Drone Routing for Post-disaster Damage Assessment

Birce Adsanver, Elvin Coban, and Burcu Balcik

Abstract We consider drones to support post-disaster damage assessment operations when the disaster-affected area is divided into grids and grids are clustered based on their attributes. Specifically, given a set of drones and a limited time for assessments, we address 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 assessed grids while ensuring that the pre-specified coverage ratio targets for the clusters are met. We adapt formulations from the literature developed for electric vehicle routing problems with recharging stations and propose two alternative mixed-integer linear programming models for our problem. We use an optimization solver to evaluate the computational difficulty of solving different formulations and show that both formulations perform similarly. We also develop a practical constructive heuristic to solve the proposed drone routing problem, which can find high-quality solutions rapidly. We evaluate the performance of the heuristic with respect to both mathematical models in a variety of instances with the different numbers of drones and grids.

Keywords Post-disaster · Drone · Routing · Damage assessment · Constructive heuristic

1 Introduction

In the last 70 years, the number and severity of disasters have risen exponentially [39]. The Emergency Events Database (EM-DAT) reports 3,751 natural disasters that occurred between 2008 and 2017 [24], and the average number of deaths is about 60,000 per year [22]. Effective and efficient disaster response operations are essential to delivering relief supplies at the right places and times to minimize

B. Adsanver (✉) · E. Coban · B. Balcik
Industrial Engineering Department, Ozyegin University, Cekmekoy, Istanbul, Turkey
e-mail: birce.adsanver@ozu.edu.tr; elvin.coban@ozyegin.edu.tr; burcu.balcik@ozyegin.edu.tr

I. S. Kotsireas et al. (eds.), *Dynamics of Disasters*, Springer Optimization and Its Applications 169, https://doi.org/10.1007/978-3-030-64973-9_1

human suffering and death [25]. Moreover, the overall success of disaster operations is highly dependent on the speed of rapid needs and damage assessment phases, during which the extent of disaster's impact on people and infrastructure is evaluated [5].

In this study, we focus on rapid damage assessment operations conducted by using drones. While drones can be used to assess the damages on the built infrastructure in the aftermath of other disasters such as floods or hurricanes, we particularly consider a post-disaster environment after an earthquake. Earthquakes are among the deadliest disasters [22]. For instance, in 2004 and 2010, earthquakes accounted for 93% and 69% of worldwide deaths due to disasters [22]. It is critical to assess the earthquake damage on buildings rapidly since the survival rate for the people who are rescued from the collapsed buildings after an earthquake is 91% during the first 30 min, while it decreases to 81% on the first day and 36% on the second day [40]. Therefore, using available resources (such as drones or other technology) effectively after a disaster to identify the damaged buildings can save many lives.

Areas that are hit by earthquakes can be assessed by a variety of visual imaging and 3D mapping technologies, such as satellite imagery or radars. However, there are limitations to using each technology; for instance, satellite mapping may not meet high-resolution requirements due to clouds blocking the image. Recently, drones are increasingly used to assess earthquake damages [14, 48]. Drones can be deployed immediately after a disaster, and high-resolution images can be generated for the scanned regions, which would help to identify highly damaged areas quickly and direct the rescue teams to the correct spots and hence prioritize the use of relief resources effectively [29, 31]. For instance, in the aftermath of the 2015 Nepal earthquake, drones assisted in creating 3D maps through image processing software to assess the widespread damage and operate search and evacuation operations. Additionally, drones were used to identify the damaged infrastructure after an F-5 tornado in Wichita, Kansas [2]. Although drones have limited battery capacity, their flight duration can be extended by allowing them to recharge at the recharging stations (RSs) that would be positioned in the affected regions. Nevertheless, it may not be possible to assess the entire disaster-affected area rapidly by using a limited number of drones. However, the amount of useful information obtained by a limited number of drones in a restricted time period can be improved by selecting which areas to scan first. For instance, information obtained by drones that scan a part of the affected region can be used to estimate the damage status of the unscanned affected regions if different regions share similar disaster risk attributes that could affect the extent of earthquake damage. This study aims to present mathematical models and a solution approach that would support managing drones to scan a disaster-affected area quickly.

We consider a post-disaster setting, in which the drones assess the physical damage after an earthquake. The disaster-affected area is divided into grids. Each grid has a set of attributes comprising its infrastructure (e.g., construction types of the buildings), geographical (e.g., elevation), geological (e.g., soil type), and socioeconomic (e.g., wealth index) conditions. We assume that the damage level of the built environment in a grid is highly correlated with its attribute values. The attributes can be associated with pre-disaster and post-disaster aspects. For instance, the construction types of the buildings and soil type may be specified before a disaster, while the distance to the earthquake epicenter is a post-disaster attribute [11, 46]. We assume that grids that have similar attributes are more likely to exhibit similar damage levels due to an earthquake [47]. Hence, grids can be clustered based on their similarities according to the relevant set of attributes by using various clustering methods such as *k*-modes clustering [47]. Since assessment of all grids in a limited amount of time may be impossible, scanning two grids belonging to different clusters may be preferred more than scanning two grids from the same cluster as information obtained from scanning a particular grid can be used to estimate the damages in the other grids of the cluster. Moreover, grids can be given different priorities based on whether a grid is densely populated and/or there is an important facility in the grid such as hospitals and schools. We focus on planning the routes of drones by considering grid clusters and priorities.

Given a set of drones and a limited time for assessments, we address the problem of determining the grids to scan and assess by each drone and the sequence of visits to the selected grids to maximize the total priority score collected from the assessed grids. We cluster grids before making routing plans based on various relevant pre- and post-disaster attributes. Then, a minimum level of coverage ratio for each cluster is targeted to achieve adequate and balanced information among all clusters, 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. Nevertheless, it may not be possible to achieve the coverage target within a limited time, which may lead to infeasible solutions. To avoid infeasibilities, we penalize the maximum unfilled coverage ratio in the objective function. Moreover, if the coverage ratio is satisfied for all clusters but there is still time in the given assessment period, additional grids can still be scanned to increase the amount of collected information and hence the total priority score.

To extend drones' flight duration, we consider several RSs already positioned in the region where drones can recharge their batteries. The drone routing problem has some similarities with the electric vehicle routing problem (E-VRP) or green vehicle routing problem (G-VRP) since recharging times and locations of drones must be considered while determining the sequence of visits to the grids. However, the proposed drone routing problem is different due to its objective function and the coverage ratio constraint, which are introduced to capture the characteristics of the post-disaster damage assessment operations. We develop two mixed-integer linear programming (MILP) models, which are adapted from [19], and propose a novel heuristic to find high-quality solutions rapidly to assist decision-makers in making effective and efficient plans for drone routing during the damage assessment phase.

The rest of the paper is organized as follows. Section 2 reviews the relevant literature. Section 3 describes the system and problem in detail and presents our MILP model formulations. Section 4 describes the proposed heuristic, whereas Sect. 5 introduces the data set developed and reports the results of the computational experiments performed. Finally, we conclude in Sect. 6 and discuss future work.

2 Literature Review

Drones are relatively inexpensive and easy to deploy and use for supporting not only military applications but also civilian applications [12]. Since they can carry cameras [42], sensors [8], and mail [13], they are utilized frequently in several sectors such as agriculture, commerce, and construction. Nowadays, they are also used in humanitarian relief operations [15, 17, 33].

