Multi-robot systems
Lab 1

Jorge Aranda
NIP: 800839

Alberto Zafra
NIP: 876628

MRGCV - 2023/2024

1 Introduction

In this Lab, we address the problem of aggregating a swarm of mobile robots (also known as Mobots) within a dynamic spatial environment characterised by multiple nests and obstacles. Leveraging insights from previous work in swarm robotics, particularly [1] and [2], we propose a novel PFSM (Probabilistic Finite State Machine) micro-behavior for Mobots to facilitate their aggregation in the large nest.

The scenario entails the creation of a spatial environment comprising two nests located in different quadrants, each with varying areas. Additionally, we introduce random obstacles and mobile obstacles to add complexity to the environment. The Mobots, numbering N=50, are initially scattered across the space, their positions randomized. Each Mobot possesses limited sensory capabilities, perceiving only the presence of other bodies nearby or in contact, without knowledge of their position or size. They are equipped with omnidirectional communication abilities with a specified range (R=1, 2, 4) and possess basic cognitive functions, including quadrant awareness, step counting, and memory for a few integer numbers.

The overarching objective is to design a PFSM micro-behavior that enables Mobots to aggregate in the largest nest autonomously. This entails the formulation of behavioral rules and decision-making processes that leverage local interactions and environmental cues to guide the collective behavior of the swarm. The design rationale for the proposed micro-behavior will be discussed in detail, highlighting considerations such as scalability, robustness, and adaptability to varying environmental conditions.

Key performance indicators (KPIs) will be defined to evaluate the efficacy of the proposed approach, including the fraction of surviving Mobots, the fraction of Mobots that are exploring, the fraction of Mobots moving towards the largest nest with an aggregation purpose (also known as nesting) and the fraction of Mobots aggregated in the large nest (also known as nested). To provide a comprehensive analysis, simulation experiments will be conducted, varying parameters such as nest areas (A), communication range (R), and tuning parameters within the PFSM micro-behavior.

In the subsequent sections, we will present the design and implementation of the PFSM micro-behavior, followed by simulation results and analysis. Through this endeavor, we aim to contribute to the advancement of swarm robotics research by elucidating strategies for autonomous aggregation in dynamic and heterogeneous environments.

2 Approach

As it has been aforementioned, the development of the behaviour of the robots have been based on the knowledge provided by the previous works of O. Soysal [1] and H. Hamann, *et al.* [2]. In this specific case, the Mobots are randomly initialized within the different quadrants of the map with an initial purpose of finding which is the largest nest within the scenario. In order to achieve that, they are focused in exploring and communicating with other Mobots, thus, they move randomly aiming to encounter more Mobots, which already know some information about the nests, or to intersect an unknown nest, measuring it during the process. The measurement of the nest is characterized by determining the time spent within it, in order words, when a Mobot enters to within a nest it starts counting the time until it goes outside the current nest. This simplistic approach may induce to poor measurements as it is not certain that the robots do these measurements through the diagonal of the nest (as they could enter through a tangent of the nest and measure only a minuscule part of the whole) and therefore, transmit it. Nonetheless, as these measurements could be replicated by other Mobots the quality of the measurement is increased drastically, as aggregating multiple measurements taken by different individuals can lead to a more accurate and reliable estimate of the true value than relying on a single measurement, also known as ”the wisdom of the crowd”.

On the other hand, the comparison of nest known sizes between two Mobots is performed by considering the maximum value as the truth, therefore, if a nest was wrongly measured, it would be easily corrected by other's measurement, as it has been previously stated. Nonetheless, this criteria of preserving only the largest value of each nest has its own drawbacks, as it might be that an incorrect measurement where the robot has been within the nest more time than expected, due to perturbations on its trajectory, such as the avoidance of other Mobots, would be almost impossible to correct. Thus, a good idea for correcting these kind of mistakes would be to use a weighted average of the information that is transmitted between users, trusting more in the data that has been broadcasted by the Mobot which has been more communicative until now, using equation 1, where $nest\_size_{MobotX}$ is the believed nest size of the Mobot X and $comms_{MobotX}$ is the number of communications that the Mobot X has been done until now. Nevertheless, despite of the supposed improved accuracy of this methodology, the convergence of the nest size belief is really slow, and not always correct, as this methodology is highly dependent on the initial measurements of the simulations, as these robots are more likely to transmit more their assumptions than Mobots that receive or measure the nest size in latter stages of the simulation. Thus, the final consensus methodology that has been used relies in transmitting the belief with a maximum nest size per quadrant as it is shown in equation 2.

