Dynamic Graph-Based UAV Swarm Simulation for Peatland Fire Monitoring with Communication Connectivity Constraints

Samantha Michelle S. Silaban - 13525013
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: samantha.michelle2007@gmail.com , 13525013@std.stei.itb.ac.id

***Abstract*—Peatland fires are difficult to monitor because fire hotspots can spread across large areas and may be difficult to detect early. These fires are highly destructive because they can damage ecosystems, produce toxic haze, release carbon emissions, and cause habitat loss. This study develops a dynamic graph-based Unmanned Aerial Vehicle (UAV) swarm simulation for peatland fire monitoring. This simulation utilizes the graph theory and path finding algorithm such as Breadth-First Search and A star algorithm. Five scenarios were evaluated using different number of UAVs, communication ranges, and map merging settings. The results show that all scenarios successfully detected all fire hotspots, while map coverage, exploration time, and communication robustness varied across scenarios. Map merging improved exploration efficiency, while larger communication ranges increased the average largest connected component ratio and one-node failure robustness.**

Keywords—unmanned aerial vehicle swarm, peatland fire monitoring, graph-based exploration.

I. Introduction

Indonesia has vast peatland areas that play an important role in supporting the wetland ecosystem. The total area of Indonesian peatland is around 20.6 million hectares [1]. However, peatland areas are vulnerable to fire, especially during the dry season. In 2019, around 1.5 million of peatland were burned in Indonesia [1]. Peatland fires are also dangerous because they can easily spread across wide areas, produce toxic haze, and cause habitat loss. Therefore, early detection and continuous monitoring are important to reduce the risk of peatland fires.

Traditional forest and peatland fire monitoring can involve several methods, such as field patrols, observation towers, cameras, satellite imagery, and hotspot data processing. These methods are useful for supporting early detection, but they may still face limitations in terms of coverage, response time, visibility, and accessibility, especially in large or difficult-to-reach peatland areas. As an alternative, unmanned aerial vehicles can support fire monitoring because they are mobile and can observe different parts of an area from above.

Using multiple unmanned aerial vehicles as a swarm can improve monitoring because several areas can be explored at the same time. However, the effectiveness of a UAV swarm is influenced by several factors, such as the number of UAVs [2], communication range, and the ability to share map information between connected UAVs [3]. These factors may affect how fast the swarm explores the area, how much of the map can be covered, and how robust the communication network is during monitoring. Therefore, this study develops a dynamic graph-based UAV swarm simulation for peatland fire monitoring. The simulation models the peatland area as a grid graph and evaluates how the number of UAVs, communication range, and map merging mechanism affect map coverage, fire detection, exploration time, and communication connectivity. Through this simulation with diverse scenarios, the study aims to analyze factors that influence UAV swarm performance in peatland fire monitoring.

The rest of this paper is organized as follows. Section II presents the theoretical background, Section III explains the methodology and evaluation metrics, Section IV presents the results and analysis, and Section V provides the conclusion and future work.

II. Theoretical Background

A. Peatland Fire

Peat is formed from the accumulation of dead plant materials that decompose slowly under waterlogged and anaerobic conditions. Because decomposition is limited, organic matter gradually builds up and forms peat soil over time.

Fire in peatlands is hazardous to the environment. According to Harrison et al. [4], besides causing habitat loss, the toxic substance affects animal and tree health. Not only on land, but reduced pH in acidic peatland rivers will cause fish death. Together with the smoke, peatland fire produces substances that cause premature mortalities in Equatorial Asia in 2015. The substance also includes a large amount of carbon that leads to global warming.

B. Graph

A graph is a mathematical structure used to represent relationships between objects. A graph is commonly defined as

(G = (V, E)), where (V) represents a set of nodes or vertices, and (E) represents a set of edges that connect pairs of nodes. Nodes can represent objects or states, while edges represent the relationship or connection between them.

