HIGH SCHOOL EDITION
Journal of Student Research
Volume 14 Issue 1 (2025)

Communication Structures in Underwater Swarm Robotics Systems

Rini Jain[1] and Kiyn Chin[#]

[1]Dublin High School, Dublin, CA, USA
[#]Advisor

ABSTRACT

A swarm of robots has unique characteristics that allow it to handle complex tasks efficiently. In this work, the deployment of a swarm of robots in a body of water is studied using computer simulations to track and identify sources of plume pollution. The water body is modeled as a section of a lake. The pollution source is a marine plume specified at a location unknown to the robots, which the robotic swarm is tasked to track efficiently. Heterogenous swarms of robots are employed which allows exploration of various behavioral archetypes of the robots as well as the distribution of archetypes among the swarm to simulate a diverse body of robots, and this is compared to homogeneous swarms of robots too. A computer simulation-based study is carried out to investigate the speed and accuracy of marine pollution plume detection by swarms with varying extents of heterogeneity. It is demonstrated that the diversity of robots can be beneficial to a swarm of robots if the archetypes are individually productive but could be harmful otherwise. Different extents of diversity are useful depending on the archetypes present in the swarm.

Introduction

Robotic swarms are defined by interactions between individual robots. In many circumstances, a swarm of robots is preferred to a single robot. Swarms are capable of handling multiple tasks, and they are scalable, cheaper than a single complex robot, and more energy efficient [1]. Many applications require coordination in problem-solving and benefit from swarm robotics. These applications include area exploration, agriculture, search-and-rescue, and surveillance, extending to space, terrestrial, aquatic, and aerial domains. Particularly, swarms of robots can be used in slave-master structure to efficiently locate a victim in a search-and-rescue mission, or they can be sent to space to coordinate to college materials and resources [2,3,4,5,6]. It is their characteristics of group behavior that makes them applicable to wide varieties of tasks in diverse environments. The design of swarm robotics software and hardware is an active research area. Many foundational algorithms are inspired by biological systems, such as Ant Colony Optimization (ACO), Genetic Algorithm, Particle Swarm Optimization (PSO), Differential Evolution, and social insects like ants and bees. The ACO method is modeled after the way in which ants in an ant colony produce several paths of pheromones leading to a source of food, and the ants will ultimately take the shortest path to the food source; this demonstrates qualities like local communication and emergent behavior of swarms. PSO follows the flocking of birds and their coexisting traits of cohesiveness, separation, and alignment [6,7,8,9]. Other proposed algorithms include uniform dispersion, robot mapping, chain-based path formation, and more [8,9].

Swarms consist of several robots that interact with neighbors and surrounding environments. They exhibit emergent intelligence: pattern organization, spatial awareness, navigation, environment monitoring, and more. Swarm members can make individual decisions as well as establish consensus with others through local interactions, rather than a centralized top-down communication structure [6]. Interactions often use wireless communications, however, acoustic communication is preferred for underwater applications due to its ability to travel large distances through water [7]. Optical or tethered communication methods exist but are not well-suited to aquatic environments [10].

The taxonomy of individual robots within a swarm can be diverse [11]. The performance of the swarm can be optimized via the behavioral diversity of component robots. Robots can further be diverse in finer details of their communication: bandwidth, ranges, and topologies [12]. Diversity is also seen in the composition of the robots, physically and function-wise. Widely diverse distribution of behaviors in a swarm of robots is shown to be more beneficial than homogeneity [13]. This study imitates the colonies of honeybees, in which there are roles of bees, including random wandering bees, seeking bees, and stationary ones in aggregation behavior. This further demonstrates the inspiration that robot swarms take from biological systems, much like the swarm we replicate in this study does.

Swarms are designed specific to the task they are designed to complete, and this heavily relies on the environment the swarm is in. Specifically to this study, the nature of marine plumes is important to understand. Marine plumes demonstrate intermittent release of particles, sinuous shapes, and time-varying characteristics, and realistic characteristics need to be employed in order to yield accurate results [14]. As a result of these plume characteristics, the robot archetypes (discussed later) can be tested realistically to the specific environment of an enclosed lake.

Methods

As mentioned before, the focus of the present work is to develop an understanding of the impact of behavioral diversity of locally interacting underwater swarms on plume-finding performance. For this purpose, a simulation of the robotic swarm and the plume was developed. The simulation was parameterized to allow various behaviors of the robots and the plume to be implemented. A series of experiments (computer simulations) were run by varying the parameters in order to establish the role of diversity on performance

Plume Model

A lake section is modeled as a two-dimensional rectangular plane with horizontal or vertical water currents. A plume source is modeled as a location that emits particles at a randomized injection velocity. Particles are defined by location, injection velocity, magnitude, deceleration rate, and maximum speed. Particles are intermittently released from the plume source. At each time step of the simulation, the location of particles is updated by adding the resultant vector from the water currents and the injection velocity. A randomly oriented velocity vector is added to simulate diffusion. The current vector is randomly initialized to be horizontal or vertical and then updated at each time step using Equation (1), as follows,

$$current = V_{plume} \times sin\left(2\pi \times \frac{time}{T}\right). \quad (1)$$

The plume current has a magnitude of V_{plume}. A sinusoidal function of period, T, is assumed in order to model the dynamics of marine plumes in time and space, releasing particles intermittently and taking sinuous shapes [14].

