Multi Drone Simulation with Fish School Search Algorithm

Guntur P. B. Knight, Hadumanro Malau, and Mulya D. G. Ginting

***Abstract*— Drone simulation is developed using the Fish School Search (FSS) algorithm. This simulation aims to optimize the coordination of multiple drones for efficient scanning of designated areas, leveraging Swarm Intelligence principles. The motivation behind this study is to determine the optimal number of autonomous systems in surveillance, environmental monitoring, and disaster management, based on the area. The objective is to determine the optimal number of drones required for scanning different area sizes, enhancing the efficiency and effectiveness of the drones used. The area used in this simulation is a rectangular area. The Python programming language is employed to implement and test the FSS algorithm. Results indicate that increasing the number of drones does not linearly decrease the scan time for each area size, highlighting the need for optimized drone coordination. Based on the scenarios used in this simulation it is concluded that the optimal scenario is 6 drones used in 17 x 17 rectangular grid which need 62.721 seconds to cover the entire rectangular area. Surely this simulation can be verified using real experiments using mapping on the C section of the experiment part.**

***Index Terms*— Fish School Search algorithm, Multi-Drone coordination, Swarm intelligence, Python simulation.**

I. Introduction

DRONE or Unmanned Aerial Vehicles (UAVs), are aircraft without a human pilot aboard. Today, UAVs are of great interest in broad areas of applications, such as military reconnaissance, firefighter operations, police pursuit, environmental monitoring, and disaster response due to their ability to operate in hazardous environments and provide real-time data [1][2]. The more and more advanced technologies enable the UAVs to perform tasks with longer distance, more accurate maneuvering, more efficient communication qualities and so forth.

The single-UAV flight distance is around 5 km, once the command has been issued by the controller, the drone executes and gives feedback via a wireless network. This end-to-end communication method has three possible design flaws:

- the communication quality is dependent on the travel distance of the drone, where the increased distance of the UAV yields a poorer connection;
- the short-range response time limits this drone from tasks that require a relatively long distance; and
- the drone is not able to respond to the change of aerial environment intelligently and give feedback in time, so that if something happened to the drone that terminated the commanding channel, the UAV-at-large might not be able to retract back to the user.

Under this circumstance, a new applicable tactic has been suggested in coping with the abovementioned disadvantages, namely multi-UAV (multi-drone) systems. The motivation for this study stems from the need to improve the efficiency and effectiveness of multi-drone operations in these applications. By employing Swarm Intelligence (SI) principles, specifically the Fish School Search (FSS) algorithm, this research aims to establish the coordination among multiple drones, which make manual coordination by human intervention unnecessary [3][4].

Swarm Intelligence is inspired by the collective behavior of social animals, where simple individual behaviors unexpectedly lead to complex group dynamics [5][6]. Swarm intelligence can be observed in birds' behaviors, school of fishes, ants, and geese flying in V shape, to name a few examples.

The swarm intelligence in this paper is the Fish School Search (FSS) Algorithm. This algorithm mimics the natural behavior of school of fish to solve optimization problems. This study focuses on developing a multi-drone simulation to test the efficacy of the FSS algorithm in coordinating multiple drones for area scanning tasks [7][8].

II. Research Methodology

A. General Description

The multi-drone simulation software is designed to perform area scanning using multiple drones. The stages of this research are depicted in Fig. 1 and are elaborated upon below.

The flowchart in Fig. 1 outlines the process flow from initializing the drones and area, implementing the FSS algorithm, performing area scanning, and analyzing the results. The narration elaborates on each step as follows: (i) Initialization: Drones and the scanning area are initialized with specific parameters; (ii) FSS Algorithm Implementation: The algorithm calculates the barycenter and updates the drones' positions iteratively; (iii) Area Scanning: Drones move to scan the assigned areas based on the FSS algorithm; (iv) Result

G. P. B. Knight is with the Electrical Engineering Study Program Institut Teknologi Del, Toba, Sumatera Utara 22381 Indonesia (corresponding author, e-mail: guntur.siboro@del.ac.id*).

