

AN INTEGRATED POST-DISASTER ASSESSMENT ROUTING PROBLEM FOR COLLECTING DAMAGE INFORMATION WITH DRONES

A Thesis

by

Birce Adsanver

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
August 2020

Copyright © 2020 by Birce Adsanver

AN INTEGRATED POST-DISASTER ASSESSMENT ROUTING PROBLEM FOR COLLECTING DAMAGE INFORMATION WITH DRONES

Approved by:

Assistant Professor Elvin Çoban, Advisor
Department of Industrial Engineering
Özyeğin University

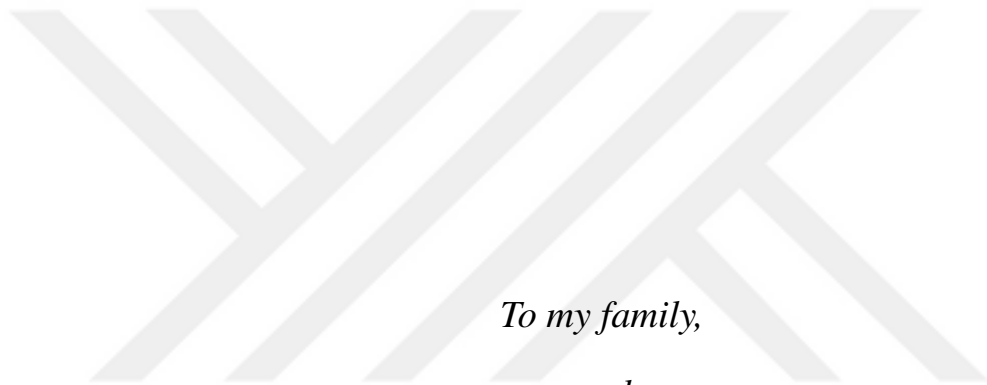
Assistant Professor İhsan Yanıkoğlu
Department of Industrial Engineering
Özyeğin University

Associate Professor Burcu Balçık,
Co-Advisor
Department of Industrial Engineering
Özyeğin University

Assistant Professor Umman Mahir
Yıldırım
Department of Industrial Engineering
İstanbul Bilgi University

Date Approved: 24 August 2020

Assistant Professor Erinc Albey
Department of Industrial Engineering
Özyeğin University



To my family,

and

to heroes who sacrificed their lives to save disaster victims...

ABSTRACT

In this study, we focus on post-disaster damage assessment operations supported by a set of drones when the disaster-affected area is divided into grids, and grids are clustered based on their attributes. We propose a two-phase methodology to assess the damage status of the built environment in grids. Specifically, given a set of drones and a limited time for an assessment interval, the first phase addresses the problem of determining the grids to scan by each drone and the sequence of visits to the selected grids. We aim to maximize the total priority score collected from the scanned grids while satisfying the predefined targeted coverage ratio. In the second phase, we aim to predict the damage status of unscanned grids by using the cluster-based information obtained from the scanned grids at the end of the assessment interval. Nevertheless, the damage status of all grids may not be assessed (by scanning or prediction) after one interval; therefore, these two phases iterate until all grids are evaluated. For the problem solved in the first phase, we adapt two formulations from the literature developed for electric vehicle routing problems. We also develop a Variable Neighborhood Descent based heuristic which can find high-quality solutions rapidly. We evaluate the performance of the alternative formulations and the heuristic in a variety of instances. For the second phase, we devise a novel imputation method and different imputation policies to predict the damage status of the unscanned grids. We also define several performance metrics to measure the efficiency and effectiveness of the proposed imputation policies. Our analyses demonstrate that using the proposed imputation policies improve the system performance as they induce a rapid detection of the damaged areas.

ÖZETÇE

Bu çalışma, bir dizi insansız hava aracı (dron) tarafından desteklenen afet sonrası hasar değerlendirme operasyonlarına odaklanmaktadır. Afetten etkilenen bölge gridlere ayrılarak ele alınmış ve bu gridler özelliklerine göre kümelenmiştir. Gridlerde bulunan yapıların hasar durumlarını belirlemek için iki aşamalı bir metodoloji önerilmektedir. İlk aşama temel olarak bir dizi dron ile sınırlı bir değerlendirme süresi içinde taranacak gridleri seçme ve seçilen gridlerin taranma sıralarını belirleme sorununu ele alır. Bu aşamadaki amaç, her bir küme için önceden belirlenmiş olan kapsama oranını karşılarken, taranan gridlerden elde edilen toplam öncelik puanlarını maksimize etmektir. İkinci aşamada ise, bir değerlendirme aralığından sonra taranan gridlerden elde edilen küme bazlı hasar bilgileri kullanılarak, taranmamış gridlerdeki hasar durumlarının tahmin edilmesi hedeflenmiştir. Ancak bütün gridlerdeki hasar durumlarının belirlenmesi (taranarak veya tahmin edilerek) tek bir değerlendirme aralığı sonunda mümkün olmayabilir. Bu nedenle önerilen iki aşama tüm gridlerdeki hasar seviyeleri belirlenene kadar yinelenir. İlk aşamada çözülen sorun için, literatürdeki elektrikli araç yönlendirme problemleri için geliştirilen iki matematiksel model uyarlanmıştır. Ayrıca, daha hızlı bir şekilde yüksek kaliteli çözümler elde edebilen bir sezgisel algoritma önerilmiştir. Matematiksel modellerin ve sezgisel algoritmanın performansları çeşitli örnekler üzerinde test edilmiştir. İkinci aşama için, taranmamış gridlerdeki hasar durumlarını tahmin etmekte kullanılmak üzere çeşitli politikalardan oluşan özgün bir veri atama yöntemi önerilmiştir. Ayrıca, bu politikaları verimlilik ve etkinlik açısından değerlendirmek için çeşitli performans ölçütleri tanımlanmıştır. Yapılan analizler önerilen politikaların kullanılmasının hasarlı alanların daha hızlı tespit edilmesini sağlayarak sistem performansını iyileştirdiğini göstermektedir.

ACKNOWLEDGEMENTS

It has been a very unusual and challenging year for me, and for the whole world. I want to take this opportunity to express my gratefulness to many people who make this time easier with their continuous support.

Firstly, I am deeply thankful to my advisors, Dr. Elvin Çoban and Dr. Burcu Balçık, for their tremendous support, guidance, inspiration, and patience. Their invaluable advice always motivates me and contributes to my academic and personal development. It is a great honor and privilege to study under their guidance. I would like to express my sincere gratitude to all professors in the Department of Industrial Engineering for their advice, feedback, passion for teaching, and willingness to help during my B.Sc. and M.Sc. study.

I would like to thank my friends and colleagues, Ahmet, Berk, Ceren, and Eren for their help, encouragement, sense of humor, and warm friendship. I owe thanks to Erdem and Gökhan for their support, valuable advice, and for reminding me to take some break from working and enjoy life. I am so grateful to my dear friend, İnci, for always listening to me, motivating me, and cheering me up. Life would be joyless without them. Thanks a bunch, to all my friends in the department, they have always been helpful, friendly, and kind to me.

I am eternally grateful to my mother and my father for their unconditional love, trust, and understandings. They taught me to respect all living beings and how to be a "good" human. They always enlighten my path and encourage me to never give up. I could never complete this study without their empathy and support in every manner. I am thankful to my brother for always making me feel protected and safe. I am truly grateful to my cousins, Pınar and Çağıl, for their continuous help and ever-smiling faces. Finally, I would like to thank all my family for always being there for me. They are my greatest treasure.

TABLE OF CONTENTS

DEDICATION	iii
ABSTRACT	iv
ÖZETÇE	v
ACKNOWLEDGEMENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	xi
I INTRODUCTION	1
II LITERATURE REVIEW	5
2.1 Humanitarian Operations with Drones	5
2.2 Humanitarian Operations Planning with Information Updates	7
2.3 Electric Vehicle Routing Problems	11
III PROBLEM AND APPROACH	13
3.1 System and Problem Description	13
3.2 Methods	18
3.2.1 Phase I. Drone routing problem and formulation	19
3.2.2 Phase II. Imputation policies (IPs) and re-clustering	29
IV A VARIABLE NEIGHBORHOOD DESCENT (VND) BASED HEURISTIC FOR THE DRONE ROUTING PROBLEM	32
4.1 Constructive Heuristic (CH)	32
4.2 Local Search	34
4.2.1 Route-improving moves.	35
4.2.2 Priority-improving moves.	36
4.2.3 Diversification.	38
V NUMERICAL ANALYSIS	39
5.1 Preliminary Analysis of the MILP formulations	39

5.1.1	Instances	40
5.1.2	Computational Results of the Preliminary Analysis	42
5.2	Numerical Analysis	46
5.2.1	Instances	46
5.2.2	Computational Results of Phase I	49
5.2.3	Computational Results of Phase II: IDARP	54
VI	CONCLUSION	63
	REFERENCES	74
	VITA	81

LIST OF TABLES

1	Computational results of the MILP formulations for 6 networks (Net.) having 16 grids varying the number of drones (D).	43
2	Computational results of the MILP formulations for 6 networks (Net.) having 25 grids varying the number of drones (D).	44
3	Computational results of the MILP formulations for 6 networks (Net.) having 36 grids varying the number of drones (D). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.	45
4	Computational results of the MILP formulation and the VND-based meta-heuristic for 2 networks (Net.) having 15 grids varying the number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method. . .	52
5	Computational results of the MILP formulation and the VND-based meta-heuristic for 2 networks (Net.) having 25 grids varying the number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method. . .	53
6	Computational results of the MILP formulation and the VND-based meta-heuristic for 4 networks (Net.) having 50 grids varying the number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method. . .	54
7	Performance metrics e_1-e_4 for P0 and all imputation policies over 20 instances for varying the targeted coverage ratio μ and the damage ratio threshold (th.)	57
8	The number of grids that are estimated likely damaged by the imputation policies over 20 instances for varying the targeted coverage ratio μ and the damage ratio threshold (th.).	61
9	Sets, parameters and decisions variables for the arc-based formulation . . .	66
10	Sets, parameters and decisions variables for the path-based formulation . .	67

- 11 Computational results of the path-based formulations with/without dominance rule for 2 networks (Net.) having 15 grids varying number of drones (D), T_{max} , and the coverage target (μ), where $\bar{\gamma}$ denotes the minimum coverage ratio across all clusters, $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method. 68
- 12 Computational results of the path-based formulations with/without dominance rule for 2 networks (Net.) having 25 grids varying number of drones (D), T_{max} , and the coverage target (μ), where $\bar{\gamma}$ denotes the minimum coverage ratio across all clusters, $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method. 69



LIST OF FIGURES

1	Overview of the two-phase approach for IDARP.	16
2	Illustration of the proposed approach and IDARP solution on an example .	18
3	Illustrative example of two types of instances for a 25 km^2 area	41



CHAPTER I

INTRODUCTION

Sudden onset natural disasters (such as earthquakes, floods, hurricanes) may cause loss of life, injuries, and infrastructural damages. Since the first hours and days after a disaster are crucial to save lives, emergency response activities must immediately begin. For instance, structural collapse is one of the main reasons of death tolls after in earthquakes causing 75% of these deaths (1), and the survival rate for the people who are rescued from the collapsed buildings after an earthquake is 91% during the first 30 minutes, while it decreases to 36% by the second day, and 19% by the fourth day (2). Hurricanes and floods can also cause significant massive structural damage in buildings (3; 4). When a disaster occurs, search and rescue (SAR) teams, which are composed of doctors, ambulance operators, engineers, firefighters, and highly trained volunteers (5), must be immediately deployed to the affected areas. SAR operations for saving trapped people from the damaged buildings usually take five to seven days (6). During this time period, it is critical to identify the most urgent areas with severe damages and direct the SAR resources accordingly (7). However since disasters may affect a large region, it may be difficult to assess the extent of structural damages in the buildings quickly.

Unmanned aerial vehicles (UAVs), also known as drones, are increasingly used to support SAR operations in disaster management (8; 9; 10). In particular, drones are used for obtaining information on the affected areas, structural damage mapping, and monitoring the post-disaster situation (11; 12). Compared to other alternatives that can provide aerial information such as airplanes and helicopters, drones take much less time to deploy, do not need runways or helipads to take off, are much cheaper to operate, and do not require a person on board (8). Moreover, compared to satellite imagery, drones can provide better

resolution independent of weather conditions and they are more flexible to use (13; 14; 15). However, one disadvantage of using drones would be related to their limited battery capacity and flight range. That is, drones need to be recharged regularly while scanning a large region, and hence given a limited number of drones, careful planning of their routes would be required to assess the entire affected region within a reasonable time period after a disaster.

In this study, we focus on rapid damage assessment operations conducted by using a set of drones in order to scan a disaster-affected area and identify the areas that include damaged buildings as quickly as possible. In the proposed system, drones, which are initially positioned at an assessment operations coordination center (AOCC), make multiple trips throughout the assessment horizon, and collect damage information by scanning selected areas. To gather and process the images collected by drones regularly (i.e., without waiting for all areas to be scanned), we require that each trip of a drone is completed within an assessment interval (such as three hours). That is, throughout the assessment horizon, drones leave the AOCC at the beginning of an assessment interval, scan their assigned area by following a pre-determined route to collect damage information and return to the AOCC after they complete their trip. Once drones return to the AOCC, collected aerial images are processed to identify the areas with structural damages, and also to make predictions about the damage status of the unscanned areas based on the similarities of the scanned and unscanned areas in terms of a set of attributes, which may accept categorical or numerical values. We assume that the damage level of the built environment is highly correlated with these attributes, which can be obtained from secondary sources such as governmental databases. For instance, in the case of earthquakes, different areas with the same structural (e.g., construction types of the buildings), geographical (e.g., elevation), geological (e.g., soil type) properties, and a similar distance to the epicenter may exhibit similar damage status (16). To predict damages, we propose using imputation methods that consider the similarities of different affected areas based on their attributes, where “imputation” refers

to filling in missing data based on available data (17).

Developing efficient plans to use the available drone fleet to scan the chosen areas and making accurate predictions about the unscanned areas are both important to complete damage assessment operations quickly. In this study, we introduce an integrated damage assessment routing problem (IDARP), which involves decisions related to selecting the areas to scan, routing the drones, and making predictions about the unscanned areas throughout the assessment horizon so that the assessments can be completed quickly and effectively. We assume that the findings of the assessment process about the damaged areas will be informed to the SAR teams over the assessment horizon to facilitate a quick and effective response; however, planning the operations of SAR teams or other emergency responders is not within the scope of this study.

To formulate IDARP, we divide the disaster-affected region into geographical grids. We assume that the attribute values of each grid are known. Moreover, each grid is assigned a priority score, which reflects the urgency of emergency response for that grid; for instance, it may be desirable to prioritize grids that have high population density at the time of the disaster and important facilities such as hospitals and schools. We consider a single AOCC and a set of recharge stations (RSs) positioned in the affected region beforehand. To assess the damages in such a post-disaster network by using a set of drones, we propose a two-phase methodology that will be applied in each assessment interval. Specifically, Phase I addresses grid selection and drone routing decisions by considering battery limitations of drones, coverage requirements, and grid priorities, and Phase II focuses on making damage predictions for the unscanned grids based on the information collected in Phase I from the scanned grids. We formulate the drone routing problem solved in Phase I as a variant of the electric vehicle routing problem (E-VRP) (see e.g., (18)). The proposed drone routing problem is different than the existing E-VRPs due to additional grid selection decisions and also its priority- and coverage-based objective function. We develop two mixed-integer linear programming (MILP) models, which are adapted from (18), for the proposed drone

routing problem, and a novel Variable Neighborhood Descent-based metaheuristic to find high-quality solutions rapidly. Additionally, we devise and implement alternative imputation policies to estimate the damage status of unscanned grids in Phase II. The two phases iterate over multiple intervals until all grids are assessed. We evaluate the performance of our approach on a set of performance measures that focus on the efficiency and the effectiveness of assessment operations.

The rest of the study is organized as follows. In Chapter 2, we review the relevant literature. In Chapter 3, we describe the system and problem in detail and present the modeling approach. In Chapter 4, we describe the proposed metaheuristic, and, in Chapter 5 we present our computational results. Finally, we conclude in Chapter 6 and discuss future work.

CHAPTER II

LITERATURE REVIEW

We discuss the literature related to the proposed assessment planning problem under three sections. We first review humanitarian logistics studies that consider using drones in Section 2.1 and planning under incomplete information in Section 2.2. We then briefly review the relevant literature on E-VRPs due to their similarities with our drone routing problem in Section 2.3.

2.1 Humanitarian Operations with Drones

Drones are relatively inexpensive and easy to deploy and are getting more popular in humanitarian relief operations (7; 10; 19; 20). They were utilized after several recent disasters for damage assessment, monitoring and situational awareness purposes; for example, to measure radiation contamination in the Daiichi nuclear power plant (21), capture the images of damaged reactors after the great Japan earthquake in 2011 (20), monitor the affected area after the 2011 Thailand floods (22), and assess the damages after the 2016 Kumamoto earthquake in Japan (23), the 2009 L'Aquila earthquake in Italy (24), and typhoon Haiyan in Philippines (25). Moreover, there is an increasing number of studies that present methods for using drones to support humanitarian operations by focusing on different problems such as locating relief facilities (26; 27; 28), distributing relief or medical supplies (29; 30), and planning post-disaster assessment operations (31).

An edge-based facility location problem where drones deliver relief items to aid recipients is studied in (28), in which aid recipients are distributed along the network edges and they can only travel to the distribution centers by using the accessible edges. The authors propose a mixed-integer non-linear programming model and metaheuristic algorithms to

find the locations of facilities minimizing the aggregate traveling time for both the aid recipients and drones. Similarly, (27) determines the locations of the distribution centers, their corresponding service regions, and the number of emergency supplies to be kept in these centers by proposing a continuous approximation model to minimize the overall distribution cost of trucks and drones. The authors consider various drone-specific aspects such as accelerating, decelerating, and velocity changes while calculating the transportation cost. Additionally, a facility coverage problem is studied in (26) considering drone energy consumption and drone flight range limitations. A location-allocation problem for positioning recharging stations is addressed by (32). In (30), the authors focus on drone routing decisions for last-mile distribution, where energy consumption is dependent on the weight of the payload. The authors develop a MILP model to minimize the total traveling distance while considering several priority policies on the locations. An MILP and metaheuristics are proposed to scan the post-disaster affected areas to minimize the total assessment costs in (33). (34) addresses a post-earthquake relief distribution system, where drones are used to serve areas that are inaccessible by ground transportation. (31) studies a post-disaster damage assessment strategy that focuses on population points (nodes) and road segments (arcs), where the damage level of nodes and arcs is assessed by drones and motorcycles. The authors consider the drones' battery time as a time limit for damage assessment operations, such that all vehicles must return to the depot before the time limit. Drones can assess the roads that are covered by debris; however, the motorcycles cannot move along the roads blocked by debris. Additionally, the authors prioritize the critical nodes and arcs in the affected network and present a multi-objective MILP model to maximize the total profit collected by visiting nodes and traversing arcs.

While drones are increasingly used to support damage assessment operations, analytical methods that would support logistical planning of assessment operations by using drones

are limited. We contribute to the humanitarian logistics literature by proposing a new problem and a novel approach for planning damage assessment operations by drones. Specifically, we propose a new drone routing problem and an efficient metaheuristic heuristic method to solve the proposed drone routing problem, as well as alternative imputation techniques to predict the damages based on partial information collected by drones over multiple time intervals. To the best of our knowledge, there is no study that addresses a post-disaster setting in which operational assessment decisions by drones are made by making predictions based on collected partial information over multiple periods.

2.2 Humanitarian Operations Planning with Information Updates

An increasing number of studies in the humanitarian logistics literature focuses on developing methods to support decision making in post-disaster environment (see (35), (36), and (37) for recent surveys on disaster response studies). Most of the existing studies assume that the required information for making logistical decisions in the post-disaster environment (e.g., such as infrastructure, damages, needs, capacity, resources, supply) are readily available. However, in most settings, data is unavailable, inaccurate, or scarce (16; 38; 39). Moreover, collecting data is challenging due to the chaotic nature of the post-disaster environments. Since the effectiveness of disaster response may highly depend on the amount and accuracy of the available information, it is important to develop methods that would support collecting information in the post-disaster environment quickly and also utilizing the available but incomplete information in the best way in decision making.

