

Article

Complete Coverage Path Planning for a Multi-UAV Response System in Post-Earthquake Assessment

Arman Nedjati ^{1,*}, Gokhan Izbirak ¹, Bela Vizvari ¹ and Jamal Arkat ²

¹ Department of Industrial Engineering, Eastern Mediterranean University, Famagusta, North Cyprus, via Mersin 10, Famagusta 99450, Turkey; gokhan.izbirak@emu.edu.tr (G.I.); bela.vizvari@emu.edu.tr (B.V.)

² Department of Industrial Engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj 15175-66177, Iran; j.arkat@uok.ac.ir

* Correspondence: arman.nedjati@cc.emu.edu.tr; Tel.: +90-392-630-1055

Academic Editor: Huosheng Hu

Received: 10 October 2016; Accepted: 17 November 2016; Published: 2 December 2016

Abstract: This paper presents a post-earthquake response system for a rapid damage assessment. In this system, multiple Unmanned Aerial Vehicles (UAVs) are deployed to collect the images from the earthquake site and create a response map for extracting useful information. It is an extension of well-known coverage path problem (CPP) that is based on the grid pattern map decomposition. In addition to some linear strengthening techniques, two mathematic formulations, 4-index and 5-index models, are proposed in the approach and coded in GAMS (Cplex solver). They are tested on a number of problems and the results show that the 5-index model outperforms the 4-index model. Moreover, the proposed system could be significantly improved by the solver-generated cuts, additional constraints, and the variable branching priority extensions.

Keywords: coverage path planning; rapid damage assessment; UAV monitoring; post-earthquake response

1. Introduction

Rapid damage assessment in post-earthquake situations plays an important role in the early response phase activities (i.e., evacuation of injured individuals, debris collection, and relief distribution). The ground-based post-earthquake inspection is extremely time-consuming, and unhelpful in severely damaged areas; therefore, at present, aerial systems are widely used for investigations. It is highly desirable for densely populated urban areas to have a pre-planned immediate and automated post-disaster mapping and monitoring system. In the past decade, scientists have made an attempt to improve high-resolution satellite imaging and laser scanning systems to evaluate the disaster damage and loss [1–8]. However, the satellite systems have many limitations for an efficient post-disaster imaging, such as weather conditions (cloudy or dust whirls), time constraints for acquiring images and uplinking the acquisition plan to the satellite, delay in satellite data delivery after collection, etc. [9,10]; therefore, much attention has been focused on utilizing small unmanned aerial vehicles (UAVs) for post-disaster mapping [11–13].

In recent times, UAVs have become a popular entertaining tool globally. UAVs of various sizes are available on the market that can carry film or photographic cameras for different applications. The classification of UAVs according to their classes, distinct applications, and characteristics is beyond the scope of this article, though interested readers can refer to earlier studies [13,14]. Moreover, Colomina and Molina [15] have reviewed significant applications of UAVs, such as imaging and remote sensing. This study focuses on the mapping applications of small UAVs in post-earthquake situations. In a UAV-mapping system, consecutive overlapping aerial images taken by the UAVs are processed to obtain a complete map or to extract useful information. In earthquake situations, the

information, such as building and bridge destructions, road blockages, and population relocation, aids the managers in organizing a more effective post-earthquake response system. Within the first 30 min, the post-earthquake survival rate is 91%, decreasing to 36.7% by the second day [11]. Hence, in addition to accuracy, time is also an important factor in post-earthquake mapping. Based on this fact, it can be said that an optimal solution to the UAV coverage path-planning problem with a time-dependent objective function will have a great effect on the efficiency of the response system.

Coverage path problem (CPP), can be considered as a variant of Vehicle Routing Problem (VRP). In VRP, a fleet of vehicles start and end their tour at a single depot while visiting all nodes on the route. Unlike the classical VRP that considers a single main depot, in multi depot vehicle routing problem (MDVRP), there are several depots and the customers can be served from any of the depots. MDVRP has widely been studied in the literature and the interested readers may refer to the extensive survey of Montoya-Torres et al. [16]. Based on the literature the UAV routing problem is also formulated as a VRP model where additional constraints need to be added to reflect the characteristics of the problem. UAV routing problem has attracted much attention in the past decade for different applications [17,18]. CPP, tries to find an appropriate path for robots while covering the pre-defined nodes. The problem has a wide range of applications in automated harvesting, vacuum cleaning, mapping, demining, monitoring, etc. It is worth mentioning that CPP is different from the Covering Tour Problem (CTP). In CTP there are two types of nodes, those that must be visited by the tour and those that must be covered i.e., it lies within a pre-defined distance from a vertex of the tour [19]. Choset [20] and Galceran and Carreras [21] conducted comprehensive surveys on CPP methods, algorithms, and recent advances.

Immediately after an earthquake, the damage and location of afflicted people can be identified by imaging and then processing the obtained images. Since time factor is very important in post-earthquake response phase, UAVs are considered to be reliable for this task. Earthquake is not the only disaster that can be monitored by UAVs, but it is the most suitable one for imaging due to absence of information about the condition of each construction over the entire area. The imaging application of UAV based on CPP necessitates further considerations. One important issue is the direction of the path while passing through a node. Another important issue that arises due to earthquake situation is that, unlike the routing problem, the maximum mission time of vehicles must be minimized. UAVs start their route from a base and end their route at the same base for refueling or at the mission completion. This study extends the CPP for an application in the post-earthquake rapid damage assessment. The CPP features on a grid-based map are adopted on a multi-depot multi-tour vehicle routing problem for an emergency situation.

Zelinsky et al. [12] were the first to investigate the CPP on a robot, starting from an origin and ending at a goal point to minimize the length, energy consumption, and travel time. Later Carvalho et al. [22] proposed an algorithm for a mobile robot in an industrial environment such that the obstacles were not pre-specified on the map. The subject of path planning has attracted a lot of attention due to the recent development of UAV systems. Li et al. [23] studied an exact cellular decomposition method for UAV path planning in a polygon region. For the purpose of precision agriculture mapping, Barrientos et al. [24] performed an experiment using an integrated tool. Initially, they subdivided the polygon area and then conducted path planning for each subarea using a multi-UAV system. Torres et al. [25] presented a path planning algorithm for a single UAV with the aim of reducing battery usage and minimizing the number of turns, coping with both convex and non-convex regions. In a study by Wang et al. [26], an algorithm was used that minimized the consumed energy by the UAV for covering 3D terrain. In addition, a distinct study [27] introduced the multi-robot boundary coverage problem with the application of inspection of blade surfaces inside a turbine. Furthermore, Galceran and Carreras [21] conducted a comprehensive survey on CPP.

