XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTIONAL METHODS IN ENGENEERING

Selective Object Aggregation using Swarm Robotics

Renata R. de Ávila[1], Nadia Nedjah[2], Luiza M. Mourelle[3]

[1]*Programa de Pós-Graduação em Engenharia Eletrônica, Universidade do Estado do Rio de Janeiro, Brazil*
prof.renata.avila@gmail.com
[2]*Dept. de Engenharia Eletrônica e Telecomunicações, Universidade do Estado do Rio de Janeiro, Brazil*
[3]*Dept. de Engenharia de Sistemas e Computação, Universidade do Estado do Rio de Janeiro, Brazil*

Abstract. Selective object aggregation refers to the autonomous clustering of objects based on specific attributes. In this context, swarm robotics, inspired by the collective behavior of natural systems, seeks to control a group of simple agents to locate and organize randomly distributed objects by similarity, without centralized control or explicit inter-agent communication. In this work, we propose an adaptation of the Lumer & Faieta algorithm to control a swarm of robots for the selective aggregation of objects by type. The approach is inspired by the self-organizing behavior observed in ant colonies, which naturally sort debris and dead bodies into clusters to maintain nest hygiene. The algorithm is chosen for its efficiency in grouping items based solely on local information, enabling decentralized operation. We present an enhanced version of this algorithm, tailored for robots with limited perception and decision-making capabilities. Despite these individual limitations, the swarm collectively achieves the formation of well-organized clusters. The implementation relies on local neighborhood perception, where the decision to pick up or drop an object is governed by the density of similar items in the robot's immediate vicinity. Adapted probability functions, derived from the original Lumer & Faieta formulation, encourage isolated objects to be picked up and deposited in regions with a higher concentration of similar types, gradually fostering cluster formation. The system is implemented in the CoppeliaSim simulation environment, utilizing Khepera-III mobile robots, chosen for their compact design and versatile sensor suite. Each robot is equipped with a gripper for object manipulation and a vision sensor for object recognition, allowing it to identify color and classify object type. To evaluate the robustness of the proposed method, we test various arena discretizations and swarm sizes. Key performance metrics include the execution time during the search and aggregation phases, as well as the quality and coherence of the final clusters. Simulation results confirm the effectiveness of the approach, demonstrating the system's ability to reproduce clustering patterns akin to those observed in biological systems. The swarm maintains selective aggregation throughout the process, underscoring the potential of simple local interactions to generate sophisticated collective behaviors.

Keywords: Swarm robotics, Object Aggregation, Lumer and Faieta Algorithm, Swarm Intelligence

1 Introduction

Ant colonies are distributed systems in which, despite the simplicity of individual agents, a social organization emerges that is capable of executing complex tasks that far exceed the capabilities of a single ant. The study of such systems and their self-organizing properties offers valuable models for engineering, especially in the development of solutions for optimization problems and distributed control. The interest in natural phenomena of this kind stems from the robustness and adaptability of their emergent behaviors, which are features that are highly desirable in a wide range of technological applications.

This work proposes an adaptation of the Lumer and Faieta's algorithm [1], inspired by ant colony behavior, to control a swarm of robots, performing selective aggregation of objects classified according to their different categories. This method is based on the collective behavior observed in real ants, where simple agents, through local interactions and decisions based solely on immediate environmental information, are able to organize cemeteries or sort larvae into distinct clusters, without any centralized control.

The first computational models of this phenomenon were proposed by Deneubourg [2] and collaborators, who simulated ants moving objects on a grid using only local perception, based on the concentration of similar items in their neighborhood. Later, Lumer and Faieta [1] generalized this model for exploratory data analysis, using agents that move data samples according to similarity and local density to form clusters. Since then, the algorithm

has served as a foundation for other researchers to develop classification and clustering techniques inspired by ant behavior, highlighting its potential in solving decentralized and scalable problems.

Motivated by the simplicity, efficiency, and low computational cost of these models, this study aims to enhance and implement the Lumer and Faieta's algorithm for distributed robot control to perform selective aggregation os objtects according to their physical characteristics, such as color. Simulated experiments are presented using the CoppeliaSim environment and Khepera III robots to selectively cluster cylindrical objects according to their categories, with the goal of assessing the performance and robustness of this approach in various scenarios.

This paper is organized as follows: Section 2 reviews the main related works in the literature; Section 3 presents the fundamentals of the Lumer and Faieta's algorithm; Section 4 describes the proposed methodology for implementing the selective aggregation system; Section 5 presents and analyzes the results obtained through simulation; and finally, Section 6 provides the conclusions of the study and outlines directions for future research.

2 Related Work

The work presented in [3] introduces AntClust, an algorithm inspired by ant behavior for image segmentation, in which agents interact with pixels as if they were items to be clustered. The main contribution lies in the implementation of new movement rules and enhanced pick-and-drop mechanisms, which improve the quality of the clusters and accelerate convergence. In [4], the increasing complexity of ant-inspired algorithms motivated the development of the Streamlined Ant-Based Clustering (SABC), a simplified and efficient version where a single agent can perform clustering with high precision by eliminating auxiliary structures.

