Evaluation of Communication Paradigms in Swarm Robotics Simulation for Search and Rescue using Reynolds Flocking Rule

Mario Jonatan
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
mariojonatan51@gmail.com

Ferry Hadary
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
ferry.hadary@ee.untan.ac.id

Jeremy Pratama Damanik
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
jeremydamanik010@gmail.com

Ibrahim Tsabit Hanif Rabbiradliya
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
ibrahimtsabithr028@gmail.com

Lisa Sabila
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
sabilationlisa@gmail.com

Rafidal Muhammad
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
rafidalmuhammad152@gmail.com

Ahmad Saifudin
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
ahmadsaifudin27112004@gmail.com

Hasan Supriadi
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
hasansupriadi911@gmail.com

***Abstract*—Natural disasters, which are often unpredictable, pose a major threat to human life as they inevitably cause casualties. One of the measures to reduce mortality in such events is the implementation of Search and Rescue (SAR) operations after a disaster occurs. However, human limitations often cause SAR operations to be slow, delayed, or even result in more casualties due to extreme weather conditions and challenging rescue terrains. This study presents a potential solution to these problems through the application of swarm robots based on Reynolds Flocking Rule. The research simulates the use of three communication paradigms—Broadcast Communication Swarm, Heterogeneous Swarm, and Base Station Swarm—and compares the advantages and disadvantages of each paradigm. From 100 simulation runs, the Broadcast Communication Swarm achieved the lowest average number of iterations at 425.82. It can be concluded that the Broadcast Communication Swarm is highly suitable for SAR operations, where time is a critical factor.**

***Keywords*—swarm robotics, SAR, Reynolds Flocking Rule, multi-robot systems, broadcast communication, heterogeneous swarm robots, base station communication**

I. INTRODUCTION

Disasters, particularly those that are unpredictable, demand an effective SAR team that can operate rapidly and accurately to minimize casualties. Victim evacuation must be completed within the first 72 hours after a disaster occurs, as survival rates can decrease drastically to only 5–10% afterward [1]. However, challenging terrains combined with the limited number of human personnel often hinder the effectiveness of conventional SAR operations. Lambin *et al.* reported that a shortage of personnel during emergency response can lead to delays in victim evacuation, thereby increasing the risk of both morbidity and mortality [2]. In addition, obstacles such as inadequate coordination and communication, as well as the lack of electronic registration systems, significantly reduce the effectiveness of SAR operations [3]. These conditions highlight the urgent need for innovation in SAR systems, including the integration of robotics to enhance the efficiency and effectiveness of victim evacuation processes. Therefore, we propose the use of swarm robots, referring to multiple small robots that work collectively and in a coordinated manner, to improve the efficiency of both manpower and time in disaster evacuation tasks.

In recent years, swarm robotics has emerged as a promising approach to support SAR operations. Inspired by collective behaviors in nature, such as those exhibited by ants, bees, and other colonial animals, swarm systems rely on simple local rules and limited communication to achieve robust global coordination. Several previous studies have demonstrated the usefulness of swarms for tasks such as area exploration, target search, and distributed decision-making [4], [5]. In the SAR context, swarm-based simulations have shown potential improvements in search efficiency within complex environments. However, most existing research has focused on leader-based assumptions, with limited systematic comparisons of different swarm communication mechanisms.

This study aims to address this research gap by comparing three swarm coordination paradigms: Broadcast Communication Swarm, Heterogeneous (leader-based) Swarm, and Base-Station Swarm. All three paradigms are implemented using the same fundamental rule, namely the Reynolds Flocking Rule. Each paradigm is simulated under identical conditions to measure efficiency in terms of the number of iterations required until the target is successfully evacuated. By treating communication mechanisms as the primary variable, this research provides an objective evaluation of the strengths and limitations of each approach.

II. RELATED WORK

A. Swarm Robots

Swarm robots are groups of autonomous robots that operate to achieve a common goal without relying on centralized control (decentralization), but instead depend on local behaviors or environmental interactions [6]. Due to this decentralized nature, individual robots within a swarm must coordinate themselves independently by relying on local sensing and communication to interact with other robots, thereby enabling the emergence of complex collective behaviors such as task allocation [7].