Most CPP algorithms are based on boustrophedon path strategy. In this strategy, the back and forth motion in a sweep direction of a polygon tries to cover an area with a minimum number of turns. Another approach divides the decomposing area into sub-areas and then finds the visiting sequence of

the sub-areas in a sweep direction base [28], describing how to find the optimal sweep direction for robots in a polygon. Huang and York claimed that the minimization of the number of turns leads to the most efficient solution (Figure 1). Li et al. [23] showed that the path with a lower number of turning motions is a more efficient coverage path for UAVs with regards to energy, distance, and travel time.

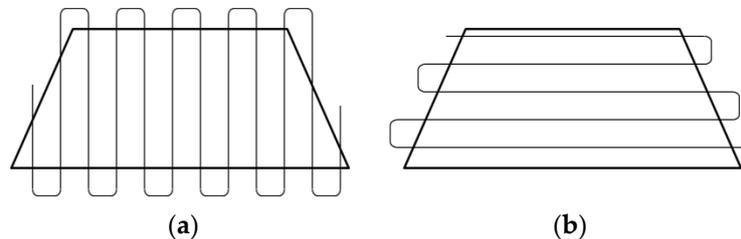


Figure 1. The right polygon is covered by lesser waste (lesser number of turns) than the left one. (a) The vertical direction with 11 turns; (b) The horizontal direction with 4 turns.

Furthermore, the map decomposition and sweep direction method were applied for a multi-UAV system [29]. Due to lack of time in a post-earthquake situation, using a multi-UAV system is potentially more beneficial and practical. Avellar et al. [30] studied a multi-UAV CPP in which the nodes were pre-defined corresponding to the boustrophedon motion. Similar to earlier studies, they decomposed the region into several sub-regions and used a VRP model to find the optimal path. However, the travel time between the sub-areas and the base to a region, as well as the possibility of turning back from the middle of a region to the base for refueling, were neglected.

Although numerous studies have been conducted on the sweep direction method, path covering based on grid-based decomposition has not yet received much attention. In a study by Wang and Li [31], the desired region was divided into grids and the amount of information collected by UAV and the path-length were maximized. According to their algorithm, passing through the center of a grid is equivalent to covering the grid; however, to have a complete image of the grid, the covering is considered as entering a grid from one side and exiting from the opposite side. Therefore, the turning motions in the grids are not considered as a part of the covered area (Figure 2). The grid size can vary depending on the altitude of the UAVs and focal length of the camera; however, by increasing the number of grids the complexity of the problem will increase.

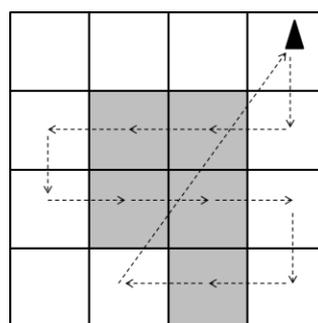


Figure 2. The gray cells indicate the cells that have been covered by the unmanned aerial vehicle (UAV), according to entrance and exit from the opposite sides. The picture depicts a feasible solution of grid decomposition-based coverage path problem (CPP).

In this study, it is assumed that the number of available UAVs, potential location of UAV bases, and the possible number of open bases are based on the restrictions prescribed by the decision maker. In addition, the total mission time limit should be specified earlier according to the endurance of UAVs. We have tried to address the following questions in the proposed post-earthquake mapping problem: (1) which potential UAV base must be open; (2) how many UAVs among all the available

ones should be assigned to each open base; (3) what must be the passing sequence of squares in each route; and (4) what is the minimum time required for covering all residential squares with the existing limitations. The proposed problem has a Minimax objective function that minimizes the maximum traveling time of each UAV in order to find the fastest possible mapping schedule. Each UAV belongs to a UAV base from where its mission starts and ends, and battery replacement or refueling is performed at the same base. Each UAV's mission time cannot exceed a maximum flight time (MT) due to its endurance. To the best of our knowledge, this problem has not been studied thus far, and the formulations presented here are the first mixed integer linear programming (MILP) mathematical models of the grid-based CPP problem. We call this problem multi-UAV multi-tour location coverage path planning (MUTLCP). The application of this model is a post-earthquake mapping system due to the Minimax objective function, and a grid-based decomposition of the residential areas.

The remainder of this study is organized as follows: Section 2 represents the problem definition and two 4-index and 5-index mathematical models. Section 3 contains the extensions and techniques used to improve the performance of the branch and cut algorithm. Section 4 includes the test problems and computational results, and Section 5 contains conclusions and future prospects of CPP.

2. Problem Definition and Mathematical Formulations

Associated with the MUTLCP problem, the sets, parameters, and decision variables are defined as follows (in Table 1):

Table 1. The definition of sets, parameters, and variables.

<i>Sets:</i>	
S_C	The set of residential squares that must be covered
S_D	The set of potential AUV base (depot) squares
S_E	The set of nonresidential squares
U	The set of available UAVs
L	The set of routes
R	The set of moving directions from one square to the other
<i>Parameters:</i>	
t_{ij}	The travel time from square i to j
d_{ij}	The distance between square i and j
TD	Total number of UAV bases (depots) that asked to be open
MT	The possible flight time of each UAV
<i>Variables:</i>	
w_{iju}^r	Binary decision variable equals to 1, if UAV u exists from r side of square i to square j in route l , and 0 otherwise
w_{ij}^{ul}	Binary decision variable equals to 1, if UAV u travels from square i to square j in route l , and 0 otherwise
v_{ij}^{ul}	Positive integer variable determines how many nodes are left to cover in route l by UAV u , after visiting i and just before visiting j
y_i	Binary variable equals to 1, if potential depot i is open and 0 otherwise
x_{ij}^r	Binary variable equals to 1, if a UAV travels from r side of square i to square j in route l
p_{iu}	Binary variable equals to 1, if UAV u belongs to depot i and 0 otherwise
g_{iu}^r	Binary variable equals to 1, if a UAV enters from r side of i and exits from the opposite side in route l
g_i^r	Binary variable equals to 1, if a UAV enters from r side of i and exits from the opposite side in route l
T	The maximum mission time of UAVs

The map of earthquake struck region was decomposed into equal sized squares, according to the UAV's camera footprint. The camera footprint (CF) depends on the camera sensor size (CSS), lens focal length (FL), and flight altitude (FA). Figure 3 shows the camera footprint, which can be calculated as follows.

$$CF = FA \frac{CSS}{FL} \quad (1)$$

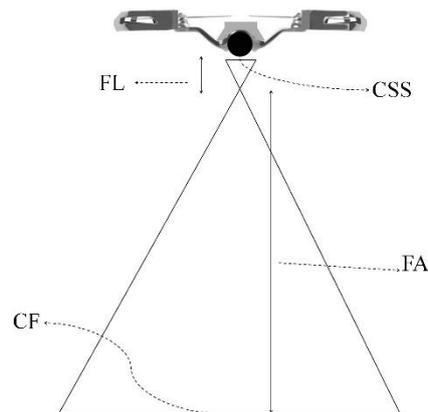


Figure 3. Camera footprint.

