.1.1 Flight direction and Flight Speed

Franco *et al.* (2016) measured battery consumption rate depending on flight direction, flight speed, acceleration, and deceleration using a quadrotor drone weighing about 2 *kg*. Figure 4.2 shows the battery consumption rate of drone when flying horizontally, ascending or descending vertically at each constant flight speed. In this graph, the x-axis represents the constant horizontal flight speed of the drone, and the y-axis represents the batter consumption rate. When the drone's flight speed is 0 m/s, that is, when it is hovering, the amount of battery consumption per second is approximately 220 *watts*. From this point to a speed of 7 *m/s*, the battery consumption rate slightly decreases. Although this outcome seems unreliable, considering the flight principle of a quadrotor drone, this is quite acceptable. When a quadrotor drone moves in any direction, the two rotors positioned in the direction the drone is moving in reduce its rotational speed, and the two rotors positioned on the opposite side maintain or increase its rotational speed (see Section2.2.1). By contrast, as the constant flight speed approaches 0 *m/s*, the four rotors rotate at approximately the same speed. In this way, as the flight speed increases from 0 *m/s* to 7 *m/s*, battery consumption rate can be reduced. However, from the speed section of 8 *m/s* or more, the battery consumption rate increases as the speed increases. In particular, from the 11 *m/s*, since the rotational speed of the two rotors increase drastically, the battery consumption rate also increases exponentially.

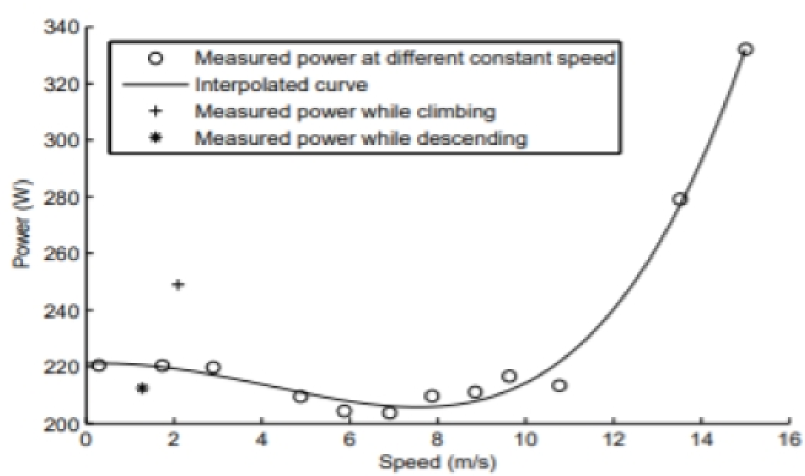


Figure 4.2: BCR by flight direction and flight speed (adapted from Franco *et al*. 2016)

Figure 4.3 shows the most energy-efficient speed by flight distance based on the battery consumption rate data according to the horizontal flight speed. Each curve represents a flight distance 50 to 1,200 *meters*, and the small black circle on this curve is the most battery-saving speed. For example, if the flight distance is no longer than 50 *meters*, the small circle is located about halfway between 6 and 8 on the x-axis. This means that if the flight distance is less than 50 meters, the horizontal speed of the drone, which will save the most battery consumption, is 7 *m/s*.

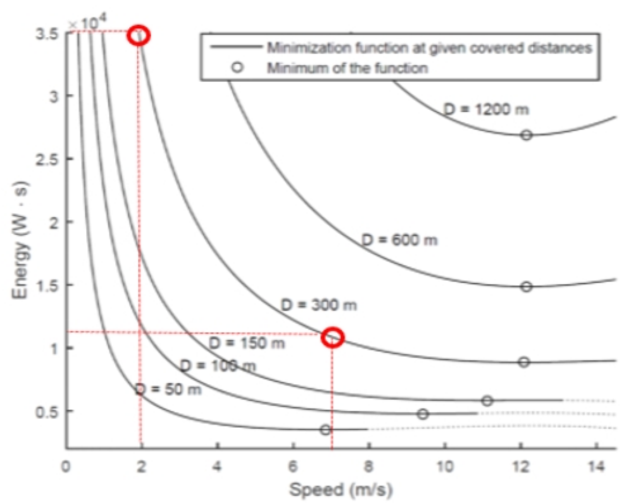


Figure 4.3: Ideal speed for each flight distance (adapted from Franco *et al.* 2016)

The following is a case of a drone flying vertically. When the drone ascends vertically at a constant speed of 2 *m/s*, its battery consumption rate is approximately 250 *watts*. This is approximately 22 *percent* higher than the that of flying horizontally at a speed of 7 *m/s* (about 205 *watts*). Conversely, when a drone descends at a speed of 1.5 *m/s*, it consumes about 215 *watts*. It can be assumed that the difference in battery consumption rate between these two motions is mostly caused by 'gravity'. In particular, when a drone is flying in a vertical direction, the battery consumption rate varies considerably depending on the vertical flight speed. In the horizontal speed range of 0∼11m/s, the battery consumption rate only changes about 15 watts. However, when the drone increases the vertical ascent speed from 2 *m/s* (7.2 *km/hr*) to 5 *m/s* (18.0 *km/hr*), the battery consumption rate increases by 110 *watts* (44 *percent*). The same applies to vertical descending case. Even as the descending speed increases from 1.5 to 4 *m/s*, the battery consumption rate increases by 85 *watts* (29 *percent*).

In this study, the battery consumption rate when the drone turns at a constant speed was also measured. If the drone rotates at a speed of 120 *deg/s*, the battery consumption rate is approximately 225 *watts*. For example, if the drone rotates a total of 720 *degrees* at a constant speed of 120 *deg/s*, it takes a total of 6 *seconds*, and the total amount of battery consumed during that time duration is about 1,350 *watts* (225 *watts* × 6 *sec*). The amount of battery consumed by the drone's turn must be considered, especially during the inspection routing model. For instance, delivery or surveillance missions using drones are performed in a relatively larger space compared to the area of structures, and the distance between the nodes that the drone should fly over is relatively longer. Therefore, no matter how many turns a drone takes, amount of battery consumption by turning can be negligible as compared to its total battery consumption. However, in the structure inspection model, the drone has to make many more turns to visit other checkpoint located other side of the structure. And, since the structure has normally much smaller space to be covered by drone than that of surveillance or delivery missions, the ratio of battery consumption due to turning to the total amount of battery consumption cannot be overlooked.

4.3.1.2 Acceleration and Deceleration

The following describes the battery consumption rate by the acceleration and deceleration of the drone. In the study of Franco *et al*. (2016), battery consumption rate was measured for a total of 35 *seconds* respectively, while the drone accelerated from 0 *m/s* to approximately 16 *m/s* and decelerated from 16 *m/s* to 0 *m/s* [25]. In the left side of Figure 4.4, the graph at the top side shows drone's acceleration for 35 *seconds*, and the graph at the lower side shows the battery consumption rate by the acceleration. We can see that the slope of the interpolated curve gradually increases in this graph. In particular, in the section accelerating from 0 *m/s* to 10 *m/s*, the battery consumption rate only increases by approximately 10 *percent* (205 $\rightarrow$ 230 *watts*). However, in the section accelerating from 10 *m/s* to 15 *m/s*, battery consumption rate rapidly increases by more than approximately 30 *percent* (230 $\rightarrow$ 300 *watts*). Similarly, in the deceleration graph (right side of Figure 4.4), when decelerating from 16 *m/s* to 10 *m/s*, the change in battery consumption rate is the largest.

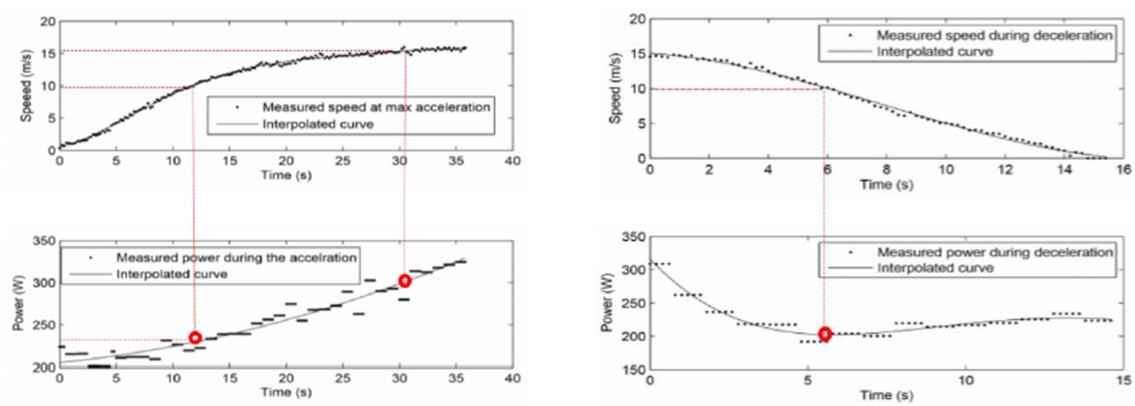


Figure 4.4: BCR by acceleration and deceleration (adapted from Franco *et al*, 2016)

However, we need to focus on the range between 0 *m/s* and 7 *m/s* of acceleration and deceleration. There will be a number of checkpoints scattered in a structure where the drone conducts inspection. Therefore, the drone must move a short distance and hover repeatedly. For precise inspection and for the maximum battery efficiency in those short flight distances, the flight speed will be set to around 7 *m/s*. Thus, it is unlikely that a drone will accelerate at a speed of 10 *m/s* or more or decelerate from a speed of 15 *m/s* while inspecting. And there is little change in the battery consumption rate when accelerating/decelerating in the range of 0 to 7 *m/s*. Therefore, we can conclude that the battery consumption rate due to acceleration and deceleration in the low-speed section is not significantly different from the battery consumption rate by constant horizontal flight speed.

4.3.1.3 Ascending / Descending Angle

Dietrich *et al*. (2017) measured the battery consumption rate depending on the diagonal ascending/descending [26]. In particular, they measured the battery consumption rate by increasing/decreasing the drone's diagonal ascending/descent angles by 22.5 degrees. Based on the vertical line (climb angle = 0 *deg*) in the right side of Figure 4.5, the red lines on the left represents the battery consumption rate when declining at each angle. Conversely, the red lines on the right shows the battery consumption when climbing diagonally. In the previous Section, we confirmed that the battery consumption rate when ascending is greater than that of descending. Similarly, in this study, the overall battery consumption rate when the drone is climbing is larger than that when the drone is declining. In addition, the steeper the drone climbs, the greater the battery consumption rate. By contrast, it is difficult to find the correlation between the battery consumption rate by declining angle on the graph.

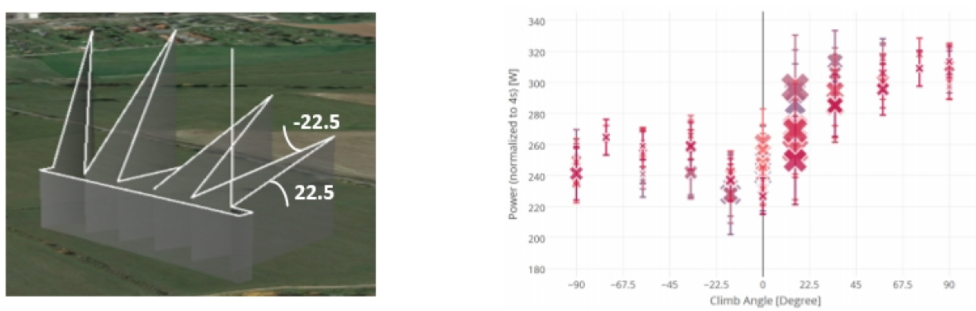


Figure 4.5: BCR by diagonal ascending and descending angle (adapted from Dietrich *et al*,2017)

It would be impossible to pinpoint the cause of these irregular measurements. Since the battery consumption rate can be changed not only by the dynamics of the drone but also by environmental factors at the time of the experiment, these irregular distribution may have been generated by winds that were not well controlled at the time of the experiment. In addition, these results might be occurred due to the difficulty of implementing diagonal flight compared to horizontal flight or vertical flight. Therefore, due to the lack of consistency of these data, they cannot be used in developing the 'battery consumption models'.

4.3.2 BCR by Drone Design and Environmental factors

In this Section, we review the changes in battery consumption rate due to drone design and environmental factors. From a physics perspective, the amount of battery consumed by drone dynamics is related to 'thrust force' or 'lift force'. By contrast, battery consumption by environmental factors or drone design is related to the force that interferes with the motion of the drone, which is called 'drag force'. For example, a drone will consume more amount of battery than expected to reach a desired speed or a desired point in a headwind. Excluding the amount of battery used by thrust from the total amount of battery consumption, the amount of battery consumed by drag will be obtained. The battery consumption due to drag will be small enough to be overlooked in stable weather conditions. However, because drones are much smaller in size and weight than general airplanes or helicopters, the drag acting on drones can increase exponentially even with small changes in weather. To find the inspection route that consumes the least amount of battery, it is necessary to carefully review methods to minimize the drag applied to the drone.

Several researchers have tried to directly measure the amount of drag acting on the drone [39]. However, since it is almost impossible to control weather factors, inconsistent or unreliable ranges of values were measured. For example, even if an experiment is conducted indoors to accurately measure the battery consumption rate for wind condition, the wind artificially generated may move unevenly depending on the air density, and the strength of the wind acting on each part of the drone may be different. Therefore, we concluded that it would be better to predict the amount of battery consumption by 'drag force' based on the laws of physics. To predict the magnitude of drag force, the drone design and environmental factors must be considered at the same time. The larger the area of friction with the atmosphere, the greater the drag. This means that even under the same environmental conditions, the drag can be changed depending on the design of the drone. In this Section, we will predict the approximate amount of battery consumed by drone design and environmental factors based on a common and well-known equation, given as follow Equation (4.1):

$$BCR_{drag} = 9.8 \times \frac{\rho C_d A V_w^3}{2} (watts). \tag{4.1}$$

Here, 'ρ' is the air density; it usually has a value of 1.15 to 1.25 kg/m^3 under normal atmospheric conditions [56]. 'C_d' denotes the drag coefficient, which can vary depending on the size and shape of the object on which the drag force is applied. 'A' refers to the area of the object to which the drag force is applied. Finally, 'V_w' represents the wind speed. In summary, 'A' and 'C_d' are related to the design of the drone, whereas 'ρ' and 'V_w are related to environmental factors [57]. We will mainly review the battery consumption in relation to these four factors in the following Sections.

4.3.2.1 Drone design

In the drag measurement formula above, the elements closely related to the design of the drone are 'A', which denotes the area in contact with the atmosphere, and 'C_d', which denotes the drag coefficient. Small drones, which generally weigh around 2 kg, have an area of about 0.02 to 0.03 m^2 in direct contact with the atmosphere during flight. The drag coefficient can vary between 0 and 1 depending on the 'A' value and the design of the drone. Moreover, even for the same drone, the values of 'A' and 'C_d' can be changed depending on its inclination. In general, for a quadrotor drone to move, each rotor must reduce or increase the rotational speed by different ratios. When the drone moves in any direction, the two rotors on the side of the moving direction reduce the rotational speed, and the two rotors on the opposite side increase the rotational speed. In this case, the two rotor sides with reduced rotational speed are slightly inclined downward because the thrust is reduced. By contrast, the side of the rotors on the opposite side, where the rotational speed is increased, rises slightly upward due to the increased thrust force (see Figure 4.6).

