Cooperated Unmanned Aerial Vehicle and Ground Vehicle in Humanitarian Operations: A Routing Problem

Maharani Rizki Larasati and Erny Rahayu Wijayanti *

Industrial Engineering Program, Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia, 55281

ABSTRACT

Disasters cause widespread destruction leading to significant loss of life and property. In such scenarios, timely and efficient disaster aid is crucial to reducing casualties and alleviating suffering. The delivery of aid and search for survivors in the aftermath of a disaster often face challenges such as damaged infrastructure and unsafe conditions, making it difficult for conventional rescue efforts. Unmanned aerial vehicle (UAV) is commonly used to solve this problem but as a disaster-prone country, Indonesian disaster relief and operations rarely use UAVs for assistance. However, due to their limited energy UAVs need to be supported by ground vehicles (GVs) to charge or replace batteries during missions. Thus, UAV and GV collaboration is necessary to perform this task. This research develops UAV and GV routing problem models using the depth-first search (DFS) method for pathfinding because it may search for feasible pathways and "backtrack" when nearby nodes are unavailable. This study represents the environment with a time-space network. The proposed model will be tested on a small-scale example and scaled up to a real-world disaster scenario, the 2010 Merapi eruption in Indonesia. The model demonstrates the effectiveness of UAV and GV collaboration in covering all target nodes with a limited of rendezvous points. The results of this study will contribute to the advancement of disaster relief efforts and provide valuable insights into the potential of UAV-GV collaboration for aid delivery in disaster scenarios.

ARTICLE INFO

Received January 31, 2023
Accepted March 3, 2023
Available online March 19, 2023

***Correspondence**

Erny Rahayu Wijayanti
erny.wijayanti@ugm.ac.id

Keywords:
disaster relief operation; unmanned aerial vehicle; ground vehicle; routing problem; depth-first search

1. Introduction

Several measures have been taken to improve the efficacy and productivity of disaster relief operations. As an example, consolidation among stakeholders (Vaillancourt, 2016). However, disasters can render facilities inaccessible. Infrastructure inaccessibility is a significant barrier to disaster assistance and operations. The inaccessible path, for instance, makes it difficult to evacuate people and deliver supplies to the impacted region.

There is a lot of potential for disasters in Indonesia, since Indonesia is home to some of the world's most active volcanoes, and one of them erupted in 2010. At least a third of a million of people were forced to flee their homes, and approximately 400 individuals lost their lives because of the disaster (Surono *et al.*, 2012). A pyroclastic flow, triggered by the eruption, travelled 17 kilometres to the South. Because of its open-conduit structure, the volcano's activity is hard to anticipate (Iguchi *et al.*, 2011). Volcanic unpredictability and associated dangers make mapping the area challenging. However, when a disaster has occurred, the map may be used to determine the extent of the damage quickly and accurately. However, satellite mapping takes too long in the aftermath of the disaster (American Red Cross, 2015).

Time is a significant constraint in disaster relief since it relates to the lives and losses of people. Especially after a natural disaster, it is essential to seek for survivors to reduce the number of casualties. However, the area is risky for humans to search for survivors once the disaster has occurred. In addition, the location is occasionally rendered inaccessible by the disaster's aftermath. In other countries, unmanned aerial vehicles or UAVs are often employed to reach inaccessible areas. Compared to land vehicles, UAVs are utilised due to their adaptability and speed. Hence, disaster relief operations become essential and time-limited to reduce loss. However, the employment of UAVs for disaster relief and operations in Indonesia has been rather limited. As a result, in-depth research on UAVs in Indonesia will be helpful for disaster relief efforts. In disaster relief efforts, unmanned aircraft play a variety of tasks, including risk assessment, logistical assistance, and search and rescue missions (American Red Cross, 2015). A UAV may do a survey task 50.4% faster than it would have taken a traditional approach, or more than half as fast (Volkmann, 2017). This proves that a UAV is suitable for the mission under time constraints. This is relevant to post-disaster activities like searching for survivors and repairing damaged infrastructure.

UAV have been used to do a variety of activities, including products delivery (Ha *et al.*, 2015; Dorling *et al.*, 2017) persistent intelligence surveillance and reconnaissance mission, and disaster relief operation (Chowdhury, 2017). UAVs may be used for a variety of tasks, e.g. operations for disaster aid. UAVs serve a variety of purposes as vehicles for disaster assistance and operation. In the four stages of a disaster—prevention, planning, reaction, and recovery—UAVs can be used (American Red Cross, 2015).

UAVs are useful for disaster relief and operation in a variety of situations, including the detection of collapsed buildings (Hua *et al.*, 2016), photogrammetric mapping (Cahyono & Zayd, 2018), providing geographic information systems for the location of facilities (Silva *et al.*, 2018), and disaster risk monitoring (Gomez & Purdie, 2016). The UAV can be used for search and rescue (SAR) in a disaster-recovery situation. According to Pólka *et al.* (2017), UAVs can be a useful technique for locating individuals in environments with challenging terrain. UAV's energy is nonetheless restricted. However, UAV was only able to fly a limited distance due to the limitation. Recent study blends UAV with GVs like GVs due to the constraint (Luo *et al.*, 2017; Luo *et al.*,2018). UAVs land at a ground vehicle, which may also be used to recharge batteries (Yu *et al.*, 2018) or switch out batteries while charging another battery (Luo *et al.*, 2018). Multiple types of studies have investigated the synchronisation of UAV and GV (Luo *et al.*, 2017; Luo *et al.*, 2018; Yu *et al.*, 2018; Ha *et al.*, 2018; Hassan *et al.*, 2022). Wei *et al.* (2020) proposed an integrated-location routing problem for depot selection and vehicle assignment. Hassan *et al.* (2022) presents a novel optimization model to distribute relief items to disaster-hit areas and the objective of the model is to optimize the location and the number of charging stations. The relative priority of locations where a preference is given to locations with higher priority levels become their consideration.

The combination of GV and UAV enables the UAV to have a larger coverage area and alters the trajectories of both vehicles. To guarantee that UAVs land on the GV at the appropriate moment, UAVs must operate concurrently with the GV. In comparison to a GV-only work, the completion time of a GV-UAV combo task can be lowered by 75% (Vaillancourt, 2016). Therefore, the trajectory of the GV and UAV is crucial. Multiple approaches have been devised to tackle the routing problem for the integration of both networks. Before a specific method can be used to solve an issue, the problem must first be stated. Some studies formulated the problem using integer programming (Ha *et al.*, 2018; Luo *et al.*, 2018; Ha *et al.*, 2018; Agatz *et al.*, 2018). A two-echelon cooperative routing issue necessitates the use of integer programming.