H. Malau is with the Department of Electronic & Electrical Engineering, University College London, London, United Kingdom (e-mail: hadumanro.malau.19@ucl.ac.uk).

M. D. G. Ginting is with the Electrical Engineering Study Program Institut Teknologi Del, Toba, Sumatera Utara 22381 Indonesia (e-mail: mulya.gianta@gmail.com).

ISSN: 1942-9703 / CC BY-NC-ND

Analysis: The performance metrics, including scan time and optimal number of drones, are analyzed.

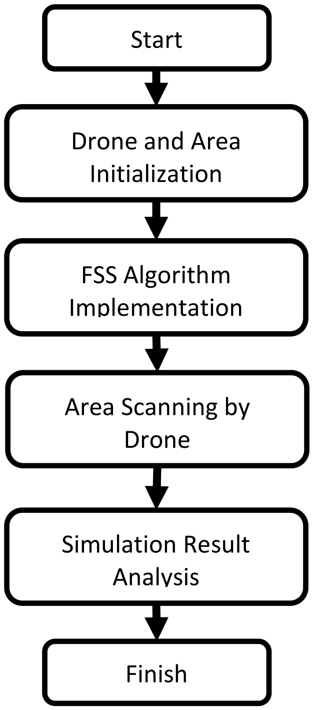

Fig. 1. Schematic of multi drone with FSS algorithm

B. *Modeling Idea*

The simulation is modeled in a 2-dimensional grid with drones categorized as main and follower drones (Fig. 2). The model operates on the following rules: Follow the main drone; Keep drone distance from the main drone; and Keep drone distance from other drones. These rules are governed by the FSS algorithm, which uses the concept of drone "weight" to manage distance maintenance, as illustrated by the equations provided.

The mechanism for maintaining distance (X_i^{iter}) is based on the weight (W_i^{iter}) of each drone, this is because the algorithm used in the drone simulation is FSS which has a weight parameter. Drones that have a lighter weight will keep their distance from heavier drones [3]. The weight is assumed to be uniform. s_v represent the maximum displacement performed by drone and r_v represent random number uniformly distributed from 0 to 1. The equation in the FSS algorithm is as follows:

$$B^{iter} = \frac{\sum_{i=1}^{N} X_i^{iter} W_i^{iter}}{\sum_{i=1}^{N} W_i^{iter}} \quad (1)$$

$$X_i^{iter+1} = X_i^{iter} - s_v r_v (X_i^{iter} - B_i^{iter}) \quad (2)$$

$$X_i^{iter+1} = X_i^{iter} + s_v r_v (X_i^{iter} - B_i^{iter}) \quad (3)$$

B^{iter} is the barycenter in FSS Algorithm where drone move into barycenter point. The FSS algorithm equations ensure that drones move towards a central point (barycenter), optimizing their positions for effective area scanning (Fig. 3).