Figure 4.6: Inclination of quadrotor drone (adapted from Felismina *et al*, 2017)

Because of this, a drone flying horizontally at a speed of less than 10 m/s has an inclination of approximately 15 *degrees*. However, if the drone is hovering or flying vertically, the four rotors rotate at the same speed. In this case, the inclination is approximately 0 *degree*. Felismina *et al*. (2017) measured the values of 'A' and 'C_d' according to the inclination of drones, as shown in Figure 4.6 [33]. When the drone has an inclination of approximately 15 *degrees*, 'A' is about 0.029 m^2 and 'C_d' is 0.0613. When the drone is flying vertically or hovering, the inclination converges to 0 and the values of 'A' and 'C_d' are 0.021 m^2 and 0.0464, respectively.

4.3.2.2 Environmental Factors

Excluding 'A' and 'C_d' from the drag measurement Equation, 'ρ' and 'V_w' are environmental factors. 'ρ' denotes the air density in the atmosphere, which can be changed depending on the temperature and the flight altitude of the drone. As shown in Figure 4.7, the air density gradually decreases as the temperature increases; an example of this can be the change in tire pressure in summer and winter. Air density also changes with altitude. In general, the altitude slot for small drones to fly is between 0 and 200 *meters* above sea level. In this case, the air density decreases as altitude increases. Wang *et al*. (2019)

showed that wind is one of the factors that affects drag the most [58]. Tseng *et al.* (2017) measured the amount of drag change caused by wind but did not obtain reliable data [39]. As shown in Figure 4.8, the change in battery consumption rate is not constant depending on the wind speed or direction. Common sense is that if the back(tail) wind blows, the drone will be able to reach the desired speed with less power, which will reduce energy consumption. However, according to the measured data, some of the data in the headwind (about 200∼290 *watts*) is smaller than that in the back(tail) wind (about 250∼270 *watts*).

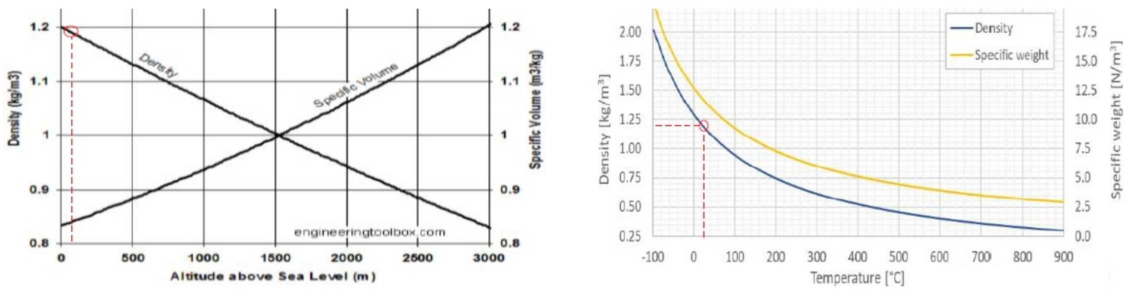


Figure 4.7: Correlation of air density with altitude and temperature (*sources*: Engr Toolbox)

In addition, in both cases, it is difficult to find a correlation between the wind speed and battery consumption rate. Therefore, a number of researchers have applied the drag measurement formula to roughly estimate the amount of battery consumption by the wind condition. Citroni *et al.* (2019) established a drag prediction formula for only drones, considering that drones are relatively smaller in size and weight than helicopters or airplanes [59]. These formulas were established under the assumption of a 4 *inch* 'microdrone'. In fact, for a microdrone that is the size of a palm, the flight can be significantly affected even by small changes in the environment.

Thus, to predict the magnitude of the drag caused by wind, we conclude that it would be best to input the wind speed into Equation 4.1. And Table 4.1 describes the drag by the wind speed. We assumed that the air density was 1.2 kg/m^3. According to drone manufacturers, the maximum wind speed that a commercial small drone can withstand is generally

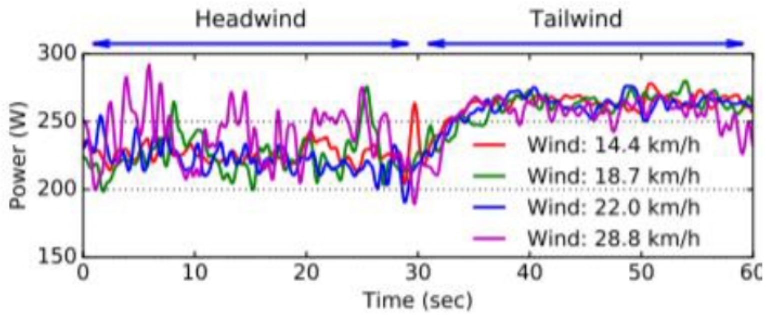


Figure 4.8: Battery consumption rate by wind conditions (adapted from Tseng *et al.*2017)

approximately 11 *m/s*. Therefore, we calculated each drag assuming the wind speed to be between 2 *m/s* and 10 *m/s*. As we explained earlier because there is no inclination when the drone is flying vertically or hovering, the values of 'A' and 'C_d' are 0.021 m^2 and 0.0464, respectively. In contrast, when flying horizontally of 7 *m/s* speed, the inclination of the drone is about 15 *degrees*; therefore, the values of 'A' and 'C_d' are 0.029 m^2 and 0.0613, respectively. In each flight dynamics, for wind speeds of 2∼6 *m/s*, the drag acting on the drone is less than 11 *joules*. However, at a wind speed of 8 *m/s*, the drag increases exponentially. In particular, when the drone flies horizontally at a wind speed of 10 *m/s*, approximately 50 *watts* of battery power is consumed by drag force.

Table 4.1: Magnitude of the BCR by drag depending on each wind speed

Wind speed	$Drag_{vertical}$ (C_d: 0.0464, A:0.021m^2)	$Drag_{horizon}$ (C_d: 0.0613, A:0.029m^2)
2 *m/s*	0.22 *watts*	0.40 *watts*
4 *m/s*	1.75*watts*	3.18 *watts*
6 *m/s*	5.89 *watts*	10.75 *watts*
8 *m/s*	13.97 *watts*	25.48 *watts*
10 *m/s*	27.28 *watts*	49.76 *watts*

4.3.3 Battery consumption model using integrated approach

In the previous Section, we confirmed that there is a clear difference in the battery consumption rate of drones depending on the drone dynamics, design, and environmental conditions. This was the first purpose of this study. However, when calculating total battery consumption in a certain flight distance, these three factors must be considered in an integrated form. We decided to use the battery consumption rate data by drone dynamics measured by Franco*et al*. (2015) for data-driven optimization. And when we estimate the the battery consumption rate by design or environmental factors using the Equation (4.1), the data from Felismina *et al*. (2017) are referenced for the 'A' and 'C_d' values. Since the values measured in the study by Franco *et al*. (2015) were generally consistent and distributed within the range acceptable to common sense, we decided not to refer to that data. In addition, even though most studies in this field have used small drones, they have slightly different in weights, areas, and drag coefficients depending on the manufacturer and the manufacturing year. If the drone needs to travel a long distance or a long time for inspection, the difference in the total battery consumption may increase because of this tiny difference, which may lead to unstable results. Therefore, to minimize such errors, we will use the BCR data and formulas from the two studies. Based on these two studies, we can calculate the total battery consumption of a drone when hovering, flying horizontally, flying vertically, or turning over a certain amount of time or a certain distance. First, before explaining each equation, we employ the notation shown as Table 4.2. And We develop battery consumption models for each of the following two conditions; 1) horizontal flight, 2) vertical flight (ascending/descending/hovering), and 3) turning in a windy condition.

When the wind speed is 'V_w' and the drone is flying horizontally over the distance 'd' meters at the speed of 'V_h', total battery consumption($E_{forward}$) is equal to Equation (4.2). 'P_t' is the function of battery consumption rate by thrust. For example, '$P_t(V_h)$' is the battery consumption rate for each horizontal flight speed. This is listed in Appendix C in

Notation	Definition
V_h	Horizontal flight speed of the drone
V_a	Vertical descending speed of the drone
V_d	Vertical descending speed of the drone
V_r	Rotational speed of the drone
V_w	Head/back(tail)wind speed acting on the drone
P_t	BCR by each flight speed and each motion of the drone
P_d	Battery consumption rate by the drag

Table 4.2: Notations for the battery consumption models

detail. In particular, when a drone is in a headwind during horizontal flight, more energy is consumed by the drag force acting on the drone. Accordingly, the total battery consumption is the sum of the thrust-induced battery consumption and the drag-induced battery consumption (e.g.,($P_t(V_d)$*d/V_d)+($P_d(V_w)$*d/V_d)). On the other hand, when the drone is in the back(tail) wind during horizontal flight, the drone can reach any speed or any distance with less battery power. Therefore, in this case, the total battery consumption is the value that the thrust-induced battery consumption minus the drag-induced battery consumption:

$$E_{horizon} = (P_t(V_h) \pm P_d(V_w))\frac{d}{V_h}. \tag{4.2}$$

And, when the wind speed is 'V_w', and the drone flies vertically over the height 'h' *meters* at the speed of 'V_a' and 'V_d', total amount of battery consumption is equal to Equation (4.3), Equation (4.4), and Equation (4.5). The battery consumption rate for each vertical flight speed(($P_t(V_a)$ or $P_t(V_d)$) is also listed in Appendix C. In particular, when the drone is flying vertically, the wind acting on the drone is considered a headwind regardless of its direction. Therefore, assuming that the wind blows at a constant speed across all altitudes, the total battery consumption is the amount of battery consumed by the lift force plus the amount of battery consumed by drag force. And when the drone hovers, the vertical or horizontal flight speed is 0 *m/s*. In this case, total flight distance is 0 *meter*, so we need to multiply the hover time 't' instead of 'd/V_d'(see Equation 4.5):

$$E_{ascend} = (P_t(V_a) + P_d(V_w))\frac{\delta h}{V_a}, \tag{4.3}$$

$$E_{descend} = (P_t(V_d) + P_d(V_w))\frac{\delta h}{V_d}, \tag{4.4}$$

$$and \quad E_{hover} = (P_t(V_h) + P_d(V_w))\, t. \tag{4.5}$$

Finally, when the wind speed is 'V_w' and the drone rotates 'θ' *degree* at the rotational speed(V_r) of 120 *deg/s*, the total battery consumption (E_{turn}) is equal to Equation (4.6).

$$E_{turn} = (P_t(V_r) + P_d(V_w))\frac{\delta\theta}{V_r} \tag{4.6}$$

4.4 Problem Description and Formulation

This study focuses on finding the optimal route to minimize the total battery consumption as well as the total flight distance. For reflecting various dynamics of drones during the inspection, we assume a situation that a drone is scanning a cuboid-type structure. The number of checkpoints may vary depending on the size, or shape of the structure and the checkpoints are distributed all over the structure. The drone is to scan all the checkpoints only once and then returns to the depot. The drone can only move from a point to an adjacent point in the horizontal or vertical direction. Since the drone flies outdoors during the inspection, weather factors such as wind and air density can also affect battery consumption. The battery consumption by wind directions other than head wind and tailwind has not been well addressed in the literature (see Section 4.3.2). Hence, the drag force is applied to the drone only in the headwind and tailwind. In other wind directions, the battery consumption due to drag force is considered as 0 *joule*.

4.4.1 Optimization models for minimizing the total flight distance(SFD) and the total battery consumption(MEC)

In this Section, we propose two MIP models to find the route with the shortest flight distance and the route with the lowest energy consumption. The following mathematical notation is defined for the model formulations.

Sets

N Set of all nodes to be inspected, $i \in N, i = 1, 2, , , , n$

A Set of all arcs that a drone can move

Parameters

d_{ij} Flight distance in arc(*i,j*)

c_{ij} Battery consumption in arc(*i,j*)

e_{ijk} Battery consumption by turning in arc(*i,j*) and arc(*j,k*)

p_i Battery consumption by hovering for scanning node i

M_c Maximum battery capacity of a drone

M_d Maximum flight distance of a drone

Decision Variables

x_{ij} 1 if the drone flies on arc(*i,j*), 0 otherwise

y_{ijk} 1 if the drone flies arc(*i,j*) and arc(*j,k*), 0 otherwise

u_i Visiting sequence of node i on a route

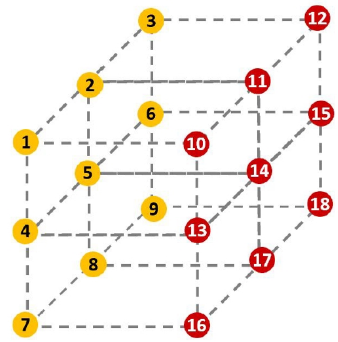


Figure 4.9: Example of cubiod structure with scanning nodes

Parameters c_{ij}, e_{ijk}, and p_i are related to the amount of battery consumed by the drone at each node or arc(s). Figure 4.9 illustrates a cuboid structure to be inspected, where c_{ij} means the amount of battery consumed by the drone while flying horizontally or vertically over the arc(*i,j*) (see equations (4.2),(4.3), and (4.4)). Parameter 'e_{ijk}' refers to the battery consumption by a turn when the drone flies in two consecutive arc(*i,j*) and arc(*j,k*) (see Equation 4.6). In a cuboid structure, if the horizontal interior angle of two consecutive arcs is 90 *degrees*, some amount of battery is consumed by drone's rotation (e.g, path $1 \rightarrow 2 \rightarrow 11$). Otherwise, i) when two consecutive arcs are horizontally and vertically straight (e.g, path $1 \rightarrow 2 \rightarrow 3$ or path $1 \rightarrow 4 \rightarrow 7$), ii) if the vertical interior angle of two consecutive arcs is 90 degrees (e.g, path $1 \rightarrow 2 \rightarrow 5$), '$e_{ijk}$' has a value of 0 *watts*. Parameter p_i is the amount of battery that the drone consumes while hovering to scan node i (see Equation 4.5), and M_c is the maximum amount of battery consumption drone can use during an inspection. The safety level is considered in case of emergency when it reaches 10-20 % of the maximum battery capacity. This is the flight speed with the smallest BCR multiplied by the maximum flight duration (=7 *m/s* * (M_c/min BCR)).