Heuristic is the typical approach to solving a given problem. Nevertheless, there are a number of heuristics methods employed in various ways to handle the problem. Ha *et al.* (2015) for instance, offered two heuristics approaches. The first technique, cluster first – route second, determines the UAV routes first and the GV routes using the UAV nodes provided. The alternate technique, route first – cluster second, involves determining the GV routes first and the UAV nodes second. Moreover, the route first – cluster second strategy is utilised by (Agatz *et al.*, 2018). At this study, GV serves not only as a mobile depot, but also as a provider for demand in the designated nodes. The issue is posed using integer programming and solved using heuristics based on greedy partitioning. The exact partitioning is contrasted to the outcome of greedy partitioning techniques. According to the research, accurate partitioning is preferable to greedy partitioning.

Exact partitioning is useful only for small-sized models and costs a considerable amount of processing effort despite producing superior results. Therefore, Luo *et al.* (2017) use evolutionary algorithms to construct a two-echelon routing issue for GV and UAV (GA). This investigation employs two sorts of heuristics. The first heuristic involves determining the UAV route before the GV path. The second heuristic is the exact opposite of the first. The outcome indicates that both heuristics are efficient and effective. Greedy Randomized Adaptive Search Procedure

(GRASP) is another heuristic technique used to tackle the problem. Ha *et al.* (2018), k-nearest neighbour, k-cheapest insertion, and random insertion are the three separate stages that make up the GRASP technique. In addition to heuristics, Wang *et al.* (2017) present an approach known as vehicle routing issue using UAV (VRPD). VRPD is like the standard vehicle routing issue, except it also accounts for unmanned aerial vehicles.

The objective of the cooperative routing issue for UAV and GV is to identify the best path that satisfies the constraints. Li and Garcia-Luna-Aceves (2006) used the depth-first search strategy to locate a plausible route. Extended depth first search (EDFS) beats other methods for solving multi-constrained paths. Duhamel *et al.* (2011) create an extended depth-first search and combine it with a greedy algorithm to solve the routing issue.

This research uses the 2010 Merapi eruption as the study case to seek for survivors to better comprehend how to utilise UAVs in disaster aid, especially in Indonesia. This research aims to tackle the routing issue for the combination of a GV and UAV. This research will serve as a roadmap for resolving additional UAV and GV routing issues, particularly in Indonesia. It is necessary to choose the best approach for the routing issue of UAV and GV based on prior study after problem identification. The goal of the research is to identify the best path for ground vehicle and unmanned aircraft system collaboration.

2. Methodology

This research focuses on determining the routing issue for a cooperative unmanned aerial system and ground vehicle for humanitarian logistics. The location of the barracks and a list of the regions damaged by the Merapi eruption in 2010 serve as the subject of this study. Affected regions will be transformed into optional rendezvous and target nodes. Along with the information gleaned from the previous catastrophe, further information is required for the operation of ground vehicles and unmanned aerial vehicles, such as the endurance of unmanned aerial vehicles and their speeds. Secondary data from a literature review was used to create the data.

2.1 Mathematical model

The objective function of the model is to reduce the amount of time required to complete the activity. The model will be implemented in two types of cases, that are small cases to validate the model and the case of Merapi eruption in 2010 for the real application. The basic framework of this research is the notion of workspace. There are two different forms of workspace: airspace and ground space. Airspace is where UAVs fly, and only UAVs can enter to that area. Airspace is comprised of air nodes, which must be traversed by the UAV. Ground space is an optional halting point for GV and the location where UAV batteries are swapped. In other words, both GVs and UAVs have access to the ground. Figure 1 illustrates the distinction between air space and ground space.

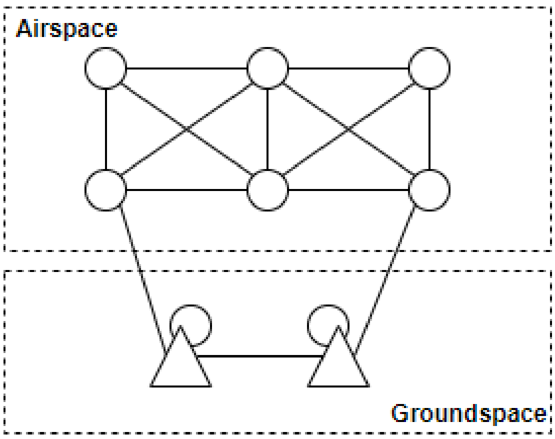


Figure 1. Airspace and groundspace illustration (Luo *et al.*, 2018)

There are known to be arcs connecting two workspaces. These arcs provide UAV operation in a single workspace. However, GV can only reach ground-level space (represented in the triangle symbol). This research will utilise this network as its foundation. This research will construct a time-space network for subsequent development.

This time-space network includes a time parameter in which each GV and UAV movement is recorded with a time-step. Figure 2 illustrates the predicted outcome of this time-space network.

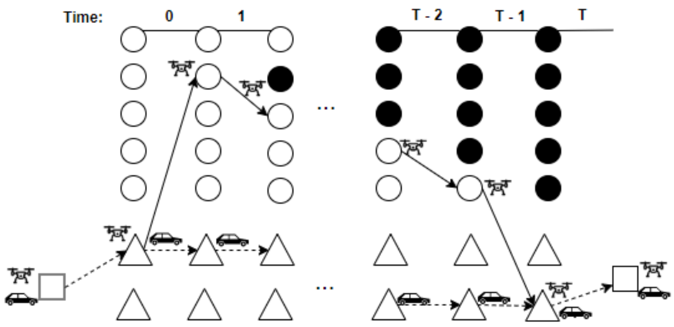


Figure 2. Airspace and groundspace illustration (Luo *et al*., 2018)

This study attempts to determine the best route for both GV and UAV based on this graphic. Both of the UAV's paths must adhere to all of the restrictions and must cover all target nodes. The mathematical framework of this study incorporates, with some adjustments, findings from work by Luo *et al*. (2018). A difficulty with cooperative routing for UAV and GV is resolved by this research. These issues and presumptions will form the basis of the mathematical model used in this study:

1. Each target node is only served once; however, UAV can visit target node more than once.
2. UAV travel with constant speed 56 km/h. This assumption is based on the research from Ha *et al*. (2018).
3. The research parameter is time duration, hence the distance between each node will be transformed into time duration.
4. Flight duration from one node to another node and service time are predetermined and consistent.
5. GV is only able to travel in ground space, while UAV can access both air space and ground space.
6. UAV has an energy limit which will be converted into flying time limit. Meanwhile, GV has no energy limitation.
7. Time to swap the battery for UAV is negligible.

We apply the following notations and its implications for the problem.

V_t the set of all target nodes,
V_r the set of all optional rendezvous nodes,
d_{ij} the distance from node i to node j,
v the velocity of UAV (assumed constant),
t_{ij} the time needed to travel from node i to node j , where $t_{ij} = \frac{d_{ij}}{v}$,
s_i the service time of node i,
T_i the time consumed in node i,
$t_{\max}$ the UAV's endurance (maximum).