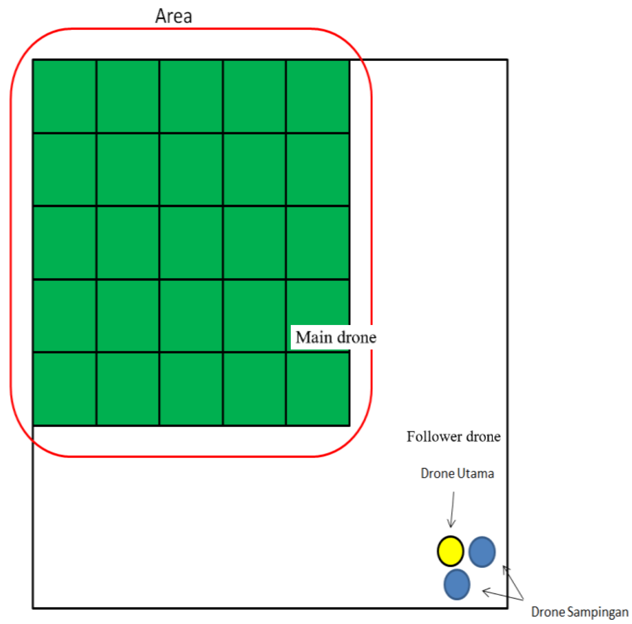

Fig. 2. Multi drone simulation modeling ideas

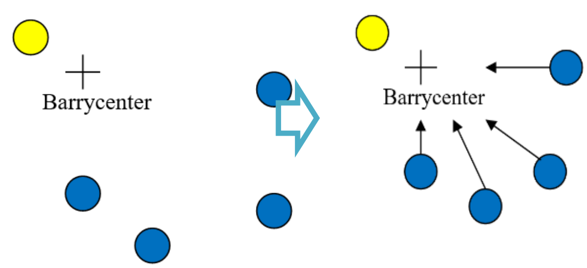

Fig. 3. All drones except the main drone are moving toward the barycenter

C. *Neighbor Concept and BFS Algorithm*

Drones use the Moore neighbor concept for area scanning, which considers 8 neighboring positions around each drone, as depicted in Fig. 4, 5, and 6. This concept is crucial for ensuring comprehensive area coverage [9].

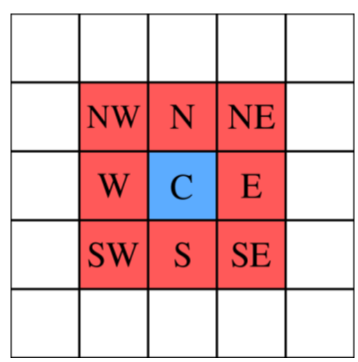

Fig. 4. Moore's neighbor concept

The BFS algorithm is employed to systematically visit all neighboring nodes, ensuring no area is left unscanned [10].

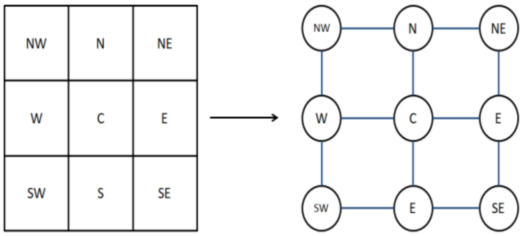

Fig. 5. Moore's neighbor concept in graph form

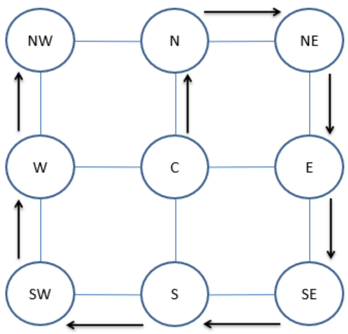

Fig. 6. BFS Search Algorithm implemented in Moore's neighborhood

D. *Hardware and Software Platform*

Following are the hardware and software specifications used to create the simulation.

1. Hardware,
   - a. PC : Dell OptiPlex 7050
   - b. CPU : Intel Core I7
   - c. RAM : 8Gb
   - d. OS : Windows 10 Pro 64-bit
2. Software used was Python version 3.10.7.

III. Result and Discussion

Multi-drone simulation has been implemented with the Python programming language and the results of the implementation are as follows.

A. *Implementation Scenario*

The implementation of the simulation includes the initialization of drones and the scanning area, the application of the FSS algorithm, and performance evaluation. In this simulation, each drone is represented by a circle (blue for the main drone and yellow circle for the follower drone, Fig 7). The area to be scanned is represented by grid cells (e.g. 15 x 15, 16 x 16 grid, Fig 8). If a drone flies by the grid (touches the grid) the cell will change color (from green to orange) which indicates the cell has been scanned. (Fig. 9).

The FSS algorithm provides drones with intelligence capabilities, enabling them to know their positions relative to others and adjust their movements accordingly as seen in Fig 10. The figure showed a condition in which two drones (main and follower drone) approached the targeted grid cell from a certain distance.