$$nest\_size = \frac{nest\_size_{Mobot1} \times comms_{Mobot1} + nest\_size_{Mobot2} \times comms_{Mobot2}}{comms_{Mobot1} + comms_{Mobot2}} \tag{1}$$

$$nest\_size = \max(nest\_size_{Mobot1}, nest\_size_{Mobot2}) \tag{2}$$

Solved the consensus problem, the next objective is to improve the convergence of the simulation by aggregating all the Mobots in the largest nest. Therefore, an information validation criteria is also needed. The initial approach to the aforementioned validation criteria was a rough evaluation of the number of communications of the Mobots, thus, if the robot had communicated with other robots more than 2000 times, its information was considered to be reliable and this Mobot changed its state from ”exploring” to ”nesting”, and therefore, its movement was directed to aggregating itself to the nest which it has been consideratedas the largest one. Nonetheless, this strict limit provoked undesired behaviours between the robots, as they could stay ”exploring” an unlimited amount of time, leading to a non-convergence of the system. On the other hand, this validation criteria may lead to a ”nesting” state before getting information of other possible nests within the environment. Therefore, in this case a probability for changing between the ”exploring” state and the ”nesting” state has been used. This probability is not only based on the number of communications that the robot has done during its lifetime, but also on the time that has passed since the simulation has started; computed by following equation 3. Where, $p(nesting)$ is the probability to change its state to ”nesting”; and $mean$ is the mean between the number of communications of the Mobot X and the time that has passed since the start of the simulation, computed as it is shown in equation 4.

$$p(nesting) = \frac{1}{1 + e^{(2000-mean)\times 0.005}} \tag{3}$$

$$mean_{comms\ and\ time} = \frac{time(s) \times 100 + n\_comms_{MobotX}}{2} \tag{4}$$

Hence, by applying the previously shown, equation 3, the probability to changing the state of a Mobot from ”exploring” to ”nesting” grows smoothly following a sigmoidal curve, as it is shown in Figure 1. On the other hand, in addition to this probability, an additional condition has to be met before changing to the ”nesting” state. This condition is that the robot has to have information of at least two nest within the environment, avoiding possible mistakes due to information scarcity.

Finally, after the Mobot has been sufficiently informed and it has verified its belief, it can direct its efforts to getting to the largest nest and, thus, aggregating there. But, another question arises, the robot is only able to know its own location and the center of the quadrants that subdivide the environment, it does not know where the rest of the elements present on the scene are located. Accordingly, the robots have to know where to go when they enter to the ”nesting” state. A simplistic approach to this problem is that,

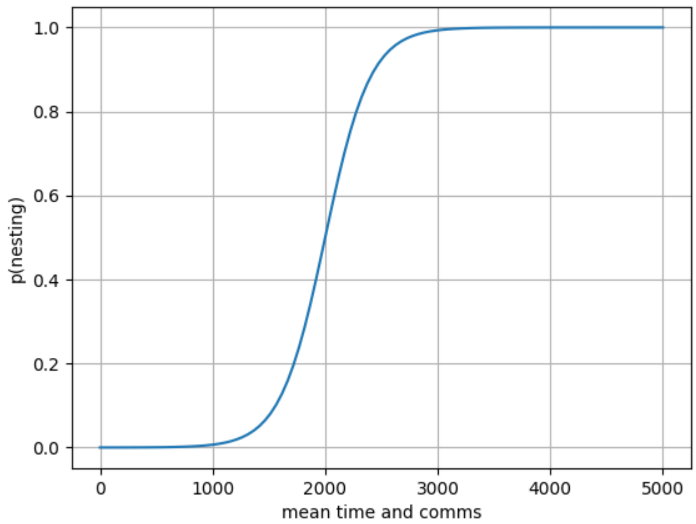

Figure 1: Growth of the probability to become nesting

during the nest measurement, the robot stores the entering and the exiting locations where it commences or concludes contact with the nest. Afterwards, the median point between the two previously mentioned locations is computed and, therefore, shared with the rest of the Mobots as the nest aggregation location. Entering to the "nested" state as soon as they get in contact with the largest nest. It is worth noting, that during the "nesting" state, the robots move towards the kown "aggregation location" following a linear and more aggressive trajectory, avoiding obstacle with a lower safety distance.