Drones were utilized in various past disasters to gather information about post-disaster situations, for example, to measure radiation contamination in the Daiichi nuclear power plant [43], capture the images of damaged reactors after the Japan East great earthquake in 2011 [33], monitor the affected area after the 2011 Thailand floods [34], and assess the damages after the 2016 Kumamoto earthquake [48], the 2009 L'Aquila earthquake [14], and Typhoon Haiyan in the Philippines [27]. Thus, integrating drones into the existing post-disaster operations may accelerate the information-gathering process and help rescuers to make timely and efficient decisions.

Integrating drones at the applications of commercial logistics is well studied in the literature [1, 21, 35, 36]; however, there are relatively few studies that focus on humanitarian applications. In particular, the studies that consider using drones in humanitarian settings address finding the location of relief facilities [7, 10, 20], distributing relief or medical items [26, 41], and post-disaster monitoring [9, 38].

An edge-based facility location problem where drones deliver relief items to aid recipients is studied in [20]. In this work, 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, in [10], the authors determine 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 [7] considering drone energy consumption and drone flight range limitations. A location-allocation problem for positioning recharging stations is addressed by [45]. In [41], 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. A MILP and metaheuristics are proposed to scan the post-disaster-affected areas to minimize the total assessment costs in [9]. In [38], the authors study 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 are restricted by the 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.

In [37], a post-earthquake relief distribution system is proposed where drones are used to serve areas that are inaccessible by ground transportation. In the proposed system, there is a data analysis center, where the information from different sources (e.g., social media, rescue employees, police reports, satellite images, and drone monitoring) must be analyzed thoroughly. After analyzing the information, the data analysis center shares relevant information with the other centers that manage the response stage. In our study, we consider a similar information-sharing system, in which the damage information of the disaster-affected region is gathered in a single operation center after drones scan the area.

To extend the flight duration of drones, we allow drones to visit recharging stations (RS) along their routes. In this respect, the drone routing problem is similar to the electric vehicle routing problems (E-VRP), which are addressed by a large number of studies (e.g., [3, 19, 28, 30, 32, 44]). Schneider et al. [44] formulate an electric vehicle routing problem with time windows and cargo capacity constraints. The vehicles are fully charged at an RS, and the recharging time is linearly dependent on the battery level at the RS. Froger et al. [19] allow partial charging and non-linear charging functions and 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 [19]. However, both the node-based and the arc-based formulations require dummy RSs (i.e., recharging station copies) to allow multiple visits; and deciding the number of dummy RSs has a critical impact on the solution [19, 32]. While a large number of dummy RSs increase the computational time, few dummy RSs may cut off the optimal solutions. On the other hand, the path-based formulation is not limited by the number of RS replications [19].

In this study, we adapt arc-based and path-based formulations from [19] for our drone routing problem that addresses post-disaster damage assessment operations. There are several differences between our problem and the existing E-VRPs. Specifically, we maximize the total priority scores collected by visiting the grids, while the objective in [19] is to minimize the total time including the traveling time and recharging time. Besides, we consider a grid network, where the disaster-related attributes of the grids are utilized to assign each grid to a cluster, and only a subset of grids may be visited under a limited assessment period. We additionally consider a pre-specified coverage target for each cluster. 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.

3 Problem Definition

We consider a post-disaster setting, in which the drones assess the physical damage in the built environment after an earthquake. The disaster-affected area is divided into grids, and 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 identical drones and a limited time for assessments. We aim to maximize the total priority score collected from the assessed grids.

Grids may have different priorities, reflecting the urgency of visiting this grid, based on whether a grid is densely populated and/or there is an important facility, such as hospitals. Moreover, we propose to utilize the disaster-related characteristics, *attributes*, of the grids to make an inference about the physical damage without covering the entire set of grids as rapid coverage of the entire set of grids may not be possible. First, grids are clustered according to their attributes so that grids in the same clusters have similar attributes. Thus, gathering information about a finite number of grids from a cluster may allow us to predict the damage level of the remaining grids in that cluster. We also define a coverage ratio target for each cluster, where *the coverage ratio* equals the ratio of the number of visited grids in a cluster to the total number of grids in that cluster. However, it may not be possible to achieve the pre-specified coverage ratio target for each cluster in a limited time. Therefore, the coverage ratio constraint is transformed into a soft constraint, such that the maximum unfulfilled coverage is penalized in the objective function with a very large number. Thus, we aim to achieve balanced and adequate information among all clusters with this single objective function.

We assume a single operation center where the scanned images are analyzed. Each drone has a battery capacity and a maximum tour duration, $T_{\max}$. Hence, drones that leave the operation center have to return within $T_{\max}$ time units. Drones are recharged at the RSs to extend their flight time, and the recharging time is calculated by assuming a linear battery charging rate. We assume that drones leave the operation center and the RSs fully charged. In addition, we assume that the RSs are located at the center of grids and the travel time between two grids is equivalent to the travel time between the center of these grids. Figure 1 illustrates an area of 16 grids, which belong to 5 clusters indicated by different shapes on the figure. For example, grids #1, #6, and #7 are in the same cluster. There exist three RSs located at the center of grids #2, #5, and #11. The priority scores of each grid are also shown on the figure in parenthesis. We present an example route, in which the drone scans and assesses seven grids in total and is recharged at the RS located at grid #2.

We model this drone routing problem by maximizing the total priority score collected from the assessed locations while ensuring sufficient coverage for each cluster by two MILP formulations, (i) an arc-based formulation and (ii) a path-based formulation, presented in Sects. 3.1 and 3.2, respectively.

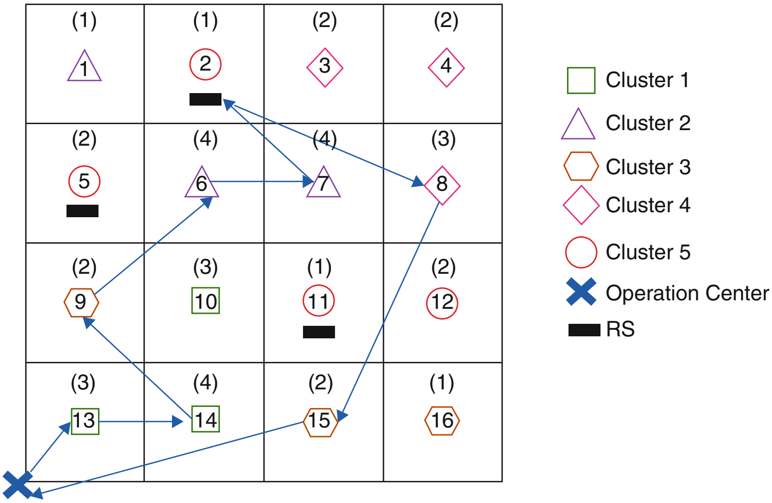


Fig. 1 Representation of a disaster-affected area divided into 16 grids and a drone's route

3.1 Arc-Based Formulation

The proposed drone routing problem is first formulated by an arc-based formulation, adapted from [19]. In the arc-based formulation, vertices represent the grid centers, and arcs connect these vertices. Let I represent the set of grids, and V_R specifies the set of RSs. V_R' defines the set of dummy RSs generated to allow multiple visits to each RS. 0 and $N+1$ denote the operation center, 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 operation center. 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 $A = \{(i, j) : i, j \in V, i \neq j\}$. C denotes the set of clusters.