The camera plays a very important role in CPPs especially for a post-earthquake situations. Despite the resolution demanded by the decision maker, the focal length of the lens and the sensor size of the camera have a great impact on the mapping schedule length. A smaller focal length with a larger sensor size ends up in a larger camera footprint (Equation (1)) and consequently a lower number of squares in the map and schedule length. Also, the horizontal flight disposition is preferred for covering residential areas by a sequence of overlapping images, although the new UAV systems allow changes in the viewing angle of the installed camera.

The ground sampling distance (GSD) is a scale of resolution in aerial photogrammetry. It is the distance between two successive pixel centers on the ground. The smaller value of the GSD leads to the higher resolution of the image and more differentiable details. At a fixed focal length, decreasing the altitude results in a smaller GSD, and at a fixed flight altitude, by increasing the focal length, the GSD value will decrease. In an earthquake situation, the GSDs of several meters may suffice for damage assessment of non-urban regions. However, it is quiet insufficient when analyzing highly populated earthquake-struck cities and detailed images of at least 1 m GSD are essential [32].

While the geographical mapping might not require a quick scheduling process, in post-earthquake mapping, time factor has a great importance. Since the use of UAVs for mapping is in its early days, there are potential improvements and considerable innovations related to unseen problems and technical challenges.

To ease the image processing, an overlap from each side is essential; thus, if the overlap is $\alpha\%$ from each side, then the square edge must be equal to $(1 - 2\alpha)\%$ of the camera footprint length. The overlap is required to discover the common details among the images, however; the minimum and maximum overlaps depend on the post-image-processing software. Considering the fact that image-processing discussion is beyond the scope of this study, and there is no universally accepted overlap standard [33], we only mention that the overall overlap for a specified-path mapping must be at least 20% of the image width [34].

The MUTLCP is defined on an undirected graph $G(S, A)$, where $S = S_O \cup S_D$ is a set of squares and $A = \{(i, j) | i, j \in S\}$ is the set of arcs. S_D is a set of potential depot squares and $S_O = S_C \cup S_E$, where S_C is the set of residential squares that must be covered and S_E is the set of empty squares. Non-negative travel time, t_{ij} , and travel distance, d_{ij} , are associated with each arc with a triangular inequality (i.e., $t_{ik} + t_{kj} \geq t_{ij}$). Moreover, the set of homogenous UAVs is denoted by $U = \{1, 2, 3, \dots, u\}$ and the set of routes by $L = \{1, 2, 3, \dots, l\}$. There is a set $R = \{Left, Right, Up, Down, Direct\}$ that specifies the leaving direction form one square to another. All the squares belonging to S_C must be covered, at least once, such that the UAV enters and exists from opposite sides (i.e., Right-Left, Left-Right, Up-Down, and Down-Up). Among the $|S_D|$ number of potential depots, TD number of them can be opened, according to the restrictions prescribed by the decision maker. The flight time limitation for each UAV is specified by MT . All UAVs start and end their tours at the same base.

Because time plays a vital role in the post-disaster response phase against cost, the proposed model must minimize the maximum mission time of all UAVs.

When M is a sufficiently large positive number, the 4-index and 5-index models of MUTLCP are presented as follows:

2.1. 4-Index Formulation

Model:

$$\text{Min } T \tag{2}$$

The objective Equation (1) is to minimize the maximum mission time of UAVs. This objective function leads to the minimum mapping schedule length for all UAVs. The MUCLTPP's constraints are as follows:

$$\text{S.t. } \sum_{i \in S_D} y_i = TD \tag{3}$$

$$\sum_{i \in S_D} p_{iu} = 1 \quad \forall u \in U \tag{4}$$

Equation (3) ensures that the total number of open bases is TD , and Equation (4) imposes that each UAV has to be allocated to one depot only.

$$\sum_{i \in S} \sum_{u \in U} w_{ij}^{ul} \leq 1 \quad \forall j \in S, i \neq j, l \in L \tag{5}$$

Equation (5) ensures that each square is passed at most once in each route.

$$\sum_{j \in S} w_{ij}^{ul} - \sum_{j \in S} w_{ji}^{ul} = 0 \quad \forall i \in S, i \neq j, l \in L, u \in U \tag{6}$$

Equation (6) guarantees that if a UAV enters a square then it must leave that square. In other words, the total number of entrances to a square is equal to the total number of exits from that square.

$$\sum_{j \in S_O} \sum_{l \in L} \sum_{u \in U} w_{ij}^{ul} - My_i \leq 0 \quad \forall i \in S_D \tag{7}$$

Equation (7) assures that the UAVs exit from an open base only. This constraint forces the w_{ij}^{ul} variable to take the value of zero if the base is close ($y_i = 0$).

$$v_{ij}^{ul} - Mw_{ij}^{ul} \leq 0 \quad \forall i, j \in S, i \neq j, u \in U, l \in L \tag{8}$$

Equation (8) assigns zero to v_{ij}^{ul} variables according to the movement variable of w_{ij}^{ul} .

$$\sum_{i \in S} v_{ij}^{ul} - \sum_{i \in S} v_{ji}^{ul} - \sum_{i \in S} w_{ij}^{ul} = 0 \quad \forall j \in S_O, i \neq j, u \in U, l \in L \tag{9}$$

Equation (9) indicates that if a UAV pass through square $i \in S_O$, then the remaining number of the nodes in route l will decrease by one.

$$\sum_{i \in S} \sum_{j \in S, j \neq i} \sum_{l \in L} t_{ij} w_{ij}^{ul} \leq MT \quad \forall l \in L \tag{10}$$

Equation (10) ensures that the duration of each route does not exceed the UAV's flight time limitation MT .