The following is a MIP model for minimizing total flight distance (SFD) of the drone during the structural inspection. The objective function(Equation 4.7) is to minimize the sum of flight distance, which visit all nodes and then back to the initial depot node(starting node) following given arcs:

$$\text{minimize} \quad \sum_{(i,j)\in A} d_{ij}x_{ij}. \tag{4.7}$$

Constraints (4.8) and (4.9) ensure that each node is visited only once. Constraint (4.10) forces that total flight distance during an inspection not to exceed M_d:

$$\sum_{i=1, i\neq j} x_{ij} = 1, \ \ \forall\, (i,j)\in A, \tag{4.8}$$

$$\sum_{j=1, j\neq i} x_{ij} = 1, \ \ \forall\, (i,j)\in A, \tag{4.9}$$

$$\textit{and} \quad \sum_{(i,j)\in A} d_{ij}x_{ij} \leq M_d, \ \ \forall\, (i,j)\in A. \tag{4.10}$$

Constraint (4.11) is for eliminating sub-tours. And inspection is terminated only when all nodes are visited by constraint (4.12).

$$u_i - u_j + nx_{ij} \leq n-1, \ \ \forall\, 2\leq i\neq j\leq n, (i,j)\in A, \tag{4.11}$$

$$0 \leq u_i \leq n-1, \ \ \forall\, 2\leq i\leq n, \tag{4.12}$$

$$\textit{and} \quad x_{ij}\in\{0,1\}, \ \ \forall\, (i,j)\in A. \tag{4.13}$$

The second model is to find the inspection route that minimizes the total battery consumption (MEC). The objective function (Equation 4.14) is to minimize the sum of the battery consumption by the horizontal or vertical flight, and plus, the sum of the battery consumption by horizontal turning as follows:

$$\text{minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in A}\sum_{(j,k)\in A} e_{ijk}y_{ijk}. \tag{4.14}$$

Constraints (4.15) and (4.16) ensure that each node is visited only once:

$$\sum_{i=1, i \neq j} x_{ij} = 1, \ \ \forall\, (i, j) \in A, \tag{4.15}$$

$$and \quad \sum_{j=1, j \neq i} x_{ij} = 1, \ \ \forall\, (i, j) \in A. \tag{4.16}$$

Constraint (4.17) and (4.18) are logical constraints for decision variable 'y_{ijk}', a 'turn penalty' variable. Constraint (4.17) forces that when 'y_{ijk}' is 1, both 'x_{ij}' and 'x_{jk}' should be '1'. That means, when 'y_{ijk}' is 1, the arc 'x_{ij}' and 'x_{jk}' should be consecutive arcs. Also, Constraint (4.18) enforces that when 'y_{ijk}' is 0, at least one of 'x_{ij}' and 'x_{jk}' has a value of 0. That means when 'y_{ijk}' is 0, 'x_{ij}' and 'x_{jk}' should not be consecutive arcs:

$$x_{ij} + x_{jk} \geq 2y_{ijk}, \ \ \forall\, (i, j) \in A, \ 1 \leq i \neq j \neq k \leq n, \tag{4.17}$$

$$and \quad x_{ij} + x_{jk} \leq 1 + y_{ijk}, \ \ \forall\, (i, j) \in A, \ 1 \leq i \neq j \neq k \leq n. \tag{4.18}$$

Constraint (4.19) guarantees an upper limit on battery consumption. The drone battery is assumed to be fully charged at the beginning of the flight. To avoid running out of battery capacity during the flight, we assume that safety portion (e.g., 10∼20 *percent* of maximum capacity). In other words, If the total energy consumed by the drone on a route (sum of the total energy consumed while flying over the arc ($\sum c_{ij}$) and the total energy consumed by the turn ($\sum e_{ijk}$), and the total energy consumed while hovering ($n \cdot p_i$)) exceeds the level of safety portion, the routes become an infeasible solution.

$$\sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{(i,j) \in A} \sum_{(j,k) \in A} e_{ijk} y_{ijk} + \sum_{i \in N} p_i \ \leq \ M_c. \tag{4.19}$$

Constraint (4.20) (4.21) are the same as previous one as follows:

$$u_i - u_j + nx_{ij} \leq n-1, \ \ \forall\, 2 \leq i \neq j \leq n, (i,j) \in A, \tag{4.20}$$

$$0 \leq u_i \leq n-1, \ \ \forall\, 2 \leq i \leq n, \tag{4.21}$$

$$x_{ij} \in \{0,1\}, \ \ \forall\, (i,j) \in A, \tag{4.22}$$

$$and \quad y_{ijk} \in \{0,1\}, \ \ \forall\, (i,j) \in A, (j,k) \in A. \tag{4.23}$$

4.4.2 Solution Approach

The OP models described in the previous section are extended versions of Travel Salesman problem (TSP), which is known to be NP-hard[60]. Therefore, this section introduces a preprocessing algorithm to find good feasible solutions to the OP models faster. The idea of the algorithm described in Section 4.4.2.1 is to remove those arcs the drone cannot reach for inspection from the solution search process because space. This drastically reduces the solution search space, which leads to much reduced computational time.

4.4.2.1 Preprocessing to reduce the solution search space

The SFD model finds a route with the shortest flight distance to complete the inspection on a graph containing physical distance information between two nodes d_{ij}. Similarly, the MEC model searches for a solution on a graph having the energy consumption data two nodes c_{ij}. Since the drone can only move to an adjacent node and is not allowed to move diagonally, there are between 3 and 5 options for a move in a horizontal direction or a vertical direction. For example, in Figure 4.9, the drone can move from the depot (node 1) only to nodes 2, 4 and 10. Hence, the arcs that the drone cannot use (e.g., arc(1,5)) can be excluded from the search space.

Algorithm 1 Preprocessing to reduce solution search space

Inputs:
Coordinate(x, y, z) of each node in node set N
BCR by flight speed($P_t(v_d)$) and by wind speed($P_d(v_w)$)

Search space reduction
for all (node $i, j \in N$)
  if two of $|x_i - x_j|$, $|y_i - y_j|$, $|z_i - z_j|$ = 0
    Create a flight network with
$$d_{ij} = |(x_i - x_j) + (y_i - y_j) + (z_i - z_j)|$$
    Create a flight network with
$$c_{ij} = (P_t(v_d) \pm P_d(v_w))\frac{d_{ij}}{v_d}$$
  else if more than two of $|x_i - x_j|$, $|y_i - y_j|$, $|z_i - z_j| \neq 0$
    Fix the decision variable with
$$x_{ij} = 0$$
  end if
end for

Algorithm 1 shows the graph formation process to reduce the search space. First, the coordinates of all nodes and BCR information for each flight status must be entered. All nodes are represented in the (x,y,z) coordinate system. For nodes i :(x_i, y_i, z_i) and node j :(x_j, y_j, z_j), if two of $|x_i - x_j|$, $|y_i - y_j|$, and $|z_i - z_j|$ are equal to 0, those nodes are horizontally or vertically adjacent nodes; hence, the drone is allowed to move to any of these nodes. For example, if we assume the coordinates of 'node 7' to be (0, 0, 0) and the coordinates of 'node 8' to be (0, 0, 50) in Figure 4.9, the values of both $|x_i - x_j|$, $|y_i - y_j|$ are equal to 0. Therefore, the arc(7,8) should be included in the drone path, and $d_{ij} = |(x_i - x_j) + (y_i - y_j) + (z_i - z_j)|$, which is used in the graph.

Similarly, the amount of battery consumed by the drone (c_{ij}) is calculated by the 'battery consumption models' discussed in Section 4.3.3 taking into account flight dynamics (v_d), wind conditions (v_w), and length of the arc(d_{ij}), and then these c_{ij} value of feasible arcs are added to the graph.

When two or more of $|x_i - x_j|$, $|y_i - y_j|$, and $|z_i - z_j|$ are not zero, such nodes are not vertically or horizontally adjacent nodes, and these nodes are removed from the search space by modifying the decision variables x_{ij} as 0. In this way, the solution search space will be reduced, and it reduces the computational time.

4.4.3 Numerical Experiments

This section begins with a description about experimental conditions to demonstrate the proposed work. In this Section, two separate experiments are conducted to show that routes with minimum flight distance and routes with minimum battery consumption are different. Moreover, we explore all the feasible solutions between the optimal solutions of SFD and MEC model and plot them as a chart in order to show the relationship between the two MIP models. The pre-processing algorithms and proposed OP models are implemented in Python[47] and solved by CPLEX 12.6.3. All experiments are made on a server running RedHat Linux 64-bit with Intel i7 core processors and 16GB RAM.

For numerical experiments, we transform large structures into cuboid shapes. There are quite a few types of large structures in the world. However, in order to reflect as much of the drone motions (horizontal/vertical flight, hovering, turning) that affect the battery consumption rate, which was reviewed in Section 4.3, we have screened some structures that can be converted into a three-dimensional form among the structures. And we simplified them into two types of cuboids (see Figure 4.10 and 4.11). The first example (Figure 4.10) is a structure in the form of a tall building. The second example (Figure 4.11) is a low-height and long-width type structure. It is similar to the shape of a large bridge. In particular, there is a path for drones to traverse inside the structure as well. Each structure has a total of 30 and 32 checkpoints, which are checked by a single drone.

The drone can move to adjacent nodes using only a designated vertical or horizontal path (solid and dotted line on the example structures). The detailed size of the building

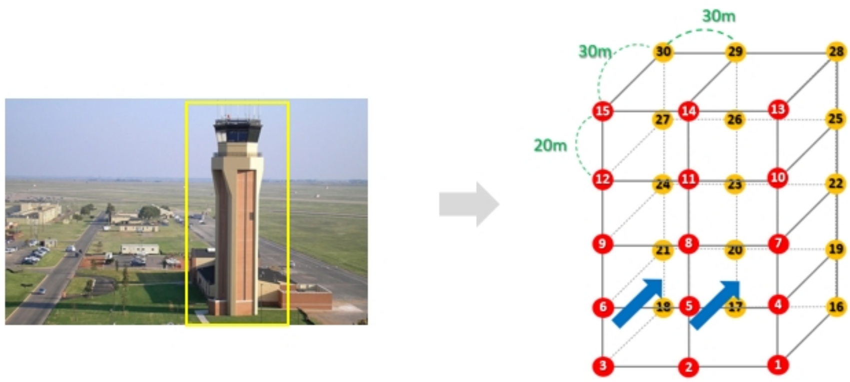


Figure 4.10: Example structure 1 (Tall-building type)

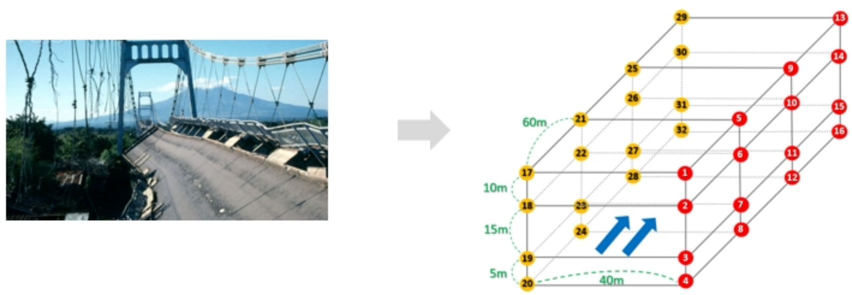


Figure 4.11: Example structure 2 (Wide-cuboid type)

and the distance between one node and adjacent nodes are described in the Figure 4.10 and 4.11. In both structures, the drone repeats hovering to scan one checkpoint and then moving to an adjacent node. Considering the distance between nodes and nodes, the maximum distance that the drone can travel at once is 50 *meters*. Therefore, as reviewed in Section 2.1, the most energy-efficient speed for horizontal flight during the inspection is 7 *m/s*. Moreover, the vertical ascent and descent speed of the drone is assumed to be 2 *m/s* and 1.5 *m/s*, respectively, taking into account that the battery consumption rate increases significantly as the vertical speed increases. In fact, there is no distinction between the front, the rear, and the side of a drone. However, it is assumed that the camera mounted on the drone is fixed to face the front for experimental purposes. Accordingly, the drone must make a 90-*degree* turn to scan the checkpoint when moving along two arcs bent horizontally (e.g., path 1→17→21 of Figure 4.11). Each node should be scanned for 3 *seconds* with a camera mounted on the drone. The drone specification used for inspection is described in Appendix B. The maximum capacity of the drone battery is about 293,040 *joules* (5,500*mAh* × 14.8*volt* × 3.6), and if the drone uses more than 85 *percent* of its maximum capacity, the drone must stop inspection and return. That is, a route that the drone consumes more than 249,084 *joules* (0.85 × max capacity) of energy will be an infeasible solution. During the inspection, we assume that the atmospheric conditions are constant. The wind blows from the north at a speed of 4*m/s* (Blue colored arrows in Figure 4.10 and 4.11). The temperature is 20 *degrees* Celsius, and since all missions are performed at an altitude of 500 *meters* or less above sea level, the air density remains at 1.20 kg/m^3.

Table 4.3 and Figure 4.12 shows the route with the shortest total flight distance and the route with the minimum total battery consumption, respectively, for inspecting the tall-building type structure in Figure 4.10. As shown in the flight sequence, the two routes are clearly different. Moreover, the route with the shortest total flight distance consumes about 24.75 *percent* more energy, despite the route is about 19.5 *percent* shorter than most battery-saving routes. Since the SFD model only considers the physical distance of the arc, regardless of the amount of battery the drone will consume in each arc, it includes a significantly larger number of vertical arcs in the flight route than the optimal solution of the MEC model (SFD: 24 arcs, MEC: 8 arcs). This can reduce the total flight distance, but it does not improve battery consumption at all. In particular, the optimal solution of the MEC model has 6 more turns (total of 540 *degrees*) than that of the SFD model, and it requires 1,012 *joules* (540 *deg* ÷ 120 *deg/s* × 225 *watts* = 1,012.5 *watts*) of additional battery consumption. Instead, by doing so, the drone can use as many horizontal arcs as possible, and it can save much more battery consumption(about 25,090 *joules*). Hence, the MEC model focuses on including as many horizontal arcs as possible in the flight route even at the expense of additional battery consumption due to turns.

Table 4.3: Numerical results of example 1 (Tall-building type)

Model	Flight sequence	Distances (*meters*)	Energy (*joules*)
SFD	1→16→19→22→25→28→29→26→23→20→ 17→18→21→24→27→30→15→12→9→6 →3→2→5→8→11→14→13→10→7→4→1 up: 12, down: 12, straight: 6, turn: 4	660	109,090
MEC	1→4→5→6→21→20→23→26→25→10→11→ 12→15→14→13→28→29→30→27→24→ 9→8→7→22→19→16→17→18→3→2→1 up: 4, down: 4, straight: 22, turn: 10	820	82,089

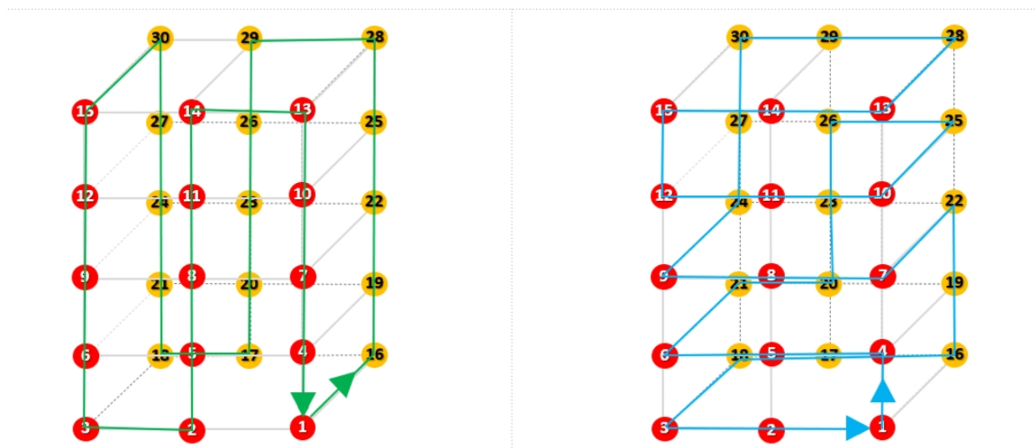


Figure 4.12: The optimal route by SFD(*left*) and MEC(*right*) for example structure 1

In the wide-cuboid type structure (Figure 4.11), we can see similar results (see Table 4.4 and Figure 4.13). The flight sequence of each optimal route by SFD and MEC model is different, and the route with the shortest total flight distance consumes more amount of battery. However, the smaller the proportion of the building height to the total circumference, that is, the flatter the shape of the building, the smaller the improvement in battery consumption by the MEC model. In the Table 4.3(numerical result for tall-building type structure), the improvement rate of total flight distance is 19.5 *percent* (820 → 660 *meters*), and the improvement rate of total battery consumption is 24.75 *percent* (109,090 → 82,089 *joules*). On the other hand, in Table 4.4 (wide-cuboid type), the improvement rate in total flight distance is 23.59 *percent* (890 → 680 *meters*), while the improvement rate in total battery consumption is only 2.42 *percent* (86,825 → 84,719 *joules*).