3 State Machine

The behaviour of the mobots follow a main PFSM of three states. Each state has influence not only on to how the mobot moves, but also on what actions is it allowed to perform. Apart form the main PFSM, we have also implemented some smaller FSMs. The main PFSM (Figure 2) has three different behavoirs, represented in three states: $Exploring$, $Nesting$ and $Nested$.

The mobot starts in **Exploring** mode, where it sets a random destination. From that state, there are three possible options depending on randomness. There are two random probabilities: α and γ. α determines whether the mobot changes its destination in exploring mode. γ determines whether the mobot explores or goes to the biggest nest **according its beliefs** ($Nesting$ state). Additionally, in order to jump to the $Nesting$ state, the amount of known nests[1] must be 2 or more. Thats why the condition for changing to the $Nesting$ state is ($\gamma \wedge n\_nests > 2$). If that condition is fulfilled, the mobot will change the state. If not, α determines if it jumps to $Exploring$ (changing the random destination) or remains in $Exploring$ (without changing anything). Note that this last two options end up in the same state with the important difference of seting a new random route or not.

If the mobot enters in the **Nesting** state, it will set a destination to the center of the biggest nest it knows. From here, the mobot can either arrive to the biggest nest or randomly change to exploration mode before arriving to the nest if α and not γ happens. In order to know if you should change to $Nested$ state, you must be both in a nest and in the quadrant of the biggest nest you know.

Once a mobot is in the **Nested** state, it can only come out to $Nesting$ if the nest in which it is is not in the quadrant of the biggest nest it knows. This can happen if an external mobot passes by and communicates the nested mobot saying that there is a bigger nest.

[1]Known nest: a nest is known to a mobot if it has a measurement of it. Either an own measurement or a measurement communicated by someone.

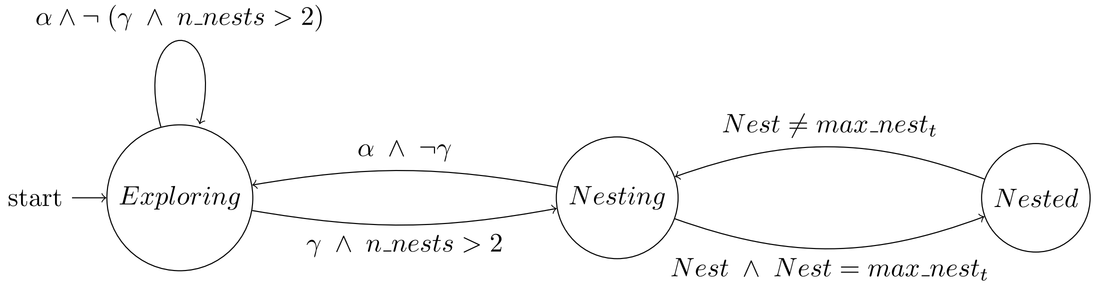

Figure 2: Main PFSM

The state not only affects to the movement of the mobot, it also sets what is it allowed to do and what not. For example, we can see in Figure 3 how the states are crucial in the mobot ability to measure a nest. In order to measure a nest, a mobot stores when did it enter to it. When the mobot comes out, it just subtracts the initial time T_{ini} to the current time, T_{end}. But, what if the mobot starts the simulation inside a nest. It will detect that it has entered a nest and store the current time. This will make the measurement smaller than it should be, that is why the mobot starts in a state where it thinks that it is in a nest, but without an initial time. This way, if the mobot apears in a nest, the measurement will not be wrong (because there will be no measurement). And, if the mobot does not apear in a nest, nothing will happen since in the next step it will just see that it is, in fact, not in a nest.

But, ¿Why maintaining that state? ¿Is it useful after the first iteration?. For several reasons, it is interesting to have two *inNest* states (one measuring and one not). Lets say the mobot is nested. It is stopped in a nest. We do not want it to calculate how much time it is inside the nest. Actually, we want it to erase when did it enter so, in case of realising it is not the biggest nest, it doesn't suddenly think it has been a huge amount amount of time inside and, therefore, think that it is big. So we are only going to allow the robot to measure only in Exploration mode. Also if the robot is exploring inside a nest and suddenly decides that it wants to go to the nest (enter in $Nesting$ mode), the movement inside that nest will not be straight. Actually, this implies that the variance of the measurement will be bigger and therefore, we cannot allow that inaccuracy. That is why it is reasonable to cancel some measurements in the middle of them.