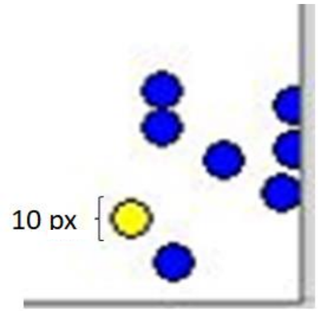

Fig. 7. Initialize 8 drones in a circle

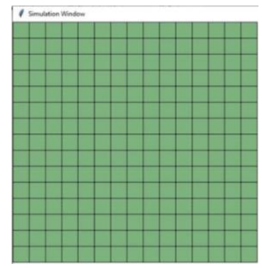

Fig. 8. Initialize 15 x 15 grid cell areas in a square

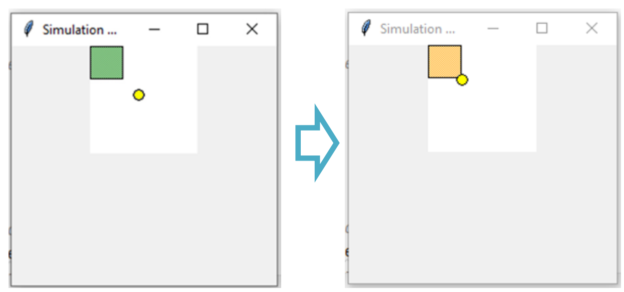

Fig. 9. The area color changes when the drone interacts with the area

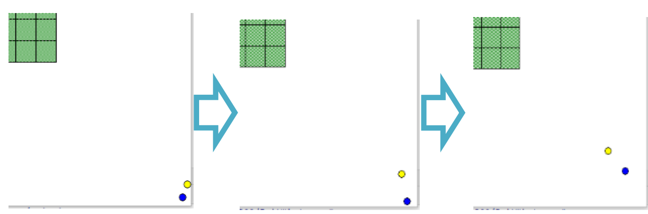

Fig. 10. Follower drones keep their distance and follow the main drone

The implementation of the simulation includes the initialization of drones and the scanning area, application of the FSS algorithm, and performance evaluation. As the drones move to the targeted grid cell, the cells that are touched by drones change color (Fig. 11). Common occurrences in these simulations are the likelihood of each cell being scanned by multiple drones. This happened because there are no specific mechanisms of movement for each drone as seen in Fig. 11.

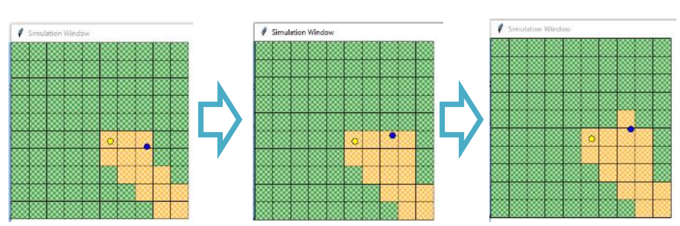

Fig. 11. Follower drones do a search when the main drone is in the middle of the entire area

B. Simulation Result

The simulation scenario tests various drone counts (from 2 to 20 drones) and area sizes (from 15×15 to 20×20 grid cells) to determine performance metrics [11].

1. Optimal number of Drones and Time

Graphs are provided to show the relationship between the number of drones and the scan time for different area sizes.

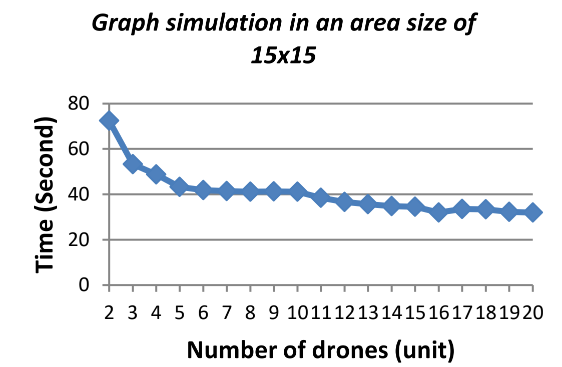

Fig. 12. Graph of scanning time against the number of drones in an area size of 15×15