Let s_j denote the survey time of grid j; 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 $T_{\max}$, such that each drone must return to the operation center within $T_{\max}$ time units. Additionally, the number of available drones is indicated by D. ρ_j represents the priority score assigned to the grid j. Moreover, ϕ_{jc} is the coverage parameter that equals 1 if grid j is assigned to cluster c and 0 otherwise. Additionally, we define τ_c which indicates the total number of grids in cluster c. μ is the coverage ratio target, and M is a sufficiently large number used to allow the model to report a feasible solution when there exists an infeasibility related to the coverage ratio target.

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 $i \in V_0$ and travels 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 surveyed. The continuous variable q_j denotes the remaining battery when the drone arrives at RS $j \in V'_R$. The continuous variable θ_j denotes the amount of time required to fully charge a drone at RS j. The continuous variables β_c and γ denote the coverage ratio of cluster $c \in C$ and the minimum coverage ratio overall clusters, respectively. Finally, the continuous variable Δ is defined as the difference between the achieved coverage ratio and the coverage ratio target. If the coverage ratio target is satisfied, Δ equals zero.

The parameters and variables of the arc-based formulation provided below are summarized in Appendix 1.

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

subject to

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

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

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

$$\sum_{j \in V_1} x_{0j} = D \tag{5}$$

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

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

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

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

$$\sum_{\substack{i \in V_0 \\ i \neq j}} y_{ij} - \sum_{\substack{i \in V_0 \\ i \neq j}} t_{ij} x_{ij} = q_j \quad \forall j \in V_R' \tag{10}$$

$$\sum_{\substack{l \in V_1 \\ l \neq j}} y_{jl} = \sum_{\substack{l \in V_1 \\ l \neq j}} B x_{jl} \quad \forall j \in V_R' \tag{11}$$

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

$$q_j \leq \sum_{\substack{i \in V_0 \\ i \neq j}} B x_{ij} \quad \forall j \in V_R' \tag{13}$$

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

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

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

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

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

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

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

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

$$\gamma \leq \beta_c \quad \forall c \in C \tag{22}$$

$$\gamma + \Delta \geq \mu \tag{23}$$

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

$$q_j \geq 0 \quad \forall j \in V_R' \tag{25}$$

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

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

$$\beta_c \geq 0 \quad \forall c \in C \tag{28}$$

$$\gamma \geq 0, \quad \Delta \geq 0 \tag{29}$$

The objective function (1) maximizes the total priority scores of the visited grids and penalizes the maximum unfulfilled coverage ratio target. 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 operation center 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 operation center. Constraints (9) denote the remaining battery after a drone traverses the arc (i, j) and surveys 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 operation center 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 [19], such that the drone must have sufficient battery to arrive at the nearest RS or the operation center after traversing arc (i, j). Constraints (21) denote the coverage ratio for each cluster. Constraints (22) indicate the minimum coverage ratio among all clusters. Constraint (23) specifies whether the coverage ratio target is satisfied by the visited grids and calculates Δ, which is the difference between the achieved and targeted coverage ratio. Note that the unsatisfied coverage ratio is penalized in the objective function. Finally, constraints (24)–(29) define the domains of the decision variables.

3.2 Path-Based Formulation

The arc-based formulation introduced in Sect. 3.1 requires one to generate several dummy RS nodes to allow multiple visits at each RS. However, increasing the number of dummy RSs increases the computational difficulty of the model. Moreover, calculating the exact number of required dummy RSs is not possible without solving the problem. To overcome these obstacles in the arc-based formulation, a path-based formulation is introduced in [19], which we adapt to the drone routing problem in this subsection.

In the path-based formulation, a path corresponds to a set of arcs between two vertices, which are not RS nodes, including or excluding possible visits to the RSs. More specifically, the starting and ending points of a path can only be the grid centers to be scanned or the operation center, and any number of RS visits can be included between the starting and the ending points of a path.

We define the drone routing problem on a directed multi-graph $\hat{G} = (\hat{V}, \hat{A})$, with the set of arcs $\hat{A}$= $\left\{(i, j) : i, j \in \hat{V}, i \neq j\right\}$ where $\hat{V} = I \cup \{0\} \cup \{N+1\}$. $\hat{V}_0 = I \cup \{0\}$ denotes the set of arrival vertices and $\hat{V}_1 = I \cup \{N+1\}$ denotes the set of departure vertices. A path represents an arc between two vertices, such that path p starts at the center of grid $i \in I \cup \{0\}$ and ends at the center of grid $j \in I \cup \{N+1\}$. A path may include varying number of visits to RSs. In other words, a path may include no visits to any RS. Then, the set of paths on the arc (i, j) is defined by P_{ij}, and the set of all paths is indicated by 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 $L_p = \left\{0, 1, \ldots, n_p - 1\right\}$. $\pi_p(l)$ denotes the RS at position $l \in L_p$. If $L_p = \emptyset$, then the drone doesn't visit any RS on path p. Traveling time of $p \in P$ is represented by t_p that is equal to drone's energy consumption.

The binary variable x_p is equal to 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, respectively, before the drone travels on path $p \in P$. 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 parameters and variables of the path-based formulation are summarized in Appendix 1. We next present the path-based formulation

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

subject to

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

$$\sum_{\substack{i \in \hat{V}_0 \\ i \neq j}} \sum_{p \in P_{ij}} x_p \leq 1 \quad \forall j \in I \tag{31}$$

$$\sum_{\substack{j \in \hat{V}_1 \\ i \neq j}} \sum_{p \in P_{ij}} x_p - \sum_{\substack{j \in \hat{V}_0 \\ i \neq j}} \sum_{p \in P_{ji}} x_p = 0 \quad \forall i \in I \tag{32}$$

$$\sum_{j \in \hat{V}_1} \sum_{p \in P_{0j}} x_p = D \tag{33}$$

$$\sum_{i \in \hat{V}_0} \sum_{p \in P_{i,N+1}} x_p = D \tag{34}$$

$$\sum_{\substack{k\in\hat{V}_1\\k\neq j}}\sum_{p\in P_{jk}} y_p = \sum_{\substack{i\in\hat{V}_0\\i\neq j}}\sum_{p\in P_{ij}}\left(y_p-(t_p+s_j)x_p+\sum_{\substack{l\in L_p\\|L_p|\neq 0}}(Bx_p-q_{pl})\right) \quad \forall j\in I \tag{35}$$

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

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

$$y_p = Bx_p \quad \forall i\in\hat{V}_1, p\in P_{0i} \tag{38}$$

$$y_p \leq Bx_p \quad \forall p\in P \tag{39}$$

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

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

$$\sum_{j\in\hat{V}_1}\sum_{p\in P_{0j}} a_p = 0 \tag{42}$$

$$\sum_{\substack{k\in\hat{V}_1\\k\neq j}}\sum_{p\in P_{jk}} a_p = \sum_{\substack{i\in\hat{V}_0\\i\neq j}}\sum_{p\in P_{ij}}\left(a_p+(t_p+s_j)x_p+\sum_{\substack{l\in L_p\\|L_p|\neq 0}}r(Bx_p-q_{pl})\right) \quad \forall j\in I \tag{43}$$

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

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

$$a_p\geq 0, y_p\geq 0 \quad \forall p\in P \tag{46}$$

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

The objective function (30) is the same as the arc-based formulation (Sect. 3.1). Constraints (31) indicate that each grid can be surveyed at most once. Constraints (32) are for balancing the flows. Constraints (33) and (34) ensure that the number of drones that depart from and return to the operation center equals to the number of available drones. Constraints (35) track the drone's battery level at each grid. Constraints (36) and (37) indicate the remaining battery level when the drone arrives at the first RS and departs to the next RS on path $p\in P$, respectively.