Swarm robotics is a subfield of multi-robot systems, which itself is a subfield of mobile robotics research. According to Shahzad *et al.*, a multi-robot system that consists of only two individuals cannot be classified as swarm robotics, even if both robots are intelligent [6]. A system can be categorized as swarm robotics if it consists of at least three autonomous robots that are capable of interaction and coordination to achieve a shared objective. In addition, swarm robotics is characterized by three fundamental properties that distinguish it from other robotic systems: robustness, meaning the system can continue functioning toward its goal even if one robot fails; flexibility, meaning the system can adapt to a variety of goals and tasks; and scalability, meaning the system remains effective and stable even when the number of robots increases or decreases [8].

B. Reynolds Flocking Rule

The Reynolds Flocking Rule was first introduced by Craig W. Reynolds in 1987 through a model known as *boids* (bird-oids). Several studies on flocking behavior design in multi-agent systems have built upon Reynolds fundamental work. To realize flocking, defined as a group moving in a coherent and orderly manner, Reynolds proposed three heuristic rules that govern the behavior of each boid:

(a) Alignment — matching the velocity with the average velocity of neighbors (*velocity matching*).

$$F_{ali} = \frac{1}{|N_i|}\sum_{j\in N_i}(v_j - v_i) \tag{1}$$

where v_i is the velocity of robot i.

(b) Cohesion — keeping close to the group (*flock centering*).

$$F_{coh} = \frac{1}{|N_i|}\sum_{j\in N_i}(p_j - p_i) \tag{2}$$

(c) Separation — avoiding collisions with neighbors (*collision avoidance*).

$$F_{sep} = -\sum_{j\in N_i}\frac{p_j - p_i}{\|p_j - p_i\|^2} \tag{3}$$

where p_i is the position of robot i, and N_i is the set of neighbors within a certain radius.

The combined effect on agent i is expressed as a linear combination of the three forces:

$$F_i = w_{ali}F_{ali} + w_{coh}F_{coh} + w_{sep}F_{sep} \tag{4}$$

where w_{ali}, w_{coh}, and w_{sep} are the weighting coefficients for each rule. These three rules form the foundation for the majority of flocking control mechanisms in the literature [9].

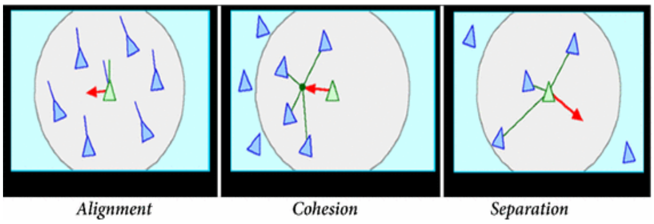

Fig 1. Reynlods Flocking Rule [9]

C. Python

Python is an interpreted programming language that offers a wide range of functionalities and is designed with a focus on code clarity and readability [10]. Python was developed by Guido van Rossum and was first introduced to the public in 1991. Today, it is one of the most widely used programming languages worldwide, due to its flexibility and applicability across multiple domains [11]. Python is also among the most common programming languages for developing various applications and simulations [12]. In addition, Python benefits from a large community, thousands of libraries, and active discussion forums.

Python operates using an interpretive system, meaning that code is executed line by line. This allows easier testing, experimentation, and debugging, making it both interactive and user-friendly. The syntax of Python is simple and close to natural language, enabling ease of understanding without the need for excessive or complex symbols. Furthermore, Python is cross-platform and can run on various operating systems, including Windows, macOS, and Linux [11].

D. Homogeneous and Heterogeneous Swarm Robots

A homogeneous swarm robot system refers to a configuration in which all robots possess identical capabilities. If uncertainties in sensing, movement, and decision-making are disregarded, these robots will exhibit the same responses when exposed to identical environmental conditions [13]. In such systems, variations in behavior and task allocation are influenced primarily by environmental factors. Furthermore, changes in the relative positions of neighboring robots affect sensor input data, which in turn leads to differences in task distribution and behavioral outcomes [14]. Since each robot has the same capabilities, the failure of one autonomous robot can be compensated for by others. However, this uniformity also presents a drawback, as homogeneous swarm robots are less suitable for missions requiring role specialization and are therefore not ideal for tasks that demand specific expertise.