Graphs can be classified based on edge direction. In a directed graph, each edge has a direction, meaning that the relationship from one node to another is not necessarily reversible. In an undirected graph, edges do not have direction, so the connection between two nodes is considered bidirectional.

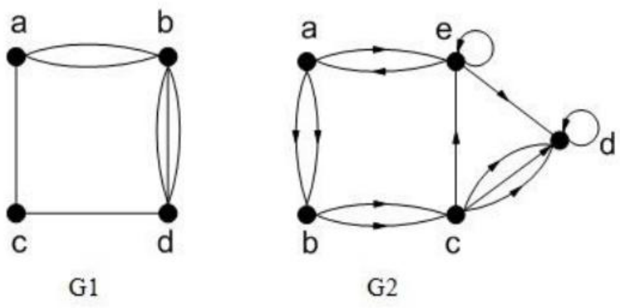

Fig 1. G1: Undirected Graph, G2: Directed Graph [5]

There are several terminologies that are used to described a graph, such as:

a) Adjacent: If two nodes are directly connected, then both nodes are adjacent to each other.

b) Incidence: If an edge directly connects a node, then the edge is incident to the node.

c) Null graph: If there are no edge on the graph, then it is called as null graph or empty graph.

d) Degree: The amount of edges that is incident to the node.

e) Path: A sequence of nodes connected by edges, where each consecutive pair of nodes has a direct connection.

f) Circuit: A path that starts and ends with the same node.

g) Connectivity: A graph is connected if there is a path from one node to other node.

C. Communication connectivity

Communication connectivity is an important aspect in UAV swarm operations because it determines whether UAVs can exchange information during the mission. In practice, UAV communication range can be affected by several factors, including the relative position between UAVs, environmental and topographical conditions, radio propagation characteristics, obstacles, multipath propagation, signal loss, quality of service requirements, and uncertainty in wireless channel prediction [3]. When UAVs move farther away from each other, the communication link may become weaker or disconnected. Environmental conditions such as terrain, smoke, and obstacles may also reduce signal quality and affect the reliability of communication. Therefore, communication range should not always be considered constant in real-world applications.

D. Breadth-First Search

Breadth-First Search (BFS) is an algorithm that starts from a source node and traverses the graph at the same level before expanding to the next level [6]. In an unweighted graph, BFS helps to compute shortest path distance from a single source node to all reachable nodes by counting the number of edges traversed. There are several ways to implement BFS, such as top-down BFS, bottom-up BFS, or even combining them (hybrid algorithm). Top-down BFS starts from the parent node and expands to the unvisited neighboring node, while bottom-up BFS checks every unvisited node whether they have neighbor in the current frontier. Bottom-up is usually used for large graphs since searching a lot of nodes like top-down BFS are expensive.

```
breadth-first-search(vertices, source)

  frontier ← {source}
  next ← {}
  parents ← [−1, −1, . . . , −1]
  while frontier ≠ {} do
    top-down-step(vertices, frontier, next, parents)
    frontier ← next
    next ← {}
  end while
  return tree
```

Fig 2. BFS Algorithm Pseudocode [6]

E. A Algorithm*

A* algorithm is a path finding algorithm that computes each candidate node using function below.

$$f(n) = g(n) + h(n) \qquad (1)$$

The symbol *g(n)* represents the cost from starting point to node *n* and h(*n)* represents estimated cost from node n to target node. The latter one is called a heuristic. By combining the actual path cost and the heuristic value, A* prioritizes nodes that are more likely to lead to the shortest path.

```
Variables:
    OPEN_list:Initialize as an empty list
      node_goal:Create the goal node;
    CLOSED_list:Initialize as an empty list
      node_start:Create start node;
Add node_start to the OPEN_list
while the OPEN_list is not empty do
  Get node n off the OPEN_list with the lowest f(n)
  Add n to the CLOSED_list
    if n is the same as node_goal then
      return Solution(n)
    Else
      Generate each successor node n′ of n
    for each successor node n′ of n do
      Set the parent of n′ to n
      Set h(n′) to be the heuristically estimate distance to node_goal
      Set g(n′) to be g(n) plus the cost to get to n′ from n
      Set f(n′) to be g(n′) plus h(n′)
        if n′ is on the OPEN_list and the existing one is as good or better then
          discard n′ and continue
        if n′ is on the CLOSED_list and the existing one is as good or better then
          discard n′ and continue
        Remove occurrences of n′ from OPEN_list and CLOSED_list
        Add n′ to the OPEN_list
```