$$\sum_{i \in S} \sum_{j \in S, j \neq i} \sum_{l \in L} t_{ij} w_{ij}^{ul} + \sum_{i \in S_D} \sum_{j \in S_O} \sum_{l \in L} ST w_{ij}^{ul} \leq T \quad \forall u \in U \tag{11}$$

The value of T which is the maximum mission time of UAVs is determined by Equation (11).

$$\sum_{j \in S_D} \sum_{l \in L} w_{ij}^{ul} - Mp_{iu} \leq 0 \quad \forall i \in S_D, u \in U \quad (12)$$

$$\sum_{i \in S_D} \sum_{j \in S_D} \sum_{u \in U} w_{ij}^{ul} \leq 1 \quad \forall l \in L \quad (13)$$

Equation (12) indicates that UAVs can only leave their own depot and Equation (13) eliminates the illegal routes.

$$\sum_{l \in L} \sum_{r \in R, r \neq \text{Direct}} \sum_{j \in S} x_{ij}^{lr} = 0 \quad \forall i \in S, d(i, j) > dm \quad (14)$$

Equation (14) prevents the formation of arcs with the length more than dm for the horizontal and vertical directions. dm is the distance between the center of two consecutive squares; the length of horizontal or vertical arcs cannot exceed it.

$$\sum_{j \in S} (x_{ij}^{lr} + x_{ji}^{lr}) - 2g_i^{lr} \geq 0 \quad \forall i \in S_C, i \neq j, l \in L, r \in R | r \neq \text{Direct} \quad (15)$$

Equation (15) specifies that the value of g_i^{lr} can be one if the entrance and exit are in the same horizontal or vertical directions.

$$\sum_{l \in L} \sum_{r \in R | r \neq \text{Direct}} g_i^{lr} \geq 1 \quad \forall i \in S_C \quad (16)$$

$$\sum_{r \in R} x_{ij}^{lr} - \sum_{u \in U} w_{ij}^{ul} = 0 \quad \forall i \neq j, j \in S, l \in L \quad (17)$$

Equation (16) ensures that all residential squares must be covered by a UAV in either horizontal or vertical directions and Equation (17) defines the value of variable x_{ij}^{lr} .

$$x_{ij}^{l \text{ Up}} = 0 \quad \forall i, j \in S | i - 1 \leq j, l \in L \quad (18)$$

$$x_{ij}^{l \text{ Down}} = 0 \quad \forall i, j \in S | i + 1 \geq j, l \in L \quad (19)$$

$$x_{ij}^{l \text{ Left}} = 0 \quad \forall i, j \in S | i \neq j + 1, l \in L \quad (20)$$

$$x_{ij}^{l \text{ Right}} = 0 \quad \forall i, j \in S | i \neq j - 1, l \in L \quad (21)$$

Equations (18)–(21) imposes the value of zero for the wrong horizontal and vertical directions. For example Equation (18) ensures that the travel in the upward (*up*) direction from i to j is zero unless $i - 1$ is less than or equal j . The travel from one square to its right square must be impossible for the *up*, *down*, and *left* directions and it is shown in Equation (21).

$$w_{ij}^{ul}, x_{ij}^{lr} \in \{0, 1\} \quad \forall i, j \in S, u \in U, l \in L, r \in R \quad (22)$$

$$y_i, p_{iu}, g_j^{lr} \in \{0, 1\} \quad \forall i \in S_D, j \in S_C, l \in L, r \in R \quad (23)$$

Finally, Equations (22) and (23) are used as the integrality constraints.

2.2. 5-Index Formulation

The 5-index model development is presented in the same structure of the 4-index formulation as follows:

$$\text{Min } T \text{ (1) S.t (2), (3), and}$$

$$\sum_{i \in S} \sum_{r \in R} \sum_{u \in U} w_{iju}^{lr} \leq 1 \quad \forall j \in S, i \neq j, l \in L \quad (24)$$

$$\sum_{j \in S} \sum_{r \in R} w_{iju}^{lr} - \sum_{j \in S} \sum_{r \in R} w_{jiu}^{lr} = 0 \quad \forall i \in S, i \neq j, l \in L, u \in U \quad (25)$$

$$\sum_{j \in S_D} \sum_{l \in L} \sum_{u \in U} \sum_{r \in R} w_{iju}^{lr} - My_i \leq 0 \quad \forall i \in S_D \quad (26)$$

$$v_{ij}^{ul} - \sum_{r \in R} Mw_{iju}^{lr} \leq 0 \quad \forall i \& j \in S, i \neq j, u \in U, l \in L \quad (27)$$

$$\sum_{i \in S} v_{ij}^{ul} - \sum_{i \in S} v_{ji}^{ul} - \sum_{i \in S} \sum_{r \in R} w_{iju}^{lr} = 0 \quad \forall j \in S_D, i \neq j, u \in U, l \in L \quad (28)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} \sum_{l \in L} \sum_{r \in R} t_{ij} w_{iju}^{lr} \leq MT \quad \forall l \in L \quad (29)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} \sum_{l \in L} \sum_{r \in R} t_{ij} w_{iju}^{lr} + \sum_{i \in S_D} \sum_{j \in S_D} \sum_{l \in L} \sum_{r \in R} ST w_{iju}^{lr} \leq T \quad \forall u \in U \quad (30)$$

$$\sum_{j \in S_D} \sum_{l \in L} \sum_{r \in R} w_{iju}^{lr} - Mp_{iu} \leq 0 \quad \forall i \in S_D, u \in U \quad (31)$$

$$\sum_{i \in S_D} \sum_{j \in S_D} \sum_{u \in U} \sum_{r \in R} w_{iju}^{lr} \leq 1 \quad \forall l \in L \quad (32)$$

$$\sum_{l \in L} \sum_{u \in U} \sum_{j \in S} \sum_{r \in R, r \neq \text{Direct}} w_{iju}^{lr} = 0 \quad \forall i \in S, d(i, j) > dm \quad (33)$$

$$\sum_{j \in S} (w_{iju}^{lr} + w_{jiu}^{lr}) - 2g_{iu}^{lr} \geq 0 \quad \forall i \in S_C, i \neq j, l \in L, u \in U, r \in R | r \neq \text{Direct} \quad (34)$$

$$\sum_{l \in L} \sum_{r \in R | r \neq \text{Direct}} \sum_{u \in U} g_{iu}^{lr} \geq 1 \quad \forall i \in S_C \quad (35)$$

$$w_{iju}^{l \text{Up}} = 0 \quad \forall i, j \in S | i - 1 \leq j, l \in L, u \in U \quad (36)$$

$$w_{iju}^{l \text{Down}} = 0 \quad \forall i, j \in S | i + 1 \geq j, l \in L, u \in U \quad (37)$$

$$w_{iju}^{l \text{Left}} = 0 \quad \forall i, j \in S | i \neq j + 1, l \in L, u \in U \quad (38)$$

$$w_{iju}^{l \text{Right}} = 0 \quad \forall i, j \in S | i \neq j - 1, l \in L, u \in U \quad (39)$$

$$w_{iju}^{lr} \in \{0, 1\} \quad \forall i \& j \in S, u \in U, l \in L, r \in R \quad (40)$$

$$y_i, p_{iu}, g_{ju}^{lr} \in \{0, 1\} \quad \forall i \in S_D, j \in S_C, l \in L, r \in R, u \in U \quad (41)$$