In contrast, heterogeneous swarm robots consist of autonomous units with diverse capabilities, enabling them to handle a wider variety of tasks. Asad *et al.* investigated this concept using an evolutionary robotics approach, where swarm heterogeneity was achieved through software differentiation. By employing two artificial neural network controllers, their study demonstrated that heterogeneous swarms exhibit greater adaptability and flexibility compared to their homogeneous counterparts [14].

E. Broadcast Communication in Multi-Robots

Broadcast is a communication mechanism in which a single transmitter sends information to all receivers within the network. In the context of multi-robot systems, broadcast is used to coordinate collective tasks since all robots require the same information. For example, in industrial applications such as automated item retrieval and warehouse management, broadcast architectures are employed to ensure the successful completion of tasks [15]. However, excessive use of broadcast can overload the robots' operation. Overuse of broadcast may degrade the robots' performance [16]. This condition also leads to a waste of resources (bandwidth and radio energy), meaning that employing too many broadcast channels can cause resource inefficiency and decreased system performance.

III. Research Methods

A. Simulation Environments

The swarm robot simulation was developed using the Python programming language, with NumPy as the primary library for numerical computation and array-based data processing. The chosen Integrated Development Environment (IDE) was Visual Studio Code (VSCode), as it provides a lightweight and flexible development environment while also supporting interactive visualization of simulation results. The simulation domain was defined as a two-dimensional square field measuring 100 × 100 units. No obstacles or barriers were included in this domain, ensuring that the robots interact only with each other and with the target. Such a simplified setup is commonly adopted in the initial stages of swarm robotics research, as it facilitates the observation of emergent behaviors derived from local robot rules without the influence of external factors.

The target in the simulation was represented as a single red star-shaped object. Only one target was placed in each run, and its position was randomized at the start of every program execution. Consequently, the swarm robots were required to adapt to varying initial conditions, thereby demonstrating the system's capability for collective target search.

At the initial stage, ten robots were positioned in the lower-left corner of the domain, near the coordinate origin (0,0). The robots were arranged in a 2 × 5 rectangular formation inside a designated *deploy box*, ensuring that their initial distances were relatively close. This formation was selected to represent a swarm being "launched" from a single gathering point. Within the deploy box, a black dot called the *drop point* was defined, serving as the final destination for the target to simulate the rescue operation performed by the swarm robots.

Each robot movement was counted as one iteration, which also represented the discrete time step of the simulation. Accordingly, the total number of iterations required for the swarm to locate and surround the target was used as the evaluation variable for measuring the performance of the implemented communication and coordination algorithms.

B. Robot Model

The robots in the simulation were modeled as circular entities with a black line indicating the front orientation. Each robot was assigned a color that dynamically changed according to its current state. Robots were assumed to have physical dimensions, and therefore collisions were possible both with other robots and with the environment boundaries.

(a) Reynolds Flocking Rule — the movement of the swarm robots followed the Reynolds Flocking Rule, consisting of three main components: separation, alignment, and cohesion. The weighting coefficients used in the simulation were:

- Separation with $w_{sep} = 0.9$
- Alignment with $w_{ali} = 0.6$
- Cohesion with $w_{coh} = 0.05$

In addition to these three basic rules, a wandering behavior was introduced with a coefficient of 0.8. This behavior enhanced the exploratory capability of the swarm, allowing robots to cover a larger search area and thereby increasing the likelihood of detecting the target.

(b) Maximum Speed — robots were assigned different speed limits depending on their operational mode:

- Search Mode: random wandering at 5.0 units per iteration
- Homing Mode: moving toward the target at 10.0 units per iteration
- Surround Mode: encircling the target at 5.0 units per iteration
- Carry Mode: transporting the target to the drop point at up to 10.0 units per iteration

(c) Other Parameters

Robots were able to detect neighboring robots within a radius of 12.0 units and detect the target within a radius of 6.0 units. The surrounding behavior around the target was activated within a radius of 8.0 units. Detection in the simulation was assumed to be perfect, with 100% accuracy and zero latency. This assumption allowed the study to focus on evaluating the effectiveness of swarm coordination rather than sensor limitations.