Constraints (38) ensure that drones leave the operation center with a full charge. Constraints (39) and (40) link the battery level variables. Constraints (41) guarantee the path feasibility, such that the drone can arrive at its destination with a sufficient battery level. Constraints (42) define the operation start time. Constraints (43) track the time at each grid. Constraints (44) indicate the operation duration. Finally, constraints (45)–(47) indicate the domain of decision variables.

In Sect. 5.2, we compare the computational times of the proposed arc-based and path-based MILP formulations and show that both MILP models may require high computational times, especially for large instances. Similar formulations are also proposed for E-VRPs that are shown to be NP-hard [19]. However, relief agencies may not be able to access any advanced tools, and the decisions must be applied immediately after the disaster [4]. To compute reasonable solutions quickly, we develop a constructive heuristic (CH) in Sect. 4.

4 A Constructive Heuristic (CH)

Our CH developed to solve the proposed drone routing problem consists of three moves in the main phase: (i) initial route generation, (ii) insertion, and (iii) exchange. Additionally, we consider two sub-phase moves: (iv) repair and (v) improvement. The improvement sub-phase includes several operators, which are performed after every main phase move. The repair sub-phase move is performed only if the initial route generation produces a solution that does not satisfy the coverage ratio target. An overview of the CH is provided in Algorithm 1.

Initial solution generation step aims to generate a set of routes that satisfies the targeted coverage ratio. In other words, our aim in this step is not to maximize the collected priority scores, but to construct routes that minimize the route duration while satisfying the coverage ratio target. Therefore, the first step terminates when the coverage ratio target is satisfied by visiting a set of grids. In this step, a new route (drone) is initiated when the previous route cannot survey any additional grid due to time limitations. A drone route starts at the operation center, and we identify a candidate grid to survey next. The candidate grid is set as the grid that has the minimum operation time where the operation time is defined as the summation of survey time and travel time depending on the current position of the drone. If there

Algorithm 1: An Overview of the CH

```
Step 0: Generate an initial solution.
If the initial solution satisfies the coverage ratio target, go to Step 2; otherwise go to Step 1.
Step 1: Apply the repair move, go to Step 2;
Step 2: Apply the insertion move, followed by the improvement operators.
Step 3: Apply the exchange move, followed by the improvement operators.
If the solution is improved and there are any unvisited grids, go to Step 2; otherwise
  terminate the algorithm.
```

are multiple grids with the same operation time, the grid with the highest priority score is selected as the next candidate grid. If the remaining battery is not enough to visit the candidate grid, the drone visits the nearest RS, and the next candidate grid to visit is selected dependent on the position of the visited RS. We present the pseudocode of the initial solution generation step in Algorithm 2. Each time a grid is appended to the route of the drone, we apply the improvement sub-phase.

The *improvement* sub-phase, whose overview is presented in Algorithm 3, includes four operators: 1-opt, 2-opt, swap, and RS removal. The 1-opt operator changes the position of one grid in the route. The 2-opt operator removes two links and replaces them with two different links. The swap operator exchanges the order of two grids in the route. Note that the previously visited RSs may need to be updated when routes are modified by these improvement operators. Therefore, after an operator terminates, we apply the RS removal operator, which removes the redundant RSs to reduce the total charging time in a route. Each operator applies the first-improvement strategy; that is, we terminate searching for a new solution once a battery-feasible solution with a lower route duration is achieved. We apply the improvement operators in the following order, which are set based on preliminary experiments: 2-opt, RS removal, 1-opt, RS removal, swap, and RS removal. We apply the improvement sub-phase iteratively following this order of operators until no improving solution is found in the last iteration. We apply the improvement operators on each drone route sequentially.

The initial solution generation step terminates once routes that satisfy the coverage ratio target are generated. However, if the initial step does not generate any feasible solutions satisfying the coverage ratio target within $T_{\max}$, the repair sub-phase is implemented, involving a remove-add operator which aims to decrease the duration of each route and an insertion operator which aims to improve coverage by inserting grids, and hence it attains a feasible solution. Specifically, we pick each cluster with unsatisfied coverage ratio targets sequentially and attempt removing a grid from the solution and insert an unvisited candidate grid to the position of the removed grid to improve the route duration. If we attain a feasible solution that yields a lower route duration, we update the route. Note that while applying the remove-add move, the feasibility of the route with respect to the battery constraint must be checked. For the battery feasibility check, we evaluate three options for each candidate grid to be added to the solution: only the candidate grid and adding the candidate grid with its nearest RS. Specifically, let i be a candidate grid, and RS_N is the nearest RS to the center of i. Then the options are $\{i\}$, $\{i, RS_N\}$, and $\{RS_N, i\}$. We select the option with the least route duration in evaluating candidate grid i's insertion to the route. The remove-add operator does not change the coverage ratios directly, but it allows to increase the number of visited grids and the coverage ratios by reducing the route duration. Hence, an insertion operator is implemented sequentially for each cluster that is covered less than the coverage ratio target. The unvisited grids in the regarding cluster are sorted in descending order according to their priority scores, and three options including the grid-RS combinations are generated. We try to insert additional grids to the routes until a cluster satisfies the coverage ratio target or there is no any feasible option to be inserted.

Algorithm 2: Initial Solution Generation

```
Initialization;
$S_+$ = Set of surveyed grids
$S_0$ = Set of candidate grids
$\tau_k$ = Route duration of drone k
$A$ = Minimum coverage ratio
$T_{\max}$ = Route duration limit
while $A <$ Coverage ratio target 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 yielded the minimum operation time;
        if insertion of z to the route of k is time-feasible and battery-feasible then
            Add z to the route of k;
            Do improvement
            Update $S_+$,$S_0$, $A$, $\tau_k$;
        else
            if Surveying an updated z is time-feasible after recharging then
                Visit the nearest RS, update z;
                Add z to the route of k;
                Do improvement
                Update $S_+$,$S_0$, $A$, $\tau_k$;
            else
                % Return to the operation center
                $\tau_k = T_{\max}$;
            end
        end
    else
        while $\tau_k \leq T_{\max}$ do
            % Add to the current route
            z:= candidate grid yielded the minimum operation time;
            if insertion of z to the route of k is time-feasible and battery-feasible then
                Add z to the route of d;
                Do improvement
                Update $S_+$,$S_0$, $A$, $\tau_k$;
            else
                if Surveying an updated z is time-feasible after recharging then
                    Visit the nearest RS, update z;
                    Add z to the route of k;
                    Do improvement
                    Update $S_+$, $S_0$, $A$, $\tau_k$;
                else
                    % Return to the operation center
                    $\tau_k = T_{\max}$;
                end
            end
        end
    end
end
```

Algorithm 3: Improvement Phase

```
Initialization;
x:=route
$O_i$:= $i$th operator for $i$=1,...,6
$impr$:=1
while $impr \neq 6$ do
    Generate a candidate solution $x'$ by $O_i$(x);
    if $x'$ is battery-feasible and better than $x$ then
        $x = x'$;
        $impr = 1$;
    else
        if $i \neq 6$ then
            $i = i + 1$;
            $impr = impr + 1$
        else
            $i = 1$;
            $impr = impr + 1$
        end
    end
end
```

If there is any unvisited grid after the initial step terminates, the *insertion* move tries to insert them into the current solution to maximize the total priority score. First, we sort the unvisited grids in descending order according to their priority scores. Then we generate three insertion options for each candidate grid-RS combination, as explained in the repair sub-phase. If an insertion option can be feasibly applied and the grid can be inserted in an existing or a new route (if an unrouted drone exists), the grid is inserted, and then the improvement sub-phase is performed. The insertion is performed until no new feasible insertions can be identified.