To simulate the physical slowdown of the particle velocity in the lake, a deceleration factor is applied as demonstrated in Equation (2),

$$deceleration = velocity \times 0.999. \quad (2)$$

Once the particles hit the edges of the lake section, they leave the lake section and are deleted. The resulting plume is shown in Figure 1. The magnitude and period variables of the sine wave equation can be altered by changing the parameters of the developed computer software to change the plume behavior. The bottom-top x and left-right y axes are the lake boundaries; each side is 15,000 units (dimensionless).

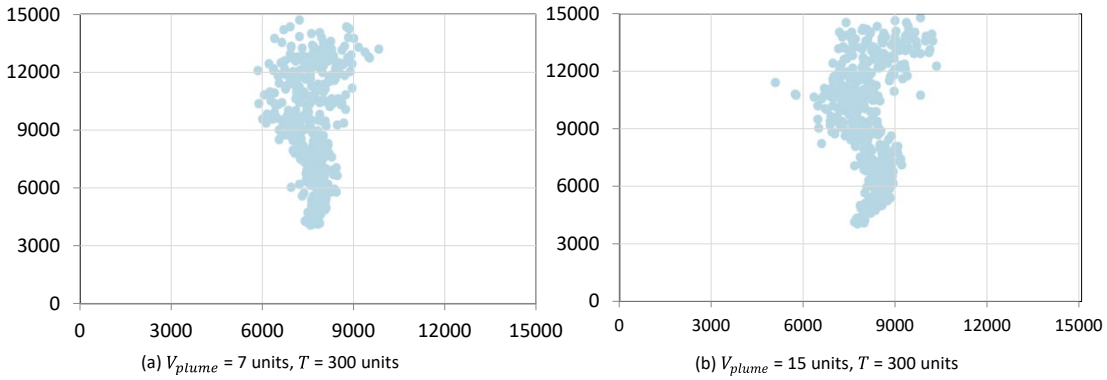


Figure 1. Plume shapes of different parameters showing different appearances.

Modeling of the Swarm of Robots

Each robot in the swarm has the following characteristics:

1) position,
2) velocity,
3) detected concentration of particles,
4) signaling status (to indicate if the robot is detecting concentration),
5) hearing dictionary (to store data of local signaling robots),
6) history of concentrations,
7) turning vector (the velocity to add if the robot is turning),
8) archetype, A (discussed below), and
9) state, S (discussed below).

Simulation Setup

At the beginning of each simulation, the plume and a set of robots are initialized with the above characteristics. The simulation then proceeds at discrete times as a series of time steps, referred to as time steps. At each time step, data is exchanged between the plume and robots and their characteristics are updated, as discussed below.

First Experiment

In the first set of experiments, each robot is designed with four possible states:

1) $s1$ = still,
2) $s2$ = finding-the-plume,
3) $s3$ = inside-the-plume, or
4) $s4$ = reacquiring-the-plume.

If the robot is initialized to be *still*, its only task is to detect the concentration of pollutant particles within a radius and send this data to local robots; they do not move. If the robot begins detecting concentration, it can change statuses. Whether it will change its status or not depends on its tendencies and the probability of changing (discussed more below). While in *finding-the-plume*, the robot will head in a randomized direction and detect particles. If there are ever plume particles detected, the robot can change statuses. If the robot's status is *inside-the-plume*, it will explore the plume. If the concentration of the robot's current detected concentration is higher than the concentration at its last position, then it will keep going in the same direction. Otherwise, the robot will head in a random direction. If the

robot detects zero concentration, it can change its status. If the robot is *reacquiring-the-plume*, it will head in the direction of the position with the highest detected concentration in its concentration history.

Robots of different archetypes, A, are defined by specifying the relative scores of these four states as a set of four numbers $A = [a_1, a_2, a_3, a_4]$ for *still*, *finding-the-plume*, *inside-the-plume*, or *reacquiring-the-plume* states, respectively. The relative scores are converted to relative probabilities, P_i, via the *softmax* function, as

$$P_i = \frac{e^{a_i}}{\sum_{j=1}^{4} e^{a_j}}. \quad (3)$$

Each number in these distributions is a score that influences the relative probability the robot has of being still, finding-the-plume, being inside-the-plume, and reacquiring-the-plume states, respectively. Three archetypes considered in this work are (1) lazybot, (2) greedybot, and (3) confusedbot. The score distributions are presented in Table 1.

Table 1. Robot archetype-score distributions.

Archetype	a_1	a_2	a_3	a_4
lazybot	6	1	3	2
greedybot	1	10	10	0
confusedbot	7	7	7	7

The lazybot has a higher probability of being still, the greedybot tends to be always looking for the plume and remaining inside it, and the confusedbot has equal probabilities.

Besides the archetype-score distribution which remains invariant during the simulation, each robot is also assigned a running-score distribution $R = [r_1, r_2, r_3, r_4]$ which is updated at each time step during the simulation, as described below. The default running-score (R) distributions of each state are given in Table 2.

Table 2. Robot running-score distributions.

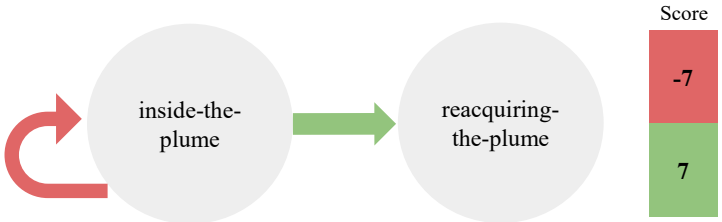
State	r_1	r_2	r_3	r_4
s_1, still	10	7	5	0
s_2, finding-the-plume	0	10	1	0
s_3, inside-the-plume	0	0	10	1
s_4, reacquiring-the-plume	0	0	1	10

When a robot begins or stops detecting particles, R is updated according to a state-transition distribution $T = [t_1, t_2, t_3, t_4]$ as in Table 3.