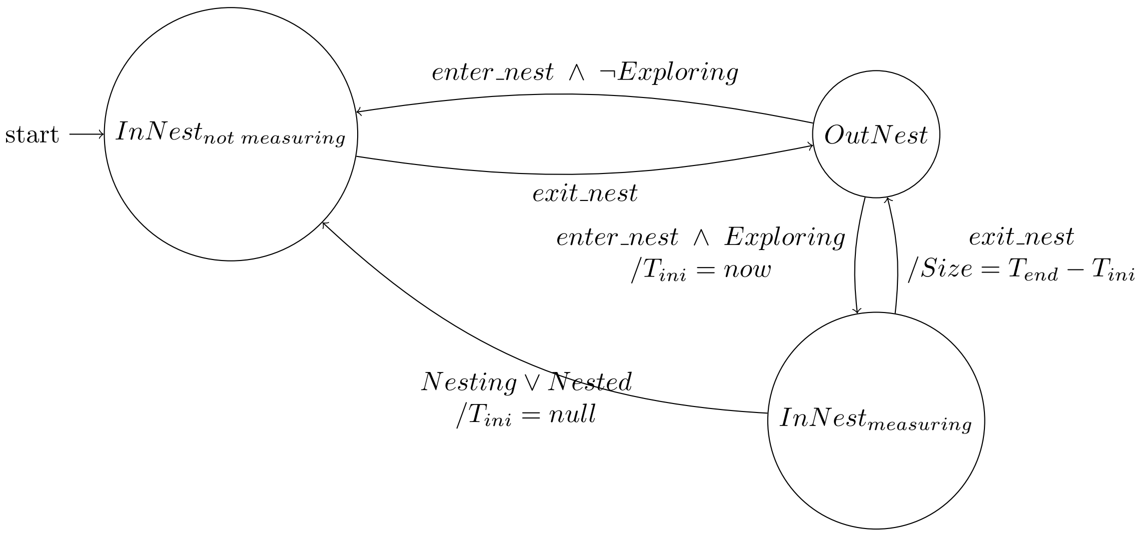

Figure 3: Measuring state machine

4 Tuning

In this section the tuning of the different parameters that have been modified during the evaluation of the implementation is analyzed and presented.

4.1 Nest Area

The first parameter that has been tested was the variation of the areas of the nests present within the environment. When changing the area of the nests two different things might happen. If we make the nests bigger, more mobots enter and the better the measurements are, this can be proved in equation 6 since the max of two random variables is bigger than the expectancy of one.

$$\chi_n = uniform\ random\ number \tag{5}$$

$$max(\chi_1, \chi_2) \geq \chi_3 \tag{6}$$

The video [3] demonstrates the previously presented hypothesis. In the KPI graph of the Figure 4 we can see that they aggregate even faster since they have information sooner. When we tried smaller nests, less mobots entered them, so the measurements were less probable to be accurate along the time. So mobots tend to share worse information and end up in the worst nest. One way of solving this problem would be augmenting the probability of γ.

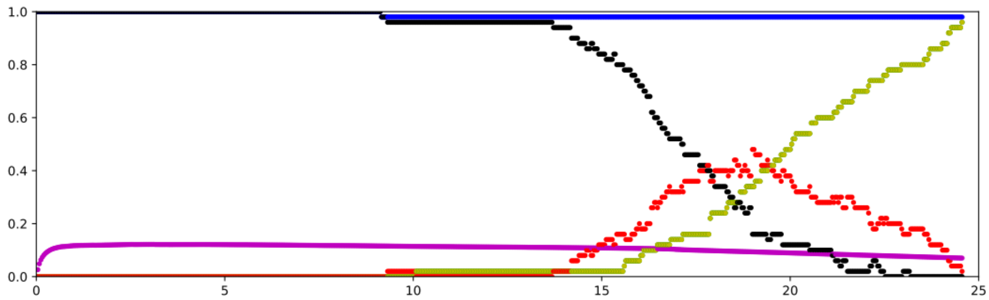

Figure 4: Big nests KPI

4.2 Communication Radius

If we change the radius of communication, something similar happens. Mobots exchange information faster and end up in **a** nest sooner. However, this implies that they tend to rush and get to conclusions sooner, which can be wrong sometimes. But as it is more probable to enter to the big nest than to the small, if a mistake is made it can be redeemed in just seconds. The video link [4] shows an example of aggregation where almost all mobots are aggregated in less than 20s (although some don't even know any nest as a consequence of the haste). With a smaller radiuis it happens the oposite. They tend to go later to the nest, as time passes, the truth tend to emerge, but splower. An example can be seen in [5] where it takes 40s and, even though, not all the mobots end up in the nest.