We also introduce the decision variables as follows:

$$x_{ij} = \begin{cases} 1, & if\ GV\ travel\ from\ node\ i\ to\ node\ j \\ 0, & otherwise \end{cases}$$

$$y_{ij} = \begin{cases} 1, & if\ UAV\ travel\ from\ node\ i\ to\ node\ j \\ 0, & otherwise \end{cases}$$

$$S_{ij} = \begin{cases} 1, & if\ node\ i\ is\ served\ from\ node\ j \\ 0, & otherwise. \end{cases}$$

2.1.1 Objective function

The goal of this problem is to give a path from start node to end node that satisfied all of the constraints. Thus, the objective function of this optimization is to minimize the total time of task completion.

$$\sum t_{ij}\, y_{ij}\,, \forall i \in \{0\} \cup \mathrm{V_t} \cup \mathrm{V_r}\,;\; j \in \mathrm{V_t} \cup \mathrm{V_r}. \qquad (1)$$

This means we will find the most efficient and quickest way to complete the whole mission while adhering to the constraints that will discuss next. The objective function takes into consideration all the factors that contribute to the total time of task completion, in this research are the service time and flight time.

2.1.2 Constraints

The UAV and GV have limitations or restrictions that are imposed on their design and operation. The constraints must be taken into account in mission planning and optimization to ensure and efficiency of operations. The constraints of this problem are described below.

a. Ensure there is an edge out from depot.
   This research assumed that index $o \in \{0\}$ is the depot. The depot must have an edge out as a start node as illustrated in Figure 3. Thus, to ensure that the binary variable (GV and UAV) out from the depot, the sum of each edge must be equal to 1 as follows:

$$\sum x_{oj} = 1\ , \forall j \in \mathrm{V_r} \qquad (2a)$$
$$\sum y_{oj} = 1\ , \forall j \in \mathrm{V_t} \cup \mathrm{V_r}\ . \qquad (2b)$$

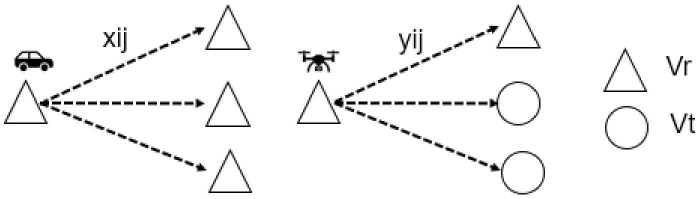


Figure 3. The depot has an edge out as a start node

b. UAV is allowed to visit the rendezvous node (GV route).
   This constraint allows UAV to visit rendezvous node both landing (y_{ij}) and take off (y_{ij}). However, it is optional for UAV to visit the rendezvous node thus the binary variable needs to be less than or equal to 1,

$$\sum y_{ij} \le 1\ , \forall\, i \in \{0\} \cup \mathrm{V_t}\ ;\ j \in \mathrm{V_r} \qquad (3a)$$
$$\sum y_{ji} \le 1\ . \forall\, i \in \{0\} \cup \mathrm{V_t}\ ;\ j \in \mathrm{V_r} \qquad (3b)$$

   These constrains can be illustrated in Figure 4.

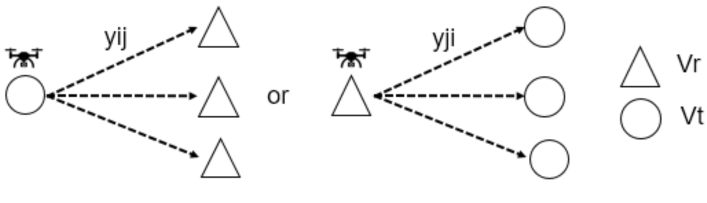


Figure 4. UAV is allowed to visit the rendezvous node

c. Ensure GV is already available when UAV finishes its mission and ensure that UAV will not exceed its endurance.
   This constraint utilises the binary UAV and GV variable and its duration to fly to and service a certain node. On the left-hand side, the GV binary variable is multiplied by the journey time from depot or rendezvous node to other nodes. The total period must be shorter than the time necessary for the UAV to fly between nodes and reach the destination node, thus

$$\sum_{i \in \{0\} \cup \mathrm{V_r}} \sum_{j \in \mathrm{V_r}} t_{ij}\, x_{ij} \le \sum_{i \in \mathrm{V_t}} \sum_{j \in \{0\} \cup \mathrm{V_r} \cup \mathrm{V_t}} S_{ij}\, s_j + \sum_{i \in \{0\} \cup \mathrm{V_t} \cup \mathrm{V_r}} \sum_{j \in \mathrm{V_t}} y_{ij}\, t_{ij} \le t_{\max}. \qquad (4)$$

The UAV binary variable y_{ij} multiplied by the flight time between nodes represents the duration of the UAV.

d. Ensure that UAV will not fly between rendezvous node.
UAV is only able to access air space (see Figure 5). Therefore, the binary variable for UAV has to be equal to 0 when accessing rendezvous node,

$$y_{ij} = 0,\ \forall i, j \in \mathrm{V_r}. \quad (5)$$

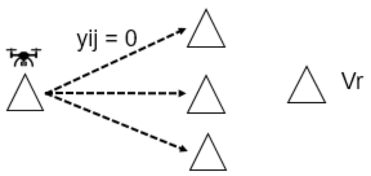


Figure 5. UAV not fly between rendezvous node

e. If GV stop in one of the rendezvous node, then that node is where UAV takes off or lands.
For x_{ij}, index i represents depot and rendezvous node and index j represent the rendezvous node. If a GV stop in a rendezvous node, the binary value of x_{ij} is 1 since node j is visited from node i. Meanwhile, index i for y_{ij} represents target node and index j represents rendezvous node. If UAV land (y_{ij}) and take off (y_{ji}) from a certain node, the binary value of y_{ij} and y_{ji} is 1,

$$\sum_{i\,\in\,\{0\}\,\cup\,V_r}\ \sum_{j\,\in V_r} x_{ij}\ \leq\ \sum_{i\,\in\ V_t}\ \sum_{j\,\in V_r} (y_{ij} + y_{ji})\ . \quad (6)$$

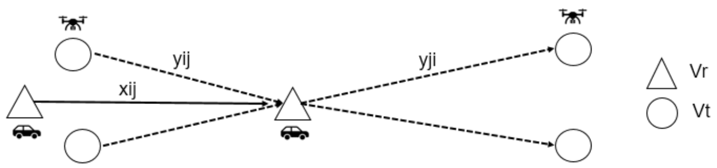


Figure 6. When GV accessed to the rendezvous node, this node is where UAV takes off or lands

Constraint (6) guarantees that if the GV has accessed to an optional stopping node, this node must be a rendezvous node at which the UAV takes off or lands as illustrated in Figure 6.

f. Ensure all target node are served.
All target nodes must be visited by UAV. Since S_{ij} means node i is visited from node j, the sum of S_{ij} has to be equal to 1,

$$\sum_{i\,\in V_t}\ \sum_{j\,\in\,\{0\}\,\cup\,V_r} S_{ij} = 1, \forall i \in\ \mathrm{V_t}\ ;\ j \in \{0\} \cup \mathrm{V_r} \cup \mathrm{V_t}. \quad (7)$$

Constraint (7) ensures that each target node is served by UAV.

g. Nodes that are not visited has $t_{ij} = 0$.
When an UAV does not fly through a certain edge the value of x_{ij} is 0. This condition is true when t_{ij} also equal to 0.

$$t_{ij}\ \leq\ M \times \sum_{i\,\in\,\{0\}\,\cup\,V_r} x_{ij}\,,\ \forall j \in \mathrm{V_r}, \quad (8)$$

where M is a sufficiently large positive number.

h. Take off from GV.
When an UAV takes off from GV in node i to node j, the value of y_{ij} and S_{ji} is 1. The consumed capacity in node j should be equal to the sum of service time in node j and flight time from node i to j,

$$M \times\ (1 - y_{ij}) \geq |\ s_j + t_{ij} - T_j\ | + |S_{ji} - 1|, \forall i \in \{0\} \cup\ V_r\ ;\ j\ \in\ V_t. \quad (9)$$

This condition can be illustrated in Figure 7.