There exist a variety of studies that propose many imputation techniques to estimate unknown information in very wide-range applications. The simple methods can be efficiently applied for the large data sets (40). For example, the mean-and-mode imputation method fills the missing values by the mean (for the numerical data) or the mode (for the categorical data) of the data. Nevertheless, this method may perform poorly since it does not utilize the underlying relationship among data (41). To overcome this, supervised (e.g., classification

(42)) and unsupervised (e.g., clustering (43)) machine learning techniques are also widely used based on the availability of the information. To use a supervised learning method, the output variables (e.g., labels, such as damage status of the areas) should be available for a part of the data to predict the unknown labels (44). The unsupervised learning methods can be utilized to group similar objects even only the input data (e.g., characteristics of the post-disaster environment) is available. However, the output variables must be available to make an imputation. For instance, a clustering algorithm (e.g., k -means clustering (45)) combined with an imputation rule to estimate the missing data is commonly used (46; 47; 48). Specifically, these techniques do not fill the missing values in the whole data set at once, but they apply the imputation methods for the subsets obtaining by clustering (40).

In a post-disaster setting, simpler methods can be preferred for a rapid response since the complicated methods demand high computational requirements (16). For example, in (16), the authors propose several techniques to estimate the unknown road damage status after a disaster. In their setting, the damage status of some roads and the characteristics (attributes) of all road segments are known in advance. Based on the available information, the authors introduce two clustering-based algorithms combining k -modes clustering with different imputation rules (modified mean-and-mode, and adjacent arc method). Specifically, all road segments (including the ones that their damage status is not known) are assigned to k clusters by using their attributes. Then, one of the proposed imputation rules is implemented to predict the unknown road damage status. As the classical mean-and-mode method assigns the same status (e.g., non-damaged) to all road segments within a cluster based on the dominant value, the authors modify this approach by utilizing the road status value that appears with the lowest percentage (called as critical value) in the data. Particularly, they compare the percentage of the critical value both for each cluster and the entire data. If the within-cluster percentage of the critical value is larger than the percentage for the entire data, then the unknown status is imputed by the critical value rather than the

mean or mode value. In the adjacent arc method, the authors use the information obtained from the adjacent arcs while estimating the unknown information. Additionally, the authors use classification trees as a supervised learning algorithm. To provide a benchmark for the proposed algorithms, they use a simple approach based on a global constant method and mean-and-mode method, such that the same constant value is given to all unknown values. In contrast with (16), we assume that partial structural damage information is not readily available, but the information is gathered by drones over the damage assessment period. Consequently, we develop a new iterative approach, such as the missing data is predicted as the information is collected over multiple periods. We also propose new and practical imputation rules.

Another relevant stream of literature to our work focuses on developing dynamic decision making approaches that incorporate information updates in planning humanitarian operations. In (49), the authors present a framework focusing on the real-time data collection process in a post-disaster environment, and they indicate the importance of integrating dynamically updated data in humanitarian operations. Nevertheless, the number of studies integrates available information in decision making dynamically are few (see reviews by (49; 50; 51)). Some studies propose using a rolling horizon framework to update decisions as new information becomes available. For instance, in (52), the authors suggest using a rolling horizon framework to capture the supply and demand uncertainties during the last mile distribution of relief items. (53) also focuses on relief distribution, and proposes a rolling horizon approach consisting of two modules. Specifically, the authors predict the relief demands and delivery times in the first module and determine the optimal distribution flows in the second module. (54) studies an integrated multi-objective optimization model combining resource allocation and emergency distribution. To cope with the frequent information updates, they present a rolling horizon approach, such that the planning horizon is divided into equal-length discrete time units where the decisions are updated based on the available information.

Some studies present dynamic (multi-period) models to incorporate information updates in planning response operations. (55) addresses an integrated emergency logistics planning problem, in which vehicles perform both pick-up and delivery of relief supplies. The authors present a MILP formulation and a solution approach consisting of decomposition and Lagrangean relaxation to support decisions on the supply and delivery types and quantities and vehicle routes. Decisions are updated at regular time intervals based on the new information on supply, demand, and availability of the vehicles. (56) also addresses a dynamic coordination problem to transport commodities from suppliers to the distribution centers and evacuate injured people from affected areas. The authors consider a short planning horizon (e.g., days or hours) to capture the frequent information changes on the number of people requiring evacuation, demand, supply and the vehicle availability. Additionally, (57) studies a dynamic relief-demand management problem where the relief demand is estimated dynamically in each time interval. First, the authors use data fusion techniques to deal with multiple sources of information since the related information may come randomly from a variety of sources in large-scale disasters. Then, they forecast the relief demands of different areas by available information and group the affected areas via a fuzzy clustering-based approach to identify their group-based relief-demand urgency. The groups having higher urgency are served earlier than others in each time interval.

The number of studies that address making operational planning in a post-disaster setting based on incomplete data is scarce. We contribute to this stream of literature by proposing a new practical approach that repeatedly plans the routes of the drones over the assessment horizon to identify the damaged areas and utilize the gathered information to predict other potentially damaged areas. We show that using the available partial information obtained by drones from the scanned parts of the network to predict the damages in the unscanned parts by implementing simple imputation techniques can reduce the overall length of the assessment operations, and hence facilitate effective emergency response.

2.3 *Electric Vehicle Routing Problems*

Planning the routes of electrical vehicles and drones are similar in the sense that both need to be recharged at battery recharging stations. Hence, the drone routing problem that must be solved in each assessment interval in our proposed approach is similar to the electric vehicle routing problems (E-VRP), or green vehicle routing problems (G-VRP), which are widely studied in the literature. Here we review the studies that are most relevant to our work.

In (58), the authors formulate an E-VRP with time windows and cargo capacity constraints to minimize traveled distance. The vehicles are fully charged at an RS, and the recharging time is linearly dependent on the battery level at the RS. The authors provide a MILP formulation which requires dummy RSs (i.e., recharging station copies) to allow multiple visits at an RS. Besides, they develop a hybrid heuristic combining Tabu Search (TS) and Variable Neighborhood Search (VNS) heuristics. (59) proposes an exact solution approach for the same problem by combining a branch-and-cut algorithm and a Simulating Annealing (SA) algorithm to minimize total distance traveled. (60) develops several constructive and local search heuristics by using a non-deterministic SA framework. The authors demonstrate that the proposed heuristics are faster than the VNS-TS heuristic proposed by (58).

In (61), the authors propose a different (path-based) formulation to avoid generating dummy RSs. The authors present an exact algorithm based on a set partitioning formulation strengthened by using valid inequalities and k-path cuts. (18) allows partial charging and non-linear charging functions, and they aim to minimize total time including the traveling time and recharging time. The authors present three formulations by utilizing different decision variables: node-based, arc-based, and path-based. The arc-based formulation is shown to perform better than the node-based formulation due to the tighter linear relaxation (18). However, both the node-based and the arc-based formulations require dummy RSs to allow multiple visits; and deciding the number of dummy RSs has a critical impact on

the solution (18; 62). While a large number of dummy RSs increases the computational time, few dummy RSs may cut off the optimal solutions. Some instances may require setting the number of dummy nodes for each RS equal to the number of nodes to be visited, which causes a significant increase in computation time, especially for large-size instances (59). On the other hand, the path-based formulation is not limited by the number of RS replications (18). An arc-based formulation and a path-based formulation addressing the post disaster damage assessment operations are compared by (63). The authors show that both formulations perform similarly, however, since the size of the arc-based formulation may significantly increase with an increasing number of nodes, the path-based formulation is concluded to perform better than the arc-based formulation. In addition to the different formulations, various heuristics and exact algorithms are proposed to solve large-size instances within a reasonable time (see (64)).

In this study, we adapt arc-based and path-based formulations from (18) for our drone routing problem that addresses post-disaster damage assessment operations. Nevertheless, there exist several differences between our problem and the existing E-VRPs primarily due to the unique characteristics of the post-disaster assessment environment. Specifically, all nodes are usually visited in the E-VRPs, while the grids to be visited in a limited assessment interval must be selected by considering various factors including priority scores and the attained coverage. In general, E-VRPs focus on minimizing total traveling time, cost, or distance. Due to these differences, new solution approaches must be developed for the proposed drone routing problem. Our study contributes to the literature by presenting a new routing problem as well as formulations and a practical heuristic to support decision making in managing post-disaster damage assessment operations via drones.

CHAPTER III

PROBLEM AND APPROACH

In this chapter, we first describe the proposed damage assessment system and define the problem (Section 3.1), and present methods to address the problem (Section 3.2).

3.1 System and Problem Description

We consider a post-disaster setting and focus on damage assessment operations, which start immediately after the occurrence of a disaster. More specifically, we focus on the assessment of the damages caused by the disaster on the built environment (e.g., residential buildings, hospitals, shopping malls, schools, etc.) with the help of drones and quickly identify the areas that have serious structural damage and need immediate attention by emergency responders. We focus on the initial critical hours after the disaster, where a rapid evaluation of the affected area is necessary to provide information to the emergency planners, who must quickly allocate critical and scarce resources (e.g., search and rescue (SAR) teams) among the affected locations.

We focus on a large geographical region affected by a disaster (such as an earthquake, flood, or hurricane). We assume that the region is divided into grids, which will be visited and scanned by the drones to collect aerial images. We only focus on the grids that include built infrastructure (i.e., residential, commercial, public buildings), which may be damaged due to the effects of the disaster and people may be trapped inside. That is, it is not necessary to visit grids that do not include built infrastructure (e.g., lakes, forests). Moreover, each grid has a priority score, which can be assigned based on the population density of the grid or the critical facilities (such as schools, hospitals) located in the grid. It is important to assess the grids with higher priority as early as possible so that critical damage information can be provided with emergency relief planners.

The assessment operations continue until all areas (grids) have been assessed. However, scanning all of the grids in the affected region a short period of time may not be possible. Therefore, in our proposed system, available drones scan some selected grids over a planning horizon, and the information from the scanned grids is used to predict the damages of the unscanned grids by considering the similarity of the grids in terms of a set of attributes (such as soil type, building characteristics, closeness to the epicenter, etc.), which are expected to affect a building to get damaged after the specific disaster at hand. Furthermore, these attributes can be associated with pre-disaster and post-disaster aspects. For instance, for an earthquake, the construction material of the majority of the buildings in a certain area can be specified before a disaster, while the closeness to the fractured fault line can be evaluated after a disaster (65; 66). If efficient drone routes can be constructed to scan the selected grids and accurate damage predictions can be made for the unscanned areas based on the available attribute values, the entire affected region can be assessed in a shorter period of time.

We focus on the problem of planning the routes of the drones over the assessment horizon to identify and predict the damaged grids. We assume a network with a single assessment operations coordination center (AOCC), where the assessment operations are planned, coordinated and the collected aerial information by the drones are processed. All drones are assumed to be initially positioned at the AOCC. Moreover, since there may be communication failures after disasters due to infrastructural damages, we assume that the assessment information collected by drones may not be shared and evaluated remotely (14; 67). Hence, in the proposed damage assessment system, the drones regularly return to the AOCC throughout the planning horizon so that the gathered information can be processed at the AOCC (68). Specifically, each drone will make multiple trips to scan the affected region, and each trip must be completed within T_{max} time units, which is called an *assessment interval*. Moreover, since the battery of a drone is limited, a drone dispatched from the AOCC may need to visit recharge stations (RSs) while scanning a set of grids in

each assessment interval. We assume that a number of RSs are positioned in the network beforehand. Hence, each route of a drone originates from the AOCC, visits a set of selected grids and RSs, and ends at the AOCC within T_{max} .

When a drone completes its trip within the assessment interval and returns to the AOCC, its battery is recharged at the AOCC and the collected aerial images are processed, which are used to make predictions for the unscanned areas. That is, the damage status of unscanned areas are predicted based on the information gathered from similar areas, where similarity is assessed based on the grid attributes. The routes of the drone fleet in an assessment interval are constructed by considering the findings in the previous assessment intervals. Specifically, grids that are scanned and predicted to be damaged (without being scanned), which are called *likely damaged*, are removed from the list of grids that need to be assessed for the next assessment interval. By processing the information gathered by drones and rerouting the drones in each assessment interval, the proposed damage assessment system aims to provide critical assessment information on the damaged and likely damaged grids with the emergency planners as quickly as possible, without waiting for the end of the assessment horizon.

Given a set of locations (grids) to assess after a disaster and a fleet of drones, we introduce the *integrated drone assessment routing problem* (IDARP), which aims to determine which grids to scan, constructs routes of drones to scan the selected grids, and make predictions on the unscanned grids in each assessment interval throughout the planning horizon in order to determine the damaged grids as quickly as possible while prioritizing the assessment of critical grids with higher priority scores. We use an iterative two-phase approach to address the IDARP, where each iteration corresponds to the problem solved at the beginning of an assessment interval. We provide an overview of our approach in Figure 1.

In the proposed approach, grids are first clustered based on their attributes by using an appropriate clustering method (such as k-modes). Given a cluster of grids, Phase I then focuses on selecting the grids to scan and constructing the routes of drones in each interval,

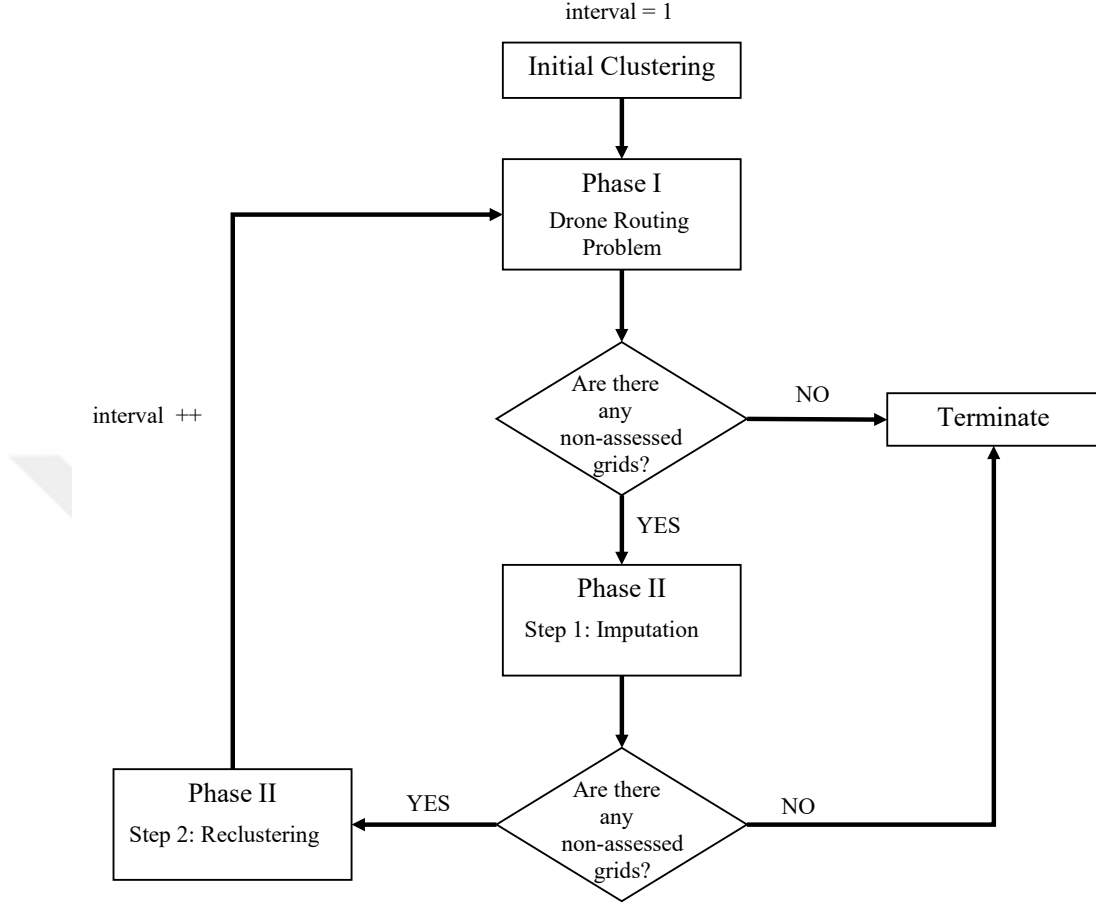


Figure 1: Overview of the two-phase approach for IDARP.

which is formulated as a *drone routing problem*. The aerial images gathered by drones are processed and analyzed by experts to identify the grids that are *damaged*. Afterward, Phase II focuses on processing the assessment information collected by drones and making predictions on the non-assessed grids based on the collected information on the scanned grids that are found to be damaged by applying basic imputation techniques, which aim to predict the *likely damaged* grids based on the scanned damaged grids. Afterward, the scanned grids and likely damaged grids are specified as assessed, and the remaining non-assessed grids are re-clustered. That is, an *assessed grid* means that the grid is either scanned (visited) by a drone or it is unscanned (unvisited) but predicted to be in likely damaged status based on the implemented imputation policy. Phase I and II iterate until all grids become assessed.

An illustration of our approach is explained in an example instance in Figure 2. In this instance, the affected region consists of 25 grids, in which 15 grids must be assessed since they involve built infrastructure. The indices of the grids to be assessed are specified in each cell. The grids are clustered into three clusters based on their attributes, which are indicated by different shapes on the figure. Moreover, the size of the shapes indicates the magnitude of grids' priority scores. For example, grids #1, #3, #4, #12, and #13 are in the same cluster, and the priority scores of grid #1 and #12 are larger. There exist two RSs in the region, one of which is located at the center of grid #4, and the other one is located at the center of a grid that does not subject to assessment. In Phase I of the first interval, shown in Figure 2(a), drones scan and assess six grids in total. One of the drones is recharged at the one of the RSs along its route. As shown in Figure 2(b), in Phase II, grids #10 and #15 are predicted to be damaged. The non-assessed six grids (#1, #3, #4, #5, #7, #8 and #13) are re-clustered, and in the second assessment interval and the drone routes are constructed to scan five of these grids (Figure 2(c)). At the end of the second interval, all grids become in assessed status, therefore the assessment operation terminates. As a result, throughout the assessment period, 11 of the 15 grids are scanned by drones, while four grids are predicted to be likely damaged.

While IDARP is decomposed into two phases, we evaluate the overall performance of the solutions obtained by our approach based on performance measures that focus on the entire assessment plan such as the number of assessment intervals necessary to assess all grids, the number of damaged grids non-assessed after each time interval, and the percentage of the inaccurate predictions made for the likely-damaged grids based on the proposed imputation policies (see Section 5.2.3).

We next describe the models and methods developed to implement the proposed two-phase approach, specifically, to select the grids to scan and optimize the routes of drones (Phase I), and to predict the likely damaged grids and re-cluster the non-assessed grids (Phase II).

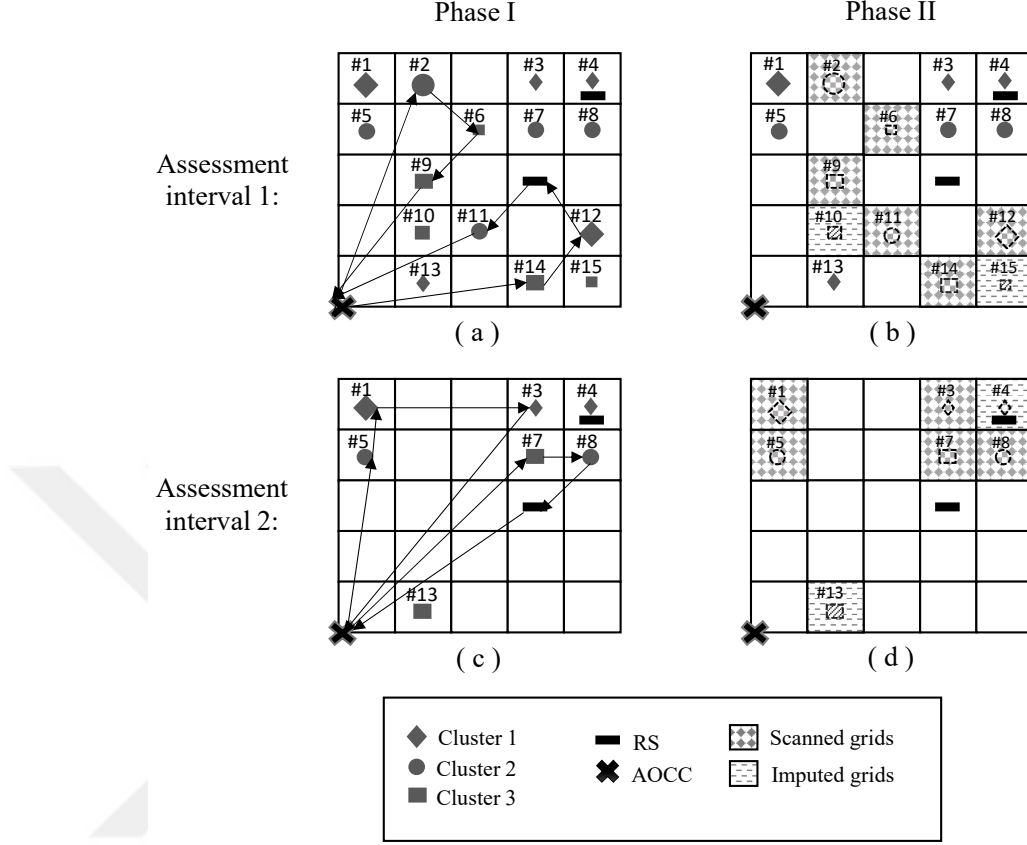


Figure 2: Illustration of the proposed approach and IDARP solution on an example

3.2 Methods

We present different methods to address the decisions to be made in each phase of IDARP. Specifically, in Phase I, we formulate the drone routing problem by proposing two mixed-integer linear programming (MILP) models: arc-based formulation (in Section 3.2.1.1) and path-based formulation (in Section 3.2.1.2). In Phase II, we devise and apply alternative imputation policies to predict the damage levels of the unscanned grids based on collected information and specify the likely damaged grids (Step 1), and then re-cluster the remaining grids by using a proper clustering algorithm (Step 2), which are explained in Section 3.2.2.