4.3 Robot Movement

Another group of parameters that have to be modified are the ones related to the robot movement. Concretely the parameters that are in charge of the safety of the robot and how they move.

As it is known, when exploring a more random movement is desired, where the destination is not as important as the area covered by the robot, as a wider area increases drastically the possibilities of further communications or measurements. Hence, initially a "zig-zag" movement with a high safety coefficient, determined by a bumper (distance to activate collision avoidance) and an Obstacle Evitation Factor (OEF), was implemented. However, the parametrization of these features was not optimal, obstructing the exploration of the whole environment, as having a high bumper (of 1.25m approximately) and a high OEF (of 0.3) provoked the isolation of some Mobots, that remained blocked in the corners of the scenario while avoiding possible obstacles. Therefore, another movement behaviour was designated for the exploration task by using a linear, also known as "keepgoing", movement towards the center of the quadrants and reducing the bumper and OEF coefficients to 0.32m and 0.15 respectively. The tuning of these parameters resulted in a better area coverage and maintained the quality of the collision avoidance as it can be seen in video [6].

On the other hand, as it has been previously mentioned, the movement of a Mobot in a "nesting" state is way more aggressive that the movement of a robot in a "exploring" state, in order to achieve this behaviour, the Mobot has to move at its maximum speed during the complete linear trajectory towards the largest nest. Nevertheless, this kind of movement would be useless if the robots are destroyed, thus, the collision avoidance feature needs to be active during the nesting process. Furthermore, as opposed as it was initially though, the collision avoidance, bumper and OEF, parameters have to be larger than when exploring, being 3.5m and 0.5 respectively. This is due to the high velocity reached by the mobot, being necessary a bigger reaction time.

4.4 Robot Consensus

Finally, the last group of parameters that is needed to be tuned are the ones related to the consensus. As it has been aforementioned, in order to validate the knowledge of the mobot, if has to have performed a minimum amount of communications. In this case 2000 communications have been selected, as different simulations showed that a lower number of communications may lead to mistaken beliefs or to a never converging algorithm, where if the robots that have been initialized in the same quadrant or hemisphere of the largest nest receive any information of the other nest, most of them would aggregate in the largest nest, keeping isolated the rest of the mobots. Whereas, a higher number of communications required leads to a slower convergence of the algorithm, as the robots must spend more time validating their knowledge and, thus, will spend more time exploring.

Another critical parameter that has to be tuned is the time validation criteria. Initially, this criteria was implemented as an absolute criteria, in other words, after an specified amount of time, all the robots that have information of more than one nest should start nesting automatically, as it is shown in figure 5. Nonetheless, as it can be easily seen in figure 5, this behaviour may lead to a huge discontinuity where, not all the robots would change state autonomously, resulting in a non-converging algorithm. Hence, in order to solve this, a proper amount of time has to be determined, in this case 20s have been selected for 50 mobots; and the time validation criteria has to be introduced within the sigmoidal curve, as it has been previously presented, resulting in a fast converging algorithm, as it is shown in figure 9.

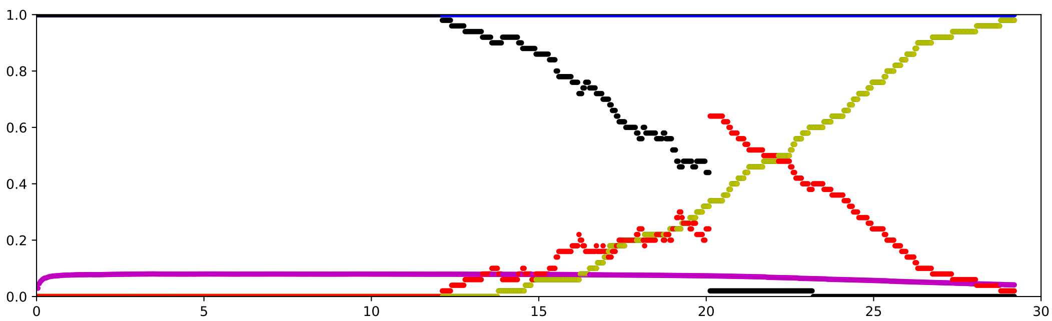

Figure 5: Time independent validation criteria KPI

5 Results

After successfully tuning the parameters a series of simulations were done in order to verify the behaviour of the swarm. These simulations where done by varying the number of Mobots present within the environment, the area and number of nests and the communication range of each robot, an example of the initial layout of the environment is presented in Figure 6.

