Gridding and Optimization of a Drone Swarm

Adithya Sanjeev Byalpi
Department of Electronics and Communication
National Institute of Engineering
Mysore, India
adithya.sanjeev@outlook.com

Prathvi Rao
Department of Electronics and Communication
National Institute of Engineering
Mysore, India
prathvirao0@gmail.com

Sanjana Raj R V
Department of Electronics and Communication
National Institute of Engineering
Mysore, India
rvsanjanaraj@gmail.com

***Abstract*— The foremost action taken during and after the occurrence of any calamity or natural disaster is the assessment of the situation. This is done through reconnaissance missions conducted by various means. A recon mission helps a command center to identify areas where the damage is the highest and where an assignment of priority is essential. On identifying these areas, it becomes efficient and easier to battle the odds and allocate resources accordingly. This is on an assumption that the required objective is to deploy a swarm of drones to conduct a reconnaissance mission. The drones have an excellent vantage point and an ample amount of edge computing resources with access to cloud computing as well. Using the drone swarm, identification on calamity-stricken areas becomes easy, with minimal human intervention. As a part of this project, we aim to develop a mathematical model which able to provide the drone command center with optimal grids allotted to each drone in the swarm. Based on the location, a larger area will be efficiently divided, by taking various parameters into account.**

Keywords—Drone Swarms, Internet of Things, Gridding

I. INTRODUCTION

The gridding model presented in the paper aims to efficiently divide the intended area into grids which can be covered by drones. This enables the drones to efficiently scout disaster-prone areas and execute reconnaissance missions. The gridding is done by a mathematical model which takes factors related to drones into consideration. These factors can include, but not be limited to – the number of drones, their battery life, the coverage area of the drones, etc. The grids will be used by the drone swarm in their reconnaissance missions in order to identify areas of interest and priority when a calamity strikes. The idea of a drone swarm is to maximize the efficiency and the delay overhead incurred in analyzing and undertaking actions to ameliorate the dangers posed. The concept proposed in this paper aims to assist the swarm by allotting each drone in the swarm its designated area. This area is calculated such that the drone can cover the grid completely.

Once an API from the command center receives a distress location, a master grid with its edge boundary points is created. These points are passed on to grid control, in a KML file. The mathematical models make use of the concept of line division and area division. These models take the above factors into consideration and give out the edge boundary points of the grids as a KML file again. This is used for further processing.

In order to accomplish the above task, QGIS [4] - a geographic platform which supports viewing, editing, and analysis of geospatial data, Python language with Geographic libraries [2] are used. The mathematical model and Global Positioning System (GPS) use the World Geodetic System (WGS84) as its reference coordinate system.

II. PROPOSED WORK

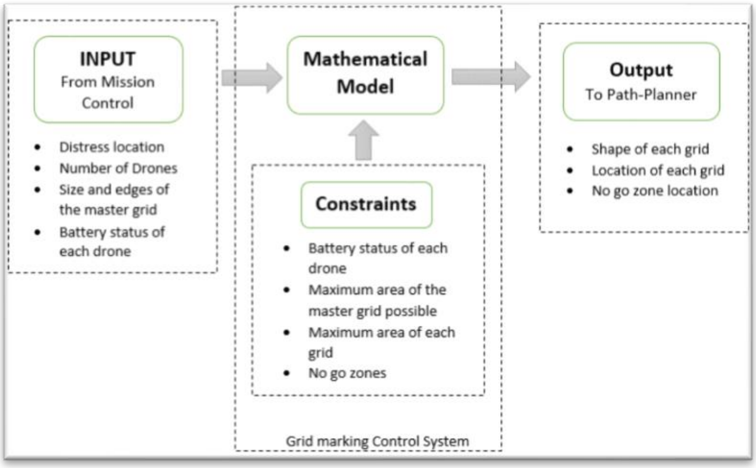

Figure 1. Gridding Model

The model achieves the objective as follows:

- The command center provides the coordinates of the master grid in the Keyhole Markup Language [5] (KML) format.
- Coordinates of the master grid are joined to form a polygon.
- The area of the master grid is compared with the maximum area that can be covered using the available number of drones.
- If at all the area cannot be covered using the available number of drones, a message is sent to the command center to intimate the nearby stations to cover the remaining area.
- A part of the master grid that can be covered is taken up and the area is divided into smaller grids. The remaining area that is to be covered by the neighboring stations also must be divided into smaller grids.
- The division of the master grid into smaller grids is based on one of the two algorithms proposed in the paper.
- Once the smaller grids are obtained, the coordinates of the smaller grids are passed to the command center in the KML format.