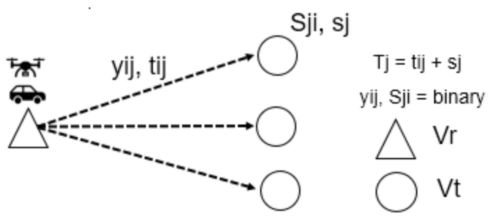


Figure 7. The taking off progress of UAV

i. Flying condition.
When an UAV flies from node i and node j, node i and node j should be in the same segment ($S_{ik} = S_{jk}$). The consumed capacity in node j should be equal to the sum of consumed capacity in node i, flying time between node i and node j and service time in node j (see Figure 8).

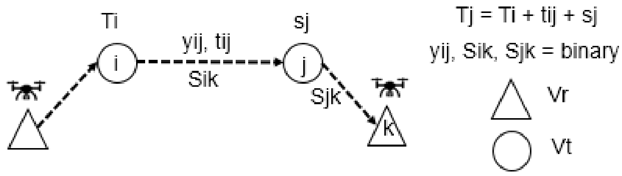


Figure 8. The flying progress of UAV between targets

Therefore, this following inequality can describe that constraint.

$$M \times (2 - y_{ij} - S_{ik}) \geq |T_i + t_{ij} + s_j - T_j| + |S_{ik} - S_{jk}|, \forall i \in V_t \; ; \; j \in V_t \; ; \; k \in \{0\} \cup V_r. \quad (10)$$

j. Landing on GV.
When an UAV land on a GV, the value of y_{ij} should be equal to 1. Figure 9 illustrate when the UAV land on GV. The duration to serve node i added with the flying time between node i and node j should be equal to UAV endurance. Hence,

$$M \times (1 - y_{ij}) \geq |s_i + t_{ij} - t_{\max}|, \; \forall i \in V_t \; ; \; j \in V_r \quad (11)$$

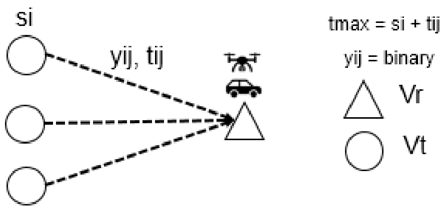


Figure 9. The landing progress

k. Ensure all of UAV's mission do not exceed the endurance
This constraint ensure that the time consumed in each node must be a positive value and must not exceed the battery endurance,

$$0 \leq T_i \leq t_{\max}, \quad \forall i \in V_t \cup V_r. \quad (12)$$

In other words this constraint describes the limit of the battery capacity consumed.

2.2 Algorithm for cooperated UAV and GV route

This method will be separated into two algorithms, one for finding the UAV path and the other for finding the GV path to determine the route for both UAV and GV. UAV is the first algorithm investigated in this study. The result of the UAV algorithm is the UAV flight route and the UAV's remaining flight time. This result will be utilised by the GV algorithm to confirm that the route from the UAV path is viable and that the GV can move along the specified route.

2.2.1 Algorithm for UAV path

The Depth First Search (DFS) method will be modified for UAV path determination. DFS is appropriate for this research since it can locate a route by searching "deep" into the neighbouring node from a given node. Once there is no longer a nearby node, this algorithm will "backtrack" to its previous node to choose an alternative path. This approach, with a few adjustments, is appropriate for this research since it can identify all feasible paths. This method requires the following information to function: start node, end node, UAV time limit, and time-space (TS) network containing all nodes. Before executing the algorithm, initialization must be performed by setting the start node to the current node, so that the algorithm will begin searching from the supplied node. Time limit (T) must also be initialised to the current time. This limit will be updated along the way when finding the path. In the initialization, the path is ∅. This 'path' will record every served node during the path traversal. Table 1 provides the algorithm for UAV path.

Table 1. Algorithm for UAV path

Algorithm 1: UAV_Path

Input Data:
TS, start_node, end_node, T_limit
Initialization:
Current_node = start_node, path = ∅, current_time = T
1. **If** *current_node = end_node* **and** *path ∈ air_node* **and** *current_time >= 0:*
2.   **print** *path, current_time*
3. **If** *next_node* **not in** *TS:*
4.   **return** *path*
5. **For** *next_node, fly_time* **in** *TS:*
6.   **if** *next_node* **not in** *path* **and** *next_node* ∈ V_t **and** *current_time >= 0*
7.   **update** *path, current_node, current_time*
8.   **if** *next_node* **not in** *path* **and** *next_node* ∈ V_r *:*
9.   **update** *path, current_node, current_time*
10. **End**

Result:
UAV_Path, UAV_Residual_Time

The first and second line will provide us with the completed route if the condition meets. A path is deemed complete when the current node and end node are identical since this method updates the current node constantly. The final path must include every air node (path air node), which indicates that all target nodes have previously been serviced. A positive value must also be included for the final path's leftover flying time. If the next node is not found in the time-space network, the third line enables us to ignore a node. This algorithm's heart, which permits us to update the node, is on the following line. If the next node is not on the route, then check if it is a part of the ground node or the air node. The distinction is made while changing the fly time restriction and determining if the node may visit that next node while still spending its remaining flying time. The current time will be updated using the formula *current time = current time - duration when the next node is an air node*, and UAV must verify that UAV is able to fly and service the next node (serving, flying). On the other hand, when the next node is a ground node, the current time will be updated to its start time since a ground node will have its initial time limit due to the swapping of the 'battery' in that node. This algorithm's result will be noted and utilised for the GV algorithm.

2.2.2 Algorithm for GV path

It is known from the output UAV Path that there is at least one segment from ground space. Consequently, verification is required to confirm that the route is accessible to GVs. Start node, end node, time-space network, and UAV path produced from the previous technique are required for this approach. Observe that the network for GV path differs from UAV path since GV can only reach ground areas. The method for GV Path is simplified due to the assumption that GV has unlimited operational time. Therefore, the duration parameter is omitted from this algorithm. However, this method functions similarly to the UAV route, but disregards time. Table 2 contains the GV route algorithm.