The problems to be solved in each phase of our approach are integrated. Specifically, in Phase I of each assessment interval, while it is important to select and scan the grids with the largest priority, it is also desirable not to ignore any clusters and also to collect sufficient

information from each cluster so that better predictions can be made in Phase II. That is, we conjecture that if the majority of the grids in a specific cluster are scanned and found damaged, then the unscanned grids of the cluster can be estimated to be likely damaged. Therefore, in Phase I, we make grid selection and drone routing decisions by targetting to achieve a minimum *coverage ratio* for each cluster, where the coverage ratio is defined as the ratio of the number of visited grids in a cluster to the total number of grids in that cluster. Then in Phase II, predictions are made for the unvisited grids by implementing imputation policies that are developed by evaluating whether the desired coverage is achieved for the cluster and/or the observed damage ratio in a cluster, which is the ratio of detected damaged grids in a cluster to the number of scanned grids. After predictions are made to specify likely-damaged grids by using imputation methods, the remaining non-assessed grids are re-clustered based on their attributes, and they are carried to the next assessment interval (Figure 1). In this way, assessment operations are performed over multiple intervals where information obtained from the previous intervals are used to make decisions in the current interval.

We next explain the methods developed to solve the problems in each phase of our approach.

3.2.1 Phase I. Drone routing problem and formulation

The drone routing problem is defined on a network divided into equisized grids, which are clustered in advance based on their similarities with respect to a set of relevant attributes that affect damage status of the grid after a specific disaster (e.g., earthquake). Moreover, each grid has a priority score indicating the urgency of visiting that grid which is based on the population density and whether the grid contains any critical facilities (such as schools, hospitals).

At the beginning of each assessment interval, the drone routing problem is solved to determine: i) the grids to be scanned, ii) assignment of grids to the drones, iii) the route

of each drone to visit the grids in that interval. There exists a limited number of identical drones. Each drone makes a single trip in an interval, which starts from the AOCC and ends at the AOCC (e.g., see Figure 2(a)). We assume that the travel time between two grids is equivalent to the travel time between the center of these grids. Moreover, when a grid is visited, the drone spends a certain time to scan the grid. The scanning times of grids may be different because although the areas of all grids are equal, the density of buildings in each grid may vary. For instance, the grids may include areas without any built infrastructure (e.g., parks, lakes, forests) and these areas do not require surveying, while dense areas may require more time to scan. Each grid can be scanned at most once throughout the assessment horizon.

Due to their battery limits, drones may need to be recharged along their routes at the battery recharge stations (RSs), which are positioned in the region beforehand. We assume that the RSs are located at the center of grids. The recharging time of a drone at a RS is calculated by assuming a linear battery charging rate similar to (58). We assume uncapacitated RSs and hence multiple drones can be recharged at the same RS simultaneously. We also assume that drones leave the AOCC and the RSs fully-charged.

The primary objective of the drone routing problem is to maximize the number of critical grids (with high priority scores) to scan in each assessment interval. Therefore, it is critical to construct routes that use battery time efficiently, which would be affected by the assignment of grids to the drones and the order of visits to the selected grids and RSs by each drone. Furthermore, as explained before (and illustrated in Figure 1 and Figure 2), Phase I and II interact. Specifically, the aerial information collected by drones from the scanned subset of grids of a cluster in Phase I will be used to make predictions on the unvisited grids of that cluster in Phase II since the grids in a cluster may exhibit similar damage status. That is, the amount of information gathered from a cluster (i.e., the number of grids scanned in a cluster) may affect the accuracy of our predictions regarding unvisited grids. That is, if a large proportion of the cluster grids are visited and their damage status

is identified, one can make a better estimation regarding the damage status of the remaining (unvisited) grids. Therefore, in constructing drone routes, we encourage increasing the number of grids visited from each cluster by defining a coverage ratio target, which is a lower bound on the ratio of the number of visited grids in a cluster to the total number of grids that belong to that cluster. Also, note that, the damage levels of a grid cannot be known before the drone that scans the grid returns to the AOCC, where the aerial images are processed. By targeting a minimum coverage ratio for each cluster in each interval, we avoid neglecting a cluster which may involve damaged grids.

Although we want to achieve a minimum coverage for each cluster in each interval, it may not be possible to visit sufficient grids from each cluster within T_{max} by using a limited number of drones. The existence of any infeasibility cannot be foreseen without solving the drone routing problem. Therefore, to prevent producing infeasible solutions in Phase I, we formulate this concern as a soft constraint rather than imposing a hard-constraint on this requirement. Specifically, we introduce a very large penalty for the maximum unfulfilled coverage in the objective function. As a result, the objective of the drone routing problem is to maximize total priority scores from the visited grids while minimizing the maximum penalty occurs due to unsatisfied coverage targets across the clusters. In Phase II, the proposed imputation policies are based on whether the coverage ratio target of a cluster is satisfied or not in Phase I (see Section 3.2.2).

To formulate the proposed drone routing problem, we adapt arc-based and the path-based formulations presented in (18) for an electric vehicle problem. Specifically, we modify the formulations in (18) to introduce our priority and coverage based objective. Moreover, while all nodes are visited in (18), a selection of a set of grids is required in our problem, which increases its complexity. In Section 5.1.2, we compare the computational times of the proposed arc-based and path-based MILP formulations. The nomenclature for the MILP formulations are provided in Appendix A.

3.2.1.1 Arc-based Formulation

The proposed drone routing problem is first formulated by an arc-based formulation. In the arc-based formulation, vertices represent the grid centers and arcs connect these vertices. Let I represent the set of grids to be scanned, and V'_R defines the set of dummy RSs generated to allow multiple visits to each RS. 0 and $N + 1$ denote the AOCC, such that each route starts at 0 and ends at $N + 1$. Let $V_0 = I \cup V'_R \cup \{0\}$ be the set of departure vertices, and $V_1 = I \cup V'_R \cup \{N + 1\}$ be the set of arrival vertices. $V' = I \cup V'_R$ denotes the set of vertices except the AOCC. We define the drone routing problem on a directed graph $G = (V, A)$, where $V = I \cup V'_R \cup \{0\} \cup \{N + 1\}$ and A is the set of arcs $A = \{(i, j) : i, j \in V, i \neq j\}$.

Let s_j denote the scanning time of grid $j \in I$; and t_{ij} is travel time when a drone travels on arc (i, j) . We represent energy consumption on arc (i, j) in terms of travel time. B specifies the battery capacity, i.e., maximum flight time, and r represents the recharging rate. The route duration limit is denoted by T_{max} . Additionally, the number of available drones is indicated by D .

We let C denotes the set of grid clusters. Let ρ_j represent the priority score assigned to the grid $j \in I$. Moreover, ϕ_{jc} is the coverage parameter that equals 1 if grid $j \in I$ is in cluster $c \in C$; 0 otherwise. Additionally, we define τ_c which indicates the total number of grids in cluster $c \in C$, and μ is the targeted coverage ratio. M is a sufficiently large number, which is used to formulate the soft constraint for achieving the targeted coverage ratio. By setting a very large M , we identify if the solution involves any unfulfilled targeted coverage ratio, as the objective becomes dominated by the resultant penalty.

Let the binary variable x_{ij} equals to 1 if a drone travels on arc (i, j) ; and 0 otherwise. The continuous variables a_{ij} and y_{ij} denote the time and the battery level, respectively, when a drone departs from vertex $i \in V_0$ to travel on arc $(i, j) \in A$. The binary variable z_j takes the value of 1 if and only if the grid $j \in I$ is scanned. The continuous variable q_j denotes the remaining battery when the drone arrives at RS $j \in V'_R$. The continuous variables β_c denote the coverage ratio achieved for cluster $c \in C$, and γ denotes the minimum

coverage ratio achieved overall clusters. Finally, the continuous variable Δ is defined as the maximum unfulfilled coverage ratio across all clusters, which is difference between the targetted coverage ratio and the minimum coverage ratio across clusters (i.e., γ). If the desired coverage target is satisfied for all clusters, Δ equals zero.

The arc-based formulation for the selective drone routing problem is as follows.

$$\max \sum_{j \in I} \rho_j z_j - M\Delta \quad (1)$$

subject to

$$\sum_{\substack{j \in V_1 \\ j \neq i}} x_{ij} \leq 1 \quad \forall i \in I \quad (2)$$

$$\sum_{\substack{j \in V_1 \\ j \neq i}} x_{ij} \leq 1 \quad \forall i \in V'_R \quad (3)$$

$$\sum_{\substack{j \in V_1 \\ j \neq i}} x_{ij} - \sum_{\substack{j \in V_0 \\ j \neq i}} x_{ij} = 0 \quad \forall i \in V' \quad (4)$$

$$\sum_{j \in V_1} x_{0j} = D \quad (5)$$

$$\sum_{i \in V_0} x_{i,N+1} = D \quad (6)$$

$$z_j - \sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} = 0 \quad \forall j \in I \quad (7)$$

$$y_{0j} = Bx_{0j} \quad \forall j \in V_1 \quad (8)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} y_{ij} - \sum_{\substack{i \in V_0 \\ i \neq j}} (t_{ij} + s_j) x_{ij} = \sum_{\substack{l \in V_1 \\ l \neq j}} y_{jl} \quad \forall j \in I \quad (9)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} y_{ij} - \sum_{\substack{i \in V_0 \\ i \neq j}} t_{ij} x_{ij} = q_j \quad \forall j \in V'_R \quad (10)$$

$$\sum_{\substack{l \in V_1 \\ l \neq j}} y_{jl} = \sum_{\substack{l \in V_1 \\ l \neq j}} Bx_{jl} \quad \forall j \in V'_R \quad (11)$$

$$y_{ij} \leq Bx_{ij} \quad \forall i \in V_0, j \in V_1, (i \neq j) \quad (12)$$

$$q_j \leq \sum_{\substack{i \in V_0 \\ i \neq j}} Bx_{ij} \quad \forall j \in V'_R \quad (13)$$

$$a_{0j} = 0 \quad \forall j \in V_1 \quad (14)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} a_{ij} + (t_{ij} + s_j)x_{ij} = \sum_{\substack{l \in V_1 \\ l \neq j}} a_{jl} \quad \forall j \in I \quad (15)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} (a_{ij} + (t_{ij} + rB)x_{ij}) - rq_j = \sum_{\substack{l \in V_1 \\ l \neq j}} a_{jl} \quad \forall j \in V'_R \quad (16)$$

$$a_{ij} \leq (T_{\max} - t_{ij} - s_j - t_{j,N+1})x_{ij} \quad \forall i \in V_0, j \in I, (i \neq j) \quad (17)$$

$$a_{ij} \leq (T_{\max} - t_{ij} - t_{j,N+1} - rB)x_{ij} + rq_j \quad \forall i \in V_0, j \in V'_R, (i \neq j) \quad (18)$$

$$a_{ij} \leq Tx_{ij} \quad \forall i \in V_0, j \in V_1, (i \neq j) \quad (19)$$

$$y_{ij} \geq \left(t_{ij} + \min_{l \in V'_R \cup \{N+1\}} \{t_{jl}\} \right) x_{ij} \quad \forall i \in V', j \in V_1 \quad (20)$$

$$\sum_{j \in I} \phi_{jc} z_j / \tau_c = \beta_c \quad \forall c \in C \quad (21)$$

$$\gamma \leq \beta_c \quad \forall c \in C \quad (22)$$

$$\gamma + \Delta \geq \mu \quad (23)$$

$$a_{ij} \geq 0, \quad y_{ij} \geq 0 \quad \forall i \in V_0, j \in V_1 \quad (24)$$

$$q_j \geq 0 \quad \forall j \in V'_R \quad (25)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in V_0, j \in V_1 \quad (26)$$

$$z_j \in \{0, 1\} \quad \forall j \in I \quad (27)$$

$$\beta_c \geq 0 \quad \forall c \in C \quad (28)$$

$$\gamma \geq 0, \quad \Delta \geq 0 \quad (29)$$

The objective function (1) maximizes the total priority scores of the visited grids and penalizes the maximum unfulfilled targeted coverage ratio. Constraints (2) and (3) imply that each grid and dummy RSs can be visited at most once. Constraints (4) are flow balance equations. Constraints (5) and (6) ensure that the number of drones that departs from and arrives at the AOCC equals the number of available drones, respectively. Constraints

(7) link the x_{ij} and the z_j variables. Constraints (8) set drones to be fully charged when they depart from the AOCC. Constraints (9) denote the remaining battery after a drone traverses the arc (i, j) and scan the grid j . Constraints (10) indicate the remaining battery level at the dummy RS j after traversing the arc (i, j) . Constraints (11) guarantee that drones leave the RSs as fully charged. Constraints (12) couple the variables x_{ij} and y_{ij} , and Constraints (13) link the variables x_{ij} and q_j . Constraints (14) specify the operation starting time. Constraints (15) and (16) track the departure time from the grids and the RSs, respectively. Constraints (17) and (18) ensure that drones return to the AOCC within T_{max} time units. Constraints (19) link the a_{ij} and the x_{ij} variables. Constraints (20) serve as valid inequalities to strengthen the formulation (18), such that the drone must have sufficient battery to arrive at the nearest RS or the AOCC after traversing arc (i, j) . Constraints (21) denote the coverage ratio achieved for each cluster. Constraints (22) indicate the minimum coverage ratio among all clusters. Constraint (23) specifies whether the coverage target is satisfied by the visited grids, and also calculates Δ , which is the difference between the minimum and targeted coverage ratios. Note that the unsatisfied coverage ratio is penalized in the objective function. Finally, Constraints (24)-(29) indicate the domain of decision variables.

3.2.1.2 Path-based Formulation

In the path-based formulation, the starting and ending points of a path can only be the grid centers to be scanned or the AOCC, and any number of RS visits can be included between the starting and the ending points of a path. All paths are enumerated beforehand, and the feasible paths are kept, where feasibility means that the drone leaves a grid center or AOCC with full capacity and arrives at the destination grid center without being out of charge.

We assume a directed graph $G = (V, A)$, with a set of vertices $V = I \cup \{0\} \cup \{N+1\}$ and arcs $A = \{(i, j) : i, j \in V, i \neq j\}$. I denotes the set of grids to be scanned, $V_0 = I \cup \{0\}$ denotes the set of arrival vertices, $V_1 = I \cup \{N+1\}$ is the set of departure vertices, and V_R is the set

of RS nodes. Path p originates from the center of grid $i \in I \cup \{0\}$ and ends at the center of grid $j \in I \cup \{N+1\}$. Then, the set of paths on arc (i, j) is represented by $P_{ij} \in P$. The starting and ending points on path p are $o(p)$ and $d(p)$, respectively. The number of RSs in path p equals n_p . The ordered set of RSs in p is represented by $L_p = \{0, 1, \dots, n_p - 1\}$. $\pi_p(l)$ denotes the RS at position $l \in L_p$. If $L_p = \emptyset$, then the drone doesn't visit any RS on path p .

We re-define the following parameters and the decision variables. We let t_p is the travelling time associated with path $p \in P$. We represent energy consumption along path p in terms of travel time. The decision variable x_p equals 1 if a drone travels on path $p \in P$, and 0 otherwise. The continuous variables a_p and y_p represent the time and remaining battery level before the drone starts travelling on path $p \in P$, respectively. The continuous variable q_{pl} denotes the remaining battery level after the drone travels on path $p \in P$ and visits $\pi_p(l)$.

The path-based formulation for the selective drone routing problem is as follows.

$$\max \sum_{j \in I} \rho_j z_j - M\Delta \quad (30)$$

subject to

$$(21)-(23), (27)-(29)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} \sum_{p \in P_{ij}} x_p \leq 1 \quad \forall j \in I \quad (31)$$

$$\sum_{\substack{j \in V_1 \\ i \neq j}} \sum_{p \in P_{ij}} x_p - \sum_{\substack{j \in V_0 \\ i \neq j}} \sum_{p \in P_{ji}} x_p = 0 \quad \forall i \in I \quad (32)$$

$$\sum_{j \in V_1} \sum_{p \in P_{0j}} x_p = D \quad (33)$$

$$\sum_{i \in V_0} \sum_{p \in P_{i, N+1}} x_p = D \quad (34)$$

$$\sum_{\substack{k \in V_1 \\ k \neq j}} \sum_{p \in P_{jk}} y_p = \sum_{\substack{i \in V_0 \\ i \neq j}} \sum_{p \in P_{ij}} \left(y_p - (t_p + s_j)x_p + \sum_{\substack{l \in L_p \\ |L_p| \neq 0}} (Bx_p - q_{pl}) \right) \quad \forall j \in I \quad (35)$$

$$y_p - t_{o(p), \pi_{p(0)}} x_p = q_{p0} \quad \forall p \in P : |L_p| \neq 0 \quad (36)$$

$$Bx_p - t_{\pi_{p(l-1)}, \pi_{p(l)}} x_p = q_{pl} \quad \forall p \in P : |L_p| \neq 0, l \in L_p \setminus \{0\} \quad (37)$$

$$y_p = Bx_p \quad \forall i \in V_1, p \in P_{0i} \quad (38)$$

$$y_p \leq Bx_p \quad \forall p \in P \quad (39)$$

$$q_{pl} \leq Bx_p \quad \forall p \in P, l \in L_p \quad (40)$$

$$y_p - t_p x_p + \sum_{l \in L_p} (Bx_p - q_{pl}) \geq 0 \quad \forall i \in I, p \in P_{i, N+1} \quad (41)$$

$$\sum_{j \in V_1} \sum_{p \in P_{0j}} a_p = 0 \quad (42)$$

$$\sum_{\substack{k \in V_1 \\ k \neq j}} \sum_{p \in P_{jk}} a_p = \sum_{\substack{i \in V_0 \\ i \neq j}} \sum_{p \in P_{ij}} \left(a_p + (t_p + s_j) x_p + \sum_{\substack{l \in L_p \\ |L_p| \neq 0}} r(Bx_p - q_{pl}) \right) \quad \forall j \in I \quad (43)$$

$$a_p \leq T_{max} - (t_p + s_{d(p)} + t_{d(p), N+1}) x_p - \sum_{\substack{l \in L_p \\ |L_p| \neq 0}} r(Bx_p - q_{pl}) \quad \forall p \in P \quad (44)$$

$$x_p \in \{0, 1\} \quad \forall p \in P \quad (45)$$

$$a_p \geq 0, y_p \geq 0 \quad \forall p \in P \quad (46)$$

$$q_{pl} \geq 0 \quad \forall p \in P, l \in L_p \quad (47)$$

The objective function (30) is the same as the arc-based formulation, which maximizes the total priority scores of the visited grids and penalizes the maximum unfulfilled targeted coverage ratio. Constraints (31) indicate that each grid can be scanned at most once. Constraints (32) are flow balance constraints. Constraints (33) and (34) ensure that the number of drones that depart from and return to the AOCC equals to the number of available drones. Constraints (35) track the drone's battery level at each grid. Constraints (36) indicate the remaining battery level when the drone arrives at the first RS ($\pi_{p(0)}$) on path p , while Constraints (37) indicate the remaining battery levels when the drone departs from an RS and

arrives at the next RS on path $p \in P$. Constraints (38) ensure that drones leave the AOCC fully charged. Constraints (39) and (40) link the battery level variables. Constraints (41) guarantee the path feasibility between a grid and the AOCC, that is, the drone arrives at its destination without being out of battery. Constraints (42) define the starting time of operations in the corresponding assessment interval. Constraints (43) track the departure time from each grid. Specifically, the departure time from grid j equals to sum of the departure time from the previous grid i , scanning time of grid j , traveling time, and the total recharging time on path $p \in P_{ij}$. Constraints (44) link the time tracking variable and the arc travel variable while indicating the route duration, such that all drones must return to the AOCC within T_{max} . Finally, constraints (45)-(47) indicate the domain of decision variables.