Table 4.4: Numerical results of example 2 (Wide-cuboid type)

Model	Flight sequence	Distances (*meters*)	Energy (*joules*)
SFD	1→5→6→7→8→12→11→10→9→13→14→15→ 16→32→31→30→29→25→26→27→28→24→ 23→22→21→17→18→19→20→4→3→2→1 up: 12, down: 12, straight: 8, turn: 3	680	86,825
MEC	1→17→18→22→6→5→21→25→9→10→26→30 →29→13→14→15→16→32→31→27→28→12 →11→7→8→24→23→19→20→4→3→2→1 up: 6, down: 6, straight: 12, turn: 10	890	84,719

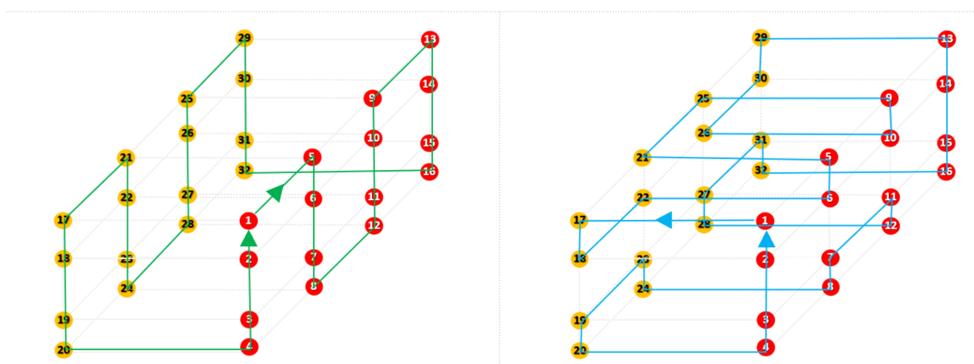


Figure 4.13: The optimal route by SFD(*left*) and MEC(*right*) for example structure 2

It has been demonstrated that the optimal solutions by the SFD and MEC models in both sample structures are not proportional to each other. However, in order to have a more robust justification for performing bi-objective optimization, we explored other feasible solutions between the optimal solutions of SFD and MEC models and illustrated the result as plot charts (see Figure 4.14 and Figure 4.15). In this graph, the x-axis is the total flight distance and the y-axis is the total battery consumption of the feasible solutions. As a result, as shown in Figure 4.14 and Figure 4.15, we plotted 5 and 11 feasible solutions for each example structure within the area where the total flight distance of the optimal solutions by the SFD and MEC models are lower and upper bound. Details of these feasible solutions are provided in Appendix D. In fact, there are much more feasible solutions in this area than these numbers. However, When we plotted feasible solutions on a chart, only one route with the minimum battery consumption per each total flight distance was illustrated. As shown in the Figure 4.14 and 4.15, trend lines in both chart have negative slope (Example 1: -173.58, Example 2: -10.66). This means that the SFD and MEC models have inverse relationship. In conclusion, we cannot minimize both criteria at the same time, and we have to find a route that compromises the two goals through bi-objective optimization.

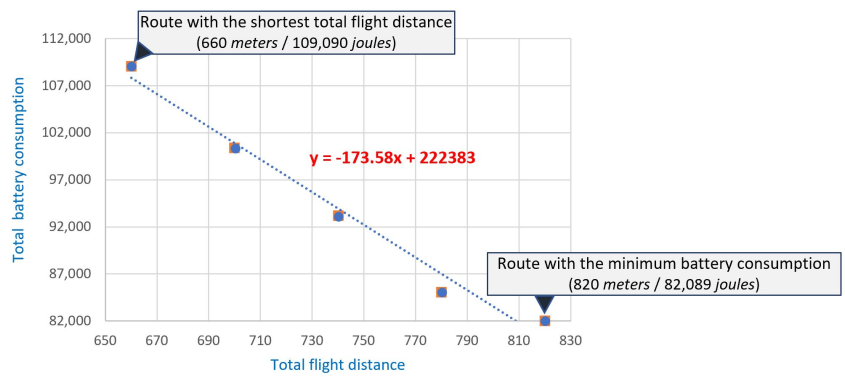


Figure 4.14: Relationship for SFD and MEC in thte tall-building type structure

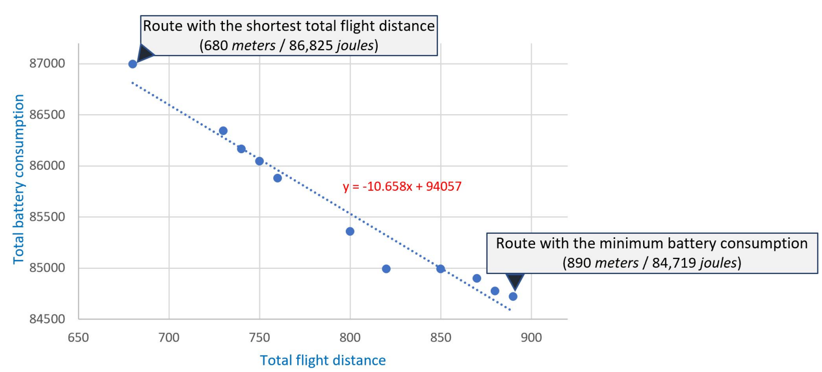


Figure 4.15: Relationship for SFD and MEC in the Wide-cuboid type structure

4.5 Bi-objective optimization

We confirm that each of the optimal solution derived from the two MIP models (SFD and MEC) in the previous Section are different. In this case, we can find a suitable solution that trades off the two criteria through bi-objective optimization model. A typical optimization model has one objective function, but it is often necessary to consider more than one objective function at the same time. For example, if we need to maximize the performance of a system while minimizing development costs, we can optimize both objective functions using multi-objective optimization. Multi-objective optimization problems can be defined by the following Notation and Equations:[61]

Notation

$f(x)$	objective functions vector
$f_i(x)$	i^{th} objective function
$g_i(x)$	i^{th} inequality constraint function
$h_j(x)$	j^{th} equality constraint function
X_{inf} , X_{sup}	lower and upper bounds of the design space
$\mathbb{S}$	feasible region in the design space
$\mathbb{R}^k$	function or criterion space

Where $f(x) = [f_1, f_2, f_3, ..., f_k]^T : X \rightarrow \mathbb{R}^k$ is a vector with the values of objective functions to be minimized. 'x' is the vector containing the decision variables, defined in the design space $\mathbb{R}^n$. '$g_i(x)$' represents the i^{th} inequality constraint function and '$h_j(x)$' the j^{th} equality constraint function. 'X_{inf}' and 'X_{sup}' are the lower and upper bounds of the decision variables. The goal of this model is as follows:

$$\text{minimize} \quad f(x) \tag{4.24}$$

Equations (4.25) to (4.27) define the region of feasible solutions, $\mathbb{S}$, in the design space $\mathbb{R}^n$.

$$g_i(x) \leq 1, \quad \forall\ i = 1,2,....,m., \tag{4.25}$$

$$h_i(x) = 0, \quad \forall\ j = 1,2,...,l., \tag{4.26}$$

$$and \quad X_{inf} \leq X \leq X_{sup}, \ \forall\ (i,j) \in A. \tag{4.27}$$

The constraint on $g_i(x)$ is a 'less than or equal to' function. And the 'greater than or equal to' function can be converted to the 'less than or equal to' type by multiplying it by '-1'. Similarly, In the problem for 'minimization' on the functions $f_i(x)$, given that functions 'maximization' can be transformed into the former by multiplying them by '-1'. The notion of 'optimum' in solving multi-objective optimization problems is known as 'Pareto frontier'. The solution is said to be 'Pareto Optimal' if there is no way to improve one goal without worsening more than one goal[61]. Due to the conflicting nature of the objective function, the Pareto optimal solution is usually scattered over the domain because all objective functions cannot be minimized simultaneously. In solving the optimization problem we obtain the Pareto set or the Pareto optimal solutions defined in the design space, and the Pareto front, an image of the objective functions, in the criterion space, calculated over the set of optimal solutions.

4.5.1 Solution method to generate Efficient frontier

In this Section, we select the appropriate bi-objective optimization method to find a compromise solution that reduces total flight distance and total battery consumption at once when inspecting previous structures. And we propose a bi-objective optimization model that applies that method. Among multi-objective optimization algorithms, the 'weighted sum method' is the most widely used because it is easy to understand and easy to apply in real life [62]. We can find a bi-objective solution by multiplying each weight for each objective function and then adding all the weighted values. Also, in this way, we can obtain the 'Pareto optimal solution' by slightly changing the weight for each objective function and arranging the found solutions in the feasible region. However, the weighted sum method may not guarantee a Pareto optimal solution for non-convex problems. Due to this drawback, this method is difficult to apply to problems with strong non-linearities[63].

Another algorithm that is often used with the weighted sum method is goal programming proposed by Charnes *et al.* (1977). This algorithm minimizes the deviation between the target value and the design value of the function. In its modified form, there are 'weighting goal programming' that weights the each goal and 'preemptive goal programming' that arranges functions according to their importance and optimizes them sequentially [64]. The 'ε-constraint method' is also one of the frequently used methods. This method sets the maximum level of all other objective functions except one given objective function and minimizes the objective function within this range [65][66]. And, optimization is performed by changing the maximum level of the constrained objective functions little by little. By repeating this process for all objective functions, we can find solutions.

However, in the case of the structural inspection, the priority(importance) for each flight distance and battery consumption may vary depending on the operating conditions. For example, when a drone is used for inspection(scanning) bridges or buildings which is supposed to be occupied by friendly military force, the shorter completion time of the inspection mission on the structure is a key factor to avoid the delay of the planning process deploying military assets. However, in situations where the focus is on operating costs, it will be more important to reduce battery consumption. In conclusion, the importance of each goal is not fixed. Therefore, instead of applying the preemptive method in which one criterion has an absolute superiority over the other, the weighted method is applied in the bi-objective optimization model.

$$\text{minimize} \quad \sum w_i \cdot f_i^s(x) \tag{4.28}$$

$$x \in \mathbb{S} \tag{4.29}$$

$$w_i \geq 0; \sum w_i = 1 \tag{4.30}$$

In this model, 'f_i^s' stands for scale normalized objective function. The approach to 'scale normalization' is described in detail in Section 4.5.2. Meanwhile, 'w_i' is the relative weight to be assigned to each normalized objective function(f_i^s) and it reflects, a 'priority', the drone user's preferences for each criteria. The objective function (Equation 4.28) is to minimize the sum of each scale normalized objective function(f_i^s) given a relative weight(w_i). Decision variables(x) must belong to a feasible design space (Equation 4.29). Depending on each situation or user's preference, 'w_i' can have a value between 0 and 1 , and the sum of 'w_i' must be 1 (Equation 4.30).

4.5.2 Scale normalization for bi-objective optimization

If the unit scale of one objective function is significantly larger or smaller than that of other objective functions, the bi-objective optimal solution is bound to be biased against the objective function with a larger unit scale regardless of the relative weight[67]. We can realize the need for scale normalization by looking at feasible routes that inspect tall-building type structure (Example structure 1). In Figure 4.14, the total flight distance of feasible solutions are distributed between 660∼820 *meters*, respectively. And the total battery consumption of these solutions ranges between 82,089∼109,090 *joules*, respectively.

Since the unit measuring battery consumption (*joules*) and the unit measuring flight distance (*meters*) are different, and the scale of total battery consumption is about 129 times ((109,090+82,089)/(660+820)=129.18) larger than that of total flight distance, no matter how much greater weight is given to the objective function of the SFD, the bi-objective optimal solution will be determined by the objective function of the MEC. Therefore, to address this issue, the unit scale of each objective function must be normalized. As a way to normalized the scale, we propose the ‘Min-max’ normalization method[68]. This method is implemented by Equation (4.31). To use this method, we need to find the minimum (min $f_i(x)$) and maximum value (max $f_i(x)$) of each model. The interval in which the SFD and MEC models are inversely proportional to each other is the range between the optimal solutions of these two models. Therefore, in order to normalize the scale of the total flight distance, ‘max $f_i(x)$’ must be the total flight distance of the optimal solution by the MEC model (Example 1: 820 *meters*, Example 2: 890 *meters*). And ‘min $f_i(x)$’ is the total flight distance of the optimal solution by the SFD model (Example 1: 660 *meters*, Example 2: 680 *meters*). And, in order to normalize the scale of the total battery consumption, ‘max $f_i(x)$’ must be the total battery consumption of the optimal solution by the SFD model (Example 1: 109,090 *joules*, Example 2: 86,825 *joules*). And ‘min $f_i(x)$’ is the total battery consumption of the optimal solution by the MEC model (Example 1: 82,089 *joules*, Example 2: 84,719 *joules*). The objective function f_i^s(x) whose scale is normalized has a value between 0 and 1 for all ‘x’. And the bi-objective solution is no longer affected by the scale of each model.

$$f_i^s = \frac{f_i(x) - Min\ \ f_i(x)}{Max\ \ f_i(x) - Min\ \ f_i(x)} \tag{4.31}$$

Figures 4.16 and 4.17 are plot charts showing feasible solutions with normalized scale. The chart on the left side of each figure shows feasible solutions that are not scale normalized, and the chart on the right shows feasible solutions with scale normalized. In the chart on the right side of each Figure, the x-axis is the normalized value of total flight distance(f_1^s)

and the y-axis is the normalized value for total battery consumption(f_2^s). As you can see, all scale normalized feasible solutions have x and y values between 0 and 1. And the scale normalized values for all feasible solutions can be viewed in detail in Appendix E. And now, since all feasible solutions have been scaled normalized, the bi-objective optimal solution will only be determined by the relative weights assigned to each objective function.