The same can be seen in Figure 2.

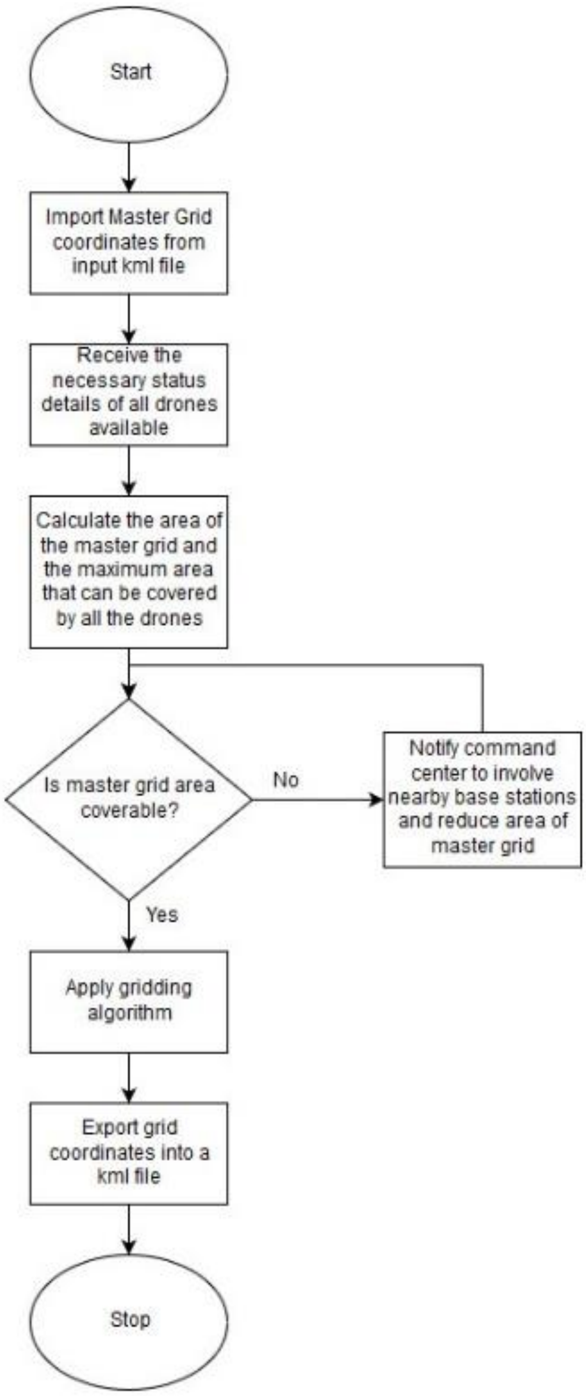

Figure 2. Working of the Gridding Model

The model makes use of these modules during its execution:

A. QGIS

QGIS, standing for Quantum Geographic Information System, is a free open-source platform used to edit, render and analyze geospatial data. It provides support for open street maps and other such maps. The algorithms presented in this paper make use of QGIS as a visual overlay to depict the coordinates provided as the input, as well as the coordinates of the smaller individual grids provided as the output.

B. Keyhole Markup Language (KML)

KML, standing for Keyhole Markup Language, is an XML notation used for describing geospatial data. In order to maintain compatibility with other existing solutions which manipulate geospatial data, the mathematical model uses KML files containing coordinates as it's input and output.

C. Python Libraries

Python libraries consist of various built-in modules that provide various functionalities. These libraries help users minimize the amount of coding that might have been required for executing various functionalities. The model makes use of geographic libraries which are python libraries for geospatial data. The model and the Global Positioning System (GPS) use WGS84 (World Geodetic System) as its reference coordinate system. Geodesic calculations are used in order to process and manipulate data represented by coordinates. In order to process KML files, PyKML and SimpleKML libraries are used.