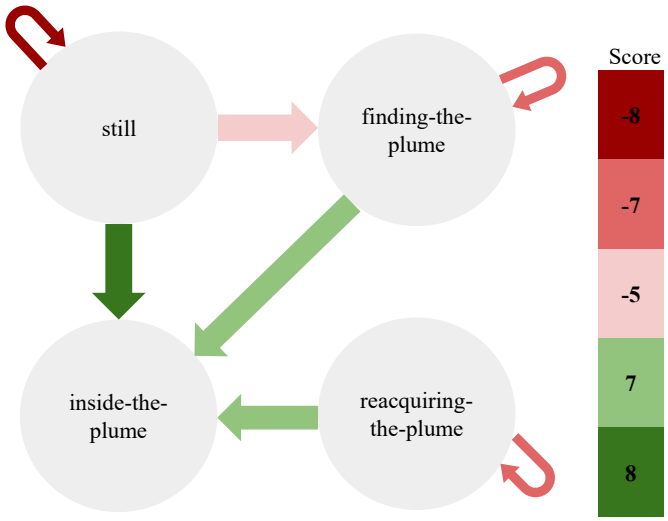
Table 3. Robot state-transition distributions.

State Transition	t_1	t_2	t_3	t_4
$s_1 \rightarrow s_3$ (still-to-inside)	-8	-5	8	0
$s_2 \rightarrow s_3$ (finding-to-inside)	0	-7	7	0
$s_3 \rightarrow s_4$ (inside-to-acquiring)	0	0	-7	7
$s_4 \rightarrow s_3$ (reacquiring-to-inside)	0	0	7	-7

Whether a robot changes statuses depends on the current status of the robot. For example, if a robot is finding-the-plume, it will not be able to change statuses if it is not detecting concentration, but it can change statuses if it starts detecting concentration. This is more elaborately described in Figure 2.



(a) Condition: Stopped detecting concentration



(b) Condition: Started detecting concentration

Figure 2. First set of experiments: transition between statuses.

The T distributions are applied to update R. For example, if the robot is in the s_2 state (finding the plume) then its score distribution is $[0, 10, 1, 0]$, as given in Table 2. However, if it begins detecting particles then its state should be s_3 (inside the plume) so the $s_2 \rightarrow s_3$ state transition scores are added to its current distribution at the next time step of the simulation, giving the update scores as $[0, 10, 1, 0] + [0, -7, 7, 0] = [0, 3, 8, 0]$. Finally, the robot's archetype score A is added to the updated R, as $P = A + R$ to get the probability distribution using Equation (3) to determine the new status.

At each time step, the robot can update its position in one of the two ways. First, the robots can hear the local robots' data and use their individual intelligence to come up with a new desired position, x_i and y_i, which will be the concentration-weighted average of all the positions, as described in Equation (4):

$$x_i = \sum_{j=1}^{n} \frac{c_j x_j}{c_{total,i}}, \quad y_i = \sum_{j=1}^{n} \frac{c_j y_j}{c_{total,i}}, \quad and \quad c_{total,i} = \sum_{j=1}^{n} c_j. \tag{4}$$

Here, c_{total} represents the total of the concentrations of all the n heard robots, c_j is the concentration of each heard robot, x_j and y_j are the coordinate positions of these heard robots. The new desired position is the position the robot should go to next, and the robot accordingly updates its velocity to head to that position. Alternatively, for when all the heard concentration data is zero, the robot can update its position through only the individual intelligence, which uses the concentration history to determine where the concentration is highest, to the best of its knowledge. If the entire history is filled with concentrations of zero, the robot will decide to keep going in the same direction.

When a robot reaches the edge or boundary of the lake, it is made to turn. The turning is simulated using the following equation,

$$v_i^{n+1} = v^n + \frac{(v_d - v_i)}{nturn}. \tag{5}$$

Here, v_d is the desired or the ending velocity after it is done turning, v_i is the initial velocity of the robot before it starts turning, $nturn$ is the number of time-steps the robot wants to turn in, v^n is the current velocity and v^{n+1} the updated velocity at the next time step.

When the experiment first runs, the robots are all initialized to the finding plume phase and they are randomly given one of the three archetypes (which is further described below). On each time step, the robot detects concentrations, sends this data, and then behaves according to the status they are in. To terminate the experiment, when the robots detect concentration on each time-step, the number of robots detecting any sort of plume particles is assumed to be inside the plume. Once 40% of the robots are inside the plume, they all report a concentration, and the position of the robot with the highest detected concentration is reported to be the detected plume source. This is the position used in the distance calculation (distance between this detected plume source and the actual plume source). Images of the simulation are shown in Figure 3.