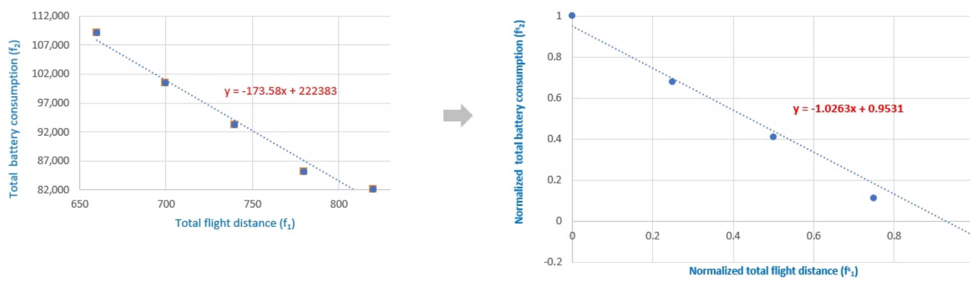


Figure 4.16: Result of Scale normalization (Example structure 1)

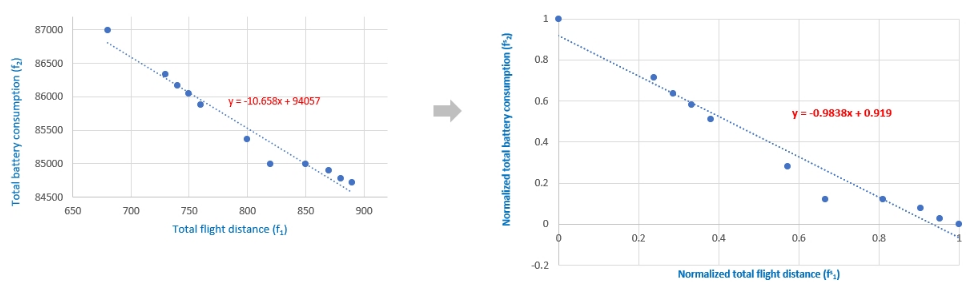


Figure 4.17: Result of Scale normalization (Example structure 2)

4.5.3 Numerical results by bi-objective optimization

In this section, bi-objective solution for each situation are suggested by the 'weighted method'. We found a bi-objective solution for a total of 101 scenarios by increasing or decreasing the relative weight by 0.01 units for each objective function whose scale was normalized. Through this method, if multiple solutions on the efficient frontier are explored, this will be an option to choose a route according to the preferences of the drone user. The detailed results of implementing the weighted method for example structures 1 and 2 are described in Appendix F.

Table 4.5: Bi-objective solution for each scenario (Example structure 1)

w_1	w_2	**Bi-objective route (Example structure 1)**	
0.00 $\sim$ 0.30	1.00 $\sim$ 0.70	Route 5	Total flight distance: 820 *meters* Total battery consumption: 82,089 *joules*
0.30 $\sim$ 0.53	0.69 $\sim$ 0.47	Route 4	Total flight distance: 780 *meters* Total battery consumption: 85,050 *joules*
0.54 $\sim$ 0.56	0.46 $\sim$ 0.44	Route 2	Total flight distance: 700 *meters* Total battery consumption: 100,359 *joules*
0.57 $\sim$ 1.00	0.43 $\sim$ 0.00	Route 1	Total flight distance: 660 *meters* Total battery consumption: 109,090 *joules*

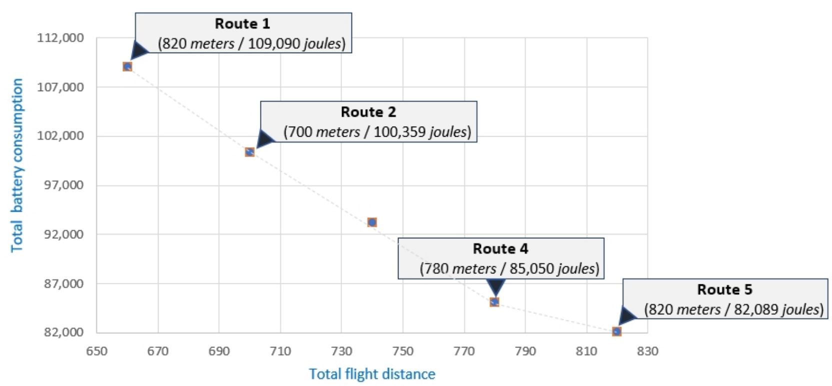


Figure 4.18: Flot chart of Bi-objective solutions (Example structure 1)

Table 4.5 and Figure 4.18 is the result of implementing a bi-objective optimization model for 5 feasible solutions (routes) that can inspect example structure 1. In the table 4.5, w_1 and w_2 represent weights for the scale normalized objective functions (f_i^s) of the SFD model and MEC model, respectively. For example, if w_1 is 0.00 and w_2 is 1.00, this is a scenario that focuses solely on minimizing total battery consumption. If w_1 and w_2 are 0.50 and 0.50 respectively, this means that reducing total flight distance and reducing total battery consumption are of equal importance. The route shown on the right refers to the bi-objective solution identified for each weighting scenario. Detailed information about each route (flight sequence, etc.) can be found in Appendix D. As you can see in the table on the above, we were able to find a total of 4 bi-objective solutions for a total of 101 weighted scenarios. Figure 4.18 shows 4 bi-objective solutions and Pareto-frontier out of a total of 5 feasible solutions. Next, as a result of implementing the weighted method for 11 feasible solutions that inspect example structure 2 (wide-cuboid type), we found 3 trade-off solutions (see Figure 4.19).

Table 4.6: Bi-objective solution for each scenario (Example structure 2)

w_1	w_2	**Bi-objective route (Example structure 2)**	
0.00 $\sim$ 0.21	1.00 $\sim$ 0.79	Route 11	Total flight distance: 890 *meters* Total battery consumption: 84,719 *joules*
0.22 $\sim$ 0.57	0.78 $\sim$ 0.43	Route 7	Total flight distance: 820 *meters* Total battery consumption: 84,932 *joules*
0.58 $\sim$ 1.00	0.42 $\sim$ 0.00	Route 1	Total flight distance: 680 *meters* Total battery consumption: 86,994 *joules*

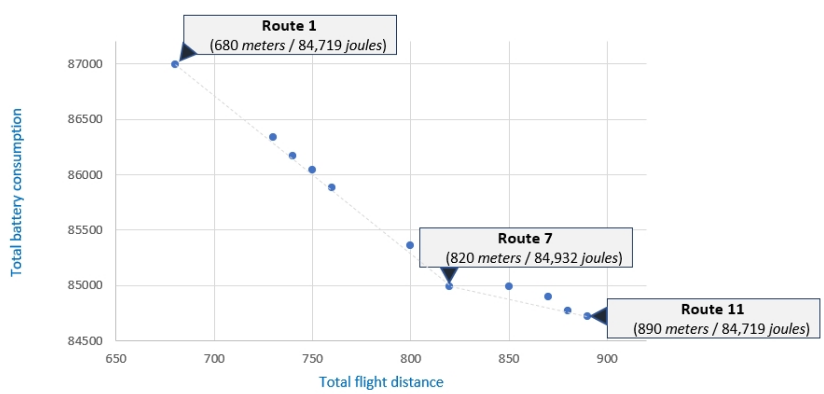


Figure 4.19: Plot chart of Bi-objective solutions (Example structure 2)

Although the example structure 2 have more number of feasible solutions (routes) for inspection than that of example 1 (Example 1: 5 routes, Example 2: 11 routes), the reason for the difference in the number of bi-objective optimal solutions can be inferred as follows. Previously, among feasible solutions of example structure 1, the difference in ratio between maximum total battery consumption and minimum total battery consumption is about 24.75 *percent* ((max battery - min battery) / max battery). And the difference between the ratio of the maximum total flight distance and the minimum total flight distance is about 19.50 *percent* ((max distance - min distance) / max distance). For example structure 2, the ratio difference between the maximum total flight distance and minimum total flight distance among feasible solutions is about 23.59 *percent*. On the other hand, the ratio difference between the maximum total battery consumption and the minimum total battery consumption is about 2.42 *percent*. In other words, in the case of a wide-cuboid type structure, no matter how much the amount of energy is saved, it is 2.42 *percent* of total battery consumption. Therefore, in terms of bi-objective optimization, even if the battery consumption increases slightly, it may be more beneficial to improve the total flight distance even more. Therefore, even a small increase in w_1, the bi-objective solution tends to reduce the total flight distance.

Table 4.7: Example of weight(w_i) determination

More important		**Conversion**		**Recommended Route**
Distance	Battery	w_1	w_2	
20 *percent*	-	0.545	0.455	Route 2 (with 700 *meters*, 100,359 *joules*)
-	50 *percent*	0.400	0.600	Route 1 (with 660 *meters*, 109,090 *joules*)

Previously, 4 and 3 bi-objective solutions for the example structures 1 and 2 found through the weighted method can be options for drone users to choose routes according to their operating conditions or preferences. However, one of the drawbacks of the weighted method is that it is not easy for the user to determine the appropriate weight for each objective function [61]. Instead, drone users will be able to judge 'which criterion is how much more important than some other criterion'. And Table 4.7 is an example that can calculate each weight according to the user's judgment. For example, as a result of judging the operational conditions for inspecting example structure 1, it is assumed that reducing the total flight distance is 20 *percent* more important. In this case, w_1 is converted to a weight of

0.565 (= 1.2/(1.2+1.0)) and w_2 is 0.455 (= 1.0/(1.2+1.0)). And Route 2 corresponding to this weighted section can be suggested as a route suitable for the preference of drone users.

4.6 Conclusion

Existing drone routing models mainly focused on minimizing flight distance (time) to overcome the limitation of short battery duration of small drones. In this study, we have shown that such models may not be sufficient in inspecting 3D structures such as bridges because drone energy consumption rate varies depending on flight dynamics, drone design, and environmental factors. We have extensively discussed the physics of battery consumption patterns depending on the type of moves and the weather conditions. Based on this, we developed a model for minimizing total energy consumption (MEC) and a model for minimizing total flight distance (SFD) while the drone inspects a three-dimensional structure. As a result, the optimal solutions of these two models were clearly different. Furthermore, total flight distance and total battery consumption have an inverse relationship. As a method to compromise these two criteria that cannot be minimized at the same time, a bi-objective optimization based on the weighting method was performed.

The results implies that the optimal solution derived by existing routing models that minimizes the flight distance (time) could be infeasible solutions in real life. For example, in the problem that a drone has to cover a wider three-dimensional space than the structures depicted in the case study in Section 4.4.3, the optimal solution by the SFD model can consume an amount of energy exceeding the maximum battery capacity of the drone. In this conjunction, the proposed path planning, taking into account the various flight dynamics and environmental conditions that a drone may have during its mission, will be a realistic approach to overcoming the battery limitations.

Chapter 5

Summary and Future Work

This thesis aims to ultimately reduce human damage and improve public safety by applying the drone routing or scheduling problem of drones to dangerous missions directly related to human life.

To achieve this goal, Chapter 3 proposed applying the Mothersip and Drone aided Routing Problem (MDRP) to medical testing kit delivery. First of all, our MDRP guarantees maximum social distancing by applying a 'CVP-D'(Carrier Vehicle Problem with Drones) model dedicated to delivery of multiple drones based on a vehicle. In order to find a high-quality solution, we first optimize the truck's travel schedule (TS), and then optimize the drone's flight schedule (FS) based on the truck's path. In a case study of 40 patients, our proposed model completes the service for all patients in 98 *minutes*. This is about 49 *percent* faster than the face-to-face diagnostics (198 *minutes*). In addition, the heuristic algorithm designed to implement this model reduces the computational time to find the optimal solution by 125 *seconds*. Therefore, it fully supports rapid decision making. Best of all, this system reduces the 'R'-number(Basic Reproduction number) to approximately 7.5 times compared to that of face-to-face diagnostics by blocking human contact opportunities. In other words, this method significantly reduces the risk of infection to a level of 13 *percent* than that of face-to-face diagnosis .

Chapter 4, proposed a model that inspects old and large structures using drones. In particular, the biggest obstacle that hinders the use of drones in this field is the short battery duration. In order to explore an energy-efficient inspection route that overcomes this shortcoming, we first review a drone's energy consumption patterns according to the flight dynamics and design of the drone and environmental factors. Based on the different BCR(Battery Consumption Rate) for each flight status, we developed 'energy consumption models' for each flight condition in which these patterns were considered in an integrated manner. The two MIP models designed based on these energy consumption models explore the inspection route to cover the checkpoints distributed in the 3-dimensional space with the shortest flight distance and the inspection route to cover the minimum battery consumption. And their optimal solutions are clearly different from each other, and rather, we have seen that the total energy consumption increases as the flight distance improves(reduced). The highlight of this study is to find a solution that compromises these two criteria through bi-objective optimization. This is an approach that has never been tried in previous studies. We could find multiple bi-objective solutions on the efficient frontier through a total of 101 weighted scenarios. This can be of great help in making rational decisions by allowing drone users to choose a route that suits their preferences and operating conditions.

The two studies proposed in this thesis can expand their use. The testing kit delivery model can be used in various medical supply fields. In particular, as diagnostic technologies advances, samples of blood or saliva and urine can provide detailed information on a person's health status or genetic information. The application of our drones and truck-aided delivery models can be extended to these areas(see Figure 5.1). However, our study mainly focuses on 'delivering' within the minimum time, because at-home testing kits for Covid-19 diagnosis allows patients to directly read the results. On the other hand, for diagnosis by blood, etc., the sample must be picked up again. Naturally, the scheduling or routing model also needs to modified for drones to perform both delivering and picking-up.

Figure 5.1: Blood sample delivery and pickup by drones (*sources*: Matternet)

The inspection routing model considering various energy consumption factors discussed in this thesis can be used not only for structure inspection, but also in missions where drones must mainly fly in vertical space. However, as mentioned in Sections 4.3.1.3 and 4.3.2.2, there are still some factors for which the energy consumption pattern is still unclear (e.g. BCR by diagonal ascending/descending or BCR by crosswind speed). If these are further reflected in the energy consumption models, we will be able to find a more realistic and pragmatic energy-efficient route for drones to cover 3-dimensional space.

References

[1] G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte, "Drone-aided routing: A literature review," *Transportation Research Part C: Emerging Technologies*, vol. 120, p. 102762, 2020.

[2] M. Rivera-Izquierdo, M. D. C. Valero-Ubierna, S. Martínez-Diz, M. Á. Fernández-García, D. T. Martín-Romero, F. Maldonado-Rodríguez, M. R. Sánchez-Pérez, L. M. Martín-delosReyes, V. Martínez-Ruiz, P. Lardelli-Claret *et al.*, "Clinical factors, preventive behaviours and temporal outcomes associated with covid-19 infection in health professionals at a spanish hospital," *International journal of environmental research and public health*, vol. 17, no. 12, p. 4305, 2020.

[3] M. Ki, "2015 mers outbreak in korea: hospital-to-hospital transmission," *Epidemiology and health*, vol. 37, 2015.

[4] "Icn confirms 1,500 nurses have died from covid-19 in 44 countries and estimates that healthcare worker covid-19 fatalities worldwide could be more than 20,000," 2020.

[5] 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.