Since the number of paths in P grows quickly when the number of RSs and the grids increase, it becomes challenging to solve proposed routing problem. To reduce its computational complexity, we apply the following dominance rules introduced in (61) and (18) and reduce the number of paths. Let $p = (i, k, \dots, l, j)$ be a path including at least one visit to an RS from $i \in V_0$ to $j \in V_1$ traversing the sub-path $(k, \dots, l)$ from $k \in V_R$ to $l \in V_R$, where $k = l$ if the path includes one visit to an RS. A path $p' = (i, k', \dots, l', j)$ from $i \in V_0$ to $j \in V_1$ traversing the sub-path $(k', \dots, l')$ from $k' \in V_R$ to $l' \in V_R$ ($k' = l'$ if the path includes one visit to an RS) dominates path p if: (i) $t_{il'} \leq t_{il}$, (ii) $t_{k'j} \leq t_{kj}$, (iii) $t_{p'} \leq t_p$ and at least one of the inequalities is strict. Note that, the paths including no visit to any RS does not involve any charging decisions, and they cannot be dominated due to the triangular inequality. However, the paths including no RS may dominate other paths.

Our experiments demonstrates that the performance of the path-based formulation without dominance rule and the arc-based formulation is very similar (see Section 5.1.2). However, as the number of dummy RSs affects the arc-based formulation's performance and computing the adequate number of RSs is nontrivial, using path-based formulation may be preferred. Besides, the path-based formulation with dominance rule is shown to perform better than the alternative arc-based formulation (see Section 5.1.2 and (18)).

Therefore, we prefer path-based formulation with dominance rule while solving the instances for Phase I (Section 5.2.2) and Phase II (Section 5.2.3). Still, the mathematical model cannot be solved quickly by using an optimization solver even eliminating dominated paths significantly reduces the solution time. Since the drone routing problem must be solved at the beginning of each assessment interval, it is important to find high-quality solutions rapidly. We develop a heuristic approach to solve the drone routing problem (Chapter 4). We compare the performances of the proposed heuristic and the path-based formulation with dominance rule in Section 5.2.2.

3.2.2 Phase II. Imputation policies (IPs) and re-clustering

After solving the drone routing problem in Phase I and identifying the damaged grids based on the collected aerial images, in Phase II, we devise and apply alternative imputation policies to predict whether the unscanned grids can be specified to be likely damaged based on the information collected from the scanned grids (Step 1). Then the remaining grids, which are not scanned or could not be predicted to be likely damaged, are re-clustered based on their attributes (Step 2) so that decisions for the next assessment interval can be made (Figure 1).

Imputation Policies. As explained before, imputation techniques in data mining are widely used in various applications to impute (infer) the missing values in a data set based on available information. Various imputations techniques are available such as mean-and-mode method, decision tree imputation, and regression imputation (69; 40). In this study, we devise new practical imputation methods that can be used to evaluate the status of unscanned grids based on the information collected from the scanned grids. Specifically, we propose three alternative imputation policies (IPs) to evaluate whether an unscanned grid can be considered as likely damaged based on the set of damaged grids. The proposed IPs differ in terms of three main criteria applied to label an unscanned grid as likely damaged. The first criterion is based on whether the *targeted coverage ratio*, μ , is satisfied for the

grid's cluster or not. The second criterion is defined based on the ratio of the number of scanned grids in a cluster that are labelled as damaged to the total number of scanned grids in that cluster, which is called the *damage ratio* of the cluster. We specify a *damage ratio threshold* for the scanned clusters to evaluate whether the visited cluster has sufficient number of damaged grids. Thirdly, we define a *cumulative damage score* for each grid, which is the sum of the damage ratios of the clusters that the grid has been assigned to since the beginning of the assessment operations till the current assessment interval. Note that in a given assessment interval, the unscanned grids in the same cluster may not have the same cumulative damage score unless their clusters have been the same throughout the all assessment intervals. Then we evaluate whether the cumulative damage score of a grid exceeds a pre-specified *damage score threshold*. We consider one of more of these three criteria in different ways in the proposed IPs.

The **imputation policy 1 (IP1)** classifies the unscanned grids in a cluster as likely damaged if and only if the targeted coverage ratio is met for that grid's cluster and also the damage ratio threshold is satisfied for the cluster. That is, according to IP1, all of the unscanned grids of a cluster can be labelled as likely damaged if the targeted coverage ratio and damage ratio threshold are both achieved for that cluster.

In **imputation policy 2 (IP2)**, we add a second criterion based on the *cumulative damage scores* of grids. Specifically, we classify an unscanned grid as likely damaged if either the criterion in IP1 holds or the cumulative damage score for the grid exceeds the damage score threshold. That is, if the grid has consistently been a member of clusters for a few assessment intervals which had damaged grids but its cluster could not reach the desired coverage or damage ratio, we relax the conditions in IP1 and classify the grid as likely damaged based on its cumulative damage score. Therefore, while IP1 makes a prediction only based on the information collected in the current assessment interval, IP2 also takes into account what we have learned about the grid since the beginning of the assessment operations. Moreover, while IP1 makes a prediction for all unscanned grids in a cluster,

IP2 additionally evaluate the grids individually.

In applying IP2, one can start applying the latter grid-based criterion based on cumulative damage score after a number of assessment intervals to avoid premature labelling of a grid as likely damaged. To specify this, we denote this policy as $IP2^i$, where i is the assessment interval that we start applying the grid-based criterion. We explore the impact of i in Section 5.2.3 while comparing the performances of the IPs.

Finally, we define **imputation policy 3 (IP3)**, which extends IP2 by considering the cumulative damage score collected by a grid as a new grid attribute, which affects the re-clustering step of Phase II (Step 2). That is, IP3 uses the same criteria with IP2 to predict the damage status of the unscanned grids; however, IP3 includes the updated cumulative damage scores of the grids as a new numerical attribute.

Re-clustering. After predicting the status of unscanned grids, the likely damaged grids are removed from the set of non-assessed grids. Then, we apply an efficient suitable clustering method to re-cluster the remaining non-assessed grids (Step 2). Specifically, we employ the k -modes clustering algorithm after IP1 and $IP2^i$ since all grid attributes are categorical. The k -modes algorithm is similar to the k -means algorithm which assigns objects into k clusters. However, while the k -means can only be used for the numerical values, k -modes can deal with the categorical values by making several modifications to the k -means algorithm (45). Particularly, the algorithm utilizes a simple matching dissimilarity measure, modes of the clusters, and a frequency-based method to find the modes. On the other hand, since we consider a new numerical attribute in IP3, we use k -prototypes algorithm which integrates the k -means and the k -modes algorithms to cope with both numerical and the categorical attributes. Here, the same steps in the k -modes algorithm are used to determine clusters, but the dissimilarity measure is modified (see (45)). The steps of the clustering algorithms are provided in Appendix C.

CHAPTER IV

A VARIABLE NEIGHBORHOOD DESCENT (VND) BASED HEURISTIC FOR THE DRONE ROUTING PROBLEM

The proposed drone routing problem, which is solved in each assessment interval, must be solved quickly so that drones can be dispatched rapidly. We develop a heuristic to solve the drone routing problem efficiently. Our heuristic involves two main parts. First, we aim to generate a feasible initial solution by using a constructive heuristic (CH). Secondly, we apply a local search procedure based on a Variable Neighborhood Descent (VND) framework that focuses on maximizing the collected priority scores. We explain the constructive heuristic in Section 4.1 and the VND process in Section 4.2.

4.1 Constructive Heuristic (CH)

We design a constructive heuristic (CH) that involves three steps: (i) route construction, (ii) repair, and (iii) insertion. The first step focuses on constructing routes to minimize the maximum unfulfilled targeted coverage ratio, and the second step is applied only if the first step could not provide a solution that satisfies the coverage target. Then, the third step tries to improve the solution in terms of total priority. We present the pseudocode of the CH in Appendix D.

In the first step of the CH, our aim is to construct routes that satisfy the coverage target for each cluster, while disregarding collected priority scores from the scanned grids. We construct drone routes sequentially such that a new route (drone) is initiated when the previously initiated routes cannot scan any additional grids due to time limitations. For each drone, which is initially positioned at the AOCC, we construct a route by identifying a candidate grid to visit and scan next. Given the current position of a drone, the next grid

to visit is the one that causes the least increase in the route duration when both traveling time and the scanning time are considered, and also one that is not in a cluster achieving the targeted coverage ratio. If there are multiple such grids, the grid with the highest priority score is selected. If the remaining battery is not enough to visit the selected grid, the drone first visits the nearest RS; afterwards, the next grid to visit is selected dependent on the position of the visited RS. The first step of the CH is terminated when the coverage targets for all visited clusters are satisfied or no grids can be added to the current solution feasibly due to time limits.

If the first step of the CH can attain a solution that satisfies the targeted coverage ratios for all clusters, then the algorithm jumps to the third step to apply an insertion operator to improve the priority scores. Otherwise, the algorithm proceeds by the second step, which involves a repair procedure that aims to achieve/converge to the coverage target for each cluster. The repair procedure involves three operators: (i) Remove-add-route, (ii) Add, (iii) Remove-add-coverage. Through the process, the feasibility of the routes with respect to the time limit constraint and the battery constraint must be checked.

The **Remove-add-route** move aims to reduce the route duration without worsening the current coverage ratios. This move is applied for each drone route in the current solution. Specifically, we attempt to remove a visited grid from a route and add an unvisited grid that belongs to the same cluster. While adding a grid, we consider all routes and insert the grid to the best position in terms of route duration. Remove-add-route move continues until route duration cannot be improved further. While the Remove-add-route operator does not improve the coverage ratios directly, it supports increasing the number of visited grids and hence the coverage ratios by the upcoming **Add** operator. The Add operator is applied for each cluster whose current coverage ratio is less than the target. We first pick the cluster with the minimum coverage ratio, and we attempt to add an unvisited candidate grid to improve the coverage ratio. While adding a grid, we evaluate different options that affect battery feasibility; specifically, we consider adding only one grid solution as well as a grid

with its nearest RS. Specifically, let i be a candidate grid, and RS_N is the nearest RS to the center of i , we evaluate inserting $\{i\}$, $\{i, RS_N\}$, and $\{RS_N, i\}$. We make the feasible insertion that leads to the smallest increase in the route duration among all candidate grids. Once a grid is added to the solution, we update the coverage ratios and keep applying the Add operator until the coverage target is achieved for all clusters or any further grids cannot be added to the solution. If the targeted ratio is achieved by Add operator, the repair process terminates. Otherwise, we next apply the **Remove-add-coverage** operator, which focuses on minimizing the maximum unfulfilled coverage target among all clusters. We attempt to remove a visited grid, which is from the cluster with the largest coverage ratio, while adding an unvisited grid from the least covered cluster. The Remove-add-coverage operator is applied until no improving solutions can be found. We present the pseudocode of the repair process in Appendix D. After the repair process, the CH proceeds with the third step.

In the third step of the CH, we apply an **Insertion move**, in which we attempt to insert an unvisited grid into the current solution to increase the total priority scores. First, we sort the unvisited grids in descending order according to their priority scores. While inserting a grid, we consider battery feasibility and insert a grid-RS pair if necessary, as explained in the Add operator. We attempt to insert a single grid or a grid-RS pair option to the best position in any existing or a new route (if an unrouted drone exist) that leads to the smallest increase in total route duration.

4.2 Local Search

After an initial solution is generated by using the CH, we apply *local search* by using the framework of a basic VND algorithm to explore the solution space. The VND is a effective metaheuristic that applies a simple strategy to systematically change the neighborhoods within the local search (70). It is one of the preferred heuristics used to solve E-VRPs (62; 71; 72).

In a classical VND, a set of neighborhood structures and their implementation order is defined beforehand. The algorithm starts with searching the solutions in the first neighborhood, and continues the search in the neighborhood until an improving solution is found. If any improving solutions cannot be identified in a neighborhood, the search continues in the next neighborhood. If an improving solution is found and accepted in a neighborhood, the search returns to the first neighborhood. The algorithm terminates when no improving solutions can be found in any neighborhoods in an iteration. Alternatively, a preset iteration or solution time limit can be considered as termination criteria.

In the local search step of our heuristic, we define two types of neighborhood moves: *route-improving moves* and *priority-improving moves*. Additionally, we define a *diversification* move to explore different parts of the solution space and hence prevent the local search process from getting stuck in the local optima. In total, we generate ten moves consisting of nine different neighborhood structures and a diversification move, where N_k , ($k = 1, \dots, 9$) denotes the finite set of different neighborhood structures and k denotes the order of the move. The pseudocode of the VND is presented in Appendix D.

4.2.1 Route-improving moves.

We devise seven moves that focus on improving the route duration. These moves do not improve the priority objective directly, but they support the forthcoming priority-improving moves to find better solutions by increasing the route efficiency. While applying a move, the feasibility of the routes with respect to the time limit constraint and the battery constraint must be checked. In all of the moves, except the Remove-RS, all grids, which are visited to be scanned or only to recharge batteries, may be affected by the move. We briefly describe the route-improving moves below.

- N_1 : **One-opt.** A classical one-opt move that switches the order of visits to the grids in each route is applied. The best one-opt move is applied for each route.
- N_2 : **Two-opt.** A classical two-opt move is applied, which switches two arcs in each

route and changes the order of grid visits. The best-improving two-opt move is accepted on each route.

- N_3 : **Swap**. This move focuses on each route and swaps the order of visiting two grids. The best-improving swap is accepted for each route.
- N_4 : **Double-swap**. In this move, two consecutive grids are swapped with another the consecutively visited pair of grids along each route. The best-improving move is made for each route.
- N_5 : **Remove-RS**. This move is applied to each route and it reduces the route duration by removing a redundantly visited RS from a route, if any. If there are multiple candidate RSs that can be removed from a route, the move that leads to better route duration is picked.
- N_6 : **Exchange**. Two routes are affected by this move, such that two grid visits in different routes are swapped. All possible exchanges are evaluated, and the best-improving one in terms of the total route duration is realized.
- N_7 : **Remove-add**. This move attempts to remove a visit from one route and add it to another route. All possible solutions are evaluated and the one that leads to the largest decrease in total route duration is accepted.

4.2.2 Priority-improving moves.

We design two moves, **Replace-t** and **Insert-t**, which focus on improving the priority objective while maintaining the feasibility of the routes with respect to the route duration and battery constraints. Both moves are performed based on the best improvement strategy such that the solution that leads to the largest increase in the collected priority scores is accepted in each neighborhood.

Our preliminary experiments have shown that cycling may occur in applying the proposed priority-improving moves. Specifically, when these moves are applied right after the

diversification step (see Section 4.2.3), which focuses on removing grids from the routes to jump to a different area in solution search space, the removed grids may be inserted to the solution immediately by the priority-improving moves, reducing the effectiveness of diversification. Therefore, we use a tabu list, adapted from a classical tabu search (TS) heuristic (73), which keeps the recently made priority-improving moves in a tabu list for a number of iterations (i.e., tabu tenure) to avoid cycling. Thus, only *the admissible solutions* (i.e., the solutions that are not classified as tabu) are evaluated in each iteration of the algorithm. Specifically, we denote the grid which is affected by the replace or insert move as tabu, and we keep that grid in the tabu list for the duration of the tabu tenure. We consider a single tabu list and the same tabu tenure for both moves. As long as the grid stays in the tabu list, Replace-t and Insert-t moves cannot be performed for that grid. However, the grid can be considered by the route-improving moves. We describe the priority-improving moves below.

- N_8 : **Replace-t.** This move attempts to remove a visited grid with a low priority score from the solution while inserting a grid that has a higher priority score by maintaining feasibility in terms of route duration and battery constraints and without worsening the current minimum coverage ratio. To perform this move, the unvisited grids, which are not in the tabu list, are sorted in descending order according to their priority scores. Then, we attempt to remove the grid having the lowest priority score while inserting the grid with the highest score. While inserting a grid to a route an RS may need to be inserted as well due to battery feasibility (as explained in Section 4.1). The selected grid is inserted to the best position in an existing route or a new route (if an unrouted drone exists) that leads to the smallest increase (or the highest decrease) in total traveling time.
- N_9 : **Insert-t.** This move is similar to the insert move explained in Section 4.1, except that we keep a tabu list and do not allow the tabu moves in local search.

4.2.3 Diversification.

If none of the route- or priority-improving moves provide a new solution in an iteration, we apply a diversification strategy that aims to make a fundamental change in the current solution. Our diversification strategy is developed based on the observation that the visited RSs play a critical role in identifying a new feasible solution in terms of battery time, and finding new feasible solutions that visits a new set of RSs may be difficult due to battery constraints. To explore the unvisited parts of the solution space, we design a move called **Relocate**, which removes the first visited RS and the grid visited just before the RS, and attempts to insert another RS to the route starting with the farthest RS to the removed RS. We continue until we find a feasible solution in terms of time and battery limitations. The Relocate move is performed for each route that involves an RS visit. If the solution obtained by the diversification does not achieve the targeted coverage, we apply the repair process (explained in Section 4.1) and then the local search moves (Section 4.2). If an improving solution is obtained, we update the current solution and continue with the VND process.

CHAPTER V

NUMERICAL ANALYSIS

In this chapter, we present the results of the numerical study conducted based on a set of test instances.

In Section 5.1, we solve 54 instances to compare the performance of the arc-based and the path-based formulation with/without dominance rule concerning the solution quality and computational time.

In Section 5.2, we consider a different set of instances to present our analyses for Phase I (48 instances) and Phase II (20 instances) by using the path-based formulation with dominance rule as it performs better. The analyses for Phase I focus on evaluating the proposed VND-based metaheuristic and the mathematical model, while the analyses for Phase II address the overall system performance considering the imputation policies.

All numerical analysis are conducted on HP 8th Generation Intel Core i7 processor with 8 GB of RAM in 64-bit mode.

5.1 Preliminary Analysis of the MILP formulations

In this section, our objective is to understand the impact of the total number of grids and the number of drones on the quality of the solutions and to analyze the sensitivity of system performance to changes in the generation of grid attributes. We solve the proposed MILP formulations by an optimization solver, CPLEX, and compare their performance respect to solution quality and computational time. The problem is solved only for one interval, such that we do not consider the iterative approach while comparing the different formulations. We describe test instances in Section 5.1.1 and evaluate the performance of the alternative formulations in Section 5.1.2.

5.1.1 Instances

We generate 54 instances with 16, 25, and 36 equal-sized grids. The instances differ in how we assign attributes to the grids and the number of drones. We assume that the affected area is divided into square grids with a one-kilometer side length. The operation center is assumed to be located at the left bottom corner of each instance (as in Figure 2).

We consider different scanning times since the density of buildings in each grid may vary. For instance, the grids may include areas without any built infrastructure, such as forests and lakes, and these areas do not require damage assessment. Thus, the scanning times are randomly generated between [10,17] minutes; here, we consider an eBee drone as a benchmark, which can cover 1 km^2 in 17 minutes (74). The priority score of each grid is randomly generated between [1,5].

We generate six different instances for each network that contains 16, 25 or 36 grids where each instance is differentiated based on the assignment of a set of attributes to the grids. We consider six different attributes having different number of categories, and we assign the attributes to the grids following two policies: (i) assign similar attribute categories based on the geographical proximity of grids, which produce C-type instances: C1, C2, and C3, and (ii) assign attribute categories randomly, which generate R-type instances: R1, R2, R3 instances. We then use k -modes algorithm and assign the grids into 5 clusters (45). Figure 3 illustrates exemplary areas of C1 and R1 instances for the area of 25 km^2 where there are five clusters represented by different shapes: a rectangle, a diamond, a triangle, a circle, or a hexagon. Thus, grids represented by the same shape belong to the same cluster.

We define the targeted coverage ratio as 0.25, and the battery capacity (i.e., maximum flight time) as 50 minutes. Similar to (33), we assume that a drone that is completely out of charge is fully charged in five minutes. Accordingly, we set the recharging time required to fill the battery for a 1-minute flight time as 0.1 minutes. The speed of a drone is set at 60 km/h (10). T_{max} is assumed to be 180 minutes, and the number of RSs equals three for each instance. Moreover, to solve the arc-based MILP formulation, we set the

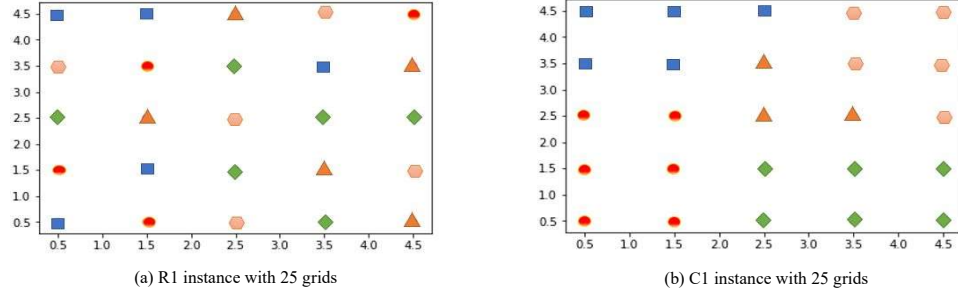


Figure 3: Illustrative example of two types of instances for a 25 km^2 area

number of dummy RSs to three, five, and seven for the instances with 16, 25, and 36 grids, respectively. Finally, we consider using one, two, and four drones.