Equations (24)–(41) are the same equations as Equations (5)–(16) and Equations (18)–(23) in the 4-index formulation. In this problem, the size of $|L|$ is problem dependent, and an increase in the number of routes results in an exponential increase in the execution time of the problem. The minimum size of $|L|$ equals to the size of $|U|$; the maximum size can be calculated according to any feasible solution that the problem solver decides. However, it is preferable to use solutions that connect the squares (horizontally or vertically) with minimum turns.

3. Methodology

The 4-index and 5-index models were coded in GAMS and the Cplex solver was selected. To improve the algorithm and make it more effective, some extensions and techniques were found to be appropriate for our problem. Branch-and-Bound is the main algorithm used for solving MIP problems in Cplex. Using well-known techniques based on the model's features can effectively reduce

the execution time of the problem and improve the performance of MIP for effective collaborative decision-making in disaster management. The following subsections shortly describe the techniques that were effective for the models.

3.1. Solver Generated Cuts and Heuristics

In the early sixties, Gomory [35] introduced the procedure of fractional cuts. The procedure was considered impractical until a study [36] proved that Gomory’s cuts can be effectively distributed over different branches of the search tree in a branch and cut framework for a mixed 0–1 problem. In the present time, most optimizers use Gomory’s cuts for MILP problems. In the Cplex optimizer, the `fraccuts` option decides whether or not Gomory’s fractional cuts should be generated for the problem; and the parameters 0, 1 and 2 control the cuts to be generated automatically, moderately, and aggressively, respectively. In the case of lack of progress in the best bound, one effective solution uses more aggressive levels of cut generation to further tighten the formulation [37]. As initial observations confirm the lack of progress in the best-found-bound, parameter 2 (aggressive cut generation) was found to be more effective for the proposed models. According to the structure of the mathematical formulations, cover cuts are also useful. Therefore, Gomory’s fractional cuts and cover cuts were aggressively generated for the models.

GAMS (General Algebraic Modeling System) is a high level modeling language designed for programming the linear, nonlinear, and mixed integer optimization models. GAMS can support different forms of linear and nonlinear programming problems. The system has a language compiler and an integrated high-performance commercial solver, e.g., Cplex. Cplex is a famous commercial optimizer that can solve linear, mixed integer, quadratic, and quadratically constrained programming problems. GAMS provides access to the Cplex solver with a variety of options to interact with it and improve the solution procedure. Cplex uses several alternative algorithms such as primal simplex, dual simplex, barrier, or sifting algorithms. For mixed integer programming problems, Cplex uses the branch-and-cut algorithm.

Cplex takes advantage of different heuristic techniques to avoid tree pollution, increasing the diversity of the branch-and-bound search, achieving integer feasible solution in shorter time, and fastening the final optimality. Based on the characteristics of the proposed problem, two heuristic features from the different heuristic procedures offered by Cplex were utilized, i.e., local branching heuristic (LBHeur), and node heuristic (NHeur). The LBHeur was introduced by Fischetti and Lodi [38] and has been widely used by commercial optimizers. We can infer the following facts based on the ILOG Cplex explanations:

- LBHeur attempts to enhance new incumbents found throughout an MIP search. The LBHeur algorithm will be appealed only if a new incumbent is identified. In cases where several incumbents are identified at a single node, the last one will be considered.
- NHeur implements the techniques to find a feasible solution from the present node in branch and bound algorithm. The parameter `HeurFreq` restricts this offer. A positive number specifies the frequency of invoking NHeur (in the number of nodes).

3.2. Additional Constraints

User cuts or additional constraints are the polynomial constraints that strengthen the mathematical formulation and reduce the solution space. These cuts are not supposed to affect the feasible solutions space. The following constraints eliminate unfavorable fractional solutions and expedite the execution process:

$$\sum_{i \in S_D} \sum_{j \in S_O} v_{ij}^{ul} - \sum_{i \in S} \sum_{j \in S_O} \sum_{r \in R} w_{iju}^{lr} = 0 \quad \forall u \in U, l \in L \quad (42)$$

$$\sum_{i \in S_O} \sum_{j \in S_D} \sum_{u \in U} \sum_{l \in L} v_{ij}^{ul} = 0 \quad (43)$$

Equation (42) imposes that the total number of squares that are to be visited by UAV u in route l right after leaving the depot is equal to the total number of squares allocated to UAV u in route l . Equation (43) ensures that the value is equal to 0 just before ending the tour at the depot. Equations (9), (42) and (43) are adapted from a previous study [39].

$$p_{iu} - y_i \leq 0 \quad \forall i \in S_D, u \in U \quad (44)$$

Equation (44) indicates that if a UAV belongs to a depot, then the depot must be open.

$$\sum_{i \in S_D} \sum_{l \in L} \sum_{u \in U} \sum_{r \in R} \sum_{j \in S_O} w_{iju}^{lr} \leq |L| \quad (45)$$

Equation (45) ensures that the total number of arcs leaving the depots is less than or equal to the number of the available routes.

$$\sum_{i \in S_N} \sum_{l \in L} \sum_{u \in U} \sum_{r \in R | r \neq \text{direct}} \sum_{j \in S_N} w_{iju}^{lr} = 0 \quad (46)$$

According to the logic of the problem and the grid-based decomposed map, the UAVs must travel directly through the nodes of the S_E set, and it is applied in Equation (46). As mentioned earlier, only the residential squares must be covered by horizontal or vertical movements. In other words, in favor of the objective function value, and in order to find the minimal path, the UAVs should travel in the *direct* direction while passing through the nonresidential squares.

$$\sum_{r \in R} \sum_{u \in U} \sum_{i \in S_C} g_{iu}^{lr} - \sum_{i \in S} \sum_{j \in S_C} \sum_{u \in U} \sum_{r \in R} w_{iju}^{lr} \leq 0 \quad \forall l \in L \quad (47)$$

Equation (47) ensures that the number of times a UAV enters a square in each route is more than or equal to the number of times the square is covered in that route.