[6] S.-H. Park and C.-H. Baek, "Strategies for establishing sustainable safety inspection

systems for existing multiple-unit dwellings in korea," *Journal of the Korea Institute of Building Construction*, vol. 13, no. 3, pp. 306–320, 2013.

[7] G. F. Burdge, M. Wu, K. Veal *et al.*, "Evaluation of osha fatality investigations," *Professional Safety*, vol. 64, no. 09, pp. 39–43, 2019.

[8] J. Wells and B. Lovelace, "Improving the quality of bridge inspections using unmanned aircraft systems (uas)," Tech. Rep., 2018.

[9] E. Mahase, "Covid-19: What is the r number?" 2020.

[10] 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.

[11] H. Savuran and M. Karakaya, "Efficient route planning for an unmanned air vehicle deployed on a moving carrier," *Soft Computing*, vol. 20, no. 7, pp. 2905–2920, 2016.

[12] 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.

[13] M. S. bin Othman, A. Shurbevski, Y. Karuno, and H. Nagamochi, "Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle and fixed carrier route," *Journal of Information Processing*, vol. 25, pp. 655–666, 2017.

[14] C. Gambella, A. Lodi, and D. Vigo, "Exact solutions for the carrier–vehicle traveling salesman problem," *Transportation Science*, vol. 52, no. 2, pp. 320–330, 2018.

[15] N. Boysen, D. Briskorn, S. Fedtke, and S. Schwerdfeger, "Drone delivery from trucks: Drone scheduling for given truck routes," *Networks*, vol. 72, no. 4, pp. 506–527, 2018.

[16] O. Dukkanci, B. Y Kara, and T. Bektas, "The drone delivery problem," *Tolga, The Drone Delivery Problem (January 10, 2019)*, 2019.

[17] A. Karak and K. Abdelghany, "The hybrid vehicle-drone routing problem for pick-up and delivery services," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 427–449, 2019.

[18] J. Wikarek, P. Sitek, and L. Zawarczyński, "An integer programming model for the capacitated vehicle routing problem with drones," in *International Conference on Computational Collective Intelligence*. Springer, 2019, pp. 511–520.

[19] X. Bai, M. Cao, W. Yan, and S. S. Ge, "Efficient routing for precedence-constrained package delivery for heterogeneous vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 248–260, 2019.

[20] S. Poikonen and B. Golden, "The mothership and drone routing problem," *INFORMS Journal on Computing*, vol. 32, no. 2, pp. 249–262, 2020.

[21] M. Moeini and H. Salewski, "A genetic algorithm for solving the truck-drone-atv routing problem," in *World Congress on Global Optimization*. Springer, 2019, pp. 1023–1032.

[22] G. H. Elkaim, F. A. P. Lie, and D. Gebre-Egziabher, "Principles of guidance, navigation, and control of uavs," *Handbook of Unmanned Aerial Vehicles*, pp. 347–380, 2015.

[23] E. Demir, T. Bektaş, and G. Laporte, "A review of recent research on green road freight transportation," *European Journal of Operational Research*, vol. 237, no. 3, pp. 775–793, 2014.

[24] 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.

[25] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of uavs," in *2015 IEEE international conference on autonomous robot systems and competitions*. IEEE, 2015, pp. 111–117.

[26] T. Dietrich, S. Krug, and A. Zimmermann, "An empirical study on generic multicopter energy consumption profiles," in *2017 Annual IEEE International Systems Conference (SysCon)*. IEEE, 2017, pp. 1–6.

[27] Y. Chen, D. Baek, A. Bocca, A. Macii, E. Macii, and M. Poncino, "A case for a battery-aware model of drone energy consumption," in *2018 IEEE International Telecommunications Energy Conference (INTELEC)*. IEEE, 2018, pp. 1–8.

[28] R. D'Andrea, "Guest editorial can drones deliver?" *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 647–648, 2014.

[29] M. A. Figliozzi, "Lifecycle modeling and assessment of unmanned aerial vehicles (drones) co2e emissions," *Transportation Research Part D: Transport and Environment*, vol. 57, pp. 251–261, 2017.

[30] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.

[31] A. J. Lohn, "What's the buzz? the city-scale impacts of drone delivery," Tech. Rep., 2017.

[32] J. Xu, *Design perspectives on delivery drones*. RAND, 2017.

[33] R. Felismina, M. Silva, A. Mateus, and C. Malça, "Study on the aerodynamic behavior of a uav with an applied seeder for agricultural practices," in *AIP Conference Proceedings*, vol. 1836, no. 1. AIP Publishing LLC, 2017, p. 020049.

[34] J. Hernandez, M. Alegre, D. Siljegovic, R. Calong, F. Attarzadeh, A. Shukla, and V. Naik, "Network controlled data acquisition drone," in *Proceedings of the 2008 ASEE Gulf-Southwest Annual Conference, The University of New Mexico–Albuquerque, NM, March 26*, vol. 28, no. 1, 2008.

[35] 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. 70–85, 2016.

[36] A. Troudi, S.-A. Addouche, S. Dellagi, and A. E. Mhamedi, "Sizing of the drone delivery fleet considering energy autonomy," *Sustainability*, vol. 10, no. 9, p. 3344, 2018.

[37] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones," *International Journal of Production Economics*, vol. 214, pp. 220–233, 2019.

[38] 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.

[39] C.-M. Tseng, C.-K. Chau, K. Elbassioni, and M. Khonji, "Autonomous recharging and flight mission planning for battery-operated autonomous drones," *arXiv preprint arXiv:1703.10049*, 2017.

[40] J. A. Guerrero and Y. Bestaoui, "Uav path planning for structure inspection in windy

environments," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 297–311, 2013.

[41] L. Lan, D. Xu, G. Ye, C. Xia, S. Wang, Y. Li, and H. Xu, "Positive rt-pcr test results in patients recovered from covid-19," *Jama*, vol. 323, no. 15, pp. 1502–1503, 2020.

[42] J. Chen, "Pathogenicity and transmissibility of 2019-ncov—a quick overview and comparison with other emerging viruses," *Microbes and infection*, vol. 22, no. 2, pp. 69–71, 2020.

[43] G. Chowell, H. Nishiura, and L. M. Bettencourt, "Comparative estimation of the reproduction number for pandemic influenza from daily case notification data," *Journal of the Royal Society Interface*, vol. 4, no. 12, pp. 155–166, 2007.

[44] K. Dietz, "The estimation of the basic reproduction number for infectious diseases," *Statistical methods in medical research*, vol. 2, no. 1, pp. 23–41, 1993.

[45] Z. Tang, X. Li, and H. Li, "Prediction of new coronavirus infection based on a modified seir model," *medRxiv*, 2020.

[46] P. Diaz, P. Constantine, K. Kalmbach, E. Jones, and S. Pankavich, "A modified seir model for the spread of ebola in western africa and metrics for resource allocation," *Applied Mathematics and Computation*, vol. 324, pp. 141–155, 2018.

[47] G. Van Rossum and F. L. Drake, "Python 3 reference manual," 2021. [Online]. Available: https://www.python.org/

[48] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com

[49] D. K. Chu, E. A. Akl, S. Duda, K. Solo, S. Yaacoub, H. J. Schünemann, A. El-harakeh, A. Bognanni, T. Lotfi, M. Loeb *et al.*, "Physical distancing, face masks, and

eye protection to prevent person-to-person transmission of sars-cov-2 and covid-19: a systematic review and meta-analysis," *The Lancet*, 2020.

[50] N. Vargas-Ramírez and J. Paneque-Gálvez, "The global emergence of community drones (2012–2017)," *Drones*, vol. 3, no. 4, p. 76, 2019.

[51] J. Zhang, J. F. Campbell, D. C. Sweeney II, and A. C. Hupman, "Energy consumption models for delivery drones: A comparison and assessment," *Transportation Research Part D: Transport and Environment*, vol. 90, p. 102668, 2020.

[52] M. R. Saleem, "Instant bridge visual inspection using a uav by image capturing and geo-tagging system and deep convolutional neural network," Ph.D. dissertation, Department of Structural Engineering, Norwegian University of Science and . . . .

[53] S. Yoon, G.-H. Gwon, J.-H. Lee, and H.-J. Jung, "Three-dimensional image coordinate-based missing region of interest area detection and damage localization for bridge visual inspection using unmanned aerial vehicles," *Structural Health Monitoring*, p. 1475921720918675, 2020.

[54] P. Lesiak, "Inspection and maintenance of railway infrastructure with the use of unmanned aerial vehicles," *Problemy Kolejnictwa*, 2020.

[55] H. D. Sherali and A. L. Soyster, "Preemptive and nonpreemptive multi-objective programming: Relationship and counterexamples," *Journal of Optimization Theory and Applications*, vol. 39, no. 2, pp. 173–186, 1983.

[56] M. C.-R. de Matos Fernandes, "Airborne wind energy systems: Modelling, simulation and economic analysis," 2018.

[57] S.-I. Kim, J. Cho, S. Park, T. Park, and S. Lim, "Characteristics comparison of a

conventional and modified spoke-type ferrite magnet motor for traction drives of low-speed electric vehicles," *IEEE Transactions on Industry Applications*, vol. 49, no. 6, pp. 2516–2523, 2013.

[58] B. H. Wang, D. B. Wang, Z. A. Ali, B. Ting Ting, and H. Wang, "An overview of various kinds of wind effects on unmanned aerial vehicle," *Measurement and Control*, vol. 52, no. 7-8, pp. 731–739, 2019.

[59] R. Citroni, F. Di Paolo, and P. Livreri, "A novel energy harvester for powering small uavs: Performance analysis, model validation and flight results," *Sensors*, vol. 19, no. 8, p. 1771, 2019.

[60] A. Zhou, L. Kang, and Z. Yan, "Solving dynamic tsp with evolutionary approach in real time," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 2. IEEE, 2003, pp. 951–957.

[61] O. B. Augusto, F. Bennis, and S. Caro, "A new method for decision making in multi-objective optimization problems," *Pesquisa Operacional*, vol. 32, no. 2, pp. 331–369, 2012.

[62] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[63] I. Y. Kim and O. L. De Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural and multidisciplinary optimization*, vol. 29, no. 2, pp. 149–158, 2005.

[64] A. Charnes and W. W. Cooper, "Goal programming and multiple objective optimizations: Part 1," *European journal of operational research*, vol. 1, no. 1, pp. 39–54, 1977.

[65] S. A. Marglin, *Public investment criteria (routledge revivals): Benefit-cost analysis for planned economic growth*. Routledge, 2014.

[66] R. E. Steuer, “Multiple criteria optimization,” *Theory, computation and applications*, 1986.

[67] N. Vafaei, R. A. Ribeiro, and L. M. Camarinha-Matos, “Normalization techniques for multi-criteria decision making: analytical hierarchy process case study,” in *doctoral conference on computing, electrical and industrial systems*. Springer, 2016, pp. 261–269.

[68] S. Patro and K. K. Sahu, “Normalization: A preprocessing stage,” *arXiv preprint arXiv:1503.06462*, 2015.

Appendices

Appendix A

Launch-Return Location Optimization Model

LRO chooses the best launch and return nodes for a current sets of patients in a way that minimizes the MDRP objective function. The difference between this model and the MDRP is that the P, SP are determined. This model mainly performs to optimize the launch, return and drone assignment to enhance the waiting time for both truck and drones. The main usage of this model, is at the early stage, when the heuristics find a solution, so *LRO* prevents generating any infeasible solution by algorithms.

$$\min_{f,w,h} \quad \text{LRO} = \sum_{(l,i,r)\in LFR} t^i_{l,r} f^i_{l,r} + \sum_{i\in P} w_i + \sum_{i\in N} h_i \tag{A.1}$$

s.t.

$$(3.7), (4.5), (3.35) \tag{A.2}$$

$$T_{p_r} - T_{p_l} + \sum_{k=l+1}^{r-1} w_{p_k} - M(1 - f^j_{p_l,p_r}) \leq t^j_{p_l,p_r} + \sum_{i\in S_j} h_i \quad \forall (p_l, j, p_r) \in FLR \tag{A.3}$$

$$t^j_{p_l,p_r} + \sum_{i\in S_j} h_i - M(1 - f^j_{p_l,p_r}) \leq T_{p_r} - T_{p_l} + \sum_{k=l}^{r} w_{p_k} \quad \forall (p_l, j, p_r) \in FLR \tag{A.4}$$

$$D_{p_{i-1}} - D_{p_i} = \sum_{(l,j,p_i)\in LFR} f^j_{l,p_i} - \sum_{(p_i,j,r)\in LFR} f^j_{p_i,r} \;, i = 2, ..., |P| \tag{A.5}$$

$$D_C + \sum_{(C,j,r)\in LFR} f^j_{C,r} = n \tag{A.6}$$

Appendix B

Specification of the drone in Numerical experiment

Weight	Area		Battery Capacity		Max speed
	no inclination	inclination(15 *deg*)	Current	Voltage	
2.0 *kg* (+Payload)	0.0464 m^2	0.0613 m^2	5,500 *mAh*	14.8 *volt*	40 *km/hr*

Appendix C

Battery consumption rate by each flight status

	Headwind (m/s)				Back(tail) wind (m/s)			
Flight Speed	0	2	4	6	0	2	4	6
horizontal(V_h)								
0 m/s	220.0	220.4	223.2	230.8	220.0	219.6	216.8	209.2
1 m/s	219.0	219.4	222.2	229.8	219.0	218.6	215.8	208.2
2 m/s	218.0	218.4	221.2	228.8	218.0	217.6	214.8	207.2
3 m/s	216.0	216.4	219.2	226.8	216.0	215.6	212.8	205.2
4 m/s	213.0	213.4	216.2	223.8	213.0	212.6	209.8	202.2
5 m/s	210.0	210.4	213.2	220.8	210.0	209.6	206.8	199.2
6 m/s	207.0	207.4	210.2	217.8	207.0	206.6	203.8	196.2
7 m/s	205.0	205.4	208.2	215.8	205.0	204.6	201.8	204.2
8 m/s	207.0	207.4	210.2	217.8	207.0	206.6	203.8	196.2
9 m/s	211.0	211.4	214.2	221.8	211.0	210.6	207.8	200.2
10 m/s	216.0	216.4	219.2	226.8	216.0	215.6	212.8	205.2
11 m/s	224.0	224.4	227.2	234.8	224.0	223.6	220.8	213.2
12 m/s	240.0	240.4	243.2	250.8	240.0	239.6	236.8	229.2
13 m/s	262.0	262.4	265.2	272.8	262.0	261.6	258.8	251.2
14 m/s	296.0	296.4	299.2	306.8	296.0	295.6	292.8	285.2
Ascending(V_a)								
2 m/s	250.0	250.2	251.7	255.9	250.0	250.2	251.7	255.9
5 m/s	364.0	364.2	365.7	369.9	364.0	364.2	365.7	369.9
Descending(V_d)								
1.5 m/s	213.0	213.2	214.7	218.9	213.0	213.2	214.7	218.9
4 m/s	299.0	299.2	300.7	304.9	299.0	299.2	300.7	304.9
Turning(V_r)								
120 deg/s	225.0	225.4	226.7	230.9	225.0	225.4	226.7	230.9