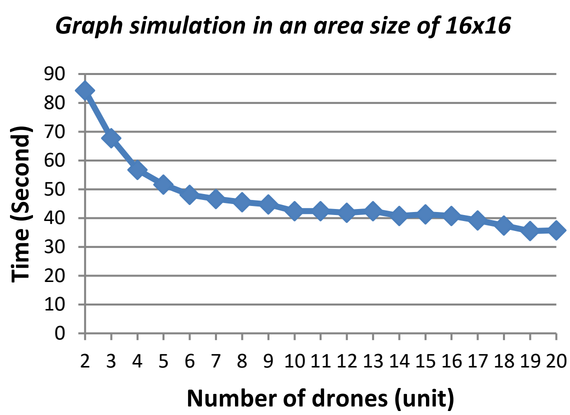

Fig. 13. Graph of scanning time against the number of drones in an area size of 16×16

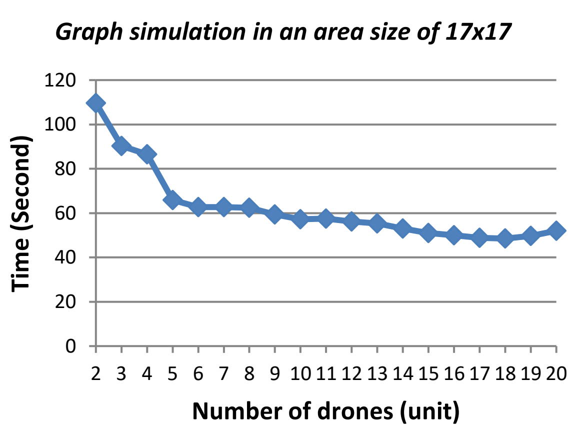

Fig. 14. Graph of scanning time against the number of drones in an area size of 17×17

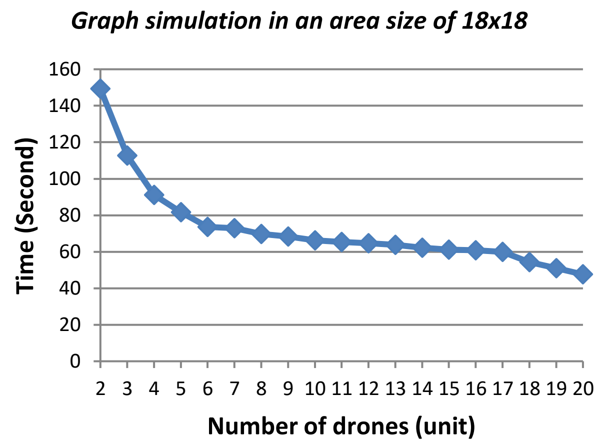

Fig. 15. Graph of scanning time against the number of drones in an area size of 18×18

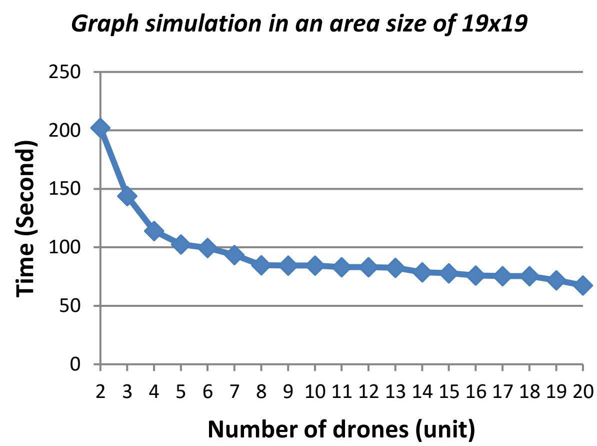

Fig. 16. Graph of scanning time against the number of drones in an area size of 19×19