The integration of techniques is addressed in [5], which combines Ant Colony Optimization (ACO) with k-means to achieve more effective clustering, particularly in datasets with overlapping regions. The use of k-means-based local search and parameter tuning significantly contributed to the performance of the hybrid algorithm. The study proposed in [2] cites Monmarché as the developer of AntClass, an algorithm based on Lumer and Faieta's model [1], in which ants operate on a toroidal grid and are capable of carrying and dropping multiple objects simultaneously. Ouedfel is also referenced for creating a version of AntClust that replaces the grid with an array of cells connected to a virtual nest, simplifying agent movement and reducing environmental complexity.

In the field of swarm robotics, [6] proposes the use of formal notations such as RoboChart to translate high-level specifications, including Finite State Machines (FSMs), into executable code for autonomous robots. This approach was successfully applied to e-puck robots using the DIRECT algorithm to achieve segregation and aggregation behaviors. In [7], the method Dynamic Interactive Self-Organizing Aggregation (DISA) is introduced, which enables robots to self-organize and cooperate through local interactions, without relying on a central controller. Simulation results show that DISA is scalable, robust, and effective even in noisy environments. The study presented in [8] investigates the influence of informed robots, which have prior knowledge about the aggregation site, on the swarm's collective decision-making. Experiments demonstrated that when at least 50% of the robots are informed, the entire swarm tends to aggregate at the desired location, breaking symmetry and improving coordination efficiency.

3 Lumer and Faieta's Algorithm

The general idea behind this algorithm is that isolated items are picked up and dropped off at locations where there are more items of the same type [1]. When there is only one type of item in the environment, the probability p_p that an unloaded agent, moving randomly, picks up an item is given by $p_p = \left(\frac{k_1}{k_1+f}\right)^2$, where f is the perceived fraction of items in the agent's neighborhood, and k_1 is a threshold constant. When $f \ll k_1$, p_p is close to 1, meaning the probability of picking up an item is high when few similar items are nearby. Conversely, when $f \gg k_1$, p_p approaches 0, making it unlikely for items to be removed from dense clusters.

Similarly, for a loaded agent that moves and finds a suitable location, the probability p_d of dropping the item is defined as $p_d = \left(\frac{f}{k_2+f}\right)^2$, where k_2 is another threshold constant. In this case, when $f \ll k_2$, p_d is close to 0, while for $f \gg k_2$, p_d approaches 1. Notably, the drop behavior follows rules that are approximately opposite to those of the pick-up behavior.

For implementation on robots, it is assumed that the local density function f is computed using a short-term memory available to each agent [9]. Each agent records the last T time units, and f is defined as the number N of objects encountered during those T units, divided by the maximum possible encounters. Assuming only one item can be perceived per time unit, then $f = N/T$. The Lumer and Faieta's algorithm aims to form clusters with low intra-cluster distances, i.e., objects with similar attributes, relative to inter-cluster variance. Initially, all items and agents are randomly distributed across a grid. During the simulation, each agent explores the environment,

making probabilistic decisions based on local perceptions. If an agent is not carrying an item and encounters one, it computes f and the pickup probability p_p, then generates a random number R to decide whether to pick up the item ($R \leq p_p$). If it is carrying an item and finds an empty location, it recalculates f and the drop probability p_d, possibly dropping the item if $R \leq p_d$. After each decision, the agent moves to a randomly selected neighboring cell not occupied by another agent. This iterative process leads to the emergent formation of clusters of similar objects. When multiple item types o_i are present, the same process applies, with f replaced by f_{o_i}, the fraction of recently encountered items of type o_i. The pickup and drop probabilities are then adjusted accordingly, with constants k_1 and k_2 regulating the sensitivity, similar to the basic model.

$$p_p(o_i) = \left(\frac{k_1}{k_1 + f(o_i)}\right)^2 \qquad p_d(o_i) \quad = \begin{cases} 2f(o_i), & \text{se } f(o_i) < k_2 \\ 1, & \text{se } f(o_i) \geq k_2 \end{cases} \tag{1}$$

4 Proposed Method

This section presents the implementation of the selective aggregation model using the Khepera III robot and the CoppeliaSim simulation environment, which enables the creation of complex virtual scenarios and provides fine-grained control over the behavior of mobile robots.