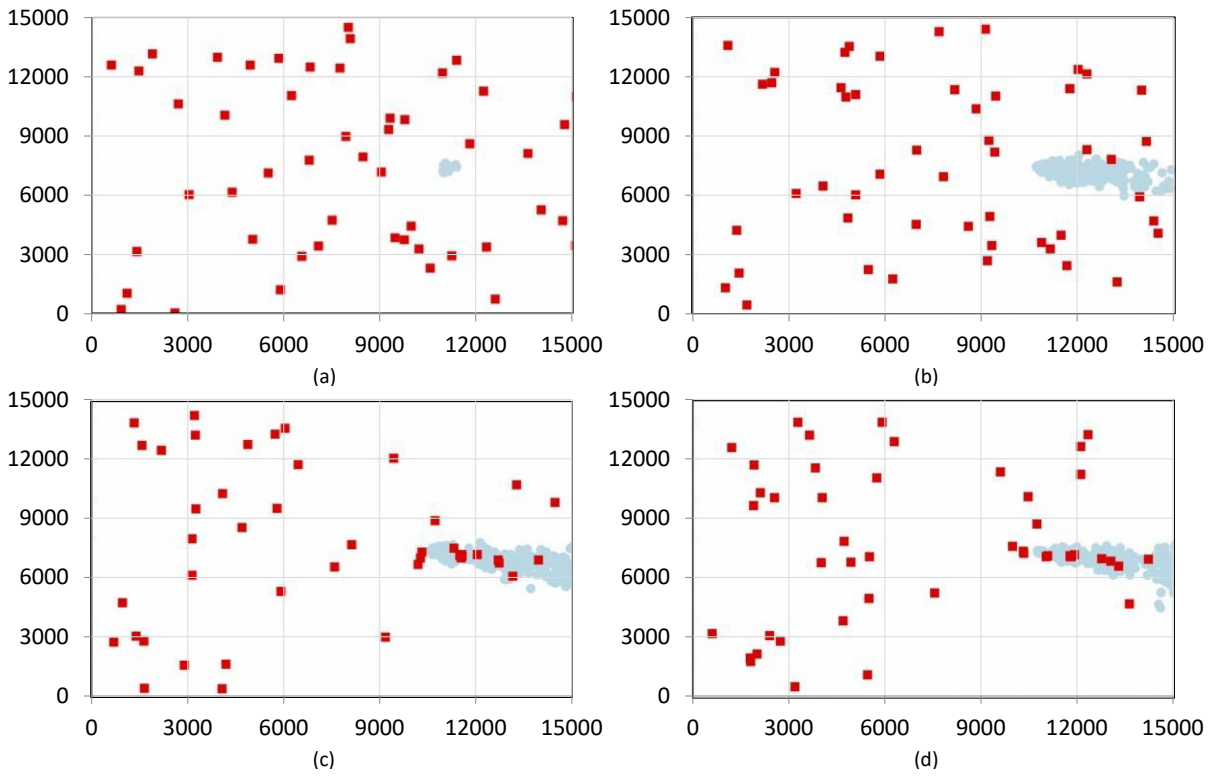


Figure 3. The images show how the robots make progress on their task. With each image from (a) to (d), they get closer to the plume source. The local interactions are shown through how the robots cluster together around the plume source.

In the first set of experiments, there were two data sets collected, one for each of the two plume sources. For each plume source, the simulation was run seven times (which represents seven trials) for each probability distribution (actual probability and not the coefficients for the softmax function) of archetypes given as $[q_{lazy}, q_{confused}, q_{greedy}]$ for lazybot, confusedbot, and greedybot, respectively. These distributions are given in Table 4 and represent different measures of diversity. The results are described in the Results section.

Table 4. Archetype distribution for the first experiment trials.

Trial #	q_{lazy}	$q_{confused}$	q_{greedy}
1 (max diversity)	1/3	1/3	1/3
2	0.3	0.2	0.5
3	0.1	0.6	0.3
4	0.5	0.1	0.4
5 (no diversity)	1.0	0.0	0.0
6 (no diversity)	0.0	1.0	0.0
7 (no diversity)	0.0	0.0	1.0

Second Experiment

The second set of experiments included improvements from the first one, as follows:

1) the *still* status and the *lazybot* archetype are eliminated, as they are found to be unhelpful (described more in the Results section). A new archetype, *seekerbot* is added.
2) the simulation termination condition was modified. Each robot stores a Boolean representing whether it found the plume or not. Each robot constantly stores how much the detected concentration changes on each time step. For a given robot, if the detected concentration on each time-step has relatively stayed the same, then the robot changes its found plume status to be true. Once enough robots that have the found plume status are near each other, the program terminates.
3) the particles have a new characteristic: an intensity, or the strength of the concentration. Each particle has an initial intensity of 1000 which is decreased at each time step subsequently, as described in Equation (6). In this way, the highest concentration of particles would be in the plume source, like in a real-world marine pollution plume.

$$\text{intensity}^{n+1} = \text{intensity}^{n} \times 0.95. \quad (6)$$

The new states are:

1) $s1$ = finding-the-plume
2) $s2$ = inside-the-plume
3) $s3$ = reacquiring-the-plume

And, the three archetypes are:

1) greedybot,
2) confusedbot, and
3) seekerbot

They are defined by specifying the relative scores of these three states as a set of three numbers $A = [a_1, a_2, a_3]$. The score distributions are presented in Table 5.

Table 5. Robot archetype-score distributions.

Archetype	a_1	a_2	a_3
greedybot	0	0	0
confusedbot	14	0	0
Seekerbot	14	-7	-7

As in the first experiment, each robot is also assigned a running-score distribution $R = [r_1, r_2, r_3]$ which is updated at each time step during the simulation. The default running-score (R) distributions of each state are given in Table 6.

Table 6. Robot running-score distributions.

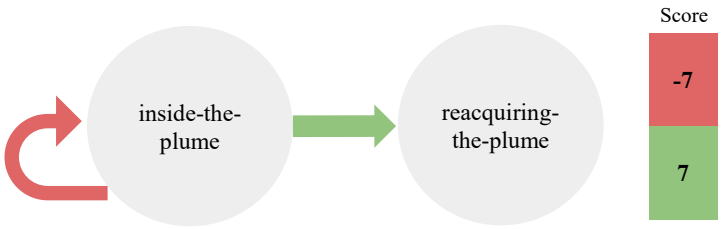
State	r_1	r_2	r_3
s_1, finding-the-plume	10	1	0
s_2, inside-the-plume	0	10	1
s_3, reacquiring-the-plume	0	1	10

And, the state transition distribution, similar to the first experiment, is given in Table 7.