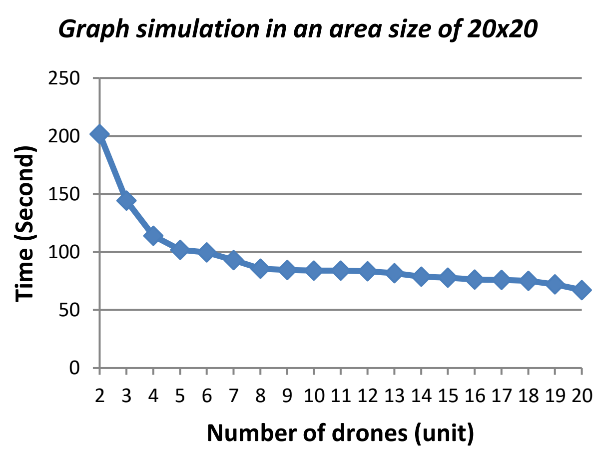

Fig. 17. Graph of scanning time against the number of drones in an area size of 20×20

2. Summary of simulations results

The summary table presents the optimal number of drones and their respective scanning times for various area sizes [12].

From all experiments on each area size, the results of data processing show that the effect of area size on the number of drones does not indicate that the area size affects the number of drones needed, this can be seen in the 19×19 and 20×20 size experiments which should decrease due to the 15×15 to 18×18 size experiment, experienced a decrease in the optimal number of drones required.

Based on the tables and graphs of previous experiments, the number of drones ranging from 2 to 5 drones experienced a steeper decrease in scanning time than the rest; this was due to the fewer number of drones searching for more areas that had not been scanned, as shown in Fig. 18, as follows.

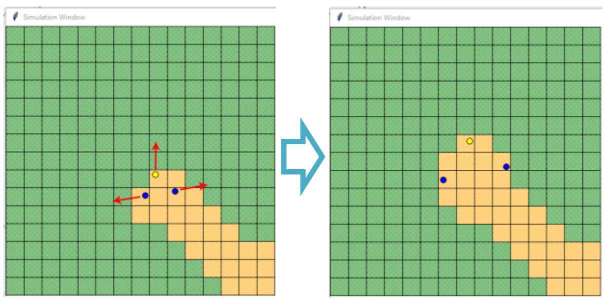

Fig. 18. Simulation of 3 drones on an area size of 15×15

TABLE I
AREA SIZE DATA AGAINST THE OPTIMAL NUMBER OF DRONES0

Area size	The optimal number of drones (Unit)	Scanning Time (Second)
15x15	8	41.126
16x16	10	42.463
17x17	6	62.721
18x18	6	73.704
19x19	9	84.536
20x20	10	84.079

Then on the number of drones starting from 8 onwards, the decrease in scanning time seems to be gradual; this is because the last drone to enter the area does not find neighboring areas that have not been scanned, so the drone will look for other areas randomly. This is another drone that will pass through the scanned area, so the following can be seen in Fig. 19 as follows.

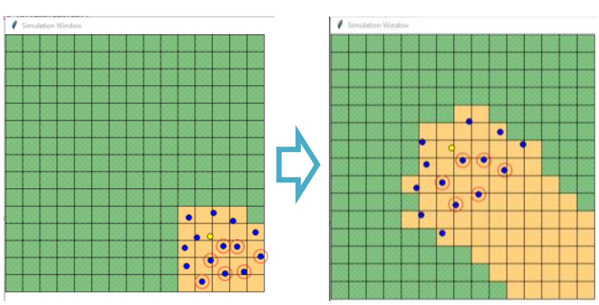

Fig. 19. Simulation of 8 drones on an area size of 15×15

3. The efficiency of the FSS algorithm in scanning the area

Efficiency in scanning the area is the number of areas that have been traversed by all drones divided by the number of areas to be scanned [13]. So that 100% efficiency is a state where all drones do not pass through the scanned area (exclusive). Efficiency is formulated in (4) as follows.

$$e = \frac{a}{\sum_{i=0}^{n} k_i} \tag{4}$$

a is the number of areas, n is the number of drones, and k is the number of areas that have been traversed by the i-th drone. Each data has been collected and processed from the simulation, showing the relationship data between scanning time and efficiency data aimed at determining which area size is optimal according to the efficiency of many drones in the following tables and graphs.