Table 2. Algorithm for ground vehicle path

Algorithm 2: GV_Path
Input Data:
TS, start_node, end_node
Initialization:
a. *Current_node = start_node*
b. **If** *current_node = end_node* **and** *path* $\notin V_t$:
c. **print** *path*
d. **If** *next_node* **not in** *TS:*
e. *return path*
f. **For** *next_node* **in** *TS:*
g. **if** *next_node* **not in** *path* :
h. **update** *path, current_node*
i. **End**
Result:
GV_Path

The algorithm will search for its neighbouring node since similar to UAV path, the start node is set as the current node. The resulting route is guaranteed to be for GV only by the first line. If the outcome indicates that the GV cannot follow the specified UAV path, the UAV path must be modified to another feasible option before being rechecked with the GV path.

2.2.3 Verification with small scale case

The proposed model must be validated to confirm that the technique is executable and capable of producing the desired outcome. The model will be implemented in a network made up of nine target nodes and four optional rendezvous nodes. The network presented in Figure 10. After the model displays the outcome of the small-scale verification, the result is compared to the Python Software Foundation's accessible algorithm (Python Patterns – Implementing Graphs) (Python Software Foundation, 2019).

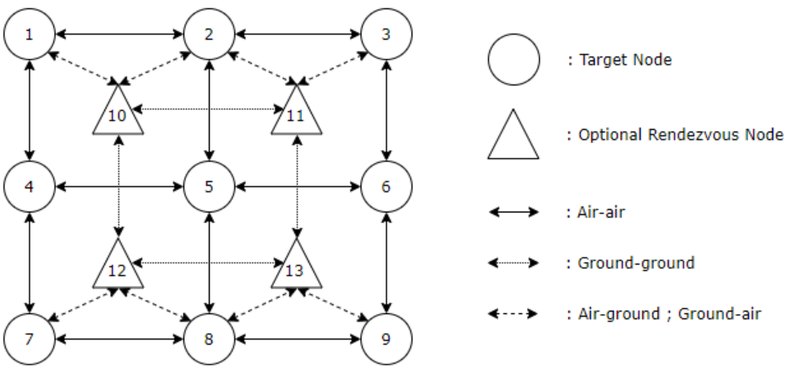


Figure 10. Small scale case

There will be a flying time between every pair of neighbouring nodes. Table 3 will illustrate the flying time from each node.

Table 3. Duration for small scale case

Start Node (i)	End Node (j)	Flying Time (min)	Start Node (i)	End Node (j)	Flying Time (min)
1	2	5	13	11	10
2	3	1	11	10	7
3	6	3	10	12	6
6	9	4	1	10	1
9	8	5	2	10	2
8	7	2	2	11	3
7	4	3	3	11	2
4	1	1	9	13	1
4	5	5	8	13	2
6	5	1	8	12	3
5	2	3	7	12	2
5	6	4	10	1	1
5	8	5	10	2	2
5	4	2	11	2	3
12	10	10	11	3	2
10	11	8	13	9	1
11	13	7	13	8	2
13	12	5	12	8	3
12	13	9	12	7	2

Assume this small-scale case is trying to find a route from node 10 to node 13 (both are ground nodes). The flight time is limited to 20 minutes. This assumption is based on research by Ha *et al*. (2018). This information will be included in the algorithm. Table 4 show the possible UAV paths.

Table 4. Possible UAV path

No	Path	The Remaining Flying Time
1	**10** – 12 – 7 – 4 – 1 – 2 – **11** – 3 – 6 – 5 – 8 – 9 – **13**	3 minutes
2	**10** – 1 – 2 – 3 – 6 – 5 – 4 – 7 – **12** – 8 – 9 – **13**	11 minutes
3	**10** – 1 – 2 – 5 – 4 – 7 – **12** – 8 – 9 – 6 – 3 – **11** – **13**	13 minutes
4	**10** – 1 – 2 – **11** – 3 – 6 – 5 – 4 – 7 – **12** – 8 – 9 – **13**	11 minutes
5	**10** – 1 – 4 – 5 – 2 – **11** – 3 – 6 – 9 – 8 – 7 – **12** – **13**	11 minutes
6	**10** – 1 – 4 – 7 – **12** – 8 – 9 – 6 – 5 – 2 – 3 – **11** – **13**	13 minutes
7	**10** – **11** – 3 – 6 – 5 – 2 – 1 – 4 – 7 – **12** – 8 – 9 – **13**	11 minutes
8	**10** – 1 – 2 – **11** – 3 – 6 – 5 – 4 – 7 – 8 – 9 – **13**	1 minute

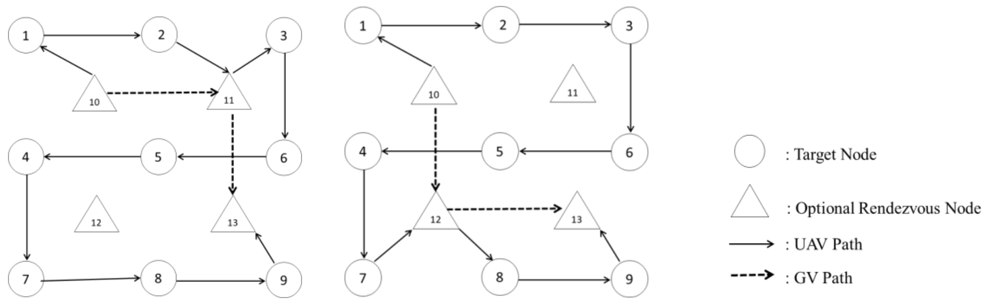
[4] The bold number means the rendezvous node

The possible UAV path result in Table 4 will be used as the input for GV Path.

Table 5. Accessible GV path

No	Path
1	10 – 11 – 13
2	10 – 12 – 13

Table 5 shows that there are two accessible GV routes. Therefore, from the UAV and GV results, it is known that there are two alternative UAV-GV routes. The possible routes are **10** – 1 – 2 – **11** – 3 – 6 – 5 – 4 – 7 – 8 – 9 – **13** with the remaining flying time is 1 minute or **10** – 1 – 2 – 3 – 6 – 5 – 4 – 7 – **12** – 8 – 9 – **13** with the remaining flying time is 11 minutes. For simplicity we call **10** – 1 – 2 – **11** – 3 – 6 – 5 – 4 – 7 – 8 – 9 – **13** as route A and **10** – 1 – 2 – 3 – 6 – 5 – 4 – 7 – **12** – 8 – 9 – **13** as route B. The final route for both GV and UAV are represented on Figure 11.



(a) (b)

Figure 11. UAV-GV Path Result for Small Scale Case. (a) Route A (b) Route B

This result is compared to the algorithm of Python Software Foundation's Algorithm (Python Software Foundation, 2019). The result of the Python programme is identical to the model's output (**10** – 1 – 2 – **11** – 3 – 6 – 5 – 4 – 7 – 8 – 9 – **13** and **10** – 1 – 2 – 3 – 6 – 5 – 4 – 7 – **12** – 8 – 9 – **13**). This indicates that the model is capable of producing identical results to the algorithm. However, the Python technique is incapable of calculating the battery usage for the specified path. Therefore, manual calculation of battery use is required. Table 6 displays the formula for calculating battery usage.