C. Communication and Coordination Model

(a) General coordination states of robots — how each of communication's pardigm swarm robots coordinating with each other on every state.

- Search state: In this state, the ten robots were deployed from the deploy box to search for the target. The search process was performed using random wandering logic with a speed of 5 units per iteration until the robots entered the detection radius of the target.

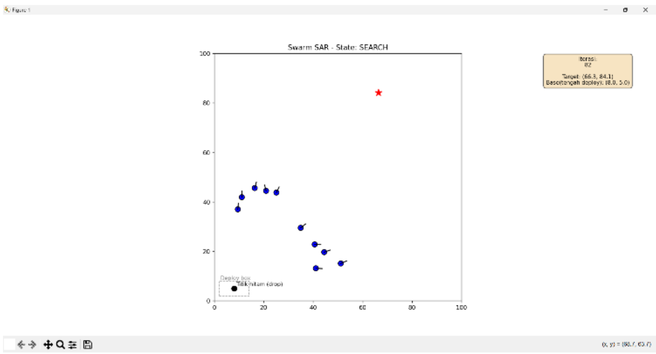

Fig 2. Search state

- Homing state: in this state, all ten robots were assumed to already know the position of the target. The robots then moved toward the target until they reached the target radius at a speed of 10 units per iteration.

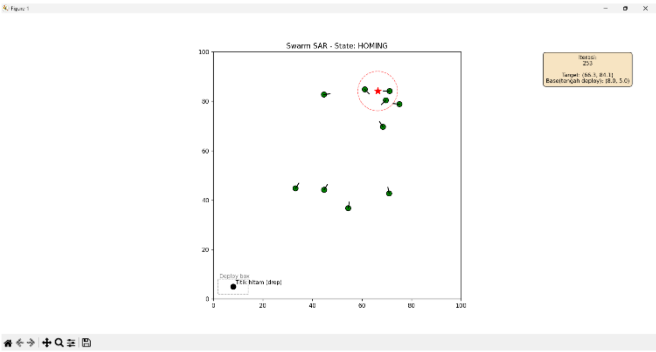

Fig 3. Homing state

- Surround state: in this state, the robots that had successfully performed homing surrounded the target by forming a circular formation around it. The surrounding behavior was executed at a speed of 5 units per iteration. The purpose of this state was to demonstrate the cooperative behavior of the robots in preparing for the rescue of the target.

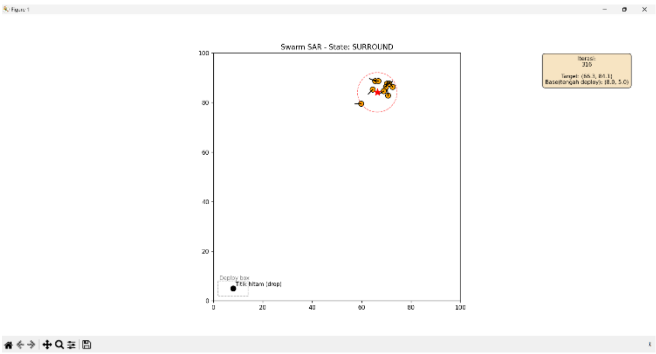

Fig 4. Surround state

- Carry state : in this state, all ten robots together with the target moved toward the drop point to deliver the target, with a maximum speed of 10 units per iteration. The objective of this state was to show that the robots were able to collaborate in formation to execute the rescue task.

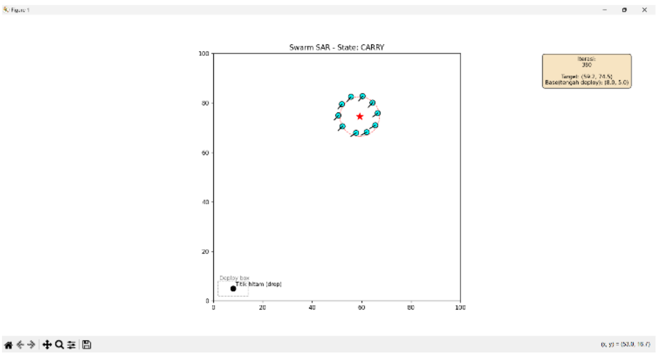

Fig 5. Carry state

(b) Robot Communication Models — how each of communication's pardigm swarm robots communicating with each other on every state. In this study, three communication paradigms for swarm robots were evaluated: Broadcast Communication Swarm, Heterogeneous (Leader-Based) Swarm, and Base-Station Swarm.