From the results of the graphic Fig. 20, it can be concluded that each area size displays each efficiency, from the data the highest efficiency is in the data to size 17×17 with an efficiency value is 50.93% so that the data is the most optimal value between sizes 15×15 to 20×20 and total drones between 2 to 20 drones.

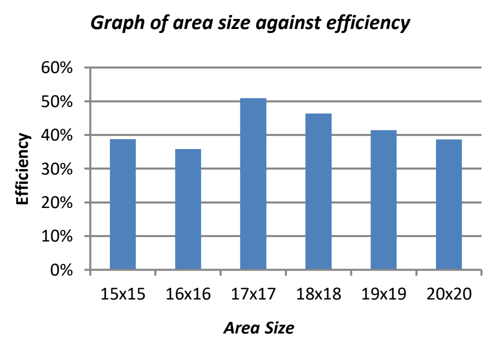

Fig. 20. Graph of area size against efficiency

C. Simulation verification to the real world

To verify the simulation into the real world, a transformation variable is needed. This variable is obtained from a comparison between the size of the object in the real world and the size in the simulation. Objects that can be compared in the real world are drones. This is because the drone object has been sized so that the size of the area in the real world can be adjusted.

TABLE 2
AREA SIZE DATA ON THE OPTIMAL NUMBER OF DRONES ALONG WITH EFFICIENCY

Area size	The optimal number of drones (Unit)	efficiency
15x15	8	38.71%
16x16	10	35.84%
17x17	6	50.93%
18x18	6	46.39%
19x19	9	41.45%
20x20	10	38.69%

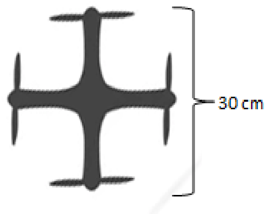

Fig. 21. The drone assumes a size of 30 cm

Assume that the size of the drone in the real world is 30 cm in diameter as shown in the Fig. 21. So that the value of the transformation variable (x) can be determined in (5) as follows with $D_s\ is\ 10\ px$ and $D_r\ is\ 30\ cm$;

$$x = \frac{D_s}{D_r} \tag{5}$$

D_s is the size of the drone in the simulation, D_r is the size of the drone in the real world. Then find the maximum distance that can be approached by drones to other drones (R_r) with the distance between drones in the simulation (R_s) is 50 pixels in (6) as:

$$R_r = R_s \times x \tag{6}$$

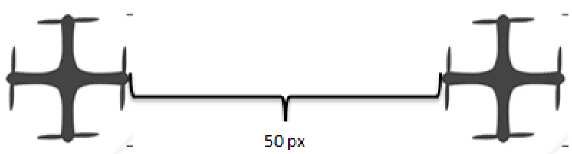

Fig. 22. Minimum distance between drones

Then find the size of the side area in the real world (A_r) using (7) as:

$$A_s = 30px \times 30px = 900\ (px)^2$$
$$A_r = A_s \times x^2 \tag{7}$$

where A_s is the side length of the area in the simulation. From (7), the area size of 15×15 in the simulation is 182.25 m^2. Then the area size verification table in the real world is given in Table 3. Finding the time value for scanning an area in the real world cannot be done because there are no parameters in the simulation in the real world that can be compared.

TABLE 3
VERIFY THE SIZE OF THE AREA

Area size	SIZE IN REAL WORLD (m^2)	Size of the area
15x15	182.25	13.5 m × 13.5 m
16x16	207.36	14.4 m × 14.4 m
17x17	234.09	15.3 m × 15.3 m
18x18	262.44	16.2 m × 16.2 m
19x19	292.41	17.1 m × 17.1 m
20x20	324	18 m × 18 m

D. Discussion

The study reveals that the FSS algorithm effectively coordinates multiple drones for area scanning. However, the optimal number of drones does not linearly correlate with the area size. This finding suggests that beyond a certain point, adding more drones does not significantly improve scan efficiency [14][15]. Factors such as drone interference and communication overhead may contribute to this phenomenon [16]. Future research could explore adaptive algorithms that dynamically adjust drone behavior based on real-time performance metrics [17].