In our instances, the locations of the RSs are set via solving a p -median problem. Let I be the set of grids in the affected region. J denotes the set of candidate RS locations. Let s_j be the time required to scan grid j . The travel time t_{ij} between grid i and grid j is computed by the Euclidean distance between the center of grids, and p is the number of RSs to be located. The binary variable x_j equals 1 if an RS is located to the center of grid j , and the binary variable y_{ij} equals 1 if grid i is assigned to RS at grid j . The p -median formulation computing the locations of the RSs is as follows.

$$\min \sum_{i \in I} \sum_{j \in J} s_i t_{ij} y_{ij} \quad (48)$$

subject to

$$\sum_{j \in J} y_{ij} = 1 \quad \forall i \in I \quad (49)$$

$$\sum_{j \in J} x_j = p \quad \forall i \in I \quad (50)$$

$$y_{ij} - x_j \leq 0 \quad \forall i \in I, j \in J \quad (51)$$

$$x_j \in \{0, 1\} \quad \forall j \in J \quad (52)$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J \quad (53)$$

The objective (48) is to minimize the demand weighted total distance, where the demand equals the scanning time. Constraints (49) imply that all grids must be assigned to exactly one RS. Constraints (50) specify the number of RSs to be located. Constraints (51) couple the x_j and y_{ij} variables. Constraints (52) and (53) define the domains of the decision variables.

5.1.2 Computational Results of the Preliminary Analysis

We solve 54 test instances by using the proposed formulations. CPLEX 12.8.0 is used to solve the proposed mathematical models by imposing a 30-minute time limit. The clustering algorithm and the path generation are coded in Python 3.6.

As explained before, the arc-based formulation requires introducing dummy RS nodes. Considering a large number of dummy RSs increases the computation time immensely, whereas, setting an insufficient number of RSs may cut off the optimal solution. Since it is impossible to compute the minimum sufficient number of dummy RSs in advance, we make experiments by setting a different number of dummy RSs. Specifically, we consider three, five, and seven dummy RSs for instances with 16, 25, and 36 grids, respectively.

Tables 1, 2 and 3 represent the computational results obtained by CPLEX for both arc-based and path-based formulations. The objective function values are denoted by obj_A , obj_P and obj_{Pd} for the arc-based, path-based without dominance rule and path-based with dominance rule (path-based-d) formulations, and $\bar{\gamma}$ denotes the minimum coverage ratio achieved among all clusters, computed via the corresponding method. $time$ indicates the computation time in seconds, and gap denotes the optimality gap. Finally, $\bar{\Delta}$ denotes the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

We note that, the path-based formulation requires generating all possible paths in advance. The paths are generated in 3 seconds, 20 seconds, and 95 seconds on average for the instances with 16, 25, and 36 grids, respectively, and they are not considered while

Table 1: Computational results of the MILP formulations for 6 networks (Net.) having 16 grids varying the number of drones (D).

Instances		Arc-based				Path-based				Path-based-d			
Net.	D	$\bar{\gamma}$	obj_A	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_P	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_{Pd}	$time$ (sec)	gap (%)
R1	1	0.34	40	25.8	0.0	0.34	40	15.5	0.0	0.34	40	5.1	0.0
R1	2	1.00	51	0.6	0.0	1.00	51	1.5	0.0	1.00	51	1.1	0.0
R1	4	1.00	51	0.6	0.0	1.00	51	1.4	0.0	1.00	51	0.9	0.0
R2	1	0.33	40	1.2	0.0	0.33	40	7.9	0.0	0.33	40	1.3	0.0
R2	2	1.00	51	0.6	0.0	1.00	51	1.5	0.0	1.00	51	1.1	0.0
R2	4	1.00	51	0.6	0.0	1.00	51	1.3	0.0	1.00	51	0.8	0.0
R3	1	0.25	41	1.2	0.0	0.25	41	1.5	0.0	0.25	41	2.7	0.0
R3	2	1.00	51	0.6	0.0	1.00	51	1.4	0.0	1.00	51	1.1	0.0
R3	4	1.00	51	0.6	0.0	1.00	51	1.3	0.0	1.00	51	0.8	0.0
C1	1	0.25	41	1.7	0.0	0.25	41	2.1	0.0	0.25	41	1.5	0.0
C1	2	1.00	51	0.6	0.0	1.00	51	1.5	0.0	1.00	51	1.1	0.0
C1	4	1.00	51	0.6	0.0	1.00	51	1.4	0.0	1.00	51	0.8	0.0
C2	1	0.50	41	2.0	0.0	0.50	41	2.2	0.0	0.50	41	2.1	0.0
C2	2	1.00	51	0.6	0.0	1.00	51	1.6	0.0	1.00	51	1.1	0.0
C2	4	1.00	51	0.6	0.0	1.00	51	1.4	0.0	1.00	51	0.8	0.0
C3	1	0.25	40	30.8	0.0	0.50	40	34.3	0.0	0.50	40	22.0	0.0
C3	2	1.00	51	0.6	0.0	1.00	51	1.5	0.0	1.00	51	1.2	0.0
C3	4	1.00	51	0.6	0.0	1.00	51	1.3	0.0	1.00	51	0.9	0.0
Max				30.8	0.0			34.3	0.0			22.0	0.0
Avg				3.9	0.0			4.5	0.0			2.6	0.0
Min				0.6	0.0			1.3	0.0			0.8	0.0

evaluating the formulations.

All instances with 16 grids are solved to optimality by both formulations within a 30-minutes time limit (usually within a couple of seconds), as shown in Table 1. When the number of drones is one, $\bar{\gamma}$ is always less than 100 %. However, $\bar{\gamma}$ becomes 100 % as more drones are used to scan the region.

For instances with 25 grids, 12 out of 18 (12 out of 18) instances are solved to optimality by the arc-based (path-based) formulation. Average computation times of the arc-based (path-based) formulation are 624 (754) and 37 (230) seconds for the all of the instances with 25 grids and for the all of the instances with 25 grids excluding the ones that cannot be solved within the computation time limit, respectively. Even if the optimality gap of the path-based formulation is higher than that of the arc-based formulation for some of the instances, the computed objective function values are all the same for both formulations. Eliminating dominated path causes a slight decrease in the optimality gap (e.g., 1.4 %),

Table 2: Computational results of the MILP formulations for 6 networks (Net.) having 25 grids varying the number of drones (D).

Instances		Arc-based				Path-based				Path-based-d			
Net.	D	$\bar{\gamma}$	obj_A	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_P	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_{Pd}	$time$ (sec)	gap (%)
R1	1	0.33	42	2.9	0.0	0.28	42	5.1	0.0	0.28	42	13.3	0.0
R1	2	0.66	68	1800	1.5	0.66	68	1800	2.9	0.66	68	1800	2.2
R1	4	1.00	73	2.0	0.0	1.00	73	3.6	0.0	1.00	73	2.7	0.0
R2	1	0.25	41	179.5	0.0	0.25	41	1093	0.0	0.25	41	176.4	0.0
R2	2	0.66	68	1800	1.5	0.60	68	1800	2.9	0.60	68	1800	2.4
R2	4	1.00	73	2.3	0.0	1.00	73	3.4	0.0	1.00	73	2.8	0.0
R3	1	0.33	40	230.5	0.0	0.33	40	1569.3	0.0	0.33	40	426.9	0.0
R3	2	0.60	68	1800	2.0	0.60	68	1800	2.9	0.60	68	1800	1.5
R3	4	1.00	73	2.2	0.0	1.00	73	3.6	0.0	1.00	73	2.7	0.0
C1	1	0.33	41	6.7	0.0	0.33	41	6.1	0.0	0.33	41	9.4	0.0
C1	2	0.33	68	1800	1.5	0.33	68	1800	3.0	0.33	68	1800	2.9
C1	4	1.00	73	2.3	0.0	1.00	73	3.6	0.0	1.00	73	2.8	0.0
C2	1	0.34	42	8.0	0.0	0.34	42	64.0	0.0	0.34	42	6.4	0.0
C2	2	0.68	68	1800	1.9	0.80	68	1800	2.9	0.80	68	1800	2.5
C2	4	1.00	73	2.6	0.0	1.00	73	3.4	0.0	1.00	73	2.9	0.0
C3	1	0.25	42	3.2	0.0	0.25	42	5.2	0.0	0.25	42	17.1	0.0
C3	2	0.51	68	1800	1.5	0.66	68	1800	2.9	0.66	68	1800	1.5
C3	4	1.00	73	2.1	0.0	1.00	73	3.8	0.0	1.00	73	2.6	0.0
Max				1800	2.0			1800	3.0			1800	2.9
Avg				624.7	0.6			753.6	1.0			637.0	0.7
Min				2.0	0.0			3.4	0.0			2.6	0.0

and the average computation time reduces from 754 seconds to 637 seconds. Additionally, scanning with two drones is not enough to reach 100% coverage in 25-grid instances. We observe that increasing the number of drones from two to four helps to solve these instances optimally. This implies that the problem may become easier to solve if there exist enough resources to visit all grids; that is, the problem structure may be more complicated when grid selection must be made along with routing decisions.

Similar results are observed for the instances with 36 grids. The main differences are that computed objective function values by the arc-based and path-based formulations are not always the same. For instances with 36 grids, 8 out of 18 (10 out of 18) instances are solved to optimality by the arc-based (path-based and path-based-d) formulation. Similar to the instances with 25 grids, increasing the number of drones helps to solve the instances to optimality for instances. In addition, average computation times of the arc-based (path-based) formulations are 1212 (1125) and 478 (586) seconds for the all of the instances

Table 3: Computational results of the MILP formulations for 6 networks (Net.) having 36 grids varying the number of drones (D). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

Instances		Arc-based				Path-based				Path-based-d			
Net.	D	$\bar{\gamma}$	obj_A	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_P	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_{Pd}	$time$ (sec)	gap (%)
R1	1	0.27	50	779.5	0.0	0.27	50	232.8	0.0	0.27	50	192.9	0.0
R1	2	0.45	86	1800	3.8	0.34	85	1800	3.5	0.45	86	1800	2.2
R1	4	1.00	113	180.9	0.0	1.00	113	334	0.0	1.00	113	48.7	0.0
R2	1	0.27	41	1800	6.3	0.28	42	1735	0.0	0.28	42	1647	0.0
R2	2	0.30	82	1800	7.3	0.30	82	1800	6.1	0.30	82	1800	6.0
R2	4	1.00	113	274.8	0.0	1.00	113	68.1	0.0	1.00	113	162.6	0.0
R3	1	0.28	45	1800	9.1	0.28	46	829.3	0.0	0.28	46	495.2	0.0
R3	2	0.40	85	1800	4.6	0.40	85	1800	3.2	0.40	85	1800	3.2
R3	4	1.00	113	282.8	0.0	1.00	113	32.8	0.0	1.00	113	43.7	0.0
C1	1	0.33	46	1800	6.6	0.33	44	1800	8.1	0.33	46	622.7	0.0
C1	2	0.33	85	1800	4.7	0.33	85	1800	3.3	0.33	85	1800	3.2
C1	4	1.00	113	260.9	0.0	1.00	113	1729	0.0	1.00	113	22.4	0.0
C2	1	0.33	46	634.5	0.0	0.22	48- $\bar{\Delta}$	1800	100	0.33	45	1800	3.8
C2	2	0.33	85	1800	4.7	0.33	85	1800	3.3	0.33	85	1800	3.2
C2	4	1.00	113	1075.9	0.0	1.00	113	64.5	0.0	1.00	113	40.4	0.0
C3	1	0.33	45	1800	8.0	0.33	46	801.7	0.0	0.33	46	1800	2.8
C3	2	0.34	85	1800	4.5	0.34	85	1800	3.3	0.34	85	1800	3.3
C3	4	1.00	113	339	0.0	1.00	113	34.4	0.0	1.00	113	53.3	0.0
Max				1800	9.1			1800	100			1800	6.0
Avg				1212	3.3			1125	7.3			985.2	1.5
Min				180.9	0.0			32.8	0.0			22.4	0.0

with 36 grids and for the all of the instances with 36 grids excluding the ones that cannot be solved within the computation time limit, respectively. Applying a dominance rule decreases the average computation time by 140 seconds. Besides, path-based formulation cannot achieve the targeted coverage ratio for one instance (Network C2 with one drone); however, path-based-d results in 3.8 % gap while satisfying the targeted coverage.

Setting the number of dummy RSs for larger instances is nontrivial when the arc-based formulation is used, hence, one may prefer utilizing the path-based formulation as both formulations show similar performances. Besides, applying dominance rule reduces the computation time and improve the solution quality. Therefore, we use path-based formulation with dominance rule for our analyses.

5.2 Numerical Analysis

In this section, we present the results of the numerical study conducted based on a set of test instances in order to (i) understand the impact of the system parameters (e.g., total number of grids, total number of drones, the targeted coverage ratio, and the length of an assessment interval) on the quality of the solutions, (ii) measure the solution quality of the proposed VND-based metaheuristic in terms of the objective function values and the CPU times, and (iii) understand the impact of imputation policies and re-clustering (Phase II of IDARP) on a set of performance measures focusing on the efficiency and the effectiveness of assessment operations.

5.2.1 Instances

We generate 48 instances only for Phase I and 20 instances for IDARP (i.e., Phase I and Phase II iterate over multiple intervals until all grids are assessed). The instances for Phase I differ in total number of grids, the total number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ), whereas, the instances for Phase II differ in μ and the damage ratio threshold.

We assume that the affected area is a square with sides of 10-kilometer and is divided into 100 equal squares (with sides of 1-kilometer). Grids are chosen randomly over the centers of these 100 squares depending on the number of required grids for each instance. The AOCC is assumed to be located at the left bottom corner of each affected area (as illustrated in Figure 2). Additionally, the location of the disaster epicenter is randomly selected among the centers of these 100 squares.

Each grid is assigned six attributes consisting of five pre-disaster attributes (e.g., the building material having four categories (such as stone or reinforced) (75), building types having three categories (such as low-rise or high-rise buildings) (76), time of construction having three categories such as (pre- 1979 or post-2000) (76), occupancy having three categories (such as residential or commercial) (16), soil type having four categories (such

as soft soil or rock) (65)), and a post-disaster attribute (e.g., distance to the epicenter (16)). We assume that the attributes overall a grid's area are the same, however one can compute median or weighted average of any attribute if that attribute has different values over a grid's area. In addition, the attributes can be transformed into the categorical values before clustering if necessary. For instance, proximity to the epicenter is a numerical attribute, but we define two categories by specifying a grid is in a high-impact (low-impact) zone if the grid's distance to the epicenter is less than (more than) the average distance of all grids to the epicenter.

We assume that the grids geographically close to each other may have similar attributes in real-world settings, such as similar housing estate. To address such settings, we apply the procedure introduced by (77) considering the geographical proximity while assigning pre-disaster attributes to the grids. In particular, we assume that close grids have two similar attributes that are randomly chosen, and the remaining three attributes are assigned to the grids randomly. The post-disaster attribute is based on the grids' proximity to the disaster epicenter, thus it is not included in the pre-disaster attribute generation procedure. The steps of this procedure are represented in Algorithm 1.

Algorithm 1: Procedure to assign pre-disaster attributes to the grids.

- Step 0.** Let D be the average distance across all grids in the network.
Calculate D to compare the proximity among grids in Step 3.
- Step 1.** Select a grid randomly and denote it as grid i .
Assign each all pre-disaster attributes to grid i randomly.
- Step 2.** Select the closest grid to grid i , and denote it as grid j .
- Step 3.** If the distance between grids i and j is less than D , go to Step 3.1;
otherwise go to Step 3.2.
- Step 3.1.** Pick two pre-disaster attributes belonged to grid i randomly.
Assign these attributes to grid j .
Assign the remaining three pre-disaster attributes to grid j randomly.
- Step 3.2.** Assign all pre-disaster attributes to grid j randomly.
- Step 4.** Set $i = j$ and go to Step 2.
-

After generating the attributes, the grids are clustered by using k -modes algorithm (as introduced in Section 3.2.2) where the number of clusters k is determined based on the

elbow-method (78).

Additionally, grids have different priorities reflecting their population density or critical facilities, such as hospitals, located in a grid. These priorities are computed based on three features (and their corresponding scores): (i) population density that is categorized as low (0), moderate (0.5) or high (1), (ii) building types indicating whether there is an important facility (1) or not (0), and (iii) damage probability that is considered as either high (1) or low (0.5). For each grid, the first two features are generated randomly, whereas, the third feature is based on the grid's attributes.

To compute the damage probability of a grid, we first define three levels for the values of each attribute: high-risk, moderate-risk or low-risk corresponding to scores 1, 0.5, and 0. For instance, soft soil (hard rock) category is assigned to the high-risk (low-risk) level, hence, corresponds to a score of 1 (0) as high levels of damage are expected for the corresponding grid. As each grid has six attributes, the highest total score computed due to a grid's attributes is 6. Then, we define the *damage probability* of a grid as the average score of the grid's attributes that equals the ratio of the grid's total attribute scores to the total number of attributes (6 in our instances). If the damage probability is less than (greater than or equal to) 0.5, the level is considered as low (high) resulting in the score of the damage probability equal 0.5 (1). In other words, the damage probability feature may increase the chance of rapid assessment of grids that are more likely to have a structural collapse in case of a disaster. Finally, the priority score of a grid equals to the sum of these three features' scores, such that each grid's priority score lies between the interval [0.5,3].

We also need to generate each grid's actual damage levels, especially while evaluating whether damaged grids are assessed rapidly. Hence, the actual damage levels of a grid is generated via a Bernoulli random variable where the probability of success, when success indicates that a grid is actually damaged, equals to the grid's damage probability score.

A grid's scanning time is generated based on each grid's population density as the number of buildings is assumed to increase with an increasing population. We consider

eBee drone as a benchmark that can cover 1 km^2 in 17 minutes (74) and generate the scanning times as 10, 15, and 20 minutes for low, moderate, high population densities, respectively.

We define the battery capacity (i.e., maximum flight time) as 50 minutes. As explained in Section 5.1.1, we set the recharging time required to fill the battery for a 1-minute flight time as 0.1 minutes, and the speed of a drone is set at 60 km/h (10).

The travel time between two grids equals the Euclidean distance between the center of these grids. There are three RSs whose locations are computed via solving a p -median problem as described in Section 5.1.1. We solve 48 test instances varying the networks altering the number of grids (6 choices), the number of drones (2 choices), T_{max} (2 choices), and the targeted coverage ratio (2 choices). CPLEX 12.8.0 is used to solve the proposed MILP formulation with a one-hour computational time limit. The heuristic and the clustering algorithms are coded in Python 3.6. Additionally, we define the targeted coverage ratio, μ , as 0.25 or 0.5, the number of drones as one or two (for networks with less than 50 grids) or six or eight (for networks with 50 grids), and T_{max} as 90 and 180 minutes.

We summarize our numerical analysis for Phase I and IDARP in Section 5.2.2 and Section 5.2.3, respectively.

5.2.2 Computational Results of Phase I

We compare the solution qualities of the proposed MILP formulation and the proposed VND-based metaheuristic in terms of their objective function values and their CPU times while routing the drones. First, we generate different networks by varying the number and the location of the grids as the grids are chosen over 100 squares randomly as described in Section 5.2.1. We generate 2 networks with 15 grids, 2 networks with 25 grids, 4 networks with 50 grids. We vary the number of drones as 1 and 2 for the networks with 15 and 25 grids, whereas, the number of drones are 6 or 8 for the networks with 50 grids.

The tabu tenure is set to four (eight) for the instances with 15 and 25 (50) grids. Our

preliminary experiments show that the best computed solution cannot be improved after 2000 iterations. Although the best solution may be found prematurely for some of the instances, we set the number of iterations to 2000 for each instance considering the trade-off between the solution quality and the computation time.