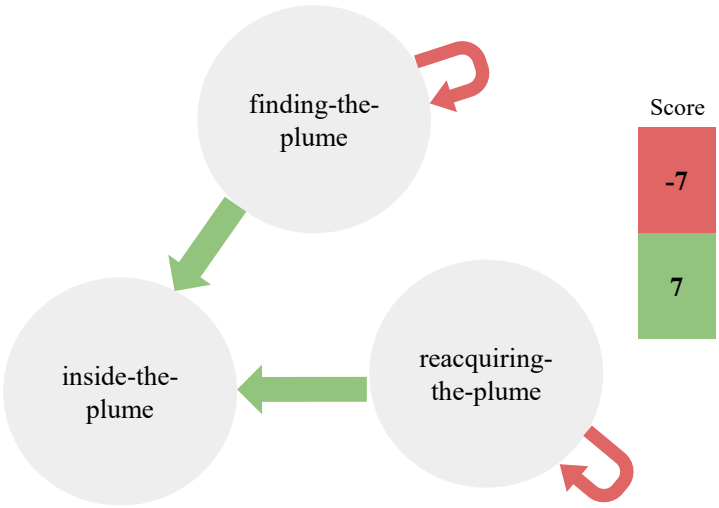
Table 7. Robot state-transition distributions.

State Transition	t_1	t_2	t_3
$s_1 \rightarrow s_2$ (finding-to-inside)	-7	7	0
$s_2 \rightarrow s_3$ (inside-to-reacquiring)	0	-7	7
$s_3 \rightarrow s_2$ (reacquiring-to-inside)	0	7	-7

Similar to the first set of experiments, the ability for a robot in a certain status is determined by the diagrams in Figure 4. This is a visual representation of the data in Table 7 and includes the conditions of changing statuses, including if the robot started to or stopped detecting concentration.



(a) Condition: Stopped detecting concentration



(b) Condition: Started detecting concentration

Figure 4. Second set of experiments: transition between statuses.

To eliminate any confounding variables, the experiments were run with ten different plume locations and velocities. The probability distributions of archetypes are given in Table 8 (similar to Table 4 in the first experiment). The results are described in the Results section.

Table 8. Archetype distribution for the second experiment trials.

Trial #	q_{laz}	$q_{confused}$	q_{greedy}
1	0.33	0.33	0.33
2	0.10	0.1	0.8
3	0.10	0.80	0.10
4	0.80	0.10	0.10
5	0.25	0.25	0.50
6	0.25	0.50	0.25
7	0.50	0.25	0.25
8	0.40	0.40	0.20
9	0.40	0.20	0.40
10	0.20	0.40	0.40
11	0.50	0.50	0.00
12	0.50	0.00	0.50
13	0.00	0.50	0.50
14	1.00	0.00	0.00
15	0.00	1.00	0.00
16	0.00	0.00	1.00

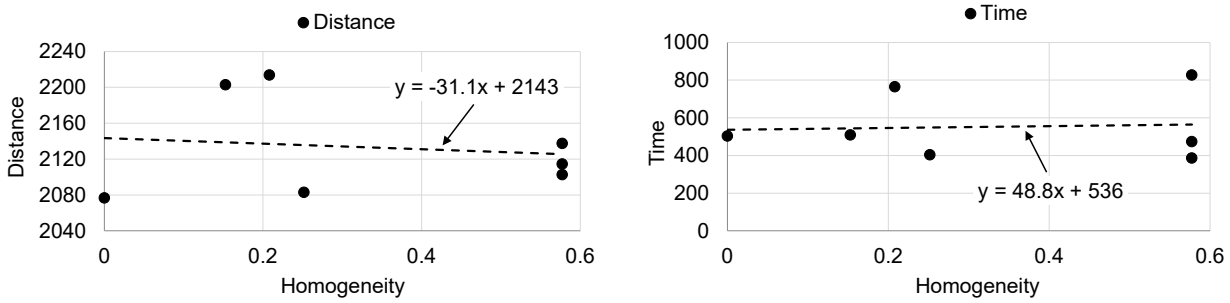
Having these variations of distributions allows us to observe the effects of the individual archetypes as well as the different variations of diversity. For each distribution at each plume location, the simulation is run ten times. The results of this experiment are presented in the Results section.

Results

First Experiment

The results of the first experiment, with the seven archetype distributions (Table 4), are described below. Each archetype distribution is run for two different plume source locations. The first plume source location at (8405, 11731) and with a velocity of (-3.4, 3.7) is in the top right corner of the section and emits plume particles towards the top of the simulated area; this is not accessible to all robots in the beginning. The second plume source location at (5830, 10906) and with a velocity of (-0.3, -5.9) was more centered around the simulation and its particles were easier for robots to encounter.