The *exchange* move searches for an improved solution in terms of priority scores. If there are unvisited grids in the network after implementing the insertion phase, and their priority scores are higher than the visited ones, we attempt to exchange the candidate grid that has the highest priority score with a visited grid with a lower priority score. While inserting a grid in a route, in this procedure, the three insertion options explained above are considered. If an exchange attempt produces a feasible solution in terms of route duration, battery, and coverage ratio target levels, we perform the exchange move, which is followed by the improvement. We apply the exchange move only for the non-empty routes (i.e., routes that visit at least one grid).

5 Numerical Analysis

In this section, we describe the results of the numerical study conducted based on a set of test instances. The objectives of our numerical study are 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 with respect to solution quality and computational time. To evaluate the heuristic performance, we compare the solution time and the objective value achieved by the proposed algorithm relative to the best-found CPLEX solution.

We describe test instances in Sect. 5.1 and evaluate the performance of the alternative formulations and the CH in Sect. 5.2.

5.1 Test Instances

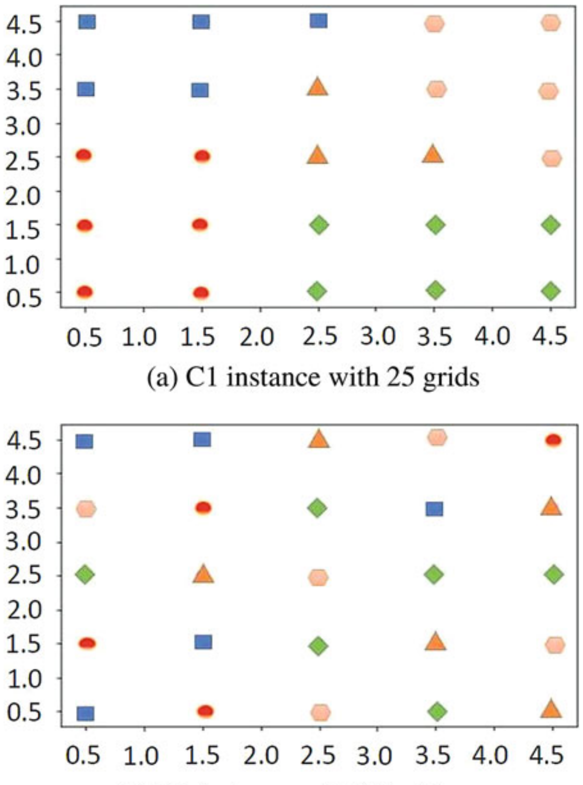
We generate 54 instances with 16, 25, and 36 equal-sized grids. The instances differ in how we assign attributes to the grids and the number of drones. We assume that the affected area is divided into square grids with a 1 kilometer side length. The operation center is assumed to be located at the left bottom corner of each instance (as in Fig. 1).

Although the area of each grid is equal, we consider different survey 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 surveying. Thus, the survey 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 min [6]. The priority score of each grid is randomly generated between [1,5].

We generate 6 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, with each one having a different number of levels (e.g., four levels for the soil type attribute such as hard rock, rock, stiff soil, and soft soil; three levels for the wealth index attribute, which are high, moderate, and low). We assign the attributes to the grids following two policies: (i) assign similar attribute levels based on the geographical proximity of grids, which produce C-type instances, C1, C2, and C3, and (ii) assign attribute levels randomly, which generate R-type instances: R1, R2, and R3 instances. We then use k-modes algorithm and assign the grids into five clusters [23]. Figure 2 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 target coverage ratio as 0.25 and the battery capacity (i.e., maximum flight time) as 50 min. Similar to [9], we assume that a drone that is completely out of charge is fully charged in 5 min. Accordingly, we set the recharging time required to fill the battery for a 1-min flight time as 0.1 min. The speed of a drone is set at 60 km/h. $T_{\max}$ is assumed to be 180 min, and the number of RSs equals three for each instance. Moreover, to solve the arc-based MILP

Fig. 2 Illustrative example of two types of instances for a 25 km^2 area



(a) C1 instance with 25 grids

(b) R1 instance with 25 grids

formulation, we set the number of dummy RSs to 3, 5, and 7 for the instances with 16, 25, and 36 grids, respectively. We consider using one, two, and four drones. Finally, we set the penalty weight "M" for the maximum unfulfilled coverage ratio target to 10,000 in the objective function. Here, we aim to dominate the objective function by the coverage penalty and distinguish the total priority score and the resultant penalty in the objective function value by setting M to a very large number. In our instances, total priority score can be at most 180 for the instances with 36 grids, and setting M to 10,000 causes a penalty more than 180 for any possible amount of unfulfilled coverage ratio.

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 survey 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.

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

subject to

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

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

$$y_{ij} - x_j \le 0 \quad \forall i \in I, j \in J \tag{51}$$

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

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

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

5.2 *Results*

We solve 54 test instances by using the proposed formulations and the CH. CPLEX 12.8.0 is used to solve the proposed mathematical models by imposing a 30-min time limit. The CH is coded in Python 3.6. All the numerical analyses are conducted on a HP 8th Generation Intel Core i7 processor with 8 GB of RAM in 64-bit mode.

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 3, 5, and 7 dummy RSs for instances with 16, 25, and 36 grids, respectively.

The path-based formulation requires generating all possible paths in advance. The paths are generated in 3, 20, and 95 s on average for the instances with 16, 25, and 36 grids, respectively. While we do not apply any strategies to eliminate the redundant paths in this study, one may strengthen the path-based formulation by identifying dominance rules (see [19]).