Computational results of the proposed MILP formulation and the proposed metaheuristic are compared based on: (i) $\bar{\gamma}$ corresponding to the computed minimum coverage ratio across all clusters, (ii) obj_{MILP} and obj_{VND-b} representing the objective function values of the MILP formulation and the VND-based metaheuristic, respectively, (iii) *time* representing CPU times in seconds, (iv) gap_{MILP} representing the optimality gap of the MILP, and (v) gap_{VND-b} that is computed as $(obj_{MILP} - obj_{VND-b}) / obj_{MILP}$ for the instances whose targeted coverage ratios are satisfied. In other words, gap_{VND-b} represents how far is the objective function of the VND-based metaheuristic from the objective function of the MILP formulation for the instances whose targeted coverage ratios are satisfied. Note that our original objective is to maximize the total priority scores overall assessed grids while satisfying the targeted coverage ratio across all clusters. However, we cannot guarantee that the targeted coverage ratio can be satisfied for each instance. For instance, it may not be possible to satisfy the targeted coverage ratio with the available number of drones and/or T_{max} . Hence, we introduce the targeted coverage ratio as a soft constraint and penalize the maximum unfulfilled coverage target ratio across the clusters at the objective function as $-M\bar{\Delta}$ where $\bar{\Delta}$ represents the maximum unfulfilled coverage target. We do not report gap_{VND-b} if the MILP formulation and/or the metaheuristic could not satisfy the targeted coverage ratio.

Table 4 represents the computational results of two networks with 15 grids varying the number of drones, the length of the assessment interval T_{max} , and the targeted coverage ratio μ . The targeted coverage ratios cannot be satisfied for 2 and 3 instances out of 16 instances by the MILP formulation and the metaheuristic, respectively (i.e., $M\bar{\Delta}$ is subtracted

from the corresponding instance's objective function value). Even if the VND-based metaheuristic satisfies the targeted coverage ratios for fewer instances than the MILP, its CPU time is much shorter than the MILP as represented in Table 4. In addition, all instances with 15 grids are solved within a one-hour time limit (usually within a couple of seconds). Solving the MILP to optimality usually results in higher targeted coverage ratios than the ones computed via the metaheuristic, hence, average gap_{VND-b} is 4.3 %, whereas, the maximum gap_{VND-b} is 13.3 %. Increasing the targeted coverage ratio and/or decreasing T_{max} and/or increasing the number of drones increases the CPU time as expected. If the number of drones or T_{max} increases, the minimum achieved targeted coverage ratio across all clusters increases, resulting in increasing objective function values as more grids can be scanned. However, increasing the targeted coverage ratio may decrease the objective function value as drones may prefer scanning other grids to satisfy the increased targeted coverage ratio instead of increasing the total priority.

Table 5 represents the computational results of two networks with 25 grids varying the number of drones, the length of the assessment interval T_{max} , and the targeted coverage ratio μ . First, solving the MILP formulation to optimality becomes harder due to the increasing number of grids. 8 and 9 instances out of 16 instances could not satisfy the targeted coverage ratios via solving the MILP and the metaheuristic, respectively. In addition, 5 instances out of 16 instances are stopped due to the computation time limit of an hour for the MILP, thus, resulting in 2.4 % average optimality gap. Note that the metaheuristic may compute better objective function values than the MILP if the MILP is stopped due to the computation time limit. Hence, gap_{VND-b} can become negative as represented for 2 instances in Table 5. Other results, such as the effect of increasing the targeted coverage ratio and/or decreasing T_{max} , are similar to the networks with 15 grids.

Increasing the number of grids to 50 prevents computing the optimal solution of the MILP formulation as all runs are terminated due to the computational time limit as shown in Table 6. Average gap_{MILP} over 12 (16) instances is 14.7 % (36.1 %) when we exclude

Table 4: Computational results of the MILP formulation and the VND-based metaheuristic for 2 networks (Net.) having 15 grids varying the number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

Instances											
Net.	D	T_{max}	μ	$\bar{\gamma}$	obj_{MILP}	$time$ (sec)	gap_{MILP} (%)	$\bar{\gamma}$	obj_{VND-b}	$time$ (sec)	gap_{VND-b} (%)
1	1	90	0.25	0.25	7.5	3.3	0.0	0.25	7.5	5.0	0.0
1	2	90	0.25	0.33	15	576.0	0.0	0.33	14	4.4	6.7
1	1	180	0.25	0.33	16	1.8	0.0	0.33	16	7.4	0.0
1	2	180	0.25	1.00	22	1.5	0.0	1.00	22	6.9	0.0
1	1	90	0.50	0.25	7.5-M $\bar{\Delta}$	16.8	0.0	0.00	8-M $\bar{\Delta}$	3.2	-
1	2	90	0.50	0.50	15	18.7	0.0	0.33	14.5-M $\bar{\Delta}$	4.8	-
1	1	180	0.50	0.50	16	3.0	0.0	0.50	15.5	10.0	3.1
1	2	180	0.50	1.00	22	1.5	0.0	1.00	22	8.0	0.0
2	1	90	0.25	0.25	8	1.2	0.0	0.25	7.5	4.5	6.3
2	2	90	0.25	0.40	15	45.4	0.0	0.25	13.5	5.3	10.0
2	1	180	0.25	0.40	16	1.1	0.0	0.40	14	10.7	12.5
2	2	180	0.25	1.00	21.5	2.1	0.0	0.80	20.5	8.0	4.7
2	1	90	0.50	0.25	8-M $\bar{\Delta}$	3.5	0.0	0.00	8-M $\bar{\Delta}$	3.1	-
2	2	90	0.50	0.50	15	17.2	0.0	0.50	13	4.1	13.3
2	1	180	0.50	0.50	15.5	2.9	0.0	0.50	15.5	7.1	0.0
2	2	180	0.50	1.00	21.5	2.1	0.0	1.00	21.5	7.9	0.0
				Max				Max			
				Avg				Avg			
				Min				Min			

Table 5: Computational results of the MILP formulation and the VND-based metaheuristic for 2 networks (Net.) having 25 grids varying the number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

Instances											
Net.	D	T_{max}	μ	$\bar{\gamma}$	obj_{MILP}	$time$ (sec)	gap_{MILP} (%)	$\bar{\gamma}$	obj_{VND-b}	$time$ (sec)	gap_{VND-b} (%)
1	1	90	0.25	0.00	10.5-M $\bar{\Delta}$	414.8	0.0	0.00	10.5-M $\bar{\Delta}$	26.8	-
1	2	90	0.25	0.28	19.5	128.5	0.0	0.25	18	52.6	7.7
1	1	180	0.25	0.25	20.5	87.7	0.0	0.28	19.5	114.4	4.9
1	2	180	0.25	0.50	34.5	3600	3.1	0.42	34	146.2	1.4
1	1	90	0.50	0.00	10.5-M $\bar{\Delta}$	387.9	0.0	0.00	10-M $\bar{\Delta}$	33.9	-
1	2	90	0.50	0.28	19.5-M $\bar{\Delta}$	3600	22.8	0.14	18-M $\bar{\Delta}$	27.8	-
1	1	180	0.50	0.33	18.5-M $\bar{\Delta}$	59.0	0.0	0.14	19.5-M $\bar{\Delta}$	20.6	-
1	2	180	0.50	0.50	34.5	3600	3.1	0.50	34	198.7	1.4
2	1	90	0.25	0.00	10-M $\bar{\Delta}$	509.0	0.0	0.00	10-M $\bar{\Delta}$	36.4	-
2	2	90	0.25	0.25	18	409.7	0.0	0.20	14.5-M $\bar{\Delta}$	46.8	-
2	1	180	0.25	0.25	19	18.6	0.0	0.25	19	134.5	0.0
2	2	180	0.25	0.33	34.5	3600	4.3	0.33	35	152.1	-1.4
2	1	90	0.50	0.00	10-M $\bar{\Delta}$	469.3	0.0	0.00	10-M $\bar{\Delta}$	34.3	-
2	2	90	0.50	0.33	18-M $\bar{\Delta}$	61.2	0.0	0.20	17.5-M $\bar{\Delta}$	46.7	-
2	1	180	0.50	0.33	18.5-M $\bar{\Delta}$	74.7	0.0	0.20	18.5-M $\bar{\Delta}$	48.2	-
2	2	180	0.50	0.50	34	3600	5.2	0.50	34.5	181.3	-1.5
					Max	3600	22.8		Max	198.7	7.7
					Avg	1288.8	2.4		Avg	81.3	1.8
					Min	18.6	0.0		Min	20.6	-1.5

:- The targeted coverage ratio cannot be satisfied for this instance via at least one of the methods.

We observe that the quality of the solution computed by the proposed VND-based metaheuristic increases if the targeted coverage ratios can be satisfied at earlier iterations. Although the MILP formulation frequently performs better than the metaheuristic for the instances with 15 and 25 grids, the metaheuristic performs better than the MILP formulation for most of the instances with 50 grids in terms of both the quality of the solution and the CPU times. Hence, one may prefer to use the proposed metaheuristic for large instances as waiting for a one-hour computation time limit may not guarantee a high-quality solution. Note that one can increase the computation time limit to increase the quality of the MILP formulation's solution, however, quick allocation of critical emergency resources

Table 6: Computational results of the MILP formulation and the VND-based metaheuristic for 4 networks (Net.) having 50 grids varying the number of drones (D), the length of an assessment interval (T_{max}), and the targeted coverage ratio (μ). $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

Instances											
Net.	D	T_{max}	μ	$\bar{\gamma}$	obj_{MILP}	$time$ (sec)	gap_{MILP} (%)	$\bar{\gamma}$	obj_{VND-b}	$time$ (sec)	gap_{VND-b} (%)
1	6	90	0.25	0.44	51	3600	13.1	0.33	51.5	156.6	-1.0
1	8	90	0.25	0.54	60.5	3600	14.0	0.33	59.5	103.4	1.7
1	6	90	0.50	0.48	51-M $\bar{\Delta}$	3600	100	0.50	38.5	100.0	-
1	8	90	0.50	0.54	60.5	3600	14.0	0.55	59	165.2	2.5
2	6	90	0.25	0.45	49.5	3600	12.9	0.40	49.5	166.9	0.0
2	8	90	0.25	0.68	61.5	3600	9.6	0.60	60	122.6	2.4
2	6	90	0.50	0.48	48.5-M $\bar{\Delta}$	3600	100	0.50	49	125.7	-
2	8	90	0.50	0.65	60	3600	12.2	0.68	62.5	170.4	-4.2
3	6	90	0.25	0.42	48.5	3600	18.1	0.30	50	174.9	-3.1
3	8	90	0.25	0.56	57.5	3600	21.0	0.63	64	142.0	-11.3
3	6	90	0.50	0.48	46.5-M $\bar{\Delta}$	3600	100	0.54	39.5	123.6	-
3	8	90	0.50	0.56	62	3600	11.9	0.60	58.5	188.6	5.6
4	6	90	0.25	0.27	51	3600	14.7	0.35	52	104.3	-2.0
4	8	90	0.25	0.42	60	3600	18.0	0.60	64	128.1	-6.7
4	6	90	0.50	0.42	52-M $\bar{\Delta}$	3600	100	0.42	43.5-M $\bar{\Delta}$	135.2	-
4	8	90	0.50	0.56	60.5	3600	17.5	0.60	63.5	161.1	-5.0
					Max	3600	100		Max	188.6	5.6
					Avg	3600	36.1		Avg	141.8	-1.7
					Min	3600	9.6		Min	100.0	-11.3

-: The targeted coverage ratio cannot be satisfied for this instance via at least one of the methods.

(e.g., search and rescue teams, medical teams) among the affected locations is vital encouraging the usage of the proposed metaheuristic.

We also present the computational results for the instances with 15 and 25 grids solved by the path-based formulation with/without dominance rule in Appendix B.

5.2.3 Computational Results of Phase II: IDARP

We focus on the recursive application of the drone routing problem, Phase I, that is followed by an imputation policy and re-clustering during Phase II, IDARP. We solve the IDARP until all grids are assessed by the MILP formulation. The objective of our numerical study for the IDARP are: to analyze the impact of (i) different imputation policies, and (ii) the

changes in the system parameters (e.g., the targeted coverage ratio and the damage ratio threshold) on the following performance metrics where the first performance metric, e_1 , is to measure the efficiency of the IDARP, whereas the remaining performance metrics, $e_2 - e_4$, are to measure the effectiveness of our proposed imputation policies.

1. **Total number of assessment intervals till all grids are assessed (e_1):** The smaller the value of this performance metric, the better it is since we prefer to assess all grids in a short period of time.
2. **Sum of time-weighted total number of non-assessed damaged grids overall assessment intervals (e_2):** Since we have a limited number of drones, it may be impossible to assess all damaged grids in an assessment interval. Hence, we would like to track the total number of non-assessed damaged grids (neither scanned nor estimated as likely damaged) at the end of each assessment interval (i.e., after applying Phase II) and compute the sum of assessment interval-weighted total number of non-assessed damaged grids overall assessment intervals. The smaller this performance metric, the better it is as it means that damaged grids are assessed rapidly.
3. **Percentage of non-damaged grids that are estimated as damaged (e_3):** The smaller this performance metric, the better it is as correct damage status estimation is important for not wasting limited relief resources on the non-damaged grids.
4. **Sum of 1/time weighted total priority of assessed damaged grids overall assessment intervals (e_4):** The higher this performance metric, the better it is as assessing damaged grids with higher priorities earlier will favor both our objective function and immediate allocation of emergency resources.

We analyze the impact of the proposed imputation policies, IP1, IP2¹, IP2², IP2³ and IP3³, on the performance metrics e_1 to e_4 over 20 instances. We set the number of grids to 25, the number of drones to one, and T_{max} to 90 minutes such that we can observe the

effect of imputation policies effectively as the IDARP continues over several assessment intervals. We generate five networks by using the same instance generation approaches explained in Section 5.2.1. We vary the targeted coverage ratio as 0.25 and 0.5, whereas, we vary the damage ratio threshold as 0.5 and 1. The damage levels of grids are generated via Bernoulli random variables as explained in Section 5.2.1. We set the damage score threshold to 1, which is considered while applying IP2 and IP3. Additionally, every drone routing problem has one-hour computational time limit while solving the IDARP.

We again use k -modes algorithm to cluster the grids while solving the IDARP utilizing IP1 and IP2 imputation policies. An additional numerical attribute is added in IP3 imputation policy, hence, we use k -prototypes algorithm for re-clustering. The k -prototypes algorithm uses a parameter (γ) to avoid favoring either the categorical or the numerical attributes (45). We set this parameter to 0.05 which allows obtaining different clusters than using the k -modes algorithm. The number of clusters is determined based on the elbow method.

To observe the effect of the targeted coverage ratio and the imputation policies on the proposed performance metrics, we first solve the proposed MILP model excluding both the targeted coverage ratio and the imputation policies. In other words, we delete the maximum unfilled targeted coverage ratio penalty from the objective function and ignore constraints 21, 22, and 23 from the proposed MILP model introduced in Section 3.2.1.2. Additionally, we do not apply any imputation policy (none of the grids are predicted as likely damaged) during the planning horizon. We call this problem P0 and report its performance in addition to the performance of the imputation policies in Table 7.

Table 7: Performance metrics e_1 - e_4 for P0 and all imputation policies over 20 instances for varying the targeted coverage ratio μ and the damage ratio threshold (th.)

Instances			Imputation Policies																						
			P0			IP1				IP2 ¹				IP2 ²				IP2 ³				IP3 ³			
Net.	μ	th.	e_1	e_2	e_4	e_1	e_2	$e_3(\%)$	e_4	e_1	e_2	$e_3(\%)$	e_4	e_1	e_2	$e_3(\%)$	e_4	e_1	e_2	$e_3(\%)$	e_4	e_1	e_2	$e_3(\%)$	e_4
1	0.25	0.5	7	73	12.0	3	16	53.8	16.1	3	4	46.7	20.5	3	12	53.8	16.8	3	16	53.8	16.3	4	19	58.3	16.1
	0.25	1				4	19	58.3	12.7	3	4	46.7	20.5	3	12	53.8	16.8	3	16	53.8	16.3	4	19	58.3	16.1
	0.5	0.5				6	60	75.0	16.3	3	9	50.0	18.5	5	34	37.5	13.3	5	34	37.5	13.3	4	25	45.5	13.7
	0.5	1				6	60	75.0	12.7	3	9	50.0	18.5	5	34	37.5	13.3	5	34	37.5	13.3	4	25	45.5	13.7
2	0.25	0.5	6	62	10.7	4	29	37.5	10.6	3	6	38.5	18.8	3	15	50.0	13.7	4	29	37.5	12.0	3	23	54.5	12.3
	0.25	1				4	29	37.5	9.8	3	6	38.5	18.8	3	15	50.0	13.7	4	29	37.5	12.0	3	23	54.5	12.3
	0.5	0.5				6	61	0.0	10.6	3	2	46.7	20.8	3	18	45.5	14.0	5	36	33.3	12.5	5	41	42.9	12.1
	0.5	1				6	61	0.0	9.8	3	2	46.7	20.8	4	18	40.0	14.0	5	36	33.3	12.5	5	41	42.9	12.1
3	0.25	0.5	7	76	11.8	5	31	14.3	11.5	4	23	40.0	15.7	5	22	12.5	15.0	5	22	12.5	15.0	4	34	22.2	14.5
	0.25	1				5	39	20.0	10.9	4	28	44.4	15.3	5	22	12.5	15.0	5	22	12.5	15.0	4	34	22.2	14.5
	0.5	0.5				6	89	0.0	11.7	4	23	44.4	16.3	5	39	14.3	12.3	5	38	28.6	12.2	5	58	16.7	11.7
	0.5	1				6	89	0.0	10.9	4	23	44.4	16.3	5	39	14.3	12.3	5	38	28.6	12.2	5	58	16.7	11.7
4	0.25	0.5	7	76	10.7	4	20	22.2	13.7	3	4	25.0	19.8	4	20	22.2	12.8	4	20	22.2	12.8	4	20	22.2	12.8
	0.25	1				4	20	22.2	11.5	3	4	25.0	19.8	4	20	22.2	12.8	4	20	22.2	12.8	4	20	22.2	12.8
	0.5	0.5				6	66	0.0	13.7	3	4	25.0	19.8	4	20	22.2	12.8	5	32	14.3	11.5	5	38	16.7	10.9
	0.5	1				6	66	0.0	11.5	3	4	25.0	19.8	4	20	22.2	12.8	5	32	14.3	11.5	5	38	16.7	10.9
5	0.25	0.5	6	67	11.2	4	22	44.4	16.9	3	9	41.7	19.3	3	9	41.7	19.3	4	22	44.4	18.5	4	22	44.4	18.5
	0.25	1				4	22	44.4	11.2	3	9	41.7	19.3	3	9	41.7	19.3	4	22	44.4	18.5	4	22	44.4	18.5
	0.5	0.5				5	36	50.0	16.9	3	10	41.7	18.5	3	11	50.0	14.3	4	26	50.0	12.8	4	32	62.5	12.6
	0.5	1				5	36	50.0	11.2	4	15	40.0	18.2	3	11	50.0	14.3	4	26	50.0	12.8	4	32	62.5	12.6

First, we summarize our findings based on increasing the *damage ratio threshold* (th.) and the *targeted coverage ratio* (μ). Increasing only the damage ratio threshold while keeping μ the same may increase e_1 , the total number of assessment intervals till all grids are assessed, as expected since increasing this threshold may prevent estimating grids because a fewer number of clusters can satisfy the increased damage ratio threshold. Thus, we may need more assessment intervals to finish assessing all grids. Similarly, increasing only the damage ratio threshold may increase e_2 , the sum of time-weighted total number of non-assessed damaged grids overall assessment intervals since fewer grids may be estimated with the increasing threshold. However, e_3 , the percentage of non-damaged grids that are estimated as likely damaged, may decrease if the grids that are estimated previously, cannot be estimated now due to the increased threshold, are actually not damaged. For example, the grids that cannot be estimated due to increasing the damage ratio threshold are actually not damaged for Network 2 when $\mu = 0.5$ when IP2² is applied. On the contrary, estimating a fewer number of grids as likely damaged may also increase e_3 if grids that cannot be estimated due to the increasing threshold are actually damaged grids, such as Network 1 when $\mu = 0.5$ when IP1 is applied. Lastly, e_4 the sum of 1/time weighted total priority of assessed damaged grids overall assessment intervals may be decreased by increasing the damage ratio threshold as fewer number of grids may be assessed and this may delay the assessment of actually damaged grids.