Fig 3. A* Algorithm Pseudocode [7]

III. Methodology

A. Simulation Environment

Simulation was made by modelling the peatland area as a two-dimensional grid. To support graph-based exploration, the grid was also represented as an undirected graph where each node corresponds to a cell coordinates and each edge represents adjacency to other cells. In parallel, the condition of each cell was modelled as a matrix. There are 3 values that represent each cell state: 0 for safe cells, 1 for obstacles, and 2 for fire hotspots. In this simulation, it is known that there are 6 fire hotspots. This representation allows the UAVs to apply graph-based operations such as neighbour checking, frontier detection, and path planning while still referring to the matrix for the condition of each cell.

B. UAV Model and Local Mapping

Each UAV is represented by an id, current position, known safe cells, known obstacle cells, known fire cells, and visited cells. At first, all UAVs did not know the real condition of the area. They will update the information of each cell by exploring step-by-step. For example, if the neighboring cell contains a value of 1 then the cell is added to known obstacle cells. Therefore, in this simulation, the UAV sensor was configured to detect cells within a radius of 2 grid units from its position.

C. Frontier-Based Exploration and Path Planning

UAVs will explore cells by checking available frontier cells on their local map. Frontier cells are safe cells that are neighbouring with unknown cells. In every time step, each UAV chooses its target frontier cells based on the shortest reachable path distance from its current position. The reachable distance is calculated by using top-down Breadth-First Search (BFS). To reduce overlapping exploration, the target selection also considers whether the frontier cells have already been chosen by other UAVs. After the selection process, the UAV will visit the targeted frontier cells by searching the shortest path using A* algorithm on a local passable graph (a graph which contains safe cells). A* algorithm is a shortest path finding algorithm by evaluating the path cost from start node to current candidate node and the estimated cost from current candidate node to targeted node by using Manhattan distance. To shorten the program, both A* and the BFS algorithm uses the networkx library.

D. Dynamic Communication Graph

Communication to other UAVs is represented by a dynamic graph since UAV positions change at every timestep. Each UAV is modelled as a node and each node connected by edge if the euclidean distance between UAV is less than or equal to communication range. In this simulation, communication range will be defined separately for each scenario.

E. Map Merging Mechanism

Map merging is used if UAVs are in the same connected component. They will share each other local map information, so that the knowledge of the UAVs are widened. The shared information includes known safe cells, known obstacle cells, known fire cells, and visited cells. If the UAVs are not connected, each UAV will continue to explore the map using their local map without additional shared information.

F. Scenarios and Metrics

To test the impact of the number of UAVS, communication range, and map merging, the simulation included 5 scenarios with a controlled environment. The first and second scenarios compare the effect of map merging using 3 UAVs within a 5-unit communication range. Meanwhile third, fourth, and fifth scenarios test the effect of communication range using 5 UAVs with map merging enabled. The metrics that will be used are map coverage percentage, fire detection ratio, average largest component ratio, one-node failure robustness, and timesteps executed. Table I below depicts each Scenario 1 to Scenario 5 conditions.

TABLE I. SIMULATION SCENARIOS

Scenario	Number of UAVs	Communication Range	Map Merging
1	3	5	No
2	3	5	Yes
3	5	5	Yes
4	5	7	Yes
5	5	10	Yes

Map coverage percentage is calculated by dividing the number of known cells by the total number of cells in the grid.