- Broadcast Communication Swarm: this paradigm assumes that all robots are capable of broadcasting target location information to every other robot in the simulation arena. When a single robot detects the target, all robots instantly receive the target's position

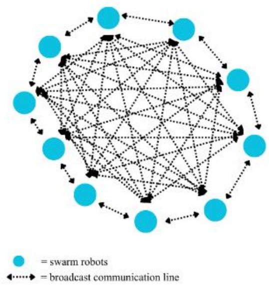

Fig 6. Broadcast communication swarm's communication lines

As shown in Fig. 6, swarm communication in this paradigm required at least 45 two-way communication links. This resulted in high communication overhead compared to the other two paradigms, which is one of the disadvantages of the broadcast approach.

- Heterogeneous Swarm (Leader-Based): this paradigm assumes that one robot becomes the leader among the swarm. In the simulation, the leader was randomly selected from the ten deployed robots. When the leader found the target, the information was broadcast to the entire swarm, allowing all robots to instantly know the target location. However, when an ordinary member robot detected the target, it continued random wandering until it encountered the leader. Once communication with the leader was established, the information was transferred to the leader, who then broadcast the target position to the swarm.

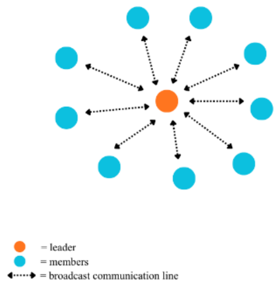

Fig 7. Heterogeneous swarm's communication lines

As shown in Fig. 7, the communication in this paradigm required at least 9 two-way communication links—significantly fewer than in the broadcast case. This reduced communication overhead is one of the key advantages of the leader-based model, as it conserves communication resources.

- Base Station Swarm: this paradigm assumes the presence of a fixed base station acting as a hub for target information. Robots that successfully detected the target reported the position to the base station, which then broadcast the information to the entire swarm.

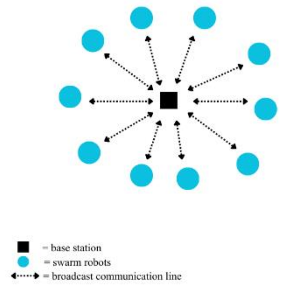

Fig 8. Base station swarm's communication lines

As shown in Fig. 8, communication in this paradigm required at least 10 two-way communication links, slightly more than the leader-based swarm. However, since the base station remained in a fixed position, it was easier for robots to locate compared to searching for a mobile leader. The drawback of this paradigm is the additional resource requirement for constructing and maintaining the base station component.

IV. Results

Based on 100 trials for each communication paradigm of swarm robots, data were obtained regarding the average number of iterations required by each paradigm, along with a graph visualizing the iteration values across experiments.

Table I. Iterations Data of Each Communication Paradigms

Swarms	Broadcast Communication	Heterogeneous	Base Station
Average Iterations	426.82	724.99	559.82
Most Iterations	1853	4748	1796
Least Iterations	113	132	127
Standard Deviation	252.82	542.13	261.43