Table 6. Manually calculated battery consumption

Route A				Route B			
Start node (i)	**End node (j)**	**Flying time (min)**	**Remaining Flying Time (min)**	**Start node (i)**	**End node (j)**	**Flying time (min)**	**Remaining Flying Time (min)**
			20				20
10	1	1	19	10	1	1	19
1	2	5	14	1	2	5	14
2	11	3	20	2	3	1	13
11	3	2	18	3	6	3	10
3	6	3	15	6	5	1	9
6	5	1	14	5	4	2	7
5	4	2	12	4	7	3	4
4	7	3	9	7	12	2	20
7	8	2	7	12	8	3	17
8	9	5	2	8	9	5	12
9	13	1	1	9	13	1	11

The result shows that the residual flying time is the same as the model (1 minute for A route and 11 minutes for B route). This indicates that the model can function and produce accurate results. Therefore, this model is suitable to be implemented in a larger case.

3. The 2010 Merapi eruption case

This research aims to address the routing problem involving a cooperative UAV and GV, particularly in humanitarian logistics. Using a real-life case study is essential for gaining a deeper understanding of the issue. The eruption of Merapi in 2010 will serve as the case study for this research. The 2010 Merapi eruption will serve as the basis for this study's air space and ground space nodes. From October 26 to December 2, 2010, there was an eruption of Merapi. The effusive kind of eruption is the most common form of eruption at Merapi. This resulted in an unanticipated pyroclastic flow; hence it is imperative that the surrounding region be evacuated.

This research implemented the previous case to aid in the development of the model. The purpose of using prior experiences to construct the model is to guarantee that it is representative and relevant for future humanitarian actions. The network's nodes must be determined before construction can begin. In this study, Kabupaten Sleman

Regional Disaster Management Agency or BPBD (Badan Penanggulangan Bencana Daerah) provides the data. According to BPBD Kabupaten Sleman (2019), 29 hamlets are affected. The scale of the damage caused by the eruption is variation about 20% up to 100%. The damaged area will be the target node that the UAV must visit to scan the area, search for survivors, and analyse the disaster's damage. This work must be completed by UAV because to the unpredictability of pyroclastic flow from the volcano at the moment. Therefore, it is unsafe for a person to perform this duty. Nevertheless, not every hamlet will be considered the target node. In this study, it is considered that UAVs are required when the damage is 100%. This indicates that facilities like roads are damaged. Therefore, the destroyed region is inaccessible to humans. The target node for UAV is given in Table 7.

Table 7. Target node for air space network

No	Sub-district	Village	Hamlet	Damage (%)
1	Cangkringan	Glagaharjo	Srunen	100
2			Kalitengah Kidul	100
3			Kalitengah Lor	100
4		Argomulyo	Bakalan	100
5		Kepuharjo	Kaliadem	100
6			Jambu	100
7			Petung	100
8			Kopeng	100
9			Batur	100
10			Manggang	100
11		Wukirsari	Ngepringan	100
12		Umbulharjo	Palemsari/Kinahrejo	100
13			Pangukrejo	100

We have to consider that the ground network must be secure, unlike the air space network. Therefore, the ground node must be situated in a region that is not near a volcano. The barracks for evacuated people are positioned at the ground node in this study. Since the barrack is presumably secure from the explosion, ground vehicles may stop. The BPBD has provided several barracks. However, just a few barracks were taken into consideration in this study. These barracks are regarded as the largest of all the barracks. The location of the barrack is given in Table 8.

Table 8. Barrack Location in Merapi Eruption 2010

No	Sub-disctrict	Village	Hamlet	Location
1	Cangkringan	Kepuharjo	Batur, Kopeng, Jambu, Kaliadem, Kepuh, Manggong, Pagerjurang, Petung	Balai Desa Wukirsari
2		Glagaharjo	Kalitengah Lor, Kalitengah Kidul	Barak Gayam
3			Srunen	SD Jiwan
4			Singlar, Gading	Balai Desa Glagaharjo
5			Jetisumur, Glagahmalang	Balai Desa Sindumartani
6		Umbulharjo	Pelemsari, Pangukrejo, Balong	Barak Plosokerep
7			Gondang, Gambretan, Plosorejo	SMP Watuadeg
8	Pakem	Hargobinangun	Kaliurang Barat, Kaliurang Timur	Balai Desa Hargobinangun

Some of the evacuees had to evacuate to a different barrack due of the pyroclastic flow from the eruption, despite the barrack's status as a safe shelter for evacuees. This occurred because of Merapi's largest eruption, which triggered a pyroclastic flow from Gendol River. This pyroclastic flow traversed the Glagaharjo safety zone. Due to the threat, residents of three barracks (Balai Desa Wukirsari, Gayam, and SD Jiwan) were forced to relocate to a different barrack. This occurrence renders unavailable the indicated barracks. Consequently, not all barracks were utilised as the optional stopping point for GV in this investigation. As an alternative halting point, just five barracks are utilised. Table 9 gives a list of barracks that may be utilised as optional halting nodes.

Table 9. Barrack location used as optional stopping nodes

No	Sub-district	Village	Hamlet	Location
1	Cangkringan	Kepuharjo	Batur, Kopeng, Jambu, Kaliadem, Kepuh, Manggong, Pagerjurang, Petung	Balai Desa Wukirsari
2		Glagaharjo	Jetisumur, Glagahmalang	Balai Desa Sindumartani
3		Umbulharjo	Pelemsari, Pangukrejo, Balong	Barak Plosokerep
4			Gondang, Gambretan, Plosorejo	SMP Watuadeg
5	Pakem	Hargobi-nangun	Kaliurang Barat, Kaliurang Timur	Balai Desa Hargobinangun

There are 13 target nodes and 5 optional rendezvous nodes based on the above information. Figure 12 illustrates the precise location of each node, with the blue node representing the target node and the red node representing the optional rendezvous node. The nodes are represented in the adjacency list, where each node contains its neighbouring node and weight. The neighbouring node of each node is presented in Table 10.