Known cells include known safe cells, known obstacle cells, and known fire cells. Fire detection ratio is calculated by dividing the number of detected fire hotspots by the total number of fire hotspots in the environment. In this simulation, the environment contains six fire hotspots.The average largest component ratio is used to measure the connectivity of the UAV communication graph. At each timestep, the largest connected component is identified, and its size is divided by the total number of UAVs. A higher value indicates that a larger portion of UAVs belongs to the same connected communication group. One-node failure robustness is used to measure the resilience of the communication graph. This metric is calculated by removing one UAV node at a time and checking whether the remaining graph stays connected. The final value is obtained by dividing the number of successful removals by the total number of UAV nodes. A higher value indicates that the communication network is more robust when one UAV fails. Timesteps executed represent the number of simulation steps required until the simulation stops. The simulation stops when map coverage reaches 100 percent or when the maximum timestep limit is reached. In this study, the maximum timestep limit is 250.

G. Simulation Workflow

The simulation process is executed iteratively based on timesteps. At the beginning of the simulation, UAVs are initialized at predefined starting positions with no prior knowledge of the complete environment. At each timestep, every UAV senses its surrounding cells based on the configured sensor radius and updates its local map with newly discovered safe cells, obstacles, and fire hotspots. After the sensing process, each UAV identifies available frontier cells from its local map and selects a reachable target frontier cell. The selected target is then used by the A* algorithm to determine the movement path. Each UAV moves one step along the generated path in every timestep.

After all UAVs update their positions, the communication graph is rebuilt based on the current UAV positions and the predefined communication range. UAVs that belong to the same connected component are allowed to merge their local map information. This process allows connected UAVs to share knowledge about safe cells, obstacles, detected fire hotspots, and visited cells. The simulation continues until all cells in the environment are known by the swarm or the maximum timestep limit is reached. During the simulation, several performance metrics are recorded to evaluate exploration performance, fire detection capability, and communication connectivity.

H. Implementation Tools

The simulation was implemented using Python with several supporting libraries. NetworkX was used to represent the grid environment and UAV communication network as graph structures. It was also used to support graph-based operations, including reachable path calculation, connected component analysis, BFS-based frontier selection, and A* path planning. NumPy was used to represent the environment as a matrix and to support numerical operations, such as coordinate calculation, distance computation, and array manipulation. Matplotlib was used to visualize the simulation results, including the grid environment, obstacle cells, fire hotspots, explored areas, UAV positions, and communication links between UAVs.

IV. Result and Analysis

A. Result

Figure 4 shows an example of the simulation result for Scenario 1. This scenario is presented as a representative visualization to illustrate how the UAV swarm explores the peatland environment, detects fire cells, and maintains communication connectivity during the monitoring process. The remaining scenario results are summarized quantitatively in Table II.

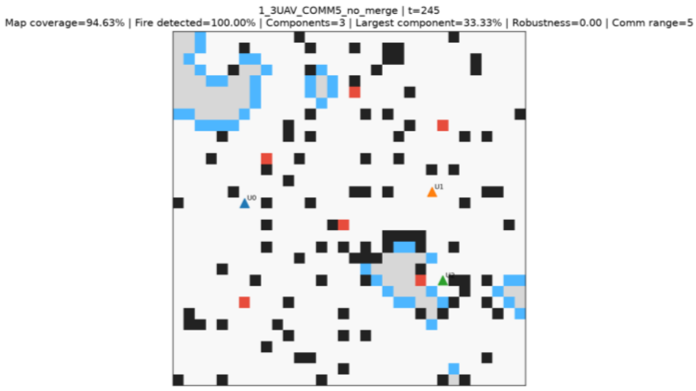

Fig 4. Representative simulation result of Scenario 1.

In the visualization, the black cells represent obstacles that cannot be crossed by the UAVs, while the red cells indicate fire locations. The blue cells represent the explored or monitored areas covered by the UAV sensors. The light gray cells represent safe areas in the environment, and the triangle markers indicate the positions of the UAVs.