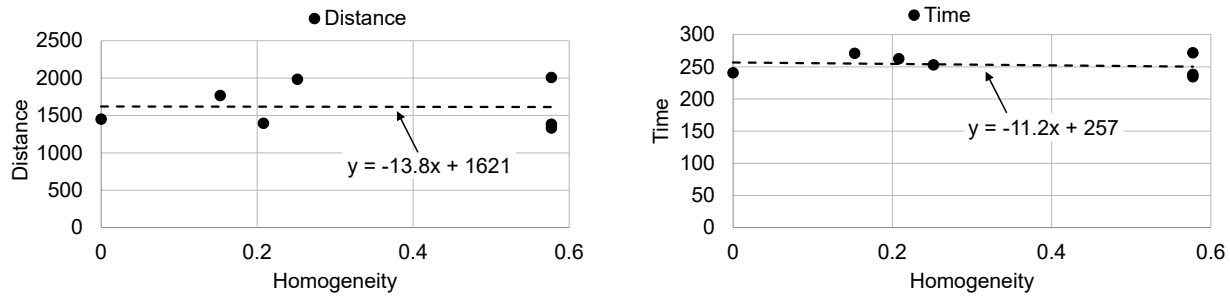
The results are displayed in Figures 5 and 6. On the x-axis is the measure of homogeneity of the robots' archetypes. It is derived from the archetype distribution (Table 4) by taking its standard deviation. In this sense, the homogeneity is the maximum when all the robots are of the same type (diversity of robots is minimum) and is minimum when they are equally mixed (e.g. 1/3 for each of the three types, so diversity is maximum). The distance and time are displayed on the y-axis. Distance is the average distance a robot travelled and time is the average time taken to reach to the plume source. Also, a linear regression is included to illustrate the general trends. With increase in homogeneity the distance is observed to decrease whereas the time to increase. Note that trials numbers 5, 6, and 7 will have the same homogeneity as each represents the case of all robots being one of each of the 3 archetypes. These are the rightmost points with the highest homogeneity. These are distinguished with each other in Figure 7.



(a) Distance versus homogeneity

(b) Time versus homogeneity

Figure 5. Plume detection results for the plume source at (8405, 11731) and a velocity of (-3.4, 3.7).



(a) Distance versus homogeneity

(b) Time versus homogeneity

Figure 6. Plume detection results for the plume source at (5830, 10906) and a velocity of (-0.3, -5.9).

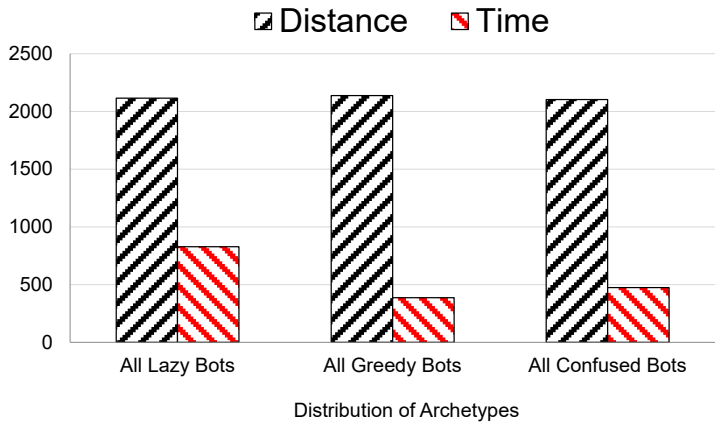


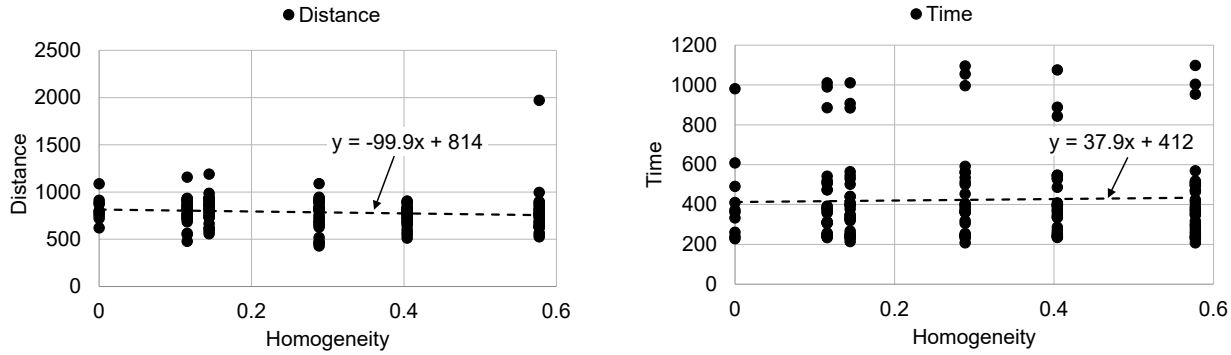
Figure 7. Performance of robots of different archetypes, first experiment.

The distance results at the first plume source (Figure 5) were between 2000 units and 2500 units while the time results were generally between 250 and 750 time-steps. However, for the second plume source location (Figure 6), the distance results were generally between 1000 and 2500 units and the time results were between 200 and 300 time-steps. Additionally, Figure 7 shows the performance of each archetype. Clearly, the lazybot performance was the worst. While the distance results were within 100 units of each other, the time results were much more varying, with the lazy bots being much slower in getting the results.

The location of the plume source significantly impacted the results of the experiment and was a determined to be confounding variable that affected the results and prevented definite conclusions to be drawn about the performance. These observations led to design of a second set of experiments, as described below.

Second Experiment

A second set of experiments was run with additional plume source locations and velocities, both randomly distributed to minimize its confounding effect, as noted above. Also, since the lazy bot proved to be unhelpful in the previous set of experiments, a new robot archetype, *seekerbot* is added for the second set of experiments, described in the next section. The results for the second set of experiments are presented in Figures 8 and 9.