As we increase the targeted coverage ratio μ while keeping the damage ratio threshold the same, the total number of assessment intervals till all grids are assessed, e_1 , may increase, whereas, the sum of time-weighted total number of non-assessed damaged grids overall assessment intervals, e_2 , usually increases except Network 2 in Table 7. In other words, we usually delay assessing the actual damaged ones since few grid estimations can be done due to a few clusters satisfying the increased targeted coverage criterion. e_3 , the percentage of non-damaged grids that are estimated as damaged, may increase or decrease due to increasing μ as the resulting decrease(s) in likely damaged grid estimations are

whether for actually damaged or non-damaged grids. For example, e_3 decreases for Network 2 when $threshold=0.5$ and IP1 is applied, whereas, e_3 increases for Network 1 when $threshold=0.5$ and IP1 is applied. Similarly, e_4 , the sum of 1/time weighted total priority of assessed damaged grids overall assessment intervals may increase or decrease or stay the same while increasing μ . The main reason is to scan different sets of grids while satisfying μ and this directly affects e_4 . For instance, when the damage ratio threshold equals 0.5 for Network 1, e_4 values increase and decrease when IP1 and IP2² are applied, respectively as μ increases to 0.5 from 0.25 as represented in Table 7.

All of the proposed IPs perform better than the results of P0 that ignores both the targeted coverage ratio and imputation policies. IP2¹ performs better than the other IPs for 19 instances out of 20 (except one instance belonging to Network 5) with respect to e_1 , the total number of assessment intervals till all grids are assessed. Note that IP2 also estimate the likely damaged grids based on the cumulative damage score of grids in addition to the criteria utilized for IP1 and this usually increases the number of estimated grids. Even if the efficiency seems better (i.e., due to fewer assessment intervals) when a policy is applied, that imputation policy cannot guarantee the effectiveness as lots of grids can be estimated as likely damaged when they are not actually damaged. Hence, to facilitate not only quick but also effective response, we need to evaluate the imputation policies based on the remaining performance metrics e_2 - e_4 .

IP2¹ performs better than the other imputation policies for 17 instances out of 20 instances with respect to e_2 , the sum of time-weighted total number of non-assessed damaged grids overall assessment intervals. For the remaining 3 instances (2 instances belonging to Network 3 and one instance belonging to Network 5 as in Table 7), IP2² outperforms IP2¹. IP1 has higher e_2 scores than the other policies. Note that the lower the e_2 score, the better it is as it means that the damaged grids are earlier. Thus, the additional grid-based estimation criterion, based on the cumulative damage score, utilized in IP2 improves the effectiveness of the policy. In other words, grids may be assigned to different clusters throughout the

planning horizon and all of the previous intervals' information is important while predicting the likely damaged ones.

Table 7 does not include e_3 performance metric for P0 model, where e_3 represents the percentage of non-damaged grids that are estimated as damaged, since we do not estimate any likely damaged grids via P0 model. IP1 performs the best with respect to e_3 as the IP1's e_3 values of 10 out of 20 instances are the lowest as represented in Table 7. IP1 is a more conservative policy as it ignores the grid-based estimation criterion based on the cumulative damage scores. This may result in fewer likely damaged grid estimations and lower misestimation percentages than the other policies. The next best performing imputation policy is IP2² with respect to e_3 . Note that all imputation policies result in the estimation of the likely damaged grids for all instances as represented in Table 8. There are instances whose e_3 values are 0 % in Table 7 but that does not mean there are no grids estimated as likely damaged. It means that all grids that are estimated likely damaged are actually damaged.

IP2¹ performs better than P0 and the other imputation policies with respect to e_4 , the sum of 1/time weighted total priority of assessed damaged grids overall assessment intervals as shown in Table 7. This means that IP2¹ assesses damaged grids with high priorities earlier than the other policies. Additionally, IP1 sometimes outperforms IP2² or vice versa with respect to e_4 . For instance, when both μ and the damage ratio threshold equal 0.5 for Network 1, e_4 values of IP1 and IP2² are 16.3 and 13.3, respectively, as well as when μ equals 0.25 and the *threshold* equals 1 for Network 1, e_4 values of IP1 and IP2² are 12.7 and 16.8, respectively. Another important observation is the dominance of IP2¹ over IP2² and IP2³ policies while IP2² dominates IP2³ with respect to e_4 . Hence, delaying the application of the grid-based criterion to 2nd or 3rd assessment intervals also delays the assessment of damaged grids. Note that e_4 may seem like a complementary performance metric to e_2 , however, the priority of the grids, reflecting the population density or critical facilities (such as, hospitals) located in grids, are also considered via e_4 prioritizing the

Table 8: The number of grids that are estimated likely damaged by the imputation policies over 20 instances for varying the targeted coverage ratio μ and the damage ratio threshold (th.).

Instances			Imputation Policies				
Net.	μ	th.	IP1	IP2 ¹	IP2 ²	IP2 ³	IP3 ³
1	0.25	0.5	13	15	13	13	12
	0.25	1	12	15	13	13	12
	0.5	0.5	4	12	8	8	11
	0.5	1	4	12	8	8	11
2	0.25	0.5	8	13	12	8	11
	0.25	1	8	13	12	8	11
	0.5	0.5	1	15	11	6	7
	0.5	1	1	15	10	6	7
3	0.25	0.5	7	10	8	8	9
	0.25	1	5	9	8	8	9
	0.5	0.5	2	9	7	7	6
	0.5	1	2	9	7	7	6
4	0.25	0.5	9	12	9	9	9
	0.25	1	9	12	9	9	9
	0.5	0.5	1	12	9	7	6
	0.5	1	1	12	9	7	6
5	0.25	0.5	9	12	12	9	9
	0.25	1	9	12	12	9	9
	0.5	0.5	4	12	14	8	8
	0.5	1	4	10	14	8	8

assessment of critical grids with higher priority scores.

The selection of the imputation policy should rely on the available SAR resources that can be directed to assessed damaged grids rapidly. Hence, wasting any resources on non-damaged grids estimated as likely damaged may result in delayed responses for the grids that really need rapid responses. Thus, all performance metrics $e_1 - e_4$ are vital in evaluating the performance of the policies. First, all imputation policies result in better performance metrics over the P0 model. Secondly, IP2¹ usually performs better than the other IPs with respect to the performance metrics $e_1 - e_4$. This result shows us that not only the current assessment interval's information is important but also the information collected over the previous assessment intervals is important while estimating the likely damaged grids. Lastly, even if IP3³ less frequently performs better than the other IPs, there is room for further

improvement, such as computing the weight of the new attribute while re-clustering.



CHAPTER VI

CONCLUSION

In this study, we consider a post-disaster setting, in which the drones assess the physical damages on the built infrastructure caused by a disaster. The disaster-affected area is firstly divided into grids where each grid has a set of attributes, such as geographical and geological conditions. We define the IDARP to assess the damage level of the grids by a limited number of drones, such that the drones collect damage information by scanning grids, and the damage level of the unscanned grids are estimated based on the collected information. To estimate the damage status of the unscanned grids, we utilize the grids attributes. That is, we assume that the damage level of a grid is correlated with its attributes, hence, the grids are assigned into clusters by using their attributes, as it is expected to observe similar damage levels in a cluster. We propose a two-phase approach aiming to assess the damage information rapidly via IDARP, such that the phases iterate until the damage status of all grids are assessed.

In the first phase, we address the problem of determining the grids to scan and assess by each drone and the sequence of visits to the selected grids given a set of drones and a limited time for assessments. We aim to maximize the total priority score collected from the assessed grids while a minimum coverage ratio for each cluster is targeted. We propose an arc-based and a path-based formulation and analyze 54 instances varying the way of assigning attributes to grids, as well as the number of drones and grids. We conclude that both formulations perform similarly in terms of the solution quality and the computation time; however, the arc-based formulation requires dummy RSs, which is an obstacle since computing the adequate number of RSs is nontrivial. Additionally, we present the effect of the dominance rule that is applied while enumerating all paths for the path-based

formulation, and we conclude that the path-based formulation with the dominance rule outperforms the other MILP alternatives. Therefore, we use the path-based formulation by applying the dominance rule for our remaining analyses. Nevertheless, the preferred MILP formulation also cannot cope with the large-sized instances. Thus, we develop a VND-based metaheuristic and compare it with the path-based formulation with the dominance rule concerning the solution quality and the computation time. We solve 48 instances varying the number of grids, number of drones, the length of an assessment interval, and the targeted coverage ratio. Our numerical study demonstrates that the metaheuristic finds high-quality solutions rapidly and outperforms the MILP formulation for most of the large-size instances based on the objective function values and computation times.

In the second phase, we predict the damage information of the unscanned grids by utilizing the partial information gathered via the scanned grids. We propose three main imputation policies (five versions in total) utilizing the cluster-based information obtained from the scanned grids to estimate the likely damaged grids. We present several performance metrics to evaluate the imputation policies concerning the efficiency and effectiveness of the assessment operations. Even if the imputation policies sometimes predict some non-damaged grids as likely damaged ones, it is still worth to utilize them as effectiveness relying on early detection of damaged grids (even when the grid priorities are considered) is much higher than not utilizing any imputation policies. Hence, we conclude that imputation policies improve the rapid detection of the damaged grids. Identifying damaged areas is very critical as the SAR resources can be directed accordingly.

Drones may play an important role in humanitarian operations by speeding up the damage assessment operations and hence supporting relief operations. There exists scarce literature that explores using drones in humanitarian logistics. Therefore, there may be several future research directions. For instance, one can extend the proposed formulations for post-disaster damage assessment by incorporating additional aspects of drones such as different speed options which may be affected by weather conditions. Another possible extension is

to consider the availability of the RS on the arrival of the drone; that is, battery charging processes can be examined in more detail. By considering the trade-off between the computation time and the solution quality, the proposed heuristic can be improved by combining different metaheuristics as hybrid heuristics result in high-quality solutions. Additionally, IP3 can be improved by making a sensitivity analysis on the weight of the new attribute. Likewise, the impact of the grid attributes can be analyzed by using the collected information and different weights can be considered for all attributes during the clustering step of the imputation policies in the later assessment intervals. Finally, different imputation techniques can be utilized to predict the damage status in future research. For instance, the performances of the simple methods, such as mean-and-mode method, and the more complex ones can be evaluated under the frequent information updates.

Appendix A. Nomenclature for the MILP formulations

Table 9: Sets, parameters and decisions variables for the arc-based formulation

Sets
0 and $N + 1$: AOCC, for departure and arrival, respectively
I : Set of grids to be scanned
C : Set of clusters
V'_R : Set of dummy recharging stations
$V_0 = I \cup V'_R \cup \{0\}$: Set of departure vertices
$V_1 = I \cup V'_R \cup \{N + 1\}$: Set of arrival vertices
$V' = I \cup V'_R$: Set of vertices except the AOCC
Parameters
s_j : Scanning time of grid $j \in I$
t_{ij} : Travelling time between grid $i \in V_0$ and grid $j \in V_1$
B : Battery capacity
r : Recharging rate
T_{max} : Route duration limit
D : Number of drones
ρ_j : Priority score of grid $j \in I$
ϕ_{jc} : A coverage parameter equals to 1 if grid $j \in I$ is in cluster $c \in C$; 0 otherwise
τ_c : Total number of grids in cluster $c \in C$
μ : Targeted coverage ratio
M : Sufficiently large number
Decision Variables
x_{ij} : 1 if a drone travels on arc $(i, j) \in A$; 0 otherwise
a_{ij} : Departure time, when a drone departs from vertex $i \in V_0$ to travel on arc $(i, j) \in A$
y_{ij} : Remaining battery, when a drone departs from vertex $i \in V_0$ to travel on arc $(i, j) \in A$
q_j : Remaining battery when a drone arrives at RS $j \in V'_R$
z_j : 1 if grid $j \in I$ is scanned; 0 otherwise
β_c : Coverage ratio of cluster $c \in C$
γ : Minimum coverage ratio achieved overall clusters
Δ : Difference between the targeted coverage ratio and the minimum coverage ratio across clusters

Table 10: Sets, parameters and decisions variables for the path-based formulation

Sets
0 and $N + 1$: AOCC, for departure and arrival, respectively
I : Set of grids to be scanned
C : Set of clusters
V_R : Set of RS nodes
$\hat{V} = I \cup \{0\} \cup \{N + 1\}$: Set of all vertices
$\hat{V}_0 = I \cup \{0\}$: Set of departure vertices
$\hat{V}_1 = I \cup \{N + 1\}$: Set of arrival vertices
P : Set of all paths
P_{ij} : Set of paths from i to j , $(i, j) \in \hat{A}$
L_p : The ordered set of RSs in path $p \in P$
Parameters
s_j : Scanning time of grid $j \in I$
t_p : Traveling time associated with path $p \in P$
B : Battery capacity
r : Recharging rate
$\pi_p(l)$: The RS at position $l \in L_p$
T_{max} : Route duration limit
D : Number of drones
ρ_j : Priority score of grid $j \in I$
ϕ_{jc} : A coverage parameter equals to 1 if grid $j \in I$ is in cluster $c \in C$; 0 otherwise
τ_c : Total number of grids in cluster $c \in C$
μ : Targeted coverage ratio
M : Sufficiently large number
Decision Variables
x_p : 1 if a drone travels on path $p \in P$; 0 otherwise
a_p : Time before the drone starts travelling on path $p \in P$
y_p : Remaining battery level before the drone starts travelling on path $p \in P$
z_j : 1 if grid $j \in I$ is scanned; 0 otherwise
q_{pl} : Remaining battery before the charging operation starts on path $p \in P$ at the l^{th} RS
β_c : Coverage ratio of cluster $c \in C$
γ : Minimum coverage ratio
Δ : Difference between the targeted coverage ratio and the minimum coverage ratio across clusters

Appendix B. Effect of dominance rule

Table 11: Computational results of the path-based formulations with/without dominance rule for 2 networks (Net.) having 15 grids varying number of drones (D), T_{max} , and the coverage target (μ), where $\bar{\gamma}$ denotes the minimum coverage ratio across all clusters, $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

Instances				Without dominance rule				With dominance rule			
Net.	D	T_{max}	μ	$\bar{\gamma}$	obj_P	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_{Pd}	$time$ (sec)	gap (%)
1	1	90	0.25	0.25	7.5	137.5	0.0	0.25	7.5	3.3	0.0
1	2	90	0.25	0.33	15	3600	8.0	0.33	15	576.0	0.0
1	1	180	0.25	0.33	16	16.1	0.0	0.33	16	1.8	0.0
1	2	180	0.25	1.00	22	28.5	0.0	1.00	22	1.5	0.0
1	1	90	0.50	0.25	$7.5-\bar{\Delta}$	80.6	0.0	0.25	$7.5-\bar{\Delta}$	16.8	0.0
1	2	90	0.50	0.50	15	3600	6.3	0.50	15	18.7	0.0
1	1	180	0.50	0.50	16	16.1	0.0	0.50	16	3.0	0.0
1	2	180	0.50	1.00	22	7.8	0.0	1.00	22	1.5	0.0
2	1	90	0.25	0.25	8	20.2	0.0	0.25	8	1.2	0.0
2	2	90	0.25	0.40	15	3600	7.2	0.40	15	45.4	0.0
2	1	180	0.25	0.40	16	137.7	0.0	0.40	16	1.1	0.0
2	2	180	0.25	1.00	21.5	34.4	0.0	1.00	21.5	2.1	0.0
2	1	90	0.50	0.25	$8-\bar{\Delta}$	80.7	0.0	0.25	$8-\bar{\Delta}$	3.5	0.0
2	2	90	0.50	0.50	15	1444.5	0.0	0.50	15	17.2	0.0
2	1	180	0.50	0.50	15.5	83.4	0.0	0.50	15.5	2.9	0.0
2	2	180	0.50	1.00	21.5	14.5	0.0	1.00	21.5	2.1	0.0
					Max	3600	8.0		Max	576.0	0.0
					Avg	806.4	1.0		Avg	43.6	0.0
					Min	7.8	0.0		Min	1.1	0.0

Table 12: Computational results of the path-based formulations with/without dominance rule for 2 networks (Net.) having 25 grids varying number of drones (D), T_{max} , and the coverage target (μ), where $\bar{\gamma}$ denotes the minimum coverage ratio across all clusters, $\bar{\Delta}$ represents the maximum unfulfilled targeted coverage ratio across the clusters, Δ , computed via the corresponding method.

Instances				Without dominance rule				With dominance rule			
Net.	D	T_{max}	μ	$\bar{\gamma}$	obj_P	$time$ (sec)	gap (%)	$\bar{\gamma}$	obj_{Pd}	$time$ (sec)	gap (%)
1	1	90	0.25	0.00	10- $\bar{\Delta}$	3600	48.5	0.00	10.5- $\bar{\Delta}$	414.8	0.0
1	2	90	0.25	0.28	19.5	3600	8.7	0.28	19.5	128.5	0.0
1	1	180	0.25	0.25	20.5	3086.2	0.0	0.25	20.5	87.7	0.0
1	2	180	0.25	0.50	33	3600	8.7	0.50	34.5	3600	3.1
1	1	90	0.50	0.00	10.5- $\bar{\Delta}$	3600	26.1	0.00	10.5- $\bar{\Delta}$	387.9	0.0
1	2	90	0.50	0.28	15- $\bar{\Delta}$	3600	23.2	0.28	19.5- $\bar{\Delta}$	3600	22.8
1	1	180	0.50	0.33	18.5- $\bar{\Delta}$	3442.6	0.0	0.33	18.5- $\bar{\Delta}$	59.0	0.0
1	2	180	0.50	0.50	34.5	1780.1	3.9	0.50	34.5	3600	3.1
2	1	90	0.25	0.00	10- $\bar{\Delta}$	3600	72.4	0.00	10- $\bar{\Delta}$	509.0	0.0
2	2	90	0.25	0.25	17	3600	12.0	0.25	18	409.7	0.0
2	1	180	0.25	0.25	19	964	0.0	0.25	19	18.6	0.0
2	2	180	0.25	0.33	34.5	3600	6.4	0.33	34.5	3600	4.3
2	1	90	0.50	0.00	10- $\bar{\Delta}$	3600	36.2	0.00	10- $\bar{\Delta}$	469.3	0.0
2	2	90	0.50	0.33	16- $\bar{\Delta}$	3600	0.0	0.33	18- $\bar{\Delta}$	61.2	0.0
2	1	180	0.50	0.33	18.5- $\bar{\Delta}$	1103.2	0.0	0.33	18.5- $\bar{\Delta}$	74.7	0.0
2	2	180	0.50	0.50	33.5	3600	8.3	0.50	34	3600	5.2
					Max	3600	72.4		Max	3600	22.8
					Avg	3123.5	15.9		Avg	1288.8	2.4
					Min	964.0	0.0		Min	18.6	0.0

Appendix C. The steps of the clustering algorithms

Algorithm 2: k -modes and k -prototypes algorithms (45)

Input: Objects to be clustered, object attributes

Step 0. Initialization;

k := Number of clusters

f := Cost function (i.e., dissimilarity measure)

Step 1. For each cluster, select k initial modes.

Until All objects are allocated to a cluster **Do** Step 2.

Step 2. Allocate an object to the nearest cluster according to the cluster's mode and f . After each allocation, update the mode of the cluster.

Until No relocation is made for any object **Do** Step 3.

Step 3. Check the dissimilarity of the objects against the modes of the clusters that are currently assigned. If there exists another cluster such that the object's nearest mode belongs to that one, relocate the object to that cluster. Update the modes of both clusters after the relocation.