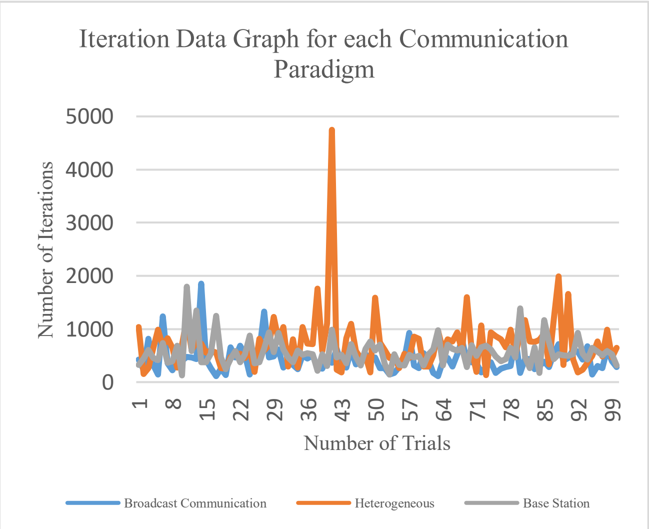

Fig 9. Graph of iterations data for each communication paradigms

From the iteration data (see Table I), the average completion time for SAR tasks was 426.82 iterations for the Broadcast Communication Swarm, 724.99 iterations for the Heterogeneous (Leader-Based) Swarm, and 559.82 iterations for the Base-Station Swarm. Based on these results , it can be concluded that the Broadcast Communication Swarm is the most effective communication paradigm for swarm robot SAR. This is because broadcast communication, which can be performed by every robot in the swarm, enables the information about the target position to be disseminated more quickly compared to the other paradigms. For the Heterogeneous (Leader-Based) Swarm, performance was consistently slower due to the necessity of performing two searches: first to locate the target, and second to locate the leader. This double the search process resulted in the longest average iteration time among the three paradigms.

As shown in Fig. 9, there was a significant difference between the iteration values of the Heterogeneous Swarm and the Base-Station Swarm. While both paradigms required "reporting" before broadcasting the target position, the Heterogeneous Swarm exhibited more fluctuation in iteration counts. This occurred because the leader robot was mobile, introducing an element of chance: if the leader happened to be near the robot that found the target, information transfer was fast; otherwise, the reporting robot could struggle to locate the leader. In contrast, the Base-Station paradigm was more stable, as the station was in a fixed position, making it easier for robots to report target locations without additional searching. In addition, based from iteration data (see Table I), the standard deviation for Heterogeneous Swarm is the highest compared to others paradigm. Then, we can conclude that Heterogeneous Swarm is the most unstable communication paradigms among these three communication paradigms based on the iterations data result.

Base Station Swarm can be considered a fairly effective communication paradigm. In terms of speed and stability, it can be regarded as average, since it is neither as fast nor as stable as the Broadcast Communication Swarm. However, this communication paradigm has an advantage, namely that it does not rely on as many broadcast channels as the Broadcast Communication Swarm, which means that it does not consume excessive resources and the robots' performance is more likely to be maintained. Just a little bit high of resource consumption, for the base station. Therefore, it can be stated that the Base Station Swarm is the most realistic paradigm among the three and has a high potential to be implemented in SAR operations.

Based on the methodology, results, and discussion, the advantages and disadvantages of each communication paradigm are summarized as follows:

Table II. Advantages and Disadvantages of Each Communication Paradigms

Swarms	Advantages	Disadvantages
Broadcast Communication	Fast and stable completion time	High resource consumption, robot performance may decrease
Heterogeneous	Slow and unstable completion time	Low resource consumption,

		stable robot performance
Base Station	Moderately fast and stable completion time, stable robot performance	A little bit high in resource consumption (for the base station)

V. Conclusions and Suggestions

A. Conclusions

Based on the simulations conducted, several important findings can be summarized as follows:

(a) All three swarm communication and coordination paradigms have potential applications in SAR scenarios, each with its own strengths and limitations.

(b) The Broadcast Communication Swarm demonstrated the most effective performance in terms of speed and stability in completing the SAR simulation. However, it requires high resource consumption and may lead to a significant decrease in robot performance if implemented.

(c) The Heterogeneous Swarm demonstrated the lowest performance in terms of speed and stability in completing the SAR simulation. Nevertheless, it requires low resource consumption and maintains stability in robot performance when implemented.

(d) The Base Station Swarm demonstrated average performance in terms of speed and stability in completing the SAR simulation. In its implementation, this communication paradigm requires slightly higher resources (for the base station) but still maintains stability in robot performance.

(e) These results remain idealized, as the simulations were conducted in an obstacle-free environment under the assumption of perfect detection accuracy and zero latency. Therefore, none of the paradigms—including the Broadcast Communication Swarm—can yet be considered a final solution for SAR applications, and further testing under more complex and realistic conditions is necessary.