To evaluate the overall performance of each scenario, several metrics were calculated, including map coverage ratio, fire detection ratio, average largest connected component, one-node failure robustness, and executed timesteps. The comparison of these metrics is presented in Table II.

TABLE II. Metric Results For Each Scenario

Scenario	**Map Coverage Ratio**	**Fire Detection Ratio**	**Average Largest Component**	**One-node Failure Robustness**	**Time Steps Executed**
1	95.02%	100%	38.53%	0.520	249
2	100%	100%	37.67%	0.0434	214
3	100%	100%	45.61%	0.0915	163
4	100%	100%	53.85%	0.0923	129
5	100%	100%	73.23%	0.1662	129

Table II shows that within fewer than 250 timesteps, only Scenario 1 did not achieve full map coverage. However, Scenario 1 still successfully detected all fire hotspots. Scenario 5 achieved the highest average largest connected component ratio and one-node failure robustness, while Scenario 2 obtained the lowest values for these metrics. The simulation results also show that the number of executed timesteps decreased from Scenario 1 to Scenario 5.

B. Analysis

1) Map Merging: By comparing scenario 1 to others, it was discovered that scenario 1 has the lowest map coverage ratio. It is plausible since the scenario does not allow map merging so that the exploration will be redundant. Unlike Scenario 2, by sharing local map information, the exploration will be faster and wider. In this case, the difference might not be significant, but if the area is wider then it probably will show a significant difference. The efficiency of map merging is also shown in timesteps executed. Scenario 1 needed more timesteps to cover the whole map than Scenario 2. Additionally, map merging affected the one-node failure robustness and average largest component ratio. For instance, Scenario 1 resulted in higher value in both metrics than Scenario 2. This indicates map merging caused wider exploration, but did not guarantee a more robust communication network.

2) Communication Range: The effect of communication range can be observed by comparing Scenarios 3, 4, and 5, since these scenarios use the same number of UAVs and enable map merging, while only the communication range is changed. The results show that increasing the communication range tends to improve the connectivity of the UAV communication graph. This is indicated by the increase in the average largest component ratio, which means that a larger portion of UAVs can remain within the same connected communication group. However, the improvement in one-node failure robustness is not always immediate. Although Scenario 4 uses a larger communication range than Scenario 3, both scenarios produce a similar robustness value. Therefore, a larger communication range can improve communication connectivity and network resilience, but its effect depends on whether the resulting communication graph has enough redundant paths.

3) Number of UAVS: The effect of the number of UAVs can be observed by comparing Scenario 2 and Scenario 3, since both scenarios use the same communication range and enable map merging. The main difference is that Scenario 2 uses 3 UAVs, while Scenario 3 uses 5 UAVs. The results show that increasing the number of UAVs improves the exploration efficiency. With more UAVs, the environment can be sensed and explored by more positions, allowing the swarm to discover frontier cells more quickly.

V. Conclusion and Future Work

This study presented a dynamic graph-based UAV swarm simulation for peatland fire monitoring with communication connectivity constraints. The simulation evaluated several scenarios based on the number of UAVs, communication range, and map merging capability. The results show that all scenarios were able to detect fire hotspots, while map coverage and exploration efficiency varied depending on the scenario configuration. Scenario 1, which did not use map merging, had the lowest map coverage and required the highest number of executed timesteps. In contrast, scenarios with map merging showed better exploration efficiency because UAVs were able to share local map information. Increasing the number of UAVs and communication range also improved the exploration process and communication connectivity. Scenario 5 achieved the best overall performance in terms of average largest connected component ratio, one-node failure robustness, and executed timesteps. These results indicate that communication range, UAV quantity, and information sharing mechanisms play important roles in improving the effectiveness of UAV swarm-based peatland fire monitoring.

This study still has several limitations. The simulation uses a simplified grid-based environment and assumes a fixed communication range for each scenario. Real-world factors such as wind direction, smoke density, signal interference, UAV battery capacity, and dynamic fire spread are not explicitly modeled. Therefore, the results should be interpreted as a simulation-based analysis rather than a complete representation of real peatland fire monitoring operations.