III. Proposed Algorithms

The gridding model makes use of any one of the mathematical algorithms proposed in the paper. The selected algorithm is used to divide the larger master area where the distress is identified into smaller grids. Each grid is assigned for a drone to cover. The proposed algorithms in this paper are the area division algorithm, line division algorithm, and the spiral algorithm.

A. Line Division Algorithm

- In this algorithm, the master grid is divided into smaller grids using a line segment.
- By considering square as an example, a line segment joining two opposite corners of the square is drawn. This divides the square into smaller triangles.
- If at all this area cannot be covered using an available number of drones then again a line segment is drawn which further divides the grids into far more smaller grids.
- In this manner, a master grid can be divided into smaller grids using the line division algorithm.
- This concept of the line division algorithm holds good for other polygons as well.

B. Area Division Algorithm

- In this algorithm, the entire master grid is divided into smaller grids by considering the area.
- In the first attempt, this algorithm divides the entire master grid into smaller grids based on area.
- For instance considering a square, the larger square that is the master grid is divided into smaller squares of equal areas.
- If at all, this area cannot be covered by an available number of drones then again the smaller squares are further divided into squares of the lesser area which can be covered by the drones.

- This concept of area division algorithm holds good for other polygons as well.
- But the major disadvantage of this algorithm is that a part of the master grid would be left uncovered while trying to divide it into smaller grids.

C. Spiral Algorithm

- The coordinates are imported from the KML file.
- The imported file consists of 4 coordinates of the required area.
- Taking the case that higher priority to the center of the area, the center position is calculated.
- The maximum area that each drone can handle is calculated based on the parameters of the drones
- The length of the 1st square is calculated considering the condition that the maximum area for the last drone to traverse must be less than the maximum possible area coverable for each drone.
- After finding the coordinates of the 1st square, other coordinates are calculated depending on their Fibonacci values. The relationship between the Fibonacci values and the squares is such that their area depends on the square of the respective Fibonacci value. Also, the length of the diagonal is proportional to the respective Fibonacci value.
- Using the above relationship, the coordinates of diagonals of individual squares are calculated. The formula to calculate the length of the diagonal is given by $Diag_{len} = m * x * \sqrt{2}$. The initial point of the diagonal is known, and the other endpoint is calculated by translating the $Diag_{len}$ distance into coordinates based on the following condition:
  Taking count as the iteration number,
    - If count%4==0 then traverse 45°
    - If count%4==1 then traverse 135°
    - If count%4==2 then traverse 225°
    - If count%4==3 then traverse 315°

  After completing the iterations, the 2 points of diagonals of individual squares are obtained.
- Using the above two points, the other two points are calculated to get all four coordinates of the square.
- This is done for all the grids assigned to the drones.
- These points are then exported into a kml file.