Tables 1 and 2 represent the computational results obtained by CPLEX for both arc-based and path-based formulations. The objective function values are denoted

Table 1 Computational results of the arc-based and path-based formulations for R-type and C-type instances with 16 grids

			Arc-based formulation					Path-based formulation				
Ins.	# of drones	# of grids	# of visits	CR %	Z^{Arc}	Gap %	Comp. time(s)	# of visits	CR %	Z^{path}	Gap %	Comp. time(s)
R1	1	16	11	34	40	0.0	25.8	11	34	40	0.0	15.5
R1	2	16	16	100	51	0.0	0.6	16	100	51	0.0	1.5
R1	4	16	16	100	51	0.0	0.6	16	100	51	0.0	1.4
R2	1	16	11	33	40	0.0	1.2	11	33	39	0.0	7.9
R2	2	16	16	100	51	0.0	0.6	16	100	51	0.0	1.5
R2	4	16	16	100	51	0.0	0.6	16	100	51	0.0	1.3
R3	1	16	10	25	41	0.0	1.2	10	25	40	0.0	1.5
R3	2	16	16	100	51	0.0	0.6	16	100	51	0.0	1.4
R3	4	16	16	100	51	0.0	0.6	16	100	51	0.0	1.3
C1	1	16	10	25	41	0.0	1.7	10	25	40	0.0	2.1
C1	2	16	16	100	51	0.0	0.6	16	100	51	0.0	1.5
C1	4	16	16	100	51	0.0	0.6	16	100	51	0.0	1.4
C2	1	16	10	50	41	0.0	2.0	10	50	41	0.0	2.2
C2	2	16	16	100	51	0.0	0.6	16	100	51	0.0	1.6
C2	4	16	16	100	51	0.0	0.6	16	100	51	0.0	1.4
C3	1	16	10	25	40	0.0	30.8	10	50	40	0.0	34.3
C3	2	16	16	100	51	0.0	0.6	16	100	51	0.0	1.5
C3	4	16	16	100	51	0.0	0.6	16	100	51	0.0	1.3

by Z^{arc} (Z^{path}) for the arc-based (path-based) formulation, and CR denotes the minimum coverage ratio achieved among all clusters. All instances with 16 grids are solved to optimality by both formulations within a 30-min time limit (usually within a couple of seconds), as shown in Table 1. When the number of drones is one, CR is always less than 100%. However, CR 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. 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

Table 2 Computational results of the arc-based and path-based formulations for R-type and C-type instances with 25 and 36 grids

			Arc-based formulation					Path-based formulation				
Ins.	# of drones	# of grids	# of visits	CR %	Z^{Arc}	Gap %	Comp. time(s)	# of visits	CR %	Z^{path}	Gap %	Comp. time(s)
R1	1	25	10	33	42	0.0	2.9	10	28	42	0.0	5.1
R1	2	25	21	66	68	1.5	1800	21	66	68	2.9	1800
R1	4	25	25	100	73	0.0	2.0	25	100	73	0.0	3.6
R2	1	25	11	25	41	0.0	179.5	11	25	41	0.0	1093
R2	2	25	21	66	68	1.5	1800	21	60	68	2.9	1800
R2	4	25	25	100	73	0.0	2.3	25	100	73	0.0	3.4
R3	1	25	10	33	40	0.0	230.5	10	33	40	0.0	1569
R3	2	25	21	60	68	1.5	1800	21	60	68	3.0	1800
R3	4	25	25	100	73	0.0	2.2	25	100	73	0.0	3.6
C1	1	25	10	33	41	0.0	6.7	10	33	41	0.0	6.1
C1	2	25	21	33	68	1.5	1800	21	33	68	3.0	1800
C1	4	25	25	100	73	0.0	2.3	25	100	73	0.0	3.6
C2	1	25	10	34	42	0.0	8.0	10	34	42	0.0	64
C2	2	25	21	68	68	1.9	1800	21	80	68	2.9	1800
C2	4	25	25	100	73	0.0	2.6	25	100	73	0.0	3.4
C3	1	25	10	25	42	0.0	3.2	10	25	42	0.0	5.2
C3	2	25	21	51	68	1.5	1800	21	66	68	2.9	1800
C3	4	25	25	100	73	0.0	2.1	25	100	73	0.0	3.8
R1	1	36	11	27	50	0.0	779.5	11	27	50	0.0	232.8
R1	2	36	20	45	**86**	3.8	1800	20	34	85	3.5	1800
R1	4	36	36	100	113	0.0	180.9	36	100	113	0.0	334
R2	1	36	11	27	41	6.3	1800	12	28	**42**	0.0	1735
R2	2	36	20	30	82	7.3	1800	20	30	82	6.1	1800
R2	4	36	36	100	113	0.0	274.8	36	100	113	0.0	68.1
R3	1	36	12	28	45	9.1	1800	12	28	**46**	0.0	829.3
R3	2	36	20	40	85	4.6	1800	20	40	85	3.2	1800
R3	4	36	36	100	113	0.0	282.8	36	100	113	0.0	32.8
C1	1	36	12	33	**46**	6.6	1800	12	33	44	8.1	1800
C1	2	36	20	33	85	4.7	1800	20	33	85	3.3	1800
C1	4	36	36	100	113	0.0	260.9	36	100	113	0.0	1730
C2	1	36	12	33	**46**	0.0	634.5	11	22	48-M[a]	100	1800
C2	2	36	20	33	85	4.7	1800	20	33	85	3.3	1800
C2	4	36	36	100	113	0.0	1076	36	100	113	0.0	64.5
C3	1	36	12	33	45	8.0	1800	12	33	**46**	0.0	802
C3	2	36	20	34	85	4.7	1800	20	34	85	3.3	1800
C3	4	36	36	100	113	0.0	339	36	100	113	0.0	34.4

[a] Coverage ratio target cannot be met within 30 min time limit and a large negative objective function value is reported to indicate the unfulfilled coverage ratio target

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) 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 1214 (1127) and 478 (586) seconds for the all of the instances 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. 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.

Post-disaster relief operations should be organized as quickly as possible. Hence, one may not prefer waiting for the 30-min time limit. To compute a route for each drone immediately, we proposed the CH algorithm introduced in Sect. 4. Even though the proposed CH is relatively simpler than the metaheuristics proposed for E-VRP (such as [16, 18, 44]), it computes high-quality and feasible solutions for our instances within less than a second. Specifically, the CH achieves the optimal solutions in 13 (7 and 6) out of 18 instances when there are 16 (25 and 36) grids, which are presented in Tables 3 and 4.

We define the *CH gap* between the CH and the arc- and path-based mathematical formulations as the ratio of the difference between the best-computed objective function value by the mathematical formulations (denoted by Z^{BEST}) and the

Table 3 Computational results of the CH and the best solution computed by the mathematical formulations for R-type and C-type instances with 16 grids

Ins.	# of drones	# of grids	# of visits	CR %	Comp. time(s)	Z^{CH}	Z^{BEST}	CH Gap%
R1	1	16	11	34	0.3	40	40	0.0
R1	2	16	16	100	0.2	51	51	0.0
R1	4	16	16	100	0.4	51	51	0.0
R2	1	16	10	33	0.2	35	40	12.5
R2	2	16	16	100	0.1	51	51	0.0
R2	4	16	16	100	0.1	51	51	0.0
R3	1	16	10	50	0.4	39	41	4.9
R3	2	16	16	100	0.2	51	51	0.0
R3	4	16	16	100	0.2	51	51	0.0
C1	1	16	10	50	0.4	40	41	2.4
C1	2	16	16	100	0.2	51	51	0.0
C1	4	16	16	100	0.2	51	51	0.0
C2	1	16	10	25	0.3	40	41	2.4
C2	2	16	16	100	0.2	51	51	0.0
C2	4	16	16	100	0.2	51	51	0.0
C3	1	16	10	50	0.3	38	40	5.0
C3	2	16	16	100	0.2	51	51	0.0
C3	4	16	16	100	0.2	51	51	0.0

CH gap equals (Z^{BEST}-Z^{CH})/Z^{BEST} where Z^{BEST} is the best-found solution obtained by CPLEX

Table 4 Computational results of CH and the best solution computed by MILP formulations for R-type and C-type instances with 25 and 36 grids