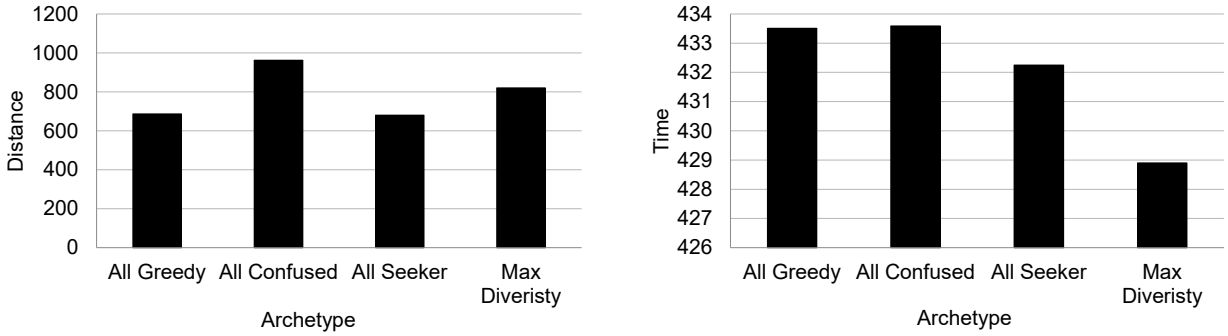
(a) Distance versus homogeneity

(b) Time versus homogeneity

Figure 8. Plume detection results for the second experiment. Each point represents the results for a specific distribution of robot archetype and a plume source. The points represent every trial, with each plume and each archetype distribution.

As evident from the linear regression, while the distance seemed to decrease as the homogeneity increased, the time-steps seemed to increase.

Figure 9 takes a closer look at the homogeneous distributions (only one archetype present compared to the diverse distribution where all three types are mixed in equal proportions. It is even from the distance plot that the confusedbot is the worst-performing.

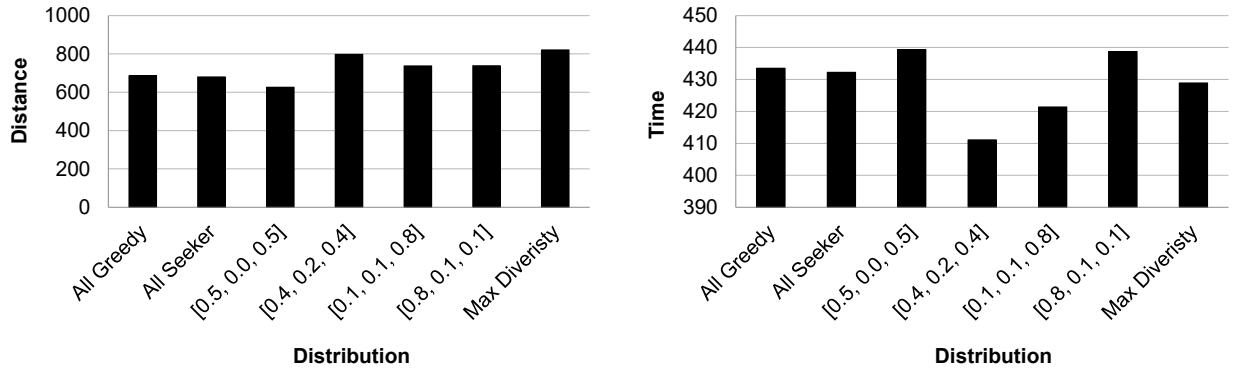


(a) Distance versus archetype

(b) Time versus archetype

Figure 9. Performance of robots of different archetypes, second experiment.

Figure 10 represents the different extents to which diversity can be present in a swarm and compares the results. Knowing that the confused archetype is the worst-performing one, only the distributions which have a minimum presence of the confusedbots are included.



(a) Distance versus diversity

(b) Time versus diversity

Figure 10. Performance of robots of different distributions, second experiment.

Discussion

The results of the first set of experiments provided insight into the helpful and non-helpful archetypes. It was discovered that the lazy bot, which simply remained still and reported concentration, was unhelpful in this task. If a still robot was far from the plume and never detected any concentration, this robot would consistently report 0.0 particles

without making progress. Knowing this, this archetype was replaced with a seeker bot. Additionally, the plume source location was found to be a confounding variable. The first plume source location at (8405, 11731) and with a velocity of (-3.4, 3.7) is in the top right corner of the section and emits plume particles towards the top of the simulated area; this is not very accessible to all robots. However, the second plume source location at (5830, 10906) and with a velocity of (-0.3, -5.9) was more centered around the simulation and its particles were easier for robots to encounter. This would explain why the swarm simulated with the first plume source was more inaccurate and took longer.

Knowing these results in the first experiments, improvements were made in the simulations. In the second set of experiments, it was noticed that more homogenous swarms tended to yield slightly better results, meaning smaller distance and time results. To reveal why this behavior occurred, performance of different archetypes and their performance were studied. Figure 9 reveals that the confused bot seems to have the worst behavior, while the seeker and greedy bots seem to perform the best. This is attributed to the behavior of each of the archetypes. A confused bot's main activity is seeking; it has an equal probability of staying in the finding plume state or the inside plume state. This proves to not be as beneficial because, if the robot has not detected particles and it goes into the inside the plume state, it will likely keep detecting no plume particles. However, the seeker bot is always seeking and has a small chance of leaving this phase. This is more productive because it is constantly moving and is likely to eventually encounter the plume.