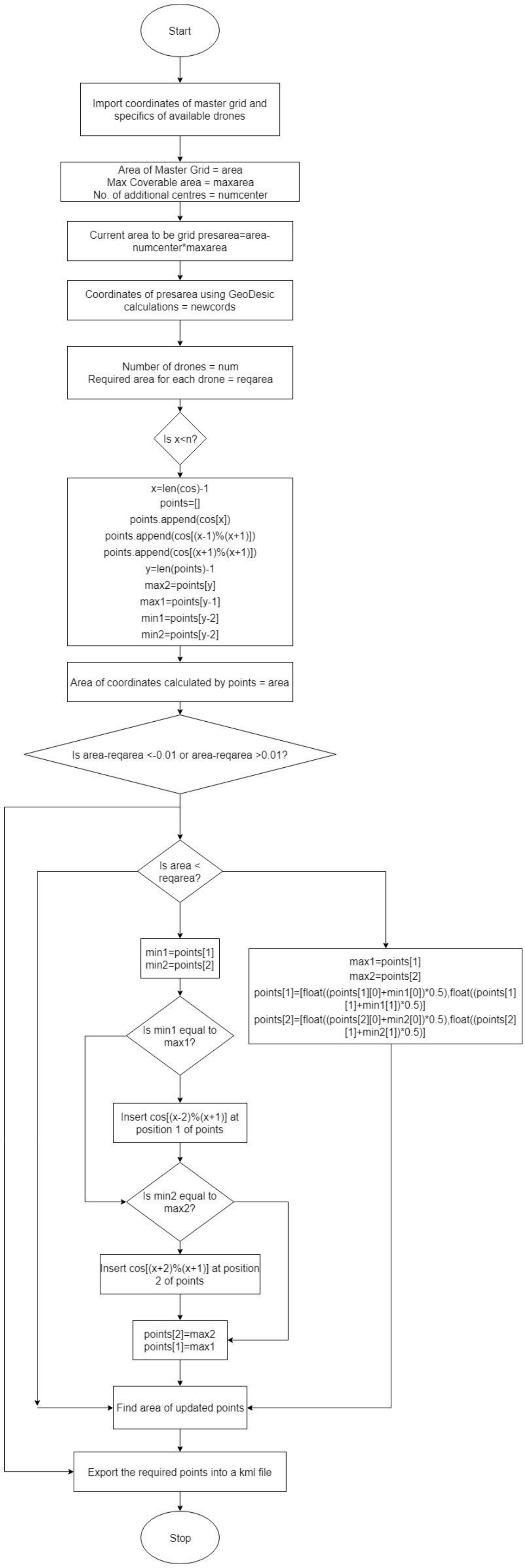

Figure 3. Line Division Algorithm

IV. Result

The outputs of all the algorithms are tabulated and depicted in this section. The coordinates of the master grid and the divided grids can be seen in Table 1, 2, 3 and 4. The same can be seen on the map overlay in Figures 4, 5 and 6.

Table 1. Master Grid

	Input
1	74.78705, 15.32038
2	74.78705, 15.321
3	74.787661, 15.321
4	74.787661, 15.32038

Table 2. Line Division Algorithm Output for 5 drones

Line Division Algorithm					
1	74.78744,15.32037	74.78704,15.32037	74.78704,15.32076		
2	74.78747,15.32037	74.78744,15.32037	74.78704,15.32076	74.78704,15.32098	74.78712,15.32098
3	74.78739,15.32052	74.78736,15.32057	74.78749,15.32098	74.78767,15.32098	74.78767,15.32081
4	74.78736,15.32057	74.78712,15.32098	74.78749,15.32098		
5	74.78767,15.32037	74.78747,15.32037	74.78739,15.32052	74.78767,15.32081	

Table 3. Area Division Algorithm Output for 5 Drones

Area Division Algorithm				
1	74.78705,15.32065	74.78705,15.32038	74.78732,15.32037	74.78732,15.32065
2	74.78732,15.32065	74.78732,15.32037	74.78760,15.32037	74.78760,15.32065
3	74.78760,15.32065	74.78760,15.32037	74.78788,15.32037	74.78788,15.32065
4	74.78705,15.32092	74.78705,15.32065	74.78732,15.32065	74.78732,15.32092
5	74.78732,15.32092	74.78732,15.32065	74.78760,15.32065	74.78760,15.32092

Table 4. Spiral Division Algorithm Output for 5 Drones

Spiral Algorithm				
1	74.78735,15.32068	74.78761,15.32068	74.78761,15.32093	74.78735,15.32093
2	74.78761,15.32093	74.78786,15.32093	74.78786,15.32068	74.78761,15.32068
3	74.78786,15.32068	74.78735,15.32068	74.78735,15.32019	74.78786,15.32019
4	74.78735,15.32019	74.78659,15.32019	74.78659,15.32093	74.78735,15.32093
5	74.78659,15.32093	74.78786,15.32093	74.78786,15.32217	74.78659,15.32217