For future work, the simulation can be extended by implementing more advanced path planning algorithms, such as Rapidly-exploring Random Tree (RRT), to support navigation in larger and more complex environments. The system can also be improved by considering dynamic environmental disturbances, such as smoke, wind direction, and reduced visibility, which may affect sensor accuracy and communication connectivity between UAVs. In addition, future research can expand the scope of the system from fire monitoring to fire suppression support, where UAVs not only detect fire locations but also assist in coordinating early extinguishing actions. Further evaluation using more realistic terrain models and dynamic fire spread behavior would also help improve the accuracy and applicability of the proposed simulation.

Video Link at Youtube

https://youtu.be/mQ7gPeZG9gg?si=6nmWiVG-XhFMM2Ma

Github Repository Link

https://github.com/michrousb/UAV-Swarm-Peatland-Monitoring-MatDis---13525013

Acknowledgment

The author would like to express sincere gratitude to God Almighty for His blessings, strength, and guidance throughout the completion of this study. The author also thanks her parents and family for their continuous prayers, encouragement, and support. Sincere appreciation is also extended to the lecturer and teaching assistants of the Discrete Mathematics IF1220 course for their guidance, feedback, and support during the development of this study. The author also appreciates the help, discussions, and motivation provided by friends, colleagues, and all other parties who contributed directly or indirectly to the completion of this paper.

REFERENCES

[1] D. Arisanty, M. Z. A. Anis, H. P. N. Putro, M. Muhaimin, and Syarifuddin, *Kebakaran Lahan Gambut: Faktor Penyebab dab Mitigasinya*. Banjarmasin: Program Studi Pendidikan IPS Fakultas Keguruan dan Ilmu Pendidikan Universitas Lambung Mangkurat, 2020. [Accessed: Jun. 19, 2026].

[2] G. Tzoumas, L. Pitonakova, L. Salinas, T. Richardson, C. Scales, and S. Hauert, "Wildfire detection in large-scale environments using force-based control for swarms of UAVs," *Swarm Intell.*, vol. 17, no. 1, pp. 89–115, Jun. 2023, doi: 10.1007/s11721-022-00218-9. [Accessed: Jun. 19, 2026].

[3] W. T. L. Teacy, J. Nie, S. McClean and G. Parr, "Maintaining connectivity in UAV swarm sensing," 2010 IEEE Globecom Workshops, Miami, FL, USA, 2010, pp. 1771-1776, doi: 10.1109/GLOCOMW.2010.5700246. [Accessed: Jun. 19, 2026].

[4] M. E. Harrison *et al.*, "Tropical forest and peatland conservation in Indonesia: Challenges and directions," *People and Nature*, vol. 2, no. 1, pp. 4–28, Mar. 2020, doi: 10.1002/pan3.10060. [Accessed: Jun. 19, 2026].

[5] R. Munir, Graf (Bag.1), lecture notes, IF1220 Matematika Diskrit, Institut Teknologi Bandung, Bandung, Jawa Barat, 2024. [Accessed: Jun. 19, 2026].

[6] S. Beamer, K. Asanović, and D. Patterson, "Direction-optimizing breadth-first search," *Scientific Programming*, vol. 21, no. 3-4, pp. 137–148, 2013, doi: 10.3233/SPR-130370. [Accessed: Jun. 19, 2026].

[7] M. Cruz, R. Carvalho, A. Costa, L. Pinto, L. Dias, P. Cerqueira, R. Machado, T. Batista, P. Castro, and J. Ribeiro, "A Contribution of Shortest Paths Algorithms to the NetworkX Python Library," *Applied Sciences*, vol. 15, no. 15, Art. no. 8273, Jul. 2025, doi: 10.3390/app15158273. [Accessed: Jun. 19, 2026].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2026

Samantha Michelle S. Silaban 13525013