IV. CONCLUSIONS

This research demonstrates the application of the FSS algorithm in a multi-drone simulation for area scanning tasks. The results show that while increasing the number of drones can improve efficiency, it is not a straightforward linear relationship. Further optimization and adaptive strategies are needed to fully leverage the potential of multi-drone systems.

Based on the scenarios used in this simulation it is concluded that the optimal scenario is 6 drones used in 17 x 17 rectangular grid which need 62.721 seconds to cover the entire rectangular area. Surely this simulation can be verified using real experiments using mapping on the C section of the experiment part.

REFERENCES

[1] R. Eberhart and J. Kennedy, “Particle Swarm Optimization,” *Proceedings of IEEE International Conference on Neural Networks*, vol. IV, pp. 1942-1948, 1995.

[2] M. Silva, J. Oliveira, and A. Santos, “Swarm Intelligence in Environmental Monitoring with UAVs,” *Environmental Monitoring and Assessment*, vol. 193, no. 4, p. 233, 2021.

[3] A. Jones, and B. Smith, “Advances in Swarm Intelligence for UAV Coordination,” *Journal of Unmanned Vehicle Systems*, vol. 9, no. 3, pp. 201-215, 2021.

[4] J. Cao, M. Chen, and Y. Li, “Recent Developments in Swarm Robotics,” *A Survey. IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 451-465, 2021.

[5] M.S Bhagat, “Swarm Intelligence: Concepts, Models and Applications,” *Swarm Intelligence*, 2018.

[6] E. Joelianto, A. Nainggolan, and Y. A. Hidayat, “Stingless bee algorithm for numerical optimization problems,” *International Journal of Innovative Computing, Information and Control*, vol. 16, no. 6, pp.2063-2081, 2020.

[7] C. de Lima Neto, F. Lins, A. Nascimento, A. Lima, and M. Bastos-Filho, “Fish School Search,” *Nature-Inspired Algorithms for Optimisation*, pp. 261-277, 2008.

[8] A. Moreira, A. Lima, and P. Pereira, "Fish School Search Algorithm Applied to UAV Path Planning.," *IEEE Access*, vol. 7, pp. 104577-104587, 2019.

[9] E.F. Moore, “The Shortest Path Through a Maze,” *Bell Telephone System Technical Publications*, 1956.

[10] S. Skiena, “Implementing Depth First Search” *The Algorithm Design Manual Springer*, 1990.

[11] C. Park, and D. Lee, “Optimizing Multi-Drone Systems Using Advanced Algorithms,” *International Journal of Robotics Research*, vol. 41, no. 5, pp. 689-702, 2022.

[12] S. Lee, and H. Kim, “Enhanced Coverage Path Planning for UAVs Using Genetic Algorithm,” *Sensors*, vol. 23, no. 2, p. 505, 2023.

[13] A. Trisanto, et. al., “Drone Waypoint Navigation Using Mission Planner with Dronekit-Python,” *Internetworking Indonesia Journal*, vol. 14, no. 2, pp. 15-19, 2022.

[14] E. Joelianto, D. Christian, A. and Samsi, “Swarm control of an unmanned quadrotor model with LQR weighting matrix optimization using genetic algorithm,” *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 11, no. 1, pp.1-10, 2020.

[15] T. Le, Q. Pham, and H. Nguyen, “Cooperative Control of Multiple UAVs for Area Coverage and Object Tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 3304-3318, 2020.

[16] V. Patel, and A. Sharma, “Communication and Coordination Strategies for UAV Networks,” *Ad Hoc Networks*, vol. 128, p. 102768, 2022.

[17] X. Lin, and Y. Wang, “Task Allocation and Optimization in Multi-Drone Systems: A Review,” *Robotics and Autonomous Systems*, vol. 150, p. 103959, 2022.