Appendix D

Feasible solutions for inspecting example structure 1 and 2

Example Structure 1 (Tall-building type)

No. of Route	Flight sequence	Total flight distance (meters)	Total battery consumption (joules)
Route 1	1 → 16 → 19 → 22 → 25 → 28 → 29 → 26 → 23→ 20 → 17 → 18 → 21 → 24 → 27 → 30 → 15 → 12 → 9 → 6 → 3 → 2 → 5 → 8 → 11 → 14 → 13 → 10 → 7 → 4 → 1	**660**	**109,090**
Route 2	1 → 4 → 7 → 8 → 5 → 2 → 3 → 6 → 9 → 12 → 11 → 10 → 13 → 14 → 15 → 30 → 27 → 24 → 21 → 18 → 17 → 20 → 23 → 26 → 29 → 28 → 25 → 22 → 19 → 16 → 1	**700**	**100,359**
Route 3	1 → 16 → 19 → 20 → 17 → 18 → 21 → 24 → 27 → 26 → 23 → 22 → 25 → 28 → 29 → 30 → 15 → 12 → 11 → 14 → 13 → 10 → 7 → 8 → 9 → 6 → 3 → 2 → 5 → 4 → 1	**740**	**93,151**
Route 4	1 → 16 → 17 → 18 → 21 → 20 → 19 → 22 → 23 → 24 → 27 → 26 → 25 → 28 → 29 → 30 → 15 → 12 → 11 → 14 → 13 → 10 → 7 → 8 → 9 → 6 → 3 → 2 → 5 → 4 → 1	**780**	**85,050**
Route 5	1 → 4 → 5 → 6 → 21 → 20 → 23 → 26 → 25→ 10 → 11 → 12 → 15 → 14 → 13 → 28 → 29 → 30 → 27 → 24 → 9 → 8 → 7 → 22 → 19 → 16 → 17 → 18 → 3 → 2 → 1	**820**	**82,089**

Example Structure 2 (Wide-cuboid type)

No of Route	Flight sequence	Total flight distance (meters)	Total battery consumption (joules)
Route 1	1 → 5 → 6 → 7 → 8 → 12 → 11 → 10 → 9→ 13 → 14 → 15 → 16 → 32 → 31 → 30 → 29 → 25 → 26 → 27 → 28 → 24 → 23 → 22 → 21 → 17 → 18 → 19 → 20 → 4 → 3 → 2 → 1	**680**	**86,994**
Route 2	1 → 5 → 6 → 10 → 9 → 13 → 14 → 15 → 16 → 32 → 31 → 30 → 29 → 25 → 26 → 27 → 28 → 12 → 11 → 7 → 8 → 24 → 23 → 22 → 21 → 17 → 18 → 19 → 20 → 4 → 3 → 2 → 1	**730**	**86,340**
Route 3	1 → 5 → 6 → 7 → 11 → 10 → 9 → 13 → 14 → 15 → 16 → 32 → 31 → 30 → 29 → 25 → 26 → 27 → 28 → 12 → 8 → 24 → 23 → 22 → 21 → 17 → 18 → 19 → 20 → 4 → 3 → 2 → 1	**740**	**86,166**
Route 4	1 → 2 → 3 → 4 → 20 → 19 → 18 → 17 → 21 → 22 → 23 → 24 → 8 → 7 → 6 → 10 → 11 → 12 → 28 → 27 → 26 → 25 → 29 → 30 → 31 → 32 → 16 → 15 → 14 → 13 → 9 → 5 → 1	**750**	**86,044**
Route 5	1 → 5 → 9 → 13 → 14 → 15 → 16 → 32 → 31 → 30 → 29 → 25 → 26 → 27 → 28 → 12 → 11 → 10 → 6 → 7 → 8 → 24 → 23 → 22 → 21 → 17 → 18 → 19 → 20 → 4 → 3 → 2 → 1	**760**	**85,880**
Route 6	1 → 2 → 3 → 7 → 6 → 5 → 9 → 10 → 11 → 15 → 14 → 13 → 29 → 30 → 31 → 32 → 16 → 12 → 28 → 27 → 26 → 25 → 21 → 22 → 23 → 24 → 8 → 4 → 20 → 19 → 18 → 17 → 1	**800**	**85,359**
Route 7	1 → 5 → 9 → 13 → 29 → 30 → 31 → 32 → 16 → 15 → 14 → 10 → 11 → 12 → 28 → 27 → 26 → 25 → 21 → 22 → 23 → 24 → 8 → 7 → 6 → 2 → 3 → 4 → 20 → 19 → 18 → 17 → 1	**820**	**84,932**
Route 8	1 → 17 → 18 → 19 → 20 → 4 → 3 → 7 → 8 → 24 → 23 → 22 → 21 → 5 → 9 → 25 → 26 → 27 → 28 → 12 → 11 → 15 → 16 → 32 → 31 → 30 → 29 → 13 → 14 → 10 → 6 → 2 → 1	**850**	**84,890**
Route 9	1 → 2 → 3 → 4 → 8 → 7 → 6 → 5 → 9 → 10 → 11 → 12 → 16 → 15 → 14 → 13 → 29 → 30 → 31 → 32 → 28 → 24 → 20 → 19 → 23 → 27 → 26 → 25 → 21 → 22 → 18 → 17 → 1	**870**	**84,899**
Route 10	1 → 2 → 3 → 4 → 8 → 7 → 6 → 5 → 9 → 10 → 11 → 12 → 16 → 15 → 14 → 13 → 29 → 30 → 31 → 32 → 28 → 27 → 23 → 24 → 20 → 19 → 18 → 22 → 26 → 25 → 21 → 17 → 1	**880**	**84,776**
Route 11	1 → 17 → 18 → 22 → 6 → 5 → 21 → 25 → 9→ 10 → 26 → 30 → 29 → 13 → 14 → 15 → 16 → 32 → 31 → 27 → 28 → 12 → 11 → 7 → 8 → 24 → 23 → 19 → 20 → 4 → 3 → 2 → 1	**890**	**84,719**

Appendix E

Computational results of Scale normalization

	Example structure 1 (Tall-building type)				**Example structure 2** (Wide-cuboid type)			
	Original Value (f_i)		Scale normalized Value (f_i^s)		Original Value (f_i)		Scale normalized Value (f_i^s)	
	Distance (*meters*)	Battery (*joules*)	Distance	Battery	Distance (*meters*)	Battery (*joules*)	Distance	Battery
Route 1	660	109,090	0.0000	1.0000	680	86,994	0.0000	1.0000
Route 2	700	100,359	0.2500	0.6774	730	86,340	0.2381	0.7125
Route 3	740	93,151	0.5000	0.4110	740	86,166	0.2857	0.6360
Route 4	780	85,050	0.7500	0.1116	750	86,044	0.3333	0.5824
Route 5	820	82,029	1.0000	0.0000	760	85,880	0.3810	0.5103
Route 6					800	85,359	0.5714	0.2813
Route 7					820	84,992	0.6667	0.1200
Route 8					850	84,990	0.8095	0.1191
Route 9					870	84,899	0.9048	0.0791
Route 10					880	84,776	0.9524	0.0251
Route 11					890	84,719	1.0000	0.0000

Appendix F

Bi-objective solutions for each scenario

For $f_i^s(x)$		Feasible solutions for example 1 (Tall-building type)					Feasible solutions for example 2 (Wide-cuboid type)										
w_1	w_2	**Route 1**	**Route 2**	**Route 3**	**Route 4**	**Route 5**	**Route 1**	**Route 2**	**Route 3**	**Route 4**	**Route 5**	**Route 6**	**Route 7**	**Route 8**	**Route 9**	**Route 10**	**Route 11**
0.00	**1.00**	1.0000	0.6774	0.4110	0.1116	0.0000	1.0000	0.7125	0.6360	0.5824	0.5103	0.2813	0.0936	0.1191	0.0791	0.0251	0.0000
0.01	**0.99**	0.9900	0.6731	0.4119	0.1180	0.0100	0.9900	0.7078	0.6325	0.5799	0.5090	0.2842	0.0994	0.1260	0.0874	0.0343	0.0100
0.02	**0.98**	0.9800	0.6688	0.4128	0.1244	0.0200	0.9800	0.7030	0.6290	0.5774	0.5077	0.2871	0.1051	0.1329	0.0956	0.0436	0.0200
0.03	**0.97**	0.9700	0.6645	0.4137	0.1308	0.0300	0.9700	0.6983	0.6255	0.5750	0.5065	0.2900	0.1108	0.1398	0.1039	0.0529	0.0300
0.04	**0.96**	0.9600	0.6603	0.4146	0.1372	0.0400	0.9600	0.6936	0.6220	0.5725	0.5052	0.2929	0.1165	0.1467	0.1122	0.0622	0.0400
0.05	**0.95**	0.9500	0.6560	0.4154	0.1436	0.0500	0.9500	0.6888	0.6185	0.5700	0.5039	0.2958	0.1223	0.1536	0.1204	0.0714	0.0500
0.06	**0.94**	0.9400	0.6517	0.4163	0.1499	0.0600	0.9400	0.6841	0.6150	0.5675	0.5026	0.2987	0.1280	0.1605	0.1287	0.0807	0.0600
0.07	**0.93**	0.9300	0.6474	0.4172	0.1563	0.0700	0.9300	0.6793	0.6115	0.5650	0.5013	0.3016	0.1337	0.1674	0.1369	0.0900	0.0700
0.08	**0.92**	0.9200	0.6432	0.4181	0.1627	0.0800	0.9200	0.6746	0.6080	0.5625	0.5000	0.3045	0.1395	0.1744	0.1452	0.0992	0.0800
0.09	**0.91**	0.9100	0.6389	0.4190	0.1691	0.0900	0.9100	0.6698	0.6045	0.5600	0.4987	0.3074	0.1452	0.1813	0.1534	0.1085	0.0900
0.10	**0.90**	0.9000	0.6346	0.4199	0.1755	0.1000	0.9000	0.6651	0.6010	0.5575	0.4974	0.3103	0.1509	0.1882	0.1617	0.1178	0.1000
0.11	**0.89**	0.8900	0.6303	0.4208	0.1819	0.1100	0.8900	0.6603	0.5975	0.5550	0.4961	0.3132	0.1567	0.1951	0.1699	0.1271	0.1100
0.12	**0.88**	0.8800	0.6261	0.4217	0.1882	0.1200	0.8800	0.6556	0.5940	0.5525	0.4948	0.3161	0.1624	0.2020	0.1782	0.1363	0.1200
0.13	**0.87**	0.8700	0.6218	0.4226	0.1946	0.1300	0.8700	0.6509	0.5905	0.5500	0.4935	0.3190	0.1681	0.2089	0.1865	0.1456	0.1300
0.14	**0.86**	0.8600	0.6175	0.4235	0.2010	0.1400	0.8600	0.6461	0.5870	0.5476	0.4922	0.3219	0.1739	0.2158	0.1947	0.1549	0.1400
0.15	**0.85**	0.8500	0.6133	0.4243	0.2074	0.1500	0.8500	0.6414	0.5835	0.5451	0.4909	0.3248	0.1796	0.2227	0.2030	0.1642	0.1500
0.16	**0.84**	0.8400	0.6090	0.4252	0.2138	0.1600	0.8400	0.6366	0.5800	0.5426	0.4896	0.3277	0.1853	0.2296	0.2112	0.1734	0.1600
0.17	**0.83**	0.8300	0.6047	0.4261	0.2202	0.1700	0.8300	0.6319	0.5765	0.5401	0.4883	0.3306	0.1910	0.2365	0.2195	0.1827	0.1700
0.18	**0.82**	0.8200	0.6004	0.4270	0.2265	0.1800	0.8200	0.6271	0.5730	0.5376	0.4870	0.3335	0.1968	0.2434	0.2277	0.1920	0.1800
0.19	**0.81**	0.8100	0.5962	0.4279	0.2329	0.1900	0.8100	0.6224	0.5695	0.5351	0.4858	0.3364	0.2025	0.2503	0.2360	0.2013	0.1900
0.20	**0.80**	0.8000	0.5919	0.4288	0.2393	0.2000	0.8000	0.6176	0.5660	0.5326	0.4845	0.3393	0.2082	0.2572	0.2443	0.2105	0.2000
0.21	**0.79**	0.7900	0.5876	0.4297	0.2457	0.2100	0.7900	0.6129	0.5625	0.5301	0.4832	0.3422	0.2140	0.2641	0.2525	0.2198	0.2100
0.22	**0.78**	0.7800	0.5833	0.4306	0.2521	0.2200	0.7800	0.6082	0.5590	0.5276	0.4819	0.3451	0.2197	0.2710	0.2608	0.2291	0.2200
0.23	**0.77**	0.7700	0.5791	0.4315	0.2585	0.2300	0.7700	0.6034	0.5555	0.5251	0.4806	0.3480	0.2254	0.2779	0.2690	0.2383	0.2300
0.24	**0.76**	0.7600	0.5748	0.4324	0.2648	0.2400	0.7600	0.5987	0.5520	0.5226	0.4793	0.3510	0.2312	0.2848	0.2773	0.2476	0.2400
0.25	**0.75**	0.7500	0.5705	0.4332	0.2712	0.2500	0.7500	0.5939	0.5485	0.5202	0.4780	0.3539	0.2369	0.2917	0.2855	0.2569	0.2500
0.26	**0.74**	0.7400	0.5662	0.4341	0.2776	0.2600	0.7400	0.5892	0.5450	0.5177	0.4767	0.3568	0.2426	0.2986	0.2938	0.2662	0.2600
0.27	**0.73**	0.7300	0.5620	0.4350	0.2840	0.2700	0.7300	0.5844	0.5415	0.5152	0.4754	0.3597	0.2483	0.3055	0.3020	0.2754	0.2700
0.28	**0.72**	0.7200	0.5577	0.4359	0.2904	0.2800	0.7200	0.5797	0.5380	0.5127	0.4741	0.3626	0.2541	0.3124	0.3103	0.2847	0.2800
0.29	**0.71**	0.7100	0.5534	0.4368	0.2968	0.2900	0.7100	0.5749	0.5344	0.5102	0.4728	0.3655	0.2598	0.3193	0.3186	0.2940	0.2900
0.30	**0.70**	0.7000	0.5492	0.4377	0.3031	0.3000	0.7000	0.5702	0.5309	0.5077	0.4715	0.3684	0.2655	0.3262	0.3268	0.3033	0.3000
0.31	**0.69**	0.6900	0.5449	0.4386	0.3095	0.3100	0.6900	0.5655	0.5274	0.5052	0.4702	0.3713	0.2713	0.3331	0.3351	0.3125	0.3100
0.32	**0.68**	0.6800	0.5406	0.4395	0.3159	0.3200	0.6800	0.5607	0.5239	0.5027	0.4689	0.3742	0.2770	0.3401	0.3433	0.3218	0.3200
0.33	**0.67**	0.6700	0.5363	0.4404	0.3223	0.3300	0.6700	0.5560	0.5204	0.5002	0.4676	0.3771	0.2827	0.3470	0.3516	0.3311	0.3300
0.34	**0.66**	0.6600	0.5321	0.4413	0.3287	0.3400	0.6600	0.5512	0.5169	0.4977	0.4663	0.3800	0.2885	0.3539	0.3598	0.3404	0.3400
0.35	**0.65**	0.6500	0.5278	0.4421	0.3351	0.3500	0.6500	0.5465	0.5134	0.4952	0.4651	0.3829	0.2942	0.3608	0.3681	0.3496	0.3500
0.36	**0.64**	0.6400	0.5235	0.4430	0.3414	0.3600	0.6400	0.5417	0.5099	0.4928	0.4638	0.3858	0.2999	0.3677	0.3764	0.3589	0.3600
0.37	**0.63**	0.6300	0.5192	0.4439	0.3478	0.3700	0.6300	0.5370	0.5064	0.4903	0.4625	0.3887	0.3057	0.3746	0.3846	0.3682	0.3700
0.38	**0.62**	0.6200	0.5150	0.4448	0.3542	0.3800	0.6200	0.5322	0.5029	0.4878	0.4612	0.3916	0.3114	0.3815	0.3929	0.3774	0.3800
0.39	**0.61**	0.6100	0.5107	0.4457	0.3606	0.3900	0.6100	0.5275	0.4994	0.4853	0.4599	0.3945	0.3171	0.3884	0.4011	0.3867	0.3900
0.40	**0.60**	0.6000	0.5064	0.4466	0.3670	0.4000	0.6000	0.5228	0.4959	0.4828	0.4586	0.3974	0.3228	0.3953	0.4094	0.3960	0.4000
0.41	**0.59**	0.5900	0.5021	0.4475	0.3734	0.4100	0.5900	0.5180	0.4924	0.4803	0.4573	0.4003	0.3286	0.4022	0.4176	0.4053	0.4100
0.42	**0.58**	0.5800	0.4979	0.4484	0.3797	0.4200	0.5800	0.5133	0.4889	0.4778	0.4560	0.4032	0.3343	0.4091	0.4259	0.4145	0.4200
0.43	**0.57**	0.5700	0.4936	0.4493	0.3861	0.4300	0.5700	0.5085	0.4854	0.4753	0.4547	0.4061	0.3400	0.4160	0.4342	0.4238	0.4300
0.44	**0.56**	0.5600	0.4893	0.4502	0.3925	0.4400	0.5600	0.5038	0.4819	0.4728	0.4534	0.4090	0.3458	0.4229	0.4424	0.4331	0.4400
0.45	**0.55**	0.5500	0.4850	0.4510	0.3989	0.4500	0.5500	0.4990	0.4784	0.4703	0.4521	0.4119	0.3515	0.4298	0.4507	0.4424	0.4500
0.46	**0.54**	0.5400	0.4808	0.4519	0.4053	0.4600	0.5400	0.4943	0.4749	0.4678	0.4508	0.4148	0.3572	0.4367	0.4589	0.4516	0.4600
0.47	**0.53**	0.5300	0.4765	0.4528	0.4117	0.4700	0.5300	0.4895	0.4714	0.4654	0.4495	0.4177	0.3630	0.4436	0.4672	0.4609	0.4700
0.48	**0.52**	0.5200	0.4722	0.4537	0.4181	0.4800	0.5200	0.4848	0.4679	0.4629	0.4482	0.4206	0.3687	0.4505	0.4754	0.4702	0.4800
0.49	**0.51**	0.5100	0.4680	0.4546	0.4244	0.4900	0.5100	0.4801	0.4644	0.4604	0.4469	0.4235	0.3744	0.4574	0.4837	0.4794	0.4900
0.50	**0.50**	0.5000	0.4637	0.4555	0.4308	0.5000	0.5000	0.4753	0.4609	0.4579	0.4456	0.4264	0.3801	0.4643	0.4919	0.4887	0.5000