3.3. Variable Branching Priority

The variable branching priority based on the mathematical formulation's specifications is one of the most efficient techniques that speeds up the solution procedure and tightens the model. Without using this feature, the solver automatically decides the appropriate variable to be branch on. It is imperative to select a suitable priority among the existing variables depending on the variables' dependence, relevance, and significance in the proposed mathematical model. In this study there are 4-index and 5-index mathematical formulations of a single problem. In this study, 4-index and 5-index mathematical formulations were used for a single problem. Based on the model's characteristics and variables' significance, the following priorities were suggested:

- I. For the 4-index model, the variable branching priority order must be g_i^{lr} , x_{ij}^{lr} , w_{ij}^{ul} , v_{ij}^{ul} , p_{iu} and y_i .
- II. The branching priority of the 5-index model should be g_i^{lr} , v_{ij}^{ul} , w_{iju}^{lr} , p_{iu} , and y_i .

4. Computational Experiments and Results

In this section, some instances are designed to test the validity of the models and features proposed in Section 3. Based on the problem's characteristics, it is assumed that the number of available UAVs, number of possible open depot squares, and maximum limit for the total traveling time of each UAV are known. Each test instance was named in the X-Y-Z-L format, where X refers to the total number of squares in the map, Y refers to the total number of residential squares, Z represents the number of potential UAV bases, and L is the number of available UAVs. The test instances are shown in Figure 4. For instance, in problem 36-11-2-3 map, where 36 is the total number of squares, 11 squares are residential areas, two squares are potential depots, and three represents the available UAVs.

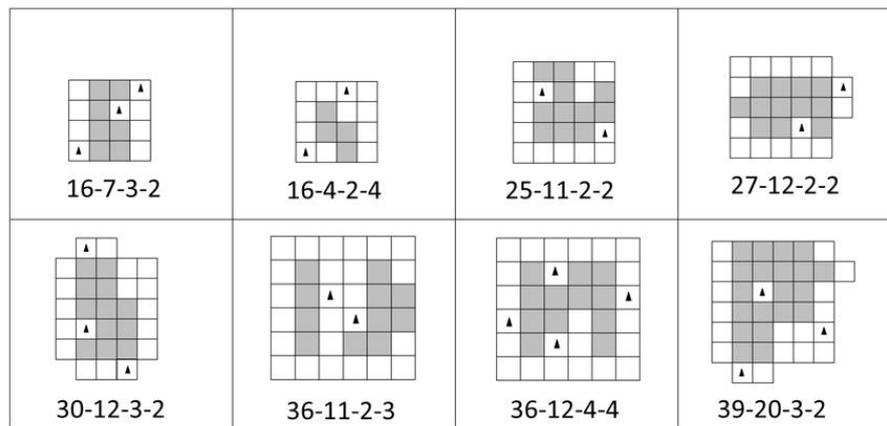


Figure 4. Test instance maps (the gray squares are assumed to be residential areas that should be covered by UAVs and squares with black triangles are potential UAV bases).

It is assumed that each small square side is one in terms of length unit and the speed of the UAVs is also one in terms of speed unit. The mathematical models were coded in GAMS 24.7.3 and solved using the Cplex 12.6.3.0 solver. The models were run on a laptop with the configuration of Intel® Core™ i7–2620M CPU 2.70 GHz, RAM 8.00 GB, and windows 10 Enterprise operating system.

To evaluate the efficiency of the techniques mentioned in Section 3, each technique was considered alone in an individual run at the first step. Subsequently, the combination of efficient ones was used. All the problems were solved in a 5-h time limit, irrespective whether a feasible or optimal solution was found or not. The reason for the 5-h time limit is that no significant improvement would occur by increasing this limit. The calculation time increases exponentially by increase in the number of squares or the number of routes. Tables 2 and 3 show the results of the 4-index and 5-index models, respectively, in a combination of techniques mentioned in Section 3.

According to the results presented in Tables 2 and 3, it can be concluded that the 5-index model shows better solutions in terms of CPU time and the absolute gap. Furthermore, as previously anticipated, the combination of the proposed LP strengthening techniques offered better solutions in a shorter amount of time. The CPU time of the 5-index problem 16-4-2-4 was 11,320 s in the original model, which decreased to 479 s by adding the techniques. In addition, for problem 36-12-4-4, a feasible solution was not obtained in the original model; however, with a combination of techniques, a solution with 6.3% relative gap was acquired. Furthermore, while no feasible solution was found for the original model of problem 39-20-3-2 in the 5-h time limit, the optimal solution was found under the influence of the techniques in 10,554 s. Figure 5 depicts the optimal solutions for problems 39-20-3-2 and 36-11-2-3 test instances as examples.

Table 2. Effects of techniques described in Sections 3.1–3.3 on the 4-index mathematical model.

4-Index Model															
Instance	Original Model			3.1, 3.2			3.1, 3.3			3.2, 3.3			3.1, 3.2, 3.3		
	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)
16-7-3-2	254	21.3	0	136	21.3	0	281	21.3	0	216	21.3	0	284	21.3	0
16-4-2-4	18,000	13.15	15.2	3850	13.15	0	4300	13.15	0	5400	13.15	0	3650	13.15	0
25-11-2-2	1400	24	0	310	24	0	343	24	0	162	24	0	84	24	0
27-12-2-2	18,000	35.98	35.1	1325	28	0	2104	28	0	1206	28	0	1618	28	0
30-12-3-2	12,587	26.32	0	2877	26.32	0	2754	26.32	0	3107	26.32	0	2639	26.32	0
36-11-2-3	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	NA	-
36-12-4-4	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	24.4	59.4
39-20-3-2	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	NA	-	15,669	36	0

Table 3. Effects of techniques described in Sections 3.1–3.3 on the 5-index mathematical model.