Hence, the three different videos [6], [7] and [8]; show the correct behaviour of the robots along the simulation. First exploring the scenario and communicating between them; and, afterwards, when they find a consensus, nesting towards the nest that they conceive as the largest one of the environment; ending with an aggregation in the pink nest, as shown in Figure 7. For clarity purposes, Figure 8, obtained from [6], shows a zoomed version of the robot aggregation.

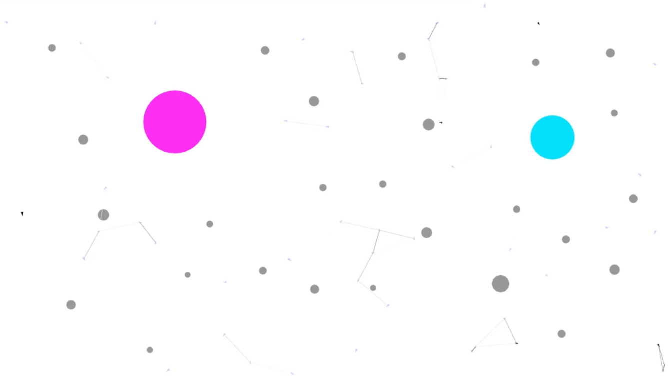

Figure 6: Initial Layout

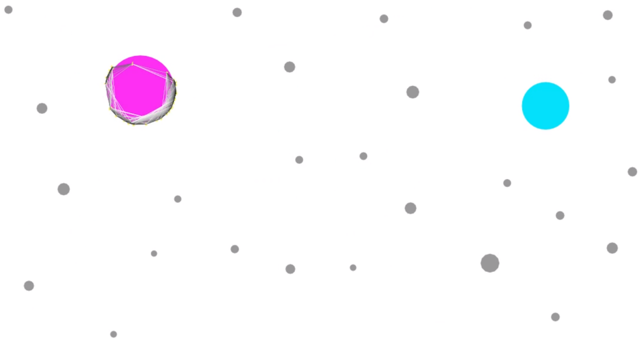

Figure 7: Robots Aggregated

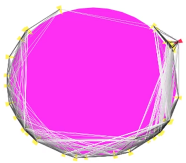

Figure 8: Zoomed Final Result of Robots Aggregated

On the other hand, the video [9] presents a simulation demonstrating the principle of "the wisdom of the crowd." It focuses on how robots, whose beliefs initially identify the blue nest as the largest within the environment, are persuaded by another Mobot to recognize the pink nest as the true largest nest. Consequently, they redirect their movement towards the real largest nest, aiming to aggregate within the pink nest.

Finally, Figure 9 displays the Key Performance Indicators (KPIs) recorded in the simulation depicted in the video by [6]. The fraction of surviving robots throughout the simulation is represented in blue, while the fraction of exploring robots is in black. The fraction of Mobots in the "nesting" state is depicted in red, and the fraction of aggregated robots within the largest nest is shown in green. The simulation time scale is indicated in magenta. As depicted in the plot, almost no robots break or cease to function during the simulation, thanks to the precise tuning of obstacle avoidance parameters at each step. It is noteworthy that, as previously mentioned and evident in the videos, Mobots exhibit more aggressive movement in the "nesting" state, yet they continue to successfully navigate around obstacles. Furthermore, as shown in Figure 9, the fraction of exploring robots remains constant until the number of communications reaches an average of 2000, a milestone achieved approximately 13 seconds after the start of the simulation. Subsequently, the number of exploring robots begins to decline as the number of nesting robots increases, resulting in a noticeable decrease after 20 seconds. This trend arises because the probability of state change is nearly 1, given that both the time and communication criteria are met. However, a small subset of Mobots continues to explore, as they have not met the minimum knowledge requirement of information about more than one nest. Conversely, as previously mentioned, the number of nesting robots peaks 20 seconds into the simulation, after which it gradually declines, indicating that the majority of robots are nesting and moving toward the largest nest for aggregation. The video of the world that corresponds to the Figure 9 can be seen in the link [10].