For $f_i^s(x)$		Feasible solutions for example 1 (Tall-building type)					Feasible solutions for example 2 (Wide-cuboid type)										
w_1	w_2	**Route 1**	**Route 2**	**Route 3**	**Route 4**	**Route 5**	**Route 1**	**Route 2**	**Route 3**	**Route 4**	**Route 5**	**Route 6**	**Route 7**	**Route 8**	**Route 9**	**Route 10**	**Route 11**
0.51	**0.49**	0.4900	0.4594	0.4564	0.4372	0.5100	0.4900	0.4706	0.4574	0.4554	0.4444	0.4293	0.3859	0.4712	0.5002	0.4980	0.5100
0.52	**0.48**	0.4800	0.4551	0.4573	0.4436	0.5200	0.4800	0.4658	0.4539	0.4529	0.4431	0.4322	0.3916	0.4781	0.5085	0.5073	0.5200
0.53	**0.47**	0.4700	0.4509	0.4582	0.4500	0.5300	0.4700	0.4611	0.4504	0.4504	0.4418	0.4351	0.3973	0.4850	0.5167	0.5165	0.5300
0.54	**0.46**	0.4600	0.4466	0.4591	0.4564	0.5400	0.4600	0.4563	0.4469	0.4479	0.4405	0.4380	0.4031	0.4919	0.5250	0.5258	0.5400
0.55	**0.45**	0.4500	0.4423	0.4599	0.4627	0.5500	0.4500	0.4516	0.4434	0.4454	0.4392	0.4409	0.4088	0.4988	0.5332	0.5351	0.5500
0.56	**0.44**	0.4400	0.4380	0.4608	0.4691	0.5600	0.4400	0.4468	0.4399	0.4429	0.4379	0.4438	0.4145	0.5057	0.5415	0.5444	0.5600
0.57	**0.43**	0.4300	0.4338	0.4617	0.4755	0.5700	0.4300	0.4421	0.4364	0.4404	0.4366	0.4467	0.4203	0.5127	0.5497	0.5536	0.5700
0.58	**0.42**	0.4200	0.4295	0.4626	0.4819	0.5800	0.4200	0.4374	0.4329	0.4380	0.4353	0.4496	0.4260	0.5196	0.5580	0.5629	0.5800
0.59	**0.41**	0.4100	0.4252	0.4635	0.4883	0.5900	0.4100	0.4326	0.4293	0.4355	0.4340	0.4525	0.4317	0.5265	0.5663	0.5722	0.5900
0.60	**0.40**	0.4000	0.4209	0.4644	0.4947	0.6000	0.4000	0.4279	0.4258	0.4330	0.4327	0.4554	0.4375	0.5334	0.5745	0.5815	0.6000
0.61	**0.39**	0.3900	0.4167	0.4653	0.5010	0.6100	0.3900	0.4231	0.4223	0.4305	0.4314	0.4583	0.4432	0.5403	0.5828	0.5907	0.6100
0.62	**0.38**	0.3800	0.4124	0.4662	0.5074	0.6200	0.3800	0.4184	0.4188	0.4280	0.4301	0.4612	0.4489	0.5472	0.5910	0.6000	0.6200
0.63	**0.37**	0.3700	0.4081	0.4671	0.5138	0.6300	0.3700	0.4136	0.4153	0.4255	0.4288	0.4641	0.4546	0.5541	0.5993	0.6093	0.6300
0.64	**0.36**	0.3600	0.4038	0.4680	0.5202	0.6400	0.3600	0.4089	0.4118	0.4230	0.4275	0.4670	0.4604	0.5610	0.6075	0.6185	0.6400
0.65	**0.35**	0.3500	0.3996	0.4688	0.5266	0.6500	0.3500	0.4041	0.4083	0.4205	0.4262	0.4699	0.4661	0.5679	0.6158	0.6278	0.6500
0.66	**0.34**	0.3400	0.3953	0.4697	0.5330	0.6600	0.3400	0.3994	0.4048	0.4180	0.4249	0.4728	0.4718	0.5748	0.6240	0.6371	0.6600
0.67	**0.33**	0.3300	0.3910	0.4706	0.5393	0.6700	0.3300	0.3947	0.4013	0.4155	0.4237	0.4757	0.4776	0.5817	0.6323	0.6464	0.6700
0.68	**0.32**	0.3200	0.3868	0.4715	0.5457	0.6800	0.3200	0.3899	0.3978	0.4130	0.4224	0.4786	0.4833	0.5886	0.6406	0.6556	0.6800
0.69	**0.31**	0.3100	0.3825	0.4724	0.5521	0.6900	0.3100	0.3852	0.3943	0.4106	0.4211	0.4815	0.4890	0.5955	0.6488	0.6649	0.6900
0.70	**0.30**	0.3000	0.3782	0.4733	0.5585	0.7000	0.3000	0.3804	0.3908	0.4081	0.4198	0.4844	0.4948	0.6024	0.6571	0.6742	0.7000
0.71	**0.29**	0.2900	0.3739	0.4742	0.5649	0.7100	0.2900	0.3757	0.3873	0.4056	0.4185	0.4873	0.5005	0.6093	0.6653	0.6835	0.7100
0.72	**0.28**	0.2800	0.3697	0.4751	0.5713	0.7200	0.2800	0.3709	0.3838	0.4031	0.4172	0.4902	0.5062	0.6162	0.6736	0.6927	0.7200
0.73	**0.27**	0.2700	0.3654	0.4760	0.5776	0.7300	0.2700	0.3662	0.3803	0.4006	0.4159	0.4931	0.5119	0.6231	0.6818	0.7020	0.7300
0.74	**0.26**	0.2600	0.3611	0.4769	0.5840	0.7400	0.2600	0.3614	0.3768	0.3981	0.4146	0.4960	0.5177	0.6300	0.6901	0.7113	0.7400
0.75	**0.25**	0.2500	0.3568	0.4777	0.5904	0.7500	0.2500	0.3567	0.3733	0.3956	0.4133	0.4989	0.5234	0.6369	0.6984	0.7206	0.7500
0.76	**0.24**	0.2400	0.3526	0.4786	0.5968	0.7600	0.2400	0.3520	0.3698	0.3931	0.4120	0.5018	0.5291	0.6438	0.7066	0.7298	0.7600
0.77	**0.23**	0.2300	0.3483	0.4795	0.6032	0.7700	0.2300	0.3472	0.3663	0.3906	0.4107	0.5047	0.5349	0.6507	0.7149	0.7391	0.7700
0.78	**0.22**	0.2200	0.3440	0.4804	0.6096	0.7800	0.2200	0.3425	0.3628	0.3881	0.4094	0.5076	0.5406	0.6576	0.7231	0.7484	0.7800
0.79	**0.21**	0.2100	0.3397	0.4813	0.6159	0.7900	0.2100	0.3377	0.3593	0.3856	0.4081	0.5105	0.5463	0.6645	0.7314	0.7576	0.7900
0.80	**0.20**	0.2000	0.3355	0.4822	0.6223	0.8000	0.2000	0.3330	0.3558	0.3832	0.4068	0.5134	0.5521	0.6714	0.7396	0.7669	0.8000
0.81	**0.19**	0.1900	0.3312	0.4831	0.6287	0.8100	0.1900	0.3282	0.3523	0.3807	0.4055	0.5163	0.5578	0.6783	0.7479	0.7762	0.8100
0.82	**0.18**	0.1800	0.3269	0.4840	0.6351	0.8200	0.1800	0.3235	0.3488	0.3782	0.4042	0.5192	0.5635	0.6853	0.7562	0.7855	0.8200
0.83	**0.17**	0.1700	0.3227	0.4849	0.6415	0.8300	0.1700	0.3187	0.3453	0.3757	0.4030	0.5221	0.5693	0.6922	0.7644	0.7947	0.8300
0.84	**0.16**	0.1600	0.3184	0.4858	0.6479	0.8400	0.1600	0.3140	0.3418	0.3732	0.4017	0.5250	0.5750	0.6991	0.7727	0.8040	0.8400
0.85	**0.15**	0.1500	0.3141	0.4867	0.6542	0.8500	0.1500	0.3093	0.3383	0.3707	0.4004	0.5279	0.5807	0.7060	0.7809	0.8133	0.8500
0.86	**0.14**	0.1400	0.3098	0.4875	0.6606	0.8600	0.1400	0.3045	0.3348	0.3682	0.3991	0.5308	0.5864	0.7129	0.7892	0.8226	0.8600
0.87	**0.13**	0.1300	0.3056	0.4884	0.6670	0.8700	0.1300	0.2998	0.3313	0.3657	0.3978	0.5337	0.5922	0.7198	0.7974	0.8318	0.8700
0.88	**0.12**	0.1200	0.3013	0.4893	0.6734	0.8800	0.1200	0.2950	0.3278	0.3632	0.3965	0.5366	0.5979	0.7267	0.8057	0.8411	0.8800
0.89	**0.11**	0.1100	0.2970	0.4902	0.6798	0.8900	0.1100	0.2903	0.3243	0.3607	0.3952	0.5395	0.6036	0.7336	0.8139	0.8504	0.8900
0.90	**0.10**	0.1000	0.2927	0.4911	0.6862	0.9000	0.1000	0.2855	0.3207	0.3582	0.3939	0.5424	0.6094	0.7405	0.8222	0.8597	0.9000
0.91	**0.09**	0.0900	0.2885	0.4920	0.6925	0.9100	0.0900	0.2808	0.3172	0.3558	0.3926	0.5453	0.6151	0.7474	0.8305	0.8689	0.9100
0.92	**0.08**	0.0800	0.2842	0.4929	0.6989	0.9200	0.0800	0.2761	0.3137	0.3533	0.3913	0.5482	0.6208	0.7543	0.8387	0.8782	0.9200
0.93	**0.07**	0.0700	0.2799	0.4938	0.7053	0.9300	0.0700	0.2713	0.3102	0.3508	0.3900	0.5511	0.6266	0.7612	0.8470	0.8875	0.9300
0.94	**0.06**	0.0600	0.2756	0.4947	0.7117	0.9400	0.0600	0.2666	0.3067	0.3483	0.3887	0.5540	0.6323	0.7681	0.8552	0.8967	0.9400
0.95	**0.05**	0.0500	0.2714	0.4956	0.7181	0.9500	0.0500	0.2618	0.3032	0.3458	0.3874	0.5569	0.6380	0.7750	0.8635	0.9060	0.9500
0.96	**0.04**	0.0400	0.2671	0.4964	0.7245	0.9600	0.0400	0.2571	0.2997	0.3433	0.3861	0.5598	0.6437	0.7819	0.8717	0.9153	0.9600
0.97	**0.03**	0.0300	0.2628	0.4973	0.7308	0.9700	0.0300	0.2523	0.2962	0.3408	0.3848	0.5627	0.6495	0.7888	0.8800	0.9246	0.9700
0.98	**0.02**	0.0200	0.2585	0.4982	0.7372	0.9800	0.0200	0.2476	0.2927	0.3383	0.3835	0.5656	0.6552	0.7957	0.8883	0.9338	0.9800
0.99	**0.01**	0.0100	0.2543	0.4991	0.7436	0.9900	0.0100	0.2428	0.2892	0.3358	0.3823	0.5685	0.6609	0.8026	0.8965	0.9431	0.9900
1.00	**0.00**	0.0000	0.2500	0.5000	0.7500	1.0000	0.0000	0.2381	0.2857	0.3333	0.3810	0.5714	0.6667	0.8095	0.9048	0.9524	1.0000