5-Index Model															
Instance	Original Model			3.1, 3.2			3.1, 3.3			3.2, 3.3			3.1, 3.2, 3.3		
	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)
25-5-3-3	888	21.3	0	467	21.3	0	782	21.3	0	159	21.3	0	129	21.3	0
16-4-2-4	11,320	13.15	0	1359	13.15	0	5527	13.15	0	650	13.15	0	479	13.15	0
25-11-2-2	850	24	0	286	24	0	186	24	0	226	24	0	48	24	0
27-12-2-2	18,000	30.82	30.9	966	28	0	18,000	30.24	30.1	570	28	0	559	28	0
30-12-3-2	6875	26.32	0	2472	26.32	0	3830	26.32	0	2958	26.32	0	1421	26.32	0
36-11-2-3	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	21.3	13.7	18,000	20.79	6.3
36-12-4-4	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	NA	-	18,000	22.82	49.2
39-20-3-2	18,000	NA	-	18,000	42.82	27.1	18,000	NA	-	18,000	36	11.7	10,554	36	0

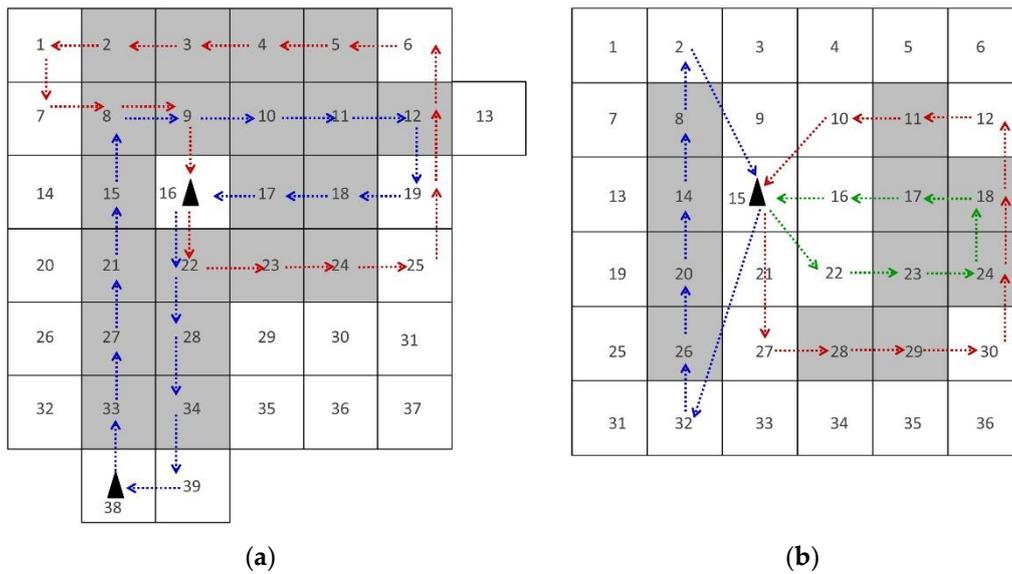


Figure 5. The optimal solutions of problems 39-20-3-2 and 36-11-2-3. (a) There are two available UAVs and each belongs to one base; (b) There are three available UAVs and base number 15 is selected to be open.

5. Conclusions and Future Prospects

A multi-UAV and multi-tour CPP problem based on the grid map decomposition was investigated. MUTLCP supports the application of post-earthquake CPP where high accuracy is required. Moreover, due to the speedy decrease in the survival rate in post-earthquake situations, the Minimax objective function was considered for the optimization problem. Two mathematical models of MUTLCP were presented and coded in GAMS, and the 5-index model outperformed the 4-index formulation in terms of solution quality and execution time. The models were improved by additional constraints and other techniques described in Section 3. On the basis of the results of the test instances, it can be affirmed that the combination of the proposed techniques sped up the execution procedure and gained solutions with a lower gap in a shorter amount of time. In the small size instance of 16-7-3-2, the combination of techniques decreased the calculation time by 85% for the 5-index model but increased by 7% for the 4-index formulation. The results depicted in Tables 2 and 3 show that the linear strengthening techniques are more effective for the 5-index model. Based on the results of Tables 2 and 3, the proposed techniques decreased the CPU time by 85%, 95%, 94%, 96%, and 79% for the first five instances of the 5-index model and decreased the time by 79%, 94%, more than 91%, and 79% for the corresponding five instances of the 4-index model. For the further three instances, while the solution is not available in the original models, by the intervention of the linear strengthen techniques optimal solutions or very good feasible solutions were found. Generally, by comparing the results (Tables 2 and 3), it is clear that the 5-index model performs much better than the 4-index formulation. Another conclusion based on the obtained results is that the solver generated cuts (Section 3.1) have less effect, and additional constraints (Section 3.2) have the maximum effect on decreasing the CPU time.

There are several interesting directions for further investigations. Some new valid inequalities can be added to the model to strengthen the MILP formulations. In addition, exact methods such as the lagrangian relaxation can be developed to find the optimal or near-optimal solutions to large real size problems. The most interesting subject for further investigations is designing a heuristic or meta-heuristic algorithm to solve large-scale problems of MUTLCP. Each randomly generated individual solution in a heuristic algorithm can be represented in such a way that each cell takes an integer <6 indicating the UAV direction (set R). Finally, one can extend the problem by reflecting the map rotation possibility to test the different covering directions.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments.