Figure 10 compares different levels of diversity while the better-performing archetypes remain the most prevalent. The distributions that minimize or eliminate the confused bots are more beneficial than increasing the number of confused bots in a swarm to maximize diversity. When compared to complete homogeneity (on the leftmost bars), some distributions with a little diversity, like the [0.1, 0.1, 0.8] and [0.8, 0.1, 0.1] distributions, perform worse while the [0.5, 0.0, 0.5] distribution, which eliminates confused bots, performs better. This suggests eliminating bad-performing archetypes and decreasing diversity is better than including them to increase diversity. Behavioral diversity is not always beneficial. If the performance of each individual archetype is known, then the diversity should accordingly be adjusted. But when the performance of individual archetypes is not known, diversity might be the safer option.

While it is true that behavioral diversity is generally beneficial as previous studies describe, we cannot ignore the few cases in which this is not true. Before creating a heterogeneous swarm and expecting better results, evaluating the different diverse behaviors and their individual effects on the overall performance is crucial. For example, in honeybees [13] where there are different roles of the bees, bees' capabilities are not used for unnecessary or harmful behaviors because it would hurt the overall goal of the swarm.

Many improvements can be made to this simulation to make it more representative of the real world. First, the environment can be three-dimensional, since we are modeling a body of water. The plume can be modeled with more characteristics resembling those of a real pollution plume, including a time-varying flow field and a better flow of the particles. Additionally, more experimentation can be done to reveal which archetypes perform the best. More distributions should be tested to find a more ideal distribution of archetypes. Also, more termination conditions can be tested to yield better results. Implementing these and similar changes can increase the performance of this simulation and lead to a more accurate representation of the real-world scenario.

Conclusion

In this experiment, we measure the performance of a swarm of robots in detecting the source of a marine pollution plume. After programming a simulation that models this environment and altering the parameters of the swarm, it was found that having higher frequencies of better-performing archetypes is better than having more behavioral diversity, where better and worse-performing archetypes have almost the same count in a swarm. But this is not to say that all robots should be the same archetype or that behavioral diversity should be eliminated. This same concept should be tested in different applications, as the archetypes will definitely differ and the results may differ. With more experimentation and testing, we can find more helpful archetypes, add new robot behaviors, and develop better distributions of robot archetypes to improve the swarm's performance.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

1. Tan, Y., & Zheng, Z. (2013). Research advance in swarm robotics. *Defence Technology*, *9*(1), 18–39. https://doi.org/10.1016/j.dt.2013.03.001.

2. Liekna, A., & Grundspenkis, J. (2014). Towards practical application of swarm robotics: overview of swarm tasks. *Engineering for rural development*, 13, 271-277. https://www.tf.llu.lv/conference/proceedings2014/Papers/46_Liekna_A.pdf.

3. Bakhshipour, M., Jabbari Ghadi, M., & Namdari, F. (2017). Swarm Robotics Search & Rescue: A novel artificial intelligence-inspired optimization approach. *Applied Soft Computing*, *57*, 708–726. https://doi.org/10.1016/j.asoc.2017.02.028.

4. Nguyen, L. A., Harman, T. L., & Fairchild, C. (2019). Swarmathon: A Swarm Robotics experiment for future Space exploration. *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, *24 2*. https://doi.org/10.1109/ismcr47492.2019.8955661.

5. Schranz, M., Umlauft, M., Sende, M., & Elmenreich, W. (2020). Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, *7*. https://doi.org/10.3389/frobt.2020.00036.

6. Shahzad, M. M., Saeed, Z., Akhtar, A., Munawar, H., Yousaf, M. H., Baloach, N. K., & Hussain, F. (2023a). A review of Swarm Robotics in a Nutshell. *Drones*, *7*(4), 269. https://doi.org/10.3390/drones7040269.

7. Vicerra, R. R., Dadios, E. P., Bandala, A. A., & Lim, L. A. (2014). Swarm robot system for Underwater Communication Network. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *18*(5), 769–775. https://doi.org/10.20965/jaciii.2014.p0769.

8. Miner, D. (2007). Swarm robotics algorithms: A survey. *Report, MAPLE lab, University of Maryland.* https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9876ba823369c86dfe83e963fa9cd1dd1e9fa1e3.

9. Wahab, M. N. A., Nefti-Meziani, S., & Atyabi, A. (2015, May 18). *A comprehensive review of swarm optimization algorithms*. PLOS ONE. https://doi.org/10.1371/journal.pone.0122827.

10. Champion, B. T., & Joordens, M. A. (2015). Underwater Swarm Robotics Review. *2015 10th System of Systems Engineering Conference (SoSE)*, *49*, 111–116. https://doi.org/10.1109/sysose.2015.7151953.

11. Hunt, E., “Swarm Robotics into The Real World,” Faculty of Engineering Research Showcase 2021 lecture series, University of Bristol, https://www.youtube.com/watch?v=7nAzbZDn0RA.

12. Dudek, G., Jenkin, MichaelR. M., Milios, E., & Wilkes, D. (1996). A taxonomy for multi-agent robotics.

Autonomous Robots, *3*(4). https://doi.org/10.1007/bf00240651.

13. Kengyel, D., Hamann, H., Zahadat, P., Radspieler, G., Wotawa, F., & Schmickl, T. (2015). Potential of heterogeneity in collective behaviors: A case study on heterogeneous swarms. *Lecture Notes in Computer Science*, 201–217. https://doi.org/10.1007/978-3-319-25524-8_13.

14. Fahad, M., Guo, Y., and Bingham, B., Simulating fine-scale marine pollution plumes for autonomous robotic environmental monitoring, *Frontiers in Robotics and AI*, 2018(5), https://doi.org/10.3389/frobt.2018.00052.