Ins.	# of drones	# of grids	# of visits	CR %	Comp. time(s)	Z^{CH}	Z^{BEST}	CH Gap%
R1	1	25	11	33	0.4	36	42	14.3
R1	2	25	20	55	0.7	66	68	2.9
R1	4	25	25	100	0.3	73	73	0.0
R2	1	25	11	28	0.4	36	41	12.2
R2	2	25	21	66	0.9	68	68	0.0
R2	4	25	25	100	0.4	73	73	0.0
R3	1	25	11	36	0.4	37	40	7.5
R3	2	25	20	60	0.8	65	68	4.4
R3	4	25	25	100	0.4	73	73	0.0
C1	1	25	11	33	0.4	32	41	21.9
C1	2	25	20	33	0.8	66	68	2.9
C1	4	25	25	100	0.4	73	73	0.0
C2	1	25	11	34	0.3	34	42	19.0
C2	2	25	20	40	0.8	65	68	4.4
C2	4	25	25	100	0.3	73	73	0.0
C3	1	25	11	25	0.4	33	42	21.4
C3	2	25	20	50	0.7	66	68	2.9
C3	4	25	25	100	0.4	73	73	0.0
R1	1	36	12	27	0.5	40	50	20.0
R1	2	36	21	45	0.6	79	86	8.1
R1	4	36	36	100	0.5	113	113	0.0
R2	1	36	12	27	0.4	37	42	11.9
R2	2	36	21	30	0.6	79	82	3.7
R2	4	36	36	100	0.6	113	113	0.0
R3	1	36	12	28	0.6	31	46	32.6
R3	2	36	21	40	0.6	73	85	14.1
R3	4	36	36	100	0.6	113	113	0.0
C1	1	36	12	33	0.7	39	46	15.2
C1	2	36	22	34	0.8	75	85	11.8
C1	4	36	36	100	0.6	113	113	0.0
C2	1	36	12	33	0.6	39	46	15.2
C2	2	36	22	34	0.8	75	85	11.8
C2	4	36	36	100	0.6	113	113	0.0
C3	1	36	12	33	0.3	35	46	23.9
C3	2	36	21	34	0.7	80	85	5.9
C3	4	36	36	100	0.5	113	113	0.0

CH gap equals (Z^{BEST}-Z^{CH})/Z^{BEST} where Z^{BEST} is the best-found solution obtained by CPLEX objective function value computed by the CH (denoted by Z^{CH}), to the best-computed objective function value by the mathematical formulations. The average percentage gaps are 2, 6, and 10% for the instances with 16, 25, and 36 grids, respectively. The CH gaps decrease as the number of drones increases since limiting the number of drones makes it more difficult to satisfy the coverage ratio target. In other words, the CH first finds a feasible solution that satisfies the coverage ratio target, and then it improves the initial solution by improvement moves. If the coverage ratio target is satisfied more easily with the help of more drones, the solution quality increases since the CH focuses on increasing the collected priority scores rather than satisfying the coverage ratio target.

To sum up, the arc-based and path-based formulations result in very similar performances concerning computation times and computed objective function values. However, some instances cannot be solved to optimality within the 30-min time interval. To overcome this limitation, we propose the CH heuristic that computes a feasible solution less than a second for each instance. The quality of the CH heuristic's solution gets better as the number of drones increases and/or the number of grids decreases. One may still prefer utilizing the CH heuristic that has an average CH gap of 6%.

6 Conclusion

In this paper, we consider a post-disaster setting, in which the drones assess the physical damages on the built infrastructure caused by an earthquake. The disaster-affected area is firstly divided into grids where each grid has a set of attributes comprising its infrastructure, geographical, geological, and socioeconomic conditions. Then, grids are clustered based on their attributes. 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 the computational performance of both formulations is very similar. 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. In addition to the proposed formulations, we propose a CH that computes a feasible solution that satisfies the coverage ratio target quickly, that is, in less than a second. The CH achieves a high level of performance in instances with a larger number of drones since it is easy to generate an initial feasible solution that satisfies the coverage ratio target in these cases.

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. Finally, coordination of damage assessment operations and relief operations can be addressed in future research.

Appendix 1

Tables 5 and 6 demonstrate the sets, parameters, and decision variables of the arc-based and path-based formulations, respectively.

Table 5 Sets, parameters, and decision variables for the arc-based formulation

Sets and parameters
0 and $N+1$: Same operation center for departure and arrival
I: Set of grids to be surveyed
C: Set of clusters
V_R: Set of recharging stations
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 operation center
s_j: Survey time of grid j
t_{ij}: Travel time between grid i and grid j
B: Battery capacity
r: Recharging rate
$T_{\max}$: Route duration limit
D: Number of drones
ρ_j: Priority score of grid j
ϕ_{jc}: A coverage parameter equals to 1 if grid j is in cluster c and 0 otherwise
τ_c: Total number of grids in cluster c
μ: Coverage ratio target
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 at grid i to grid j, $(i, j) \in A$
y_{ij}: Remaining battery departing at grid i to grid j, $(i, j) \in A$
z_j: 1 if grid $j \in I$ is surveyed; 0 otherwise
q_j: Remaining battery before the charging operation at RS $j \in V'_R$
β_c: Coverage ratio of cluster $c \in C$
γ: Minimum coverage ratio
Δ: Difference between the achieved coverage ratio and the coverage ratio target

Table 6 Sets, parameters, and decision variables for the path-based formulation

Sets and parameters
0 and $N+1$: Same operation center for departure and arrival
I: Set of grids to be surveyed
C: Set of clusters
$\hat{V} = I \cup \{0\} \cup \{N+1\}$: 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, $(i, j) \in \hat{A}$
s_j: Survey time of grid j
t_p: Travel time of path $p \in P$
B: Battery capacity
r: Recharging rate
$T_{\max}$: Route duration limit
D: Number of drones
ρ_j: Priority score of grid j
ϕ_{jc}: A coverage parameter equals to 1 if grid j is in cluster c and 0 otherwise
τ_c: Total number of grids in cluster c
μ: Coverage ratio target
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 travels on path $p \in P$
y_p: Remaining battery before the drone travels on path $p \in P$
z_j: 1 if grid $j \in I$ is surveyed; 0 otherwise
q_{pl}: Remaining battery before the charging operation on path $p \in P$ at the $l^{\text{th}} RS$
β_c: Coverage ratio of cluster $c \in C$
γ: Minimum coverage ratio
Δ: Difference between the achieved coverage ratio and the coverage ratio target

References

1. Agatz, N., Bouman, P., Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. Transportation Science, 52(4), 965–981.
2. American Red Cross. (2015). Drones for disaster response and relief operations. Retrieved from https://www.issuelab.org/resources/21683/21683.pdf Accessed 17 May 2020
3. Andelmin, J., Bartolini, E. (2017). An exact algorithm for the green vehicle routing problem. Transportation Science, 51(4), 1288–1303.
4. Balcik, B. (2016, June). Selective routing for post-disaster needs assessments. In International Conference on Dynamics of Disasters (pp. 15–36). Springer, Cham.
5. Balcik, B., Beamon, B. M. (2008). Facility location in humanitarian relief. International Journal of Logistics, 11(2), 101–121.