B. Suggestions

Given the limitations of this study, several suggestions for future research are proposed:

(a) Extending experiments to more realistic conditions, such as environments with obstacles, limited communication, delayed information transfer, and imperfect sensor accuracy.

(b) Expanding the scope of application beyond SAR to include swarm robotics for efficient transportation, logistics distribution, or non-SAR operations.

(c) Investigating multi-target scenarios, where the swarm must detect, evaluate, and allocate search and surround tasks for more than one target simultaneously.

(d) Conducting experiments with varying numbers of robots and adjusting the coefficients in Reynolds Flocking Rule to assess their impact on swarm coordination performance.

(e) Implementing the paradigms using physical robots, enabling direct comparison between simulation results and real-world performance in complex environments.

Acknowledgment

The authors would like to express their gratitude to the Digital Control and Computation Laboratory, Electrical Engineering Department, Faculty of Engineering, Universitas Tanjungpura, for providing laboratory resources and a supportive research environment.

References

[1] M. Hakami, A. Al-Ahmadi, and H. K. Al-Masri, “Search and rescue time-critical operations in disaster response: A survival rate perspective,” Journal of Emergency Management, vol. 12, no. 3, pp. 201–210, n.d.

[2] E. Lambin, J. T. Garcia, and P. H. Rios, “Human resource limitations in emergency response and their impact on evacuation outcomes,” *International Journal of Disaster Risk Reduction*, vol. 78, pp. 103–118, 2024.

[3] T. Sofuoglu, K. Demir, and L. Karaca, “Barriers to effective search and rescue operations: Coordination, communication, and digital registration systems,” *Safety Science*, vol. 165, pp. 105–120, 2025.

[4] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: A review from the swarm engineering perspective,” *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013

[5] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, and G. Baldassarre, *et al.*, “Evolving self-organizing behaviors for a swarm-bot,” *Autonomous Robots*, vol. 17, no. 2–3, pp. 223–245, 2005.

[6] M. Shahzad, A. Hussain, and T. H. Nguyen, “Swarm robotics: Decentralized coordination and applications,” *Robotics and Autonomous Systems*, vol. 167, pp. 104–115, 2023.

[7] L. Kuckling, “Task allocation in swarm robotic systems: A review of mechanisms and strategies,” *Journal of Intelligent and Robotic Systems*, vol. 108, no. 2, pp. 1–20, 2023.

[8] . Dias, R. Ventura, P. Lima, and L. Marques, “Fundamental properties of swarm robotics: Robustness, flexibility, and scalability,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 1–25, 2021.

[9] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’87)*, vol. 21, no. 4, pp. 25–34, 1987.

[10] D. Triono, A. Wijaya, and R. Setiawan, “Python sebagai bahasa pemrograman interpretatif,” *Jurnal Teknologi Informasi Indonesia*, vol. 15, no. 2, pp. 45–52, 2023. [in Indonesian]

[11] A. Buwono, M. Rahardjo, and L. Putra, “Perkembangan Python: Fleksibilitas, lintas platform, dan adopsi global,” *Jurnal Ilmu Komputer dan Aplikasi*, vol. 30, no. 1, pp. 10–19, 2025. [in Indonesian]

[12] N. S. Arvi, F. Salsabila, and R. Pratama, “Python dalam pengembangan aplikasi dan simulasi: Sebuah tinjauan,” *Prosiding Konferensi Nasional Informatika (KNIF)*, pp. 112–118, 2024. [in Indonesian]

[13] G. A. Kaminka and D. Douchan, “Homogeneous swarm robotics: Behavioral equivalence and limitations,” *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 39, no. 2, pp. 145–160, 2025.

[14] M. Asad, F. Rehman, and H. Khan, “Evolving heterogeneous swarm robots using neural network controllers,” *Swarm and Evolutionary Computation*, vol. 78, pp. 102–118, 2023.

[15] J. Gielis, M. Garzón, F. Andriella, and A. Billard, “A Critical Review of Communications in Multi-Robot Systems,” *Current Robotics Reports*, vol. 3, pp. 213–225, 2022.

[16] B. Al Nahas, “Network-wide Consensus Utilizing the Capture Effect in Low-power Wireless Networks,” *Proceedings of the 17th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2019, pp. 366–367.