Author Contributions: Arman Nedjati conceived the problem, designed the mathematical models, and wrote the paper. Gokhan Izbirak, Bela Vizvari and Jamal Arkat revised the models and the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pesaresi, M.; Gerhardinger, A.; Haag, F. Rapid damage assessment of built-up structures using VHR satellite data in tsunami-affected areas. *Int. J. Remote Sens.* **2007**, *28*, 3013–3036. [[CrossRef](#)]
2. Chang, K.T.; Wang, E.H. Developing procedures for post-earthquake structural evaluation by laser scanning techniques. *Insight-Non-Destr. Test. Cond. Monit.* **2012**, *54*, 562–567. [[CrossRef](#)]
3. Szantoi, Z.; Malone, S.; Escobedo, F.; Misas, O.; Smith, S.; Dewitt, B. A tool for rapid post-hurricane urban tree debris estimates using high resolution aerial imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2012**, *18*, 548–556. [[CrossRef](#)]
4. Olsen, M.J.; Chen, Z.; Hutchinson, T.; Kuester, F. Optical techniques for multiscale damage assessment. *Geomat. Nat. Hazards Risk* **2013**, *4*, 49–70. [[CrossRef](#)]
5. Kashani, A.G.; Crawford, P.S.; Biswas, S.K.; Graettinger, A.J.; Grau, D. Automated Tornado Damage Assessment and Wind Speed Estimation Based on Terrestrial Laser Scanning. *J. Comput. Civ. Eng.* **2015**, *29*, 4014051. [[CrossRef](#)]
6. Jiang, S.; Friedland, C.J. Automatic urban debris zone extraction from post-hurricane very high-resolution satellite and aerial imagery. *Geomat. Nat. Hazards Risk* **2016**, *7*, 933–952. [[CrossRef](#)]
7. Janalipour, M.; Mohammadzadeh, A. Building Damage Detection Using Object-Based Image Analysis and ANFIS From High-Resolution Image (Case Study: BAM Earthquake, Iran). *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 1937–1945. [[CrossRef](#)]
8. Fernandez Galarreta, J.; Kerle, N.; Gerke, M. UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. *Nat. Hazards Earth Syst. Sci.* **2015**, *15*, 1087–1101. [[CrossRef](#)]
9. Al-Wassai, F.A.; Kalyankar, N.V. Major Limitations of Satellite images. Available online: <https://arxiv.org/ftp/arxiv/papers/1307/1307.2434.pdf> (accessed on 29 November 2016).
10. Grandoni, D. Advantages and Limitations of Using Satellite Images for Flood Mapping. Available online: http://www.copernicus.eu/sites/default/files/documents/User_uptake/Emergency_Events/Athens/03_Session_01_AdvantageAndLimitationsOfUsingSatelliteImagery.pdf (accessed on 29 November 2016).
11. Qi, J.; Song, D.; Shang, H.; Wang, N.; Hua, C.; Wu, C.; Qi, X.; Han, J. Search and Rescue Rotary-Wing UAV and Its Application to the Lushan Ms 7.0 Earthquake. *J. Field Robot.* **2016**, *33*, 290–321. [[CrossRef](#)]
12. Zelinsky, A.; Jarvis, R.A.; Byrne, J.C.; Yuta, S. Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot. In Proceedings of the International Conference on Advanced Robotics, Tokyo, Japan, 1–2 November 1993; pp. 533–538.
13. Nedjati, A.; Vizvari, B.; Izbirak, G. Post-earthquake response by small UAV helicopters. *Nat. Hazards* **2015**, *80*, 1669–1688. [[CrossRef](#)]
14. Kendoul, F. Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems. *J. Field Robot.* **2012**, *29*, 315–378. [[CrossRef](#)]
15. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [[CrossRef](#)]
16. Montoya-Torres, J.R.; López Franco, J.; Nieto Isaza, S.; Felizzola Jiménez, H.; Herazo-Padilla, N. A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **2015**, *79*, 115–129. [[CrossRef](#)]
17. Oberlin, P.; Rathinam, S.; Darbha, S. Today's Traveling Salesman Problem. *IEEE Robot. Autom. Mag.* **2010**, *17*, 70–77. [[CrossRef](#)]
18. Ho, H.-M.; Ouaknine, J. The Cyclic-Routing UAV Problem is PSPACE-Complete. In *Foundations of Software Science and Computation Structures*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 328–342.
19. Gendreau, M.; Laporte, G.; Semet, F. The Covering Tour Problem. *Oper. Res.* **1997**, *45*, 568–576. [[CrossRef](#)]
20. Choset, H. Coverage for robotics—A survey of recent results. *Ann. Math. Artif. Intell.* **2001**, *31*, 113–126. [[CrossRef](#)]

21. Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. [[CrossRef](#)]
22. De Carvalho, R.N.; Vidal, H.A.; Vieira, P.; Ribeiro, M.I. Complete coverage path planning and guidance for cleaning robots. In Proceedings of the 1997 IEEE International Symposium on Industrial Electronics, Guimaraes, Portugal, 7–11 July 1997; pp. 677–682.
23. Li, Y.; Chen, H.; Joo Er, M.; Wang, X. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics* **2011**, *21*, 876–885. [[CrossRef](#)]
24. Barrientos, A.; Colorado, J.; Cerro del, J.; Martinez, A.; Rossi, C.; Sanz, D.; Valente, J. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *J. Field Robot.* **2011**, *28*, 667–689. [[CrossRef](#)]
25. Torres, M.; Pelta, D.A.; Verdegay, J.L.; Torres, J.C. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Syst. Appl.* **2016**, *55*, 441–451. [[CrossRef](#)]
26. Wang, X.; Sun, T.; Li, D. Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles. *Electron. Lett.* **2016**, *52*, 699–701.
27. Easton, K.; Burdick, J. A Coverage Algorithm for Multi-robot Boundary Inspection. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation; Barcelona, Spain, 18–22 April 2005; pp. 727–734.
28. Huang, W.H.; York, N. Optimal Line-sweep-based Decompositions for Coverage Algorithms. In Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Seoul, Korea, 21–26 May 2001; pp. 27–32.
29. Maza, I.; Ollero, A. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In *Distributed Autonomous Robotic Systems 6*; Springer: Tokyo, Japan, 2007; pp. 221–230.
30. Avellar, G.; Pereira, G.; Pimenta, L.; Iscold, P. Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time. *Sensors* **2015**, *15*, 27783–27803. [[CrossRef](#)] [[PubMed](#)]
31. Wang, X.; Li, D. Coverage path planning for UAVs in unknown directional regions. *Int. J. Wirel. Mob. Comput.* **2015**, *8*, 285–293. [[CrossRef](#)]
32. Lemmens, M. Geo-information Technology—What It Is, How It Was and Where It Is Heading to. In *Geo-Information*; Springer: Dordrecht, The Netherlands, 2011; pp. 1–22.
33. Greenwood, F. How To Make Maps With Drones. In *Drones and Aerial Observation: New Technologies for Property Rights, Human Rights, and Global Development a Primer*; New America: Washington, DC, USA, 2015; pp. 35–48.
34. Falkner, E.; Morgan, D. *Aerial Mapping: Methods and Applications*; Lewis Publishers: Boca Raton, FL, USA, 2002.
35. Gomory, R.E. *An Algorithm for the Mixed Integer Problem*; RAND Corporation: Santa Monica, CA, USA, 1960.
36. Balas, E.; Ceria, S.; Cornuéjols, G.; Natraj, N. Gomory cuts revisited. *Oper. Res. Lett.* **1996**, *19*, 1–9. [[CrossRef](#)]
37. Klotz, E.; Newman, A.M. Practical guidelines for solving difficult mixed integer linear programs. *Surv. Oper. Res. Manag. Sci.* **2013**, *18*, 18–32. [[CrossRef](#)]
38. Fischetti, M.; Lodi, A. Local branching. *Math. Program.* **2003**, *98*, 23–47. [[CrossRef](#)]
39. Karaoglan, I.; Altıparmak, F. A memetic algorithm for the capacitated location-routing problem with mixed backhauls. *Comput. Oper. Res.* **2015**, *55*, 200–216. [[CrossRef](#)]