Output: Clusters

Appendix D. The pseudocode for the VND-based heuristic

Algorithm 3: CH: Route Construction

```

Initialization;
 $S_+ :=$  Set of scanned grids,  $S_0 :=$  Set of candidate grids to be scanned
 $\tau_k :=$  Route duration of drone  $k$ ,  $T_{max} :=$  Route duration limit
while  $S_0 \neq \emptyset$  and  $\exists \tau_{k'} \leq T_{max}$  do
    if  $S_+ == \emptyset$  or  $\tau_k == T_{max}$  and  $\exists \tau_{k'} == 0$  then
        %Start a new route
         $k :=$  an unused drone;
         $z :=$  candidate grid  $\in S_0$ ;
         $\beta_c :=$  coverage ratio of cluster  $c$  that  $z \in c$ ;
        if insertion of  $z$  to the route of  $k$  is feasible then
            Add  $z$  to the route of  $k$ ;
            Update  $S_+$ ,  $S_0$ ,  $\tau_k$ ,  $\beta_c$ ;
            if  $\beta_c \geq$  the coverage target then
                Remove all grids  $\in c$  from  $S_0$ ;
            end
        else
            if scanning an updated  $z$  is feasible after recharging then
                Visit the nearest RS, update  $z$ ;
                Add  $z$  to the route of  $k$ ;
                Update  $S_+$ ,  $S_0$ ,  $\tau_k$ ,  $\beta_c$ ;
                if  $\beta_c \geq$  the coverage target then
                    Remove all grids  $\in c$  from  $S_0$ ;
                end
            else
                 $\tau_k = T_{max}$  % Return to the AOCC;
            end
        end
    else
        while  $\tau_k \leq T_{max}$  do
            %Add to the current route
             $z :=$  candidate grid  $\in S_0$ ;
             $\beta_c :=$  coverage ratio of cluster  $c$  that  $z \in c$ ;
            if insertion of  $z$  to the route of  $k$  is feasible then
                Add  $z$  to the route of  $k$ ;
                Update  $S_+$ ,  $S_0$ ,  $\tau_k$ ,  $\beta_c$ ;
                if  $\beta_c \geq$  the coverage target then
                    Remove all grids  $\in c$  from  $S_0$ ;
                end
            else
                if scanning an updated  $z$  is feasible after recharging then
                    Visit the nearest RS, update  $z$ ;
                    Add  $z$  to the route of  $k$ ;
                    Update  $S_+$ ,  $S_0$ ,  $\tau_k$ ,  $\beta_c$ ;
                    if  $\beta_c \geq$  the coverage target then
                        Remove all grids  $\in c$  from  $S_0$ ;
                    end
                else
                    % Return to the AOCC  $\tau_k = T_{max}$ ;
                end
            end
        end
    end
end

```

Algorithm 4: Repair Process

```

Initialization;
imp = 0
while  $\exists$  a better feasible solution do
     $\tau$  := total routes' duration of the best solution
    for each route  $r$  in the best solution do
        for  $grid \in r$  do
             $c$  := cluster of  $grid$ 
             $r' = \text{Remove } grid \text{ from } r$ ;
            for  $(grid' \notin r) \in c$  do
                 $x = \text{Add } grid' \text{ to the best position in } r'$ ;
                 $\tau' := \text{total routes' duration of the candidate solution } x$ 
                if  $\tau' < \tau$  and  $x$  is feasible then
                    incumbent solution =  $x$ ;
                     $\tau = \tau'$ ;
                    imp = 1
                end
            end
        end
    end
    the best solution = incumbent solution
end
 $A := \text{Minimum coverage ratio}$ 
if  $imp == 1$  then
    while  $A < \text{coverage target}$  and  $\exists$  a better feasible solution do
         $\tau = T_{max}$ 
         $c := \text{cluster having the minimum ratio}$ 
        for each unvisited  $grid \in c$  do
            Generate three options of  $grid$ ;
            for each route  $r$  in the best solution do
                 $r' = \text{Add the best option to the best position in route } r$ ;
                 $\tau' := \text{route duration of } r'$ ;
                if  $\tau' < \tau$  and  $r'$  is feasible then
                    incumbent solution := Change  $r$  by  $r'$  in the best solution;
                     $\tau = \tau'$ ;
                end
            end
        end
        the best solution = incumbent solution
        Update  $A$ 
    end
end
if  $A < \text{coverage target}$  then
    while  $A < \text{coverage target}$  and  $\exists$  a better feasible solution do
         $c := \text{cluster having the minimum ratio}$ 
         $c' := \text{cluster having the maximum ratio}$ 
        for each visited  $grid \in c'$  in the solution do
             $solution' = \text{Remove } grid$ ;
             $solution' = \text{Add an unvisited } (grid' \in c) \text{ to the best position}$ ;
            if  $solution'$  is feasible then
                Update  $A$ ;
                 $solution = solution'$ ;
                break
            end
        end
    end
end
end

```

Algorithm 5: Local Search

```
Initialization;
x:= The initial solution
 $N_k$ := Set of neighborhood structures for  $k = 1, \dots, k_{max}$ 
 $max_{iter}$ := Set the maximum iteration number
 $k = 1$ 
 $iter = 0$ 
while  $iter < max_{iter}$  do
    Find the best neighbor  $x'$  of  $x$  in  $N_k$ ;
    if  $x'$  is better than  $x$  then
        |  $x = x'$ ;
        |  $k = 1$ ;
    else
        |  $k = k + 1$ ;
    end
    if  $k == k_{max}$  then
        Do diversification;
        if the targeted coverage ratio is not achieved then
            | Do repair process
        end
         $k = 1$ ;
    end
     $iter = iter + 1$ ;
end
```

Bibliography

- [1] A. W. Coburn, R. J. Spence, and A. Pomonis, “Factors determining human casualty levels in earthquakes: Mortality prediction in building collapse,” in *Proceedings of the Tenth World Conference on Earthquake Engineering*, vol. 10, pp. 5989–5994, Balkema Rotterdam, 1992.
- [2] J. Qi, D. Song, H. Shang, N. Wang, C. Hua, C. Wu, X. Qi, and J. Han, “Search and rescue rotary-wing UAV and its application to the Lushan MS 7.0 earthquake,” *Journal of Field Robotics*, vol. 33, no. 3, pp. 290–321, 2016.
- [3] J. O. Arumala, “Impact of large-scale disasters on the built environment,” *Leadership and management in engineering*, vol. 12, no. 3, pp. 147–150, 2012.
- [4] J.-P. Pinelli, E. Simiu, K. Gurley, C. Subramanian, L. Zhang, A. Cope, J. J. Filliben, and S. Hamid, “Hurricane damage prediction model for residential structures,” *Journal of Structural Engineering*, vol. 130, no. 11, pp. 1685–1691, 2004.
- [5] TNH, “The New Humanitarian (TNH): Rescue people trapped in a collapsed building,” 2009. (Accessed 29 July 2020).
- [6] BBC, “Nepal earthquake: How does the search and rescue operation work?,” 2015. (Accessed 29 July 2020).
- [7] M. Erdelj and E. Natalizio, “UAV-assisted disaster management: Applications and open issues,” in *2016 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1–5, IEEE, 2016.
- [8] ALTI, “Search And Rescue Drones: An Irreplaceable Tool For The SAR Industry,” n.d. (Accessed 29 July 2020).
- [9] M. Półka, S. Ptak, and Ł. Kuziora, “The use of UAV’s for search and rescue operations,” *Procedia Engineering*, vol. 192, pp. 748–752, 2017.
- [10] UAVIATORS, “Uaviators: Humanitarian UAV network,” n.d. (Accessed 14 August 2020).
- [11] J. Chen, H. Liu, J. Zheng, M. Lv, B. Yan, X. Hu, and Y. Gao, “Damage degree evaluation of earthquake area using UAV aerial image,” *International Journal of Aerospace Engineering*, vol. 2016, 2016.
- [12] N. Kerle, F. Nex, M. Gerke, D. Duarte, and A. Vetrivel, “UAV-based structural damage mapping: A review,” *ISPRS International Journal of Geo-information*, vol. 9, no. 1, p. 14, 2020.
- [13] S. Adams, C. Friedland, and M. Levitan, “Unmanned aerial vehicle data acquisition for damage assessment in hurricane events,” in *Proceedings of the 8th International Workshop on Remote Sensing for Disaster Management, Tokyo, Japan*, vol. 30, 2010.

- [14] D. Soesilo, P. Meier, A. Lessard-Fontaine, J. Du Plessis, C. Stuhlberger, and V. Fabbroni, "Drones in Humanitarian Action: A guide to the use of airborne systems in humanitarian crises," *FSD Report*, pp. 1–60, 2016.
- [15] M. Kakooei and Y. Baleghi, "Fusion of satellite, aircraft, and UAV data for automatic disaster damage assessment," *International Journal of Remote Sensing*, vol. 38, no. 8–10, pp. 2511–2534, 2017.
- [16] K. Y. Sokat, I. S. Dolinskaya, K. Smilowitz, and R. Bank, "Incomplete information imputation in limited data environments with application to disaster response," *European Journal of Operational Research*, vol. 269, no. 2, pp. 466–485, 2018.
- [17] D. T. Larose and C. D. Larose, *Discovering knowledge in data: An introduction to data mining*, vol. 4. John Wiley & Sons, 2014.
- [18] A. Froger, J. E. Mendoza, O. Jabali, and G. Laporte, "Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions," *Computers & Operations Research*, vol. 104, pp. 256–294, 2019.
- [19] M. A. R. Estrada and A. Ndoma, "The uses of unmanned aerial vehicles–UAV’s-(or drones) in social logistic: Natural disasters response and humanitarian relief aid," *Procedia Computer Science*, vol. 149, pp. 375–383, 2019.
- [20] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.
- [21] Y. Sato, S. Ozawa, Y. Terasaka, M. Kaburagi, Y. Tanifuji, K. Kawabata, H. N. Miyamura, R. Izumi, T. Suzuki, and T. Torii, "Remote radiation imaging system using a compact gamma-ray imager mounted on a multicopter drone," *Journal of Nuclear Science and Technology*, vol. 55, no. 1, pp. 90–96, 2018.
- [22] R. R. Murphy, B. A. Duncan, T. Collins, J. Kendrick, P. Lohman, T. Palmer, and F. Sanborn, "Use of a small unmanned aerial system for the SR-530 mudslide incident near Oso, Washington," *Journal of Field Robotics*, vol. 33, no. 4, pp. 476–488, 2016.
- [23] F. Yamazaki, K. Kubo, R. Tanabe, and W. Liu, "Damage assessment and 3d modeling by UAV flights after the 2016 Kumamoto, Japan earthquake," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 3182–3185, IEEE, 2017.
- [24] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Towards a more efficient detection of earthquake induced facade damages using oblique UAV imagery," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 93, 2017.
- [25] K. Kim, P. Pant, and E. Yamashita, "Disasters, drones, and crowdsourced damage assessment," in *Proceedings of Computers in Urban Planning and Urban Management Conference, Cambridge, Massachusetts*, 2015.

- [26] D. Chauhan, A. Unnikrishnan, and M. Figliozzi, "Maximum coverage capacitated facility location problem with range constrained drones," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 1–18, 2019.
- [27] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S. G. Nurre, and L. Bian, "Drones for disaster response and relief operations: A continuous approximation model," *International Journal of Production Economics*, vol. 188, pp. 167–184, 2017.
- [28] M. Golabi, S. M. Shavarani, and G. Izbirak, "An edge-based stochastic facility location problem in UAV-supported humanitarian relief logistics: A case study of Tehran earthquake," *Natural Hazards*, vol. 87, no. 3, pp. 1545–1565, 2017.
- [29] 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.
- [30] B. Rabta, C. Wankmüller, and G. Reiner, "A drone fleet model for last-mile distribution in disaster relief operations," *International Journal of Disaster Risk Reduction*, vol. 28, pp. 107–112, 2018.
- [31] B. E. Oruc and B. Y. Kara, "Post-disaster assessment routing problem," *Transportation Research Part B: Methodological*, vol. 116, pp. 76–102, 2018.
- [32] S. M. Shavarani, "Multi-level facility location-allocation problem for post-disaster humanitarian relief distribution," *Journal of Humanitarian Logistics and Supply Chain Management*, 2019.
- [33] S. Chowdhury, *Drone routing and optimization for post-disaster inspection*. PhD thesis, Mississippi State University, 2020.
- [34] A. Nedjati, B. Vizvari, and G. Izbirak, "Post-earthquake response by small UAV helicopters," *Natural Hazards*, vol. 80, no. 3, pp. 1669–1688, 2016.
- [35] R. Dubey, A. Gunasekaran, and T. Papadopoulos, "Disaster relief operations: Past, present and future," *Annals of Operations Research*, vol. 283, no. 1-2, pp. 1–8, 2019.
- [36] A. M. Caunhye, X. Nie, and S. Pokharel, "Optimization models in emergency logistics: A literature review," *Socio-economic Planning Sciences*, vol. 46, no. 1, pp. 4–13, 2012.
- [37] G. Galindo and R. Batta, "Review of recent developments in OR/MS research in disaster operations management," *European Journal of Operational Research*, vol. 230, no. 2, pp. 201–211, 2013.
- [38] L. Chen and E. Miller-Hooks, "Optimal team deployment in urban search and rescue," *Transportation Research Part B: Methodological*, vol. 46, no. 8, pp. 984–999, 2012.
- [39] S. Argollo, R. Bandeira, and V. Campos, "Operations research in humanitarian logistics decisions," in *Selected Proceedings of 13th World Conference on Transportation Research*, 2013.

- [40] Y. Fujikawa and T. Ho, "Cluster-based algorithms for dealing with missing values," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 549–554, Springer, 2002.
- [41] P. Schmitt, J. Mandel, and M. Guedj, "A comparison of six methods for missing data imputation," *Journal of Biometrics & Biostatistics*, vol. 6, no. 1, p. 1, 2015.
- [42] A. K. Dwivedi, "Performance evaluation of different machine learning techniques for prediction of heart disease," *Neural Computing and Applications*, vol. 29, no. 10, pp. 685–693, 2018.
- [43] K. Shobha and N. Savarimuthu, "Clustering based imputation algorithm using unsupervised neural network for enhancing the quality of healthcare data," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2020.
- [44] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging Artificial Intelligence Applications in Computer Engineering*, vol. 160, no. 1, pp. 3–24, 2007.
- [45] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
- [46] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Dealing with missing data: Algorithms based on fuzzy set and rough set theories," in *Transactions on Rough Sets IV*, pp. 37–57, Springer, 2005.
- [47] B. M. Patil, R. C. Joshi, and D. Toshniwal, "Missing value imputation based on k-mean clustering with weighted distance," in *International Conference on Contemporary Computing*, pp. 600–609, Springer, 2010.
- [48] A. Purwar and S. K. Singh, "Empirical evaluation of algorithms to impute missing values for financial dataset," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 652–656, IEEE, 2014.
- [49] K. Y. Sokat, R. Zhou, I. S. Dolinskaya, K. Smilowitz, and J. Chan, "Capturing real-time data in disaster response logistics," *Journal of Operations and Supply Chain Management*, vol. 9, no. 1, pp. 23–54, 2016.
- [50] M. Ortuño, P. Cristóbal, J. Ferrer, F. Martín-Campo, S. Muñoz, G. Tirado, and B. Vitoriano, "Decision aid models and systems for humanitarian logistics. A survey," in *Decision Aid Models for Disaster Management and Emergencies*, pp. 17–44, Springer, 2013.
- [51] L. Özdamar and M. A. Ertem, "Models, solutions and enabling technologies in humanitarian logistics," *European Journal of Operational Research*, vol. 244, no. 1, pp. 55–65, 2015.

- [52] B. Balcik, B. M. Beamon, and K. Smilowitz, "Last mile distribution in humanitarian relief," *Journal of Intelligent Transportation Systems*, vol. 12, no. 2, pp. 51–63, 2008.
- [53] C.-C. Lu, K.-C. Ying, and H.-J. Chen, "Real-time relief distribution in the aftermath of disasters—A rolling horizon approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 93, pp. 1–20, 2016.
- [54] K. Huang, Y. Jiang, Y. Yuan, and L. Zhao, "Modeling multiple humanitarian objectives in emergency response to large-scale disasters," *Transportation Research Part E: Logistics and Transportation Review*, vol. 75, pp. 1–17, 2015.
- [55] L. Özdamar, E. Ekinçi, and B. Küçükyazıcı, "Emergency logistics planning in natural disasters," *Annals of Operations Research*, vol. 129, no. 1-4, pp. 217–245, 2004.
- [56] W. Yi and L. Özdamar, "A dynamic logistics coordination model for evacuation and support in disaster response activities," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1177–1193, 2007.
- [57] J.-B. Sheu, "Dynamic relief-demand management for emergency logistics operations under large-scale disasters," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 1, pp. 1–17, 2010.
- [58] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.
- [59] Ç. Koç and I. Karaoglan, "The green vehicle routing problem: A heuristic based exact solution approach," *Applied Soft Computing*, vol. 39, pp. 154–164, 2016.
- [60] Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado, "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges," *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, pp. 111–128, 2014.
- [61] J. Andelmin and E. Bartolini, "An exact algorithm for the green vehicle routing problem," *Transportation Science*, vol. 51, no. 4, pp. 1288–1303, 2017.
- [62] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, "The electric vehicle routing problem with nonlinear charging function," *Transportation Research Part B: Methodological*, vol. 103, pp. 87–110, 2017.
- [63] B. Adsanver, E. Coban, and B. Balcik, "Drone routing for post-disaster damage assessment," 2020. Submitted to: Dynamics of Disasters—Key Concepts, Models, Algorithms, and Insights, Springer.
- [64] M. Keskin, G. Laporte, and B. Çatay, "Electric vehicle routing problem with time-dependent waiting times at recharging stations," *Computers & Operations Research*, vol. 107, pp. 77–94, 2019.

- [65] R. A. Davidson and H. C. Shah, *An urban earthquake disaster risk index*. John A. Blume Earthquake Engineering Center Stanford University, 1997.
- [66] I. M. Shohet, H. Wei, R. Levy, S. Shapira, O. Levi, T. Levi, *et al.*, “Analytical-empirical model for the assessment of earthquake casualties and injuries in a major city in Israel—the case of Tiberias,” *Final Report, submitted to the Ministry of Science and Technology, grant*, no. 39618, 2015.
- [67] D. Pamukcu and B. Balcik, “A multi-cover routing problem for planning rapid needs assessment under different information-sharing settings,” *OR Spectrum*, vol. 42, no. 1, pp. 1–42, 2020.
- [68] C. A. F. Ezequiel, M. Cua, N. C. Libatique, G. L. Tangonan, R. Alampay, R. T. Labuguen, C. M. Favila, J. L. E. Honrado, V. Canos, C. Devaney, *et al.*, “UAV aerial imaging applications for post-disaster assessment, environmental management and infrastructure development,” in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 274–283, IEEE, 2014.
- [69] G. B. Durrant *et al.*, “Imputation methods for handling item-nonresponse in the social sciences: A methodological review,” *ESRC National Centre for Research Methods and Southampton Statistical Sciences Research Institute. NCRM Methods Review Papers NCRM/002*, 2005.
- [70] P. Hansen, N. Mladenović, and D. Perez-Britos, “Variable neighborhood decomposition search,” *Journal of Heuristics*, vol. 7, no. 4, pp. 335–350, 2001.
- [71] J. Hof, M. Schneider, and D. Goeke, “Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops,” *Transportation Research Part B: Methodological*, vol. 97, pp. 102–112, 2017.
- [72] S. Zhang, M. Chen, W. Zhang, and X. Zhuang, “Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations,” *Expert Systems with Applications*, vol. 145, p. 113123, 2020.
- [73] F. Glover, “Tabu search—part i,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [74] P. Boccardo, F. Chiabrando, F. Dutto, F. G. Tonolo, and A. Lingua, “UAV deployment exercise for mapping purposes: Evaluation of emergency response applications,” *Sensors*, vol. 15, no. 7, pp. 15717–15737, 2015.
- [75] D. H. Lang, A. Kumar, S. Sulaymanov, and A. Meslem, “Building typology classification and earthquake vulnerability scale of Central and South Asian building stock,” *Journal of Building Engineering*, vol. 15, pp. 261–277, 2018.
- [76] M. Erdik, N. Aydinoglu, Y. Fahjan, K. Sesetyan, M. Demircioglu, B. Siyahi, E. Durukal, C. Ozbey, Y. Biro, H. Akman, *et al.*, “Earthquake risk assessment for Istanbul

- metropolitan area,” *Earthquake Engineering and Engineering Vibration*, vol. 2, no. 1, pp. 1–23, 2003.
- [77] B. Balcik, “Site selection and vehicle routing for post-disaster rapid needs assessment,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 101, pp. 30–58, 2017.
 - [78] P. Bholowalia and A. Kumar, “EBK-means: A clustering technique based on elbow method and k-means in WSN,” *International Journal of Computer Applications*, vol. 105, no. 9, 2014.
 - [79] B. Balcik and B. M. Beamon, “Facility location in humanitarian relief,” *International Journal of Logistics*, vol. 11, no. 2, pp. 101–121, 2008.
 - [80] IFRC, “The International Federation of Red Cross and Red Crescent Societies (IFRC) World Disasters Report 2018: Leaving no one behind,” 2018. (Accessed 14 July 2020).
 - [81] Y. Jiang, Y. Yuan, K. Huang, and L. Zhao, “Logistics for large-scale disaster response: Achievements and challenges,” in *2012 45th Hawaii International Conference on System Sciences*, pp. 1277–1285, IEEE, 2012.
 - [82] V. Leggieri and M. Haouari, “A practical solution approach for the green vehicle routing problem,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 104, pp. 97–112, 2017.
 - [83] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, “Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey,” *Networks*, vol. 72, no. 4, pp. 411–458, 2018.

VITA

Birce Adsanver graduated from Beyoğlu Anatolian High School in 2012. She earned her B.Sc. degree in Industrial Engineering from Özyeğin University in June 2018. She has been pursuing her M.Sc. degree in Industrial Engineering at Özyeğin University since September 2018, under the supervision of Dr. Elvin Çoban and Dr. Burcu Balçık. Her research interests include supply chain management, humanitarian logistics, and healthcare operations.