Table 10. Adjacency list of each node

Index	Target Node	Adjacent Node	Fly Time (min)	Service Time (min)	Total (min)
1	Srunen	Kaliadem	0.49	3.9	4.39
		Kalitengah Kidul	1.19	3.9	5.09
		Kalitengah Lor	2.65	3.9	6.55
		Jambu	1.21	3.9	5.11
		Kopeng	2.82	3.9	6.72
2	Kalitengah Kidul	Kalitengah Lor	0.96	4.44	5.4
		Srunen	1.19	4.44	5.63
		Kaliadem	3.21	4.44	7.65
3	Kalitengah Lor	Kalitengah Kidul	0.96	5.36	6.32
		Srunen	2.65	5.36	6.55
4	Bakalan	Ngepringan	2.36	4.19	6.55
		Balai Desa Wukirsari	5.83	-	5.83
		Balai Desa Sindumartani	9.2	-	9.2
5	Kaliadem	Srunen	0.49	4.18	4.67
		Jambu	0.61	4.18	4.79
		Palemsari	1.08	4.18	5.26
		Kalitengah Kidul	5.7	4.18	9.88
		Barak Plosokerep	8.22	-	8.22
6	Jambu	Kaliadem	0.61	5.19	5.8
		Petung	1.31	5.19	6.49
		Palemsari	1.08	5.19	6.27
		Srunen	0.04	5.19	5.23
		Kopeng	0.59	5.19	5.78
		Pangukrejo	1.32	5.19	6.51
7	Petung	Kopeng	0.66	4.78	5.44

Index	Target Node	Adjacent Node	Fly Time (min)	Service Time (min)	Total (min)
		Batur	0.68	4.78	5.46
		Jambu	1.31	4.78	6.09
		Pangukrejo	1.49	4.78	6.27
		Barak Plosokerep	6.93	-	6.93
8	Kopeng	Petung	0.66	4	4.66
		Batur	0.38	4	4.38
		Manggang	2.33	4	6.32
		Jambu	1.78	4	5.78
		Srunen	2.68	4	6.68
9	Batur	Petung	0.68	3.5	4.18
		Kopeng	0.38	3.5	3.88
		Manggang	5.45	3.5	8.96
		Barak Plosokerep	4	-	5.58
		SMP Watuadeg	4	-	8.33
10	Manggang	Batur	5.45	3.91	9.36
		Ngepringan	2.5	3.91	6.41
		Kopeng	2.33	3.91	6.24
11	Ngepringan	Bakalan	2.36	5.3	7.66
		Manggang	2.5	5.3	7.8
		SMP Watuadeg	6.03	-	6.03
		Balai Desa Hargobinangun	7.8	-	7.8
12	Palemsari	Kaliadem	1.08	4.29	5.37
		Pangukrejo	0.76	4.29	5.05
		Jambu	1.08	4.29	5.37
13	Pangukrejo	Petung	2.54	4.74	7.28
		Palemsari	0.76	4.74	5.5
		Barak Plosokerep	6.88	-	6.88
		Jambu	1.74	4.74	6.48
14	Balai Desa Wukirsari	Bakalan	1.64	4.19	5.83
		SMP Watuadeg	1.48	-	1.48
		Balai Desa Sindumartani	5.01	-	5.01
15	Balai Desa Sindumartani	Bakalan	5.01	4.19	9.2
		Balai Desa Wukirsari	5.01	-	5.01
16	Barak Plosokerep	Petung	2.15	4.78	6.93
		Batur	2.35	3.5	5.85
		Pangukrejo	2.14	4.74	6.88
		Balai Desa Hargobinangun	2.23	-	2.23
		SMP Watuadeg	5.4	-	5.4
		Kaliadem	4.72	3.5	8.22
17	SMP Watuadeg	Ngepringan	5.3	0.73	6.03

Index	Target Node	Adjacent Node	Fly Time (min)	Service Time (min)	Total (min)
		Balai Desa Wukirsari	1.48	-	1.48
		Balai Desa Hargobinangun	2.38	-	2.38
		Barak Plosokerep	5.36	-	5.36
		Batur	4.2	4.3	8.5
18	Balai Desa Hargobinangun	Ngepringan	2.5	5.3	7.8
		Barak Plosokerep	2.23	-	2.23
		SMP Watuadeg	2.38	-	2.38

The nodes above are converted into a network, and then implemented using in the model to find the optimal route for both UAV and GV. The start and end node has to be the optional rendezvous node, we will find all of possibilities routes from each start and end nodes. The flying limitation time is 60 minutes. This assumption is based on research by Półka *et al.* (2017) will be included in the algorithm.

Table 11. Possible UAV Path Result and Remaining Flying Time

No	Path	The Remaining Flying Time
1	**14** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17**	21.97 minutes
2	**14** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17** – **18**	57.62 minutes
3	**14** – **15** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17** – **18**	57.62 minutes
4	**15** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17** – **14**	58.52 minutes
5	**15** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17**	21.97 minutes
6	**15** – **14** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17**	21.97 minutes
7	**15** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17** – **18**	57.62 minutes
8	**15** – 4 – **14** – **17** – 9 – 7 – 6 – 12 – 13 – **16** – 5 – 2 – 3 – 1 – 8 – 10 – 11 – **18**	2.70 minutes
9	**15** – **14** – 4 – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17** – **18**	57.62 minutes
10	**17** – 9 – 7 – 6 – 12 – 13 – **16** – 5 – 2 – 3 – 1 – 8 – 10 – 11 – 4 – **15** – **14**	54.99 minutes
11	**17** – 9 – 7 – 6 – 12 – 13 – **16** – 5 – 2 – 3 – 1 – 8 – 10 – 11 – 4 – **14** – **15**	54.99 minutes
12	**18** – **17** – 9 – 7 – 6 – 12 – 13 – **16** – 5 – 2 – 3 – 1 – 8 – 10 – 11 – 4 – **15** – **14**	54.99 minutes
13	**18** – 11 – 10 – 8 – 1 – 3 – 2 – 5 – **16** – 13 – 12 – 6 – 7 – 9 – **17** – **14** – 4 – **15**	44.97 minutes
14	**18** – **17** – 9 – 7 – 6 – 12 – 13 – **16** – 5 – 2 – 3 – 1 – 8 – 10 – 11 – 4 – **14** – **15**	54.99 minutes

[12] The bold number means the rendezvous node

The UAV possible path result in Table 11 will be used as the input for GV Path.

Table 12. Accessible GV Path of Merapi Case

No	Path
1	15 – 14 – 17 – 16 – 18

Therefore, the path result with some optional rendezvous nodes are in bold as follows:

15 – 4 – **14** – **17** – 9 – 7 – 6 – 12 – 13 – **16** – 5 – 2 – 3 – 1 – 8 – 10 – 11 – **18**.

Hence, it is known from the result that there is only one feasible route as the final route for UAV and GV, that is: **Balai Desa Sindumartani** – Bakalan – **Balai Desa Wukirsari** – **SMP Watuadeg** – Batur – Petung – Jambu – Palemsari – Pangukrejo – **Barak Plosokerep** – Kaliadem – Kalitengah Kidul – Kalitengah Lor – Srunen – Kopeng – Manggang – Ngepringan – **Balai Desa Hargobinangun** with GV route is in the bold. Final route for UAV and GV is illustrated in Figure 12.