6. Boccardo, P., Chiabrando, F., Dutto, F., Tonolo, F. G., Lingua, A. (2015). UAV deployment exercise for mapping purposes: Evaluation of emergency response applications. Sensors, 15(7), 15717–15737.
7. Chauhan, D., Unnikrishnan, A., Figliozzi, M. (2019). Maximum coverage capacitated facility location problem with range constrained drones. Transportation Research Part C: Emerging Technologies, 99, 1–18.
8. Chmaj, G., and Selvaraj, H. (2015). Distributed processing applications for UAV/drones: a survey. In Progress in Systems Engineering (pp. 449–454). Springer, Cham.
9. Chowdhury, S. (2018). Drone routing and optimization for post-disaster inspection. Mississippi State University.
10. Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S. G., Bian, L. (2017). Drones for disaster response and relief operations: A continuous approximation model. International Journal of Production Economics, 188, 167–184.
11. Davidson, R. A., Shah, H. C. (1997). An urban earthquake disaster risk index. Standford University: John A. Blume Earthquake Engineering Center.
12. Dhein, G., Zanetti, M. S., de Araújo, O. C. B., and Cardoso Jr, G. (2019). Minimizing dispersion in multiple drone routing. Computers and Operations Research, 109, 28–42.
13. Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S. (2016). Vehicle routing problems for drone delivery. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(1), 70–85.
14. Duarte, D., Nex, F., Kerle, N., Vosselman, G. (2017). 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, 42, 93.
15. Erdelj, M., Natalizio, E. (2016, February). UAV-assisted disaster management: Applications and open issues. In 2016 international conference on computing, networking and communications (ICNC) (pp. 1–5). IEEE.
16. Erdoğan, S., and Miller-Hooks, E. (2012). A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 48(1), 100–114.
17. Estrada, M. A. R., Ndoma, A. (2019). The uses of unmanned aerial vehicles–UAV's-(or drones) in social logistic: Natural disasters response and humanitarian relief aid. Procedia Computer Science, 149, 375–383.
18. Felipe, Á., Ortuño, M. T., Righini, G., and Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transportation Research Part E: Logistics and Transportation Review, 71, 111–128.
19. Froger, A., Mendoza, J. E., Jabali, O., Laporte, G. (2019). Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Computers and Operations Research, 104, 256–294.
20. Golabi, M., Shavarani, S. M., Izbirak, G. (2017). An edge-based stochastic facility location problem in UAV-supported humanitarian relief logistics: a case study of Tehran earthquake. Natural Hazards, 87(3), 1545–1565.
21. Ham, A. M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. Transportation Research Part C: Emerging Technologies, 91, 1–14.
22. Ritchie, H. and Roser, M. (2020) - "Natural Disasters". Published online at OurWorldInData.org. Retrieved from: https://ourworldindata.org/natural-disasters Accessed 17 May 2020
23. Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery, 2(3), 283–304.
24. IFRC (International Federation of Red Cross and Red Crescent Societies): World disasters report 2018. https://media.ifrc.org/ifrc/wp-content/uploads/sites/5/2018/10/B-WDR-2018-EN-LR.pdf. Accessed 12 March 2020
25. IFRC (International Federation of Red Cross and Red Crescent Societies): Humanitarian logistics and procurement. (n.d.). Retrieved from https://www.ifrc.org/en/what-we-do/logistics/ Accessed 17 May 2020

26. Kim, S. J., Lim, G. J., Cho, J., Côté, M. J. (2017). Drone-aided healthcare services for patients with chronic diseases in rural areas. Journal of Intelligent and Robotic Systems, 88(1), 163–180.
27. Kim, K., Pant, P., Yamashita, E. (2015). Disasters, drones, and crowdsourced damage assessment. In Proceedings of Computers in Urban Planning and Urban Management Conference, Cambridge, Massachusetts.
28. Koç, Ç., and Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. Applied Soft Computing, 39, 154–164.
29. Leetaru, K., (2015). How Drones Are Changing Humanitarian Disaster Response. Retrieved from https://www.forbes.com/sites/kalevleetaru/2015/11/09/how-drones-are-changing-humanitarian-disaster-response/ Accessed 17 May 2020
30. Leggieri, V., and Haouari, M. (2017). A practical solution approach for the green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 104, 97–112.
31. Meier, P., (2015). Crisis Mapping Nepal with Aerial Robotics. Retrieved from https://irevolutions.org/2015/11/04/crisis-mapping-nepal-aerial-robotics/ Accessed 17 May 2020
32. Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological, 103, 87–110.
33. Motlagh, N. H., Taleb, T., Arouk, O. (2016). Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. IEEE Internet of Things Journal, 3(6), 899–922.
34. Murphy, R. R., Duncan, B. A., Collins, T., Kendrick, J., Lohman, P., Palmer, T., Sanborn, F. (2016). Use of a Small Unmanned Aerial System for the SR-530 Mudslide Incident near Oso, Washington. Journal of field Robotics, 33(4), 476–488.
35. Murray, C. C., Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies, 54, 86–109.
36. Murray, C., Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones, Transportation Research Part C: Emerging Technologies, 110, 368–398.
37. Nedjati, A., Vizvari, B., Izbirak, G. (2016). Post-earthquake response by small UAV helicopters. Natural Hazards, 80(3), 1669–1688.
38. Oruc, B. E., Kara, B. Y. (2018). Post-disaster assessment routing problem. Transportation research part B: methodological, 116, 76–102.
39. Özdamar, L., and Ertem, M. A. (2015). Models, solutions and enabling technologies in humanitarian logistics. European Journal of Operational Research, 244(1), 55–65.
40. Qi, J., Song, D., Shang, H., Wang, N., Hua, C., Wu, C., ... and Han, J. (2016). Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake. Journal of Field Robotics, 33(3), 290–321.
41. Rabta, B., Wankmüller, C., Reiner, G. (2018). A drone fleet model for last-mile distribution in disaster relief operations. International Journal of Disaster Risk Reduction, 28, 107–112.
42. Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., and Sarazzi, D. (2011). UAV photogrammetry for mapping and 3d modeling–current status and future perspectives. International archives of the photogrammetry, remote sensing and spatial information sciences, 38(1), C22.
43. Sato, Y., Ozawa, S., Terasaka, Y., Kaburagi, M., Tanifuji, Y., Kawabata, K., ..., Torii, T. (2018). Remote radiation imaging system using a compact gamma-ray imager mounted on a multicopter drone. Journal of Nuclear Science and Technology, 55(1), 90–96.
44. Schneider, M., Stenger, A., Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. Transportation Science, 48(4), 500–520.
45. Shavarani, S. M. (2019). Multi-level facility location-allocation problem for post-disaster humanitarian relief distribution. Journal of Humanitarian Logistics and Supply Chain Management.

46. Shohet, I. M., Levi, L. A. D. T., Levy, R., Salamon, A., Vilnay, O., Ornai, D., . . . , Levi, S. S. O. (2015). Analytical-Empirical Model for the Assessment of Earthquake Casualties and Injuries in a Major City in Israel–The Case of Tiberias. Contract, 3, 9618.
47. Sokat, K. Y., Dolinskaya, I. S., Smilowitz, K., Bank, R. (2018). Incomplete information imputation in limited data environments with application to disaster response. European Journal of Operational Research, 269(2), 466–485.
48. Yamazaki, F., Kubo, K., Tanabe, R., Liu, W. (2017, July). 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.