Figure 4. Master Grid Overlay

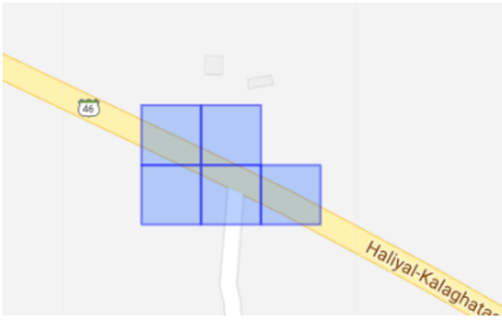

Figure 5. Area Division output overlay

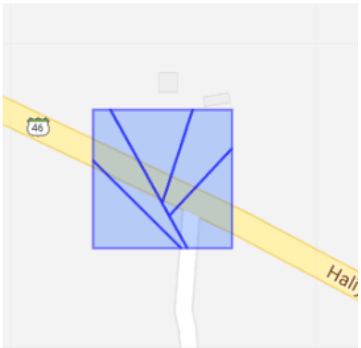

Figure 6. Line Division Overlay

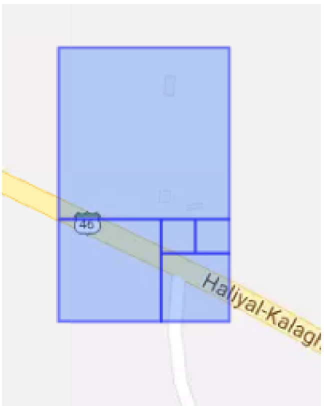

Figure 7. Spiral Algorithm Overlay

V. CONCLUSION

In order to compare the algorithms presented in this paper, we take the following factors into consideration. Each factor is considered over a range of 5, 10 and 15 drones. For each number of drones, 5 iterations are run, and the average result value is taken.

A. *Operation time*

Since the model is going to be applied in mission-critical situations, the amount of time taken to process the output is highly paramount as delays incurred can deteriorate the success of the objective.

The run time is shown in Table 5.

Number of drones	Time in seconds
Area Division Algorithm	
5	1.000037193
10	1.251970291
15	1.375807762
Line Division Algorithm	
5	1.391474009
10	1.883895636
15	2.509415388
Spiral Algorithm	
5	0.83661437
10	0.922720432
15	1.088864565

The results are calculated on a computer which has the following parameters: 2.5 gigabytes of Random-Access Memory, 4% 0f 0.75 Gigahertz Central Processing Unit running 88 processes and 1100 (± 25) threads.

As we can see, the processing time increases in a fairly linear fashion for the area division and the spiral algorithm. Whereas, we can see an exponential increase in the processing time of the spiral algorithm.

B. *Loss of area*

The algorithms presented can sometimes not account for the complete area present in the master grid. The area division algorithm and the spiral algorithm both have loss of areas as the output grids are restricted to the shape of a square. Due to this, the loss of area covered in the edges of the master grid is possible. This can be seen in Figures 6 and 7.

Figure 8. Loss of area due to the Area Division Algorithm

C. *Variation in the areas allocated to drones*

The area division algorithm and the line division algorithm have no difference in the area allocated to each drone. The spiral algorithm can have output grids whose areas vary based on the equation $(m^2 - 1) * x^2$ where m is (Number of drones + 1) and x is the length of the side of the grid.

D. *The shape of the grids created*

The shape of the output grids in the spiral algorithm and the area division algorithm is restricted to squares, while the line division algorithm is able to provide grids in terms of any number of edges of the polygon.

ACKNOWLEDGMENT

We acknowledge the guidance of Mr. Rohith Prasanna, Assistant Professor, Department of Electronics and Communication, National Institute of Engineering, Mysore for his insights and guidance throughout the course of formulating this paper.

REFERENCES

[1] M´arton G´otzy, D´aniel Het´enyi, L´aszl´o Bl´azovics, “Aerial Surveillance System with Cognitive Swarm Drones”, 28th International Conference 2016 Cybernetics and Informatics (K&I), Levoca, Slovakia

[2] PyQGIS Developer Cookbook: https://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/

[3] Polygon Splitting Algorithm: http://www.khetarpal.org/polygon-splitting/

[4] QGIS: https://qgis.org/en/docs/index.html

[5] KML: https://developers.google.com/kml/documentation/kmlreference