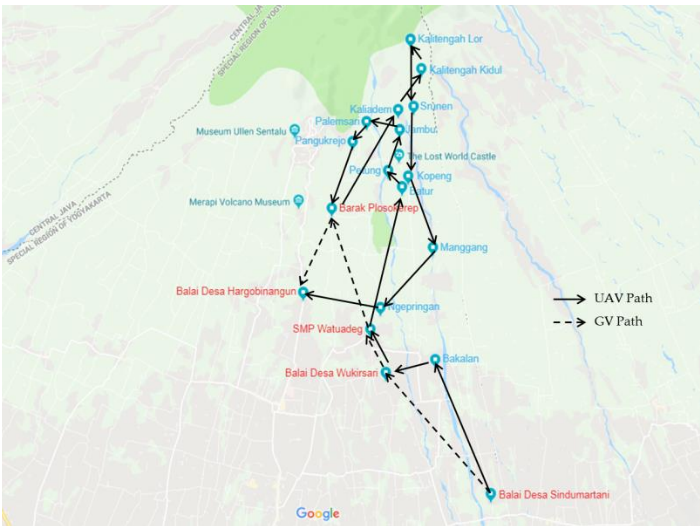


Figure 12. Route for cooperated UAV – GV in the 2010 Merapi eruption

4. Conclusion

The aim of this research is to determine the best path for a collaborated system of ground vehicle (GV) and unmanned aircraft. This research has effectively modelled the cooperative routing problem of ground vehicle (GV) and unmanned aerial vehicle (UAV) using Depth First Search (DFS) method. This investigation comprises two workspaces: airspace that is only accessible to UAV and ground space that can be accessed by GV and UAV. This method is separated into two algorithms for finding UAV path and GV path. We first find the UAV routes and GV routes with some considerations of UAV limitations, such as flight time restrictions, and then we used the result to confirm that the route from UAV paths is viable so that the GV can move along to the specified route.

This research seeks to identify a path that fulfils the restriction. The model is constrained by vehicle condition; ensuring that there is a means for vehicles to leave the depot, ensuring that UAVs can fly in both airspace and ground space networks while GVs can only access ground space networks, ensuring that UAVs only fly when their battery capacity is sufficient, and ensuring that all target nodes are served.

The model is implemented in the actual humanitarian case which is Merapi Eruption in 2010. The data consist of 13 target node and 5 optional rendezvous nodes. The result of this research is the UAV has the following path: Balai Desa Sindumartani – Bakalan – Balai Desa Wukirsari – SMP Watuadeg – Batur – Petung – Jambu – Palemsari – Pangukrejo – Barak Plosokerep – Kaliadem – Kalitengah Kidul – Kalitengah Lor – Srunen – Kopeng – Manggang – Ngepringan – Balai Desa Hargobinangun. The result shows that all the target nodes are served with 2.7 minutes of UAV's battery left.

Declaration of Conflicting Interests. The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4), 965-981.

American Red Cross. (2015). *Drones for Disaster Response and Relief Operations*. Retrieved Feb 1, 2019, from https://www.issuelab.org/resources/21683/21683.pdf

Badan Penanggulangan Bencana Daerah Kabupaten Sleman. (2012). *Rencana Kontijensi Erupsi Gunung Merapi*, BPBD Sleman, Yogyakarta.

Cahyono, A. B., & Zayd, R. A. (2018, March). Rapid mapping of landslide disaster using UAV-photogrammetry. *In journal of physics: conference series* (Vol. 974, No. 1, p. 012046). IOP Publishing.

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.

Duhamel, C., Lacomme, P., & Prodhon, C. (2011). Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, 38(4), 723-739.

Dorling, K., Heinrichs, J., Messier, G. G., & Magierowski, S. (2016). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1), 70-85.

Gomez, C., & Purdie, H. (2016). UAV-based photogrammetry and geocomputing for hazards and disaster risk monitoring–a review. *Geoenvironmental Disasters*, 3, 1-11.

Ha, Q. M., Deville, Y., Pham, Q. D., & Ha, M. H. (2015). Heuristic methods for the traveling salesman problem with drone. arXiv preprint arXiv:1509.08764.

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 86, 597-621.

Hassan, Z., Ali Shah, S. I., & Sarwar Rana, A. (2022). Charging Station Distribution Optimization Using Drone Fleet in a Disaster. *Journal of Robotics*, 2022, 1-13.

Hua, C. S., Qi, J. T., Shang, H., Hu, W., & Han, J. (2016). Detection of collapsed buildings with the aerial images captured from UAV. *Science China Information Sciences*, 59(3), 1-15.

Iguchi, M., Ishihara, K., & Hendrasto, M. (2011). Learn from 2010 Eruptions at Merapi and Sinabung Volcanoes in Indonesia. *Annuals of Disaster Prevention Research Institute*, 54, 185-194.

de Oliveira Silva, L., de Mello Bandeira, R. A., & Campos, V. B. G. (2017). The use of UAV and geographic information systems for facility location in a post-disaster scenario. *Transportation Research Procedia*, 27, 1137-1145.

Półka, M., Ptak, S., & Kuziora, Ł. (2017). The use of UAV's for search and rescue operations. Procedia engineering, 192, 748-752.

Vaillancourt, A. (2016). A theoretical framework for consolidation in humanitarian logistics. *Journal of Humanitarian Logistics and Supply Chain Management*, 6(1), 2-23.

Volkmann, W. (2017). Small unmanned aerial system mapping versus conventional methods. *CTA Working Paper 17/07*, CTA.

Li, Z., & Garcia-Luna-Aceves, J. J. (2007). Finding multi-constrained feasible paths by using depth-first search. *Wireless Networks*, 13(3), 323-334.

Luo, Z., Liu, Z., Shi, J., Wang, Q., Zhou, T., & Liu, Y. (2018, June). The mathematical modeling of the two-echelon ground vehicle and its mounted unmanned aerial vehicle cooperated routing problem. *In 2018 IEEE intelligent vehicles symposium (IV)* (pp. 1163-1170). IEEE.

Luo, Z., Liu, Z., & Shi, J. (2017). A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. *Sensors*, 17(5), 1144.

Python Software Foundation. (2019, 5 April). Python Patterns - Implementing Graphs, https://www.python.org/doc/essays/graphs/

Surono., Jousset, P., Pallister, J., Boichu, M., Buongiorno, M. F., Budisantoso, A., Costa, F., & Humaida, H. (2012). The 2010 explosive eruption of Java's Merapi volcano – A '100-year' event. *Journal of Volcanology and Geothermal Research*, 241-242, 121-135.

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11, 679-697.

Wei, X., Qiu, H., Wang, D., Duan, J., Wang, Y., & Cheng, T. C. E. (2020). An integrated location-routing problem with post-disaster relief distribution. *Computers & Industrial Engineering*, 147, 106632.

Yu, K., Budhiraja, A. K., & Tokekar, P. (2018, May). Algorithms for routing of unmanned aerial vehicles with mobile recharging stations. *In 2018 IEEE international conference on robotics and automation* (ICRA) (pp. 5720-5725). IEEE.

To Cite This Article: Larasati, M. R. & Wijayanti, E. R. (2023). Cooperated unmanned aerial vehicle and ground vehicle in humanitarian operations: A routing problem. *Journal of Industrial Engineering and Education*, 1(1), 90-106.