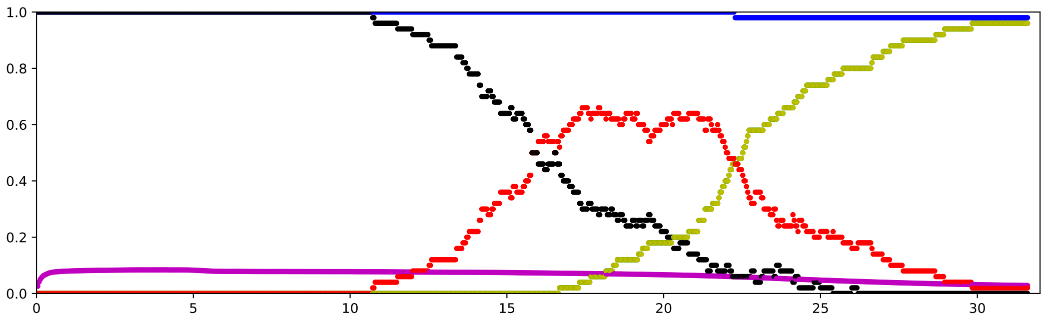

Figure 9: Key performance indicators (KPIs) of [10]

6 Conclusions

In conclusion, the presented implementation highlights the effectiveness of employing the principle of "the wisdom of the crowd" in guiding the behavior of autonomous agents towards a common goal. By leveraging communication and shared beliefs, the Mobots successfully navigate their environment, identify the largest nest, and aggregate within it. The results demonstrate the robustness of the proposed PFSM micro-behavior, which enables Mobots to adapt their actions based on local information and collective knowledge.

Moreover, as it has been aforementioned in the analysis of Key Performance Indicators (KPIs), the fraction of surviving robots remains consistently high throughout the simulation, underscoring the efficacy of obstacle avoidance mechanisms. Furthermore, the transition of Mobots from exploration to nesting behavior occurs smoothly, driven by the accumulation of knowledge and the convergence of beliefs about the location of the largest nest.

Finally, the evolution from simulation [9] to simulation [6] reveals the importance of parameter tuning, particularly in relation to communication range and obstacle avoidance. Fine-tuning these parameters significantly impacts the speed and efficiency of the aggregation process, highlighting the need for careful calibration in real-world implementations.

References

[1] O. Soysal and E. Sahin, "Probabilistic aggregation strategies in swarm robotic systems," in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, June 2005, pp. 325–332.

[2] H. Hamann, T. Schmickl, H. Wörn, and K. Crailsheim, "Analysis of emergent symmetry breaking in collective decision making," *Neural Computing and Applications*, vol. 21, no. 2, pp. 207–218, 2012.

[3] A. Zafra and J. Aranda, "Simulation case 0: Big nests." [Online]. Available: https://drive.google.com/file/d/1MODN_17MxyeUWJCKPcakjgnFytBQjINN/view?usp=drive_link

[4] ——, "Simulation case 1: Big radius." [Online]. Available: https://drive.google.com/file/d/1D22kxeYGoFAlsuwDFlNWaPhEmyXYDzwS/view?usp=sharing

[5] ——, “Simulation case 2: Small radius.” [Online]. Available: https://drive.google.com/file/d/1byxSMsz1MppjteOPMJVo2XgUXlul1pqC/view?usp=drive_link

[6] ——, “Simulation case 3: Aggregation in nests.” [Online]. Available: https://drive.google.com/file/d/1YT4HYiJkayw-txeAcNrq37-xJXzHPC-K/view?usp=sharing

[7] ——, “Simulation case 4: Aggregation in nests.” [Online]. Available: https://drive.google.com/file/d/1vyafZllkVy1uMu4xoayDgtLlwPIdGeOI/view?usp=sharing

[8] ——, “Simulation case 5: Aggregation in nests.” [Online]. Available: https://drive.google.com/file/d/1O6gNHMf-WWkf4Ys2diFl2uovOp76ojS9/view?usp=sharing

[9] ——, “Simulation case 6: Consensus use case.” [Online]. Available: https://drive.google.com/file/d/1zuBKCkCeP8ibUW59SSwC0e0qVxzB3yso/view?usp=sharing

[10] ——, “Video of the kpi.” [Online]. Available: https://drive.google.com/file/d/188hnMrouCynOYayEnzzRMmVcYlWYZDVv/view?usp=sharing