Algorithm 1 Algorithm for a Single Object Type in Robot$_i$

```
Initialize the arena; State_i ← unloaded
for t = 0 to t_max do
    Navigate while checking the robot's surroundings; Read infrared (IR) sensors
    if Robot detects an obstacle then
        Stop the robot; Capture image using Vision Sensor and identify the type
        if Obstacle is a robot then
            Avoid Obstacle
        else
            Count objects in front, to the right and left
            if State_i = unloaded then
                Compute f and p_p using Equation 1; R ← random real number in [0, 1]
                if R ≤ p_p then
                    Pickup object by robot_i; State_i ← loaded
                else
                    Avoid Obstacle
            else
                Compute f and p_d using Equation 2; R ← random real number in [0, 1]
                if R ≤ p_d then
                    Drop object by robot_i; State_i ← unloaded
                else
                    Avoid obstacle
```

The Khepera III is a compact mobile robot widely used in research on autonomous and swarm robotics. Equipped with differential wheels, infrared sensors, ultrasonic sensors, and expansion modules, it offers great versatility in both simulations and real-world applications. It is capable of performing object manipulation tasks, such as picking up, transporting, and dropping items in controlled environments, thanks to an attached gripper. The gripper is driven by dedicated motors and is sensitive enough to grasp small objects, allowing for the simulation of bio-inspired behaviors with high fidelity. In this study, all objects used are cylindrical in shape, selected for their geometric uniformity and ease of manipulation by the robot's gripper. Although the work discusses the impact of object diversity on system performance, such diversity refers specifically to visual and categorical differentiation via object color, not geometric variation. The use of uniformly shaped objects aimed to minimize physical variability that could interfere with the analysis of type-based categorization, enabling a more precise evaluation of selective clustering behavior based solely on visual attributes.

The developed algorithm1 is an adaptation of the Lumer and Faieta's algorithm, designed to simulate the process of selective clustering of objects of the same type. This process is carried out by autonomous robots that navigate the arena, detect objects, make probabilistic decisions about picking up or dropping items, and dynamically interact with the environment.

Algorithm 2 simulates the process of selective aggregation in an environment containing objects of diferent types. During execution, the robot explores the arena, detects objects, determines the type of each object, and calculates the probabilities associated with pickup (p_p) or drop-off (p_d), based on Equation 1. This strategy enables the robot to pick up and group objects of the same type it is currently carrying, promoting the formation of homogeneous clusters.

Algorithm 2 Algorithm for Multiple Object Types in Robot$_i$

```
Initialize the arena; State_i ← unloaded
for t = 0 to t_max do
    Navigate while checking the robot's surroundings; Read infrared (IR) sensors
    if Robot detects an obstacle then
        Stop the robot; Capture image using Vision Sensor and identify the type
        if Obstacle is a robot then
            Avoid Obstacle
        else
            IdentifiedType ← Color_{o_i}; Count objects in front, to the right and left that are the same color as o_i
            if State_i = unloaded then
                Compute f(o_i) and p_p(o_i) using Equation 3; R ← random real number in [0, 1]
                if R ≤ p_p(o_i) then
                    Pickup object by robot_i; CarriedType_i ← detectedType; State_i ← loaded
                else
                    Avoid Obstacle
            else
                Compute f(o_i) and p_d(o_i) using Equation 3; R ← random real number in [0, 1]
                if R ≤ p_d(o_i) and IdentifiedColor = CarriedType_i then
                    Drop object by robot_i; State_i ← unloaded
                else
                    Avoid Obstacle
```

Algorithm 3 details the operations performed by the robot to pick up an object found in the arena. Initially, the robot positions itself correctly in relation to the detected object and identifies its type, which will be useful for future clustering decisions. The gripper arm is lowered, the gripper is closed to grasp the item, and then the arm is lifted to ensure the object is transported safely. Finally, the robot's state is updated to reflect that it is now loaded.

Algorithm 4 describes the procedure adopted by the robot to deposit the carried object. Initially, the robot positions itself at the drop-off location, lowers the gripper arm, and opens it to release the object in a controlled manner. Then, the robot slightly moves backward to avoid interfering with the placed object, raises the arm again, and updates its internal state to indicate it is now unloade, i.e. no longer carrying an object.

Algorithm 3 Pick up object by robot$_i$

```
Position the robot relative to the object
Move the gripper arm down
Close the gripper
Move the gripper arm up
```

Algorithm 4 Drop object by robot$_i$

```
Position the robot relative to the drop-off location
Lower the gripper arm; Open the gripper
Move the robot slightly backward
Raise the gripper arm
```

5 Results

This section presents the results obtained from the simulation of the proposed algorithm for the selective aggregation of objects using a swarm of robots. The experiments were conducted in the CoppeliaSim 4.1 virtual environment, employing Khepera III robots, each equipped with proximity sensors, a vision camera, and a gripper for object manipulation. The simulations aimed to evaluate the formation of homogeneous clusters, execution time, and the degree of selectivity achieved during the aggregation process. The simulations for the case studies can be accessed at: `https://linktr.ee/agregacao`[1]. The behavior observed, in which robots frequently pick up and drop items in nearby locations, is an expected outcome of the probabilistic nature of the Lumer and Faieta

[1]The source codes will be made available upon request for reproducibility and comparison purposes.

inspired algorithm. Since each robot operates based on local perception and lacks global knowledge, pick-up and drop-off decisions are driven by the local density of similar objects. Although this behavior may appear redundant, it is essential for the emergence of global organization through local interactions and aligns with self-organizing principles observed in natural systems.

5.1 Impact of the Number of Objects

The first case study aimed to investigate the impact of the number of objects on the performance of the selective aggregation system. For this purpose, a single robot operated in an arena containing only one type of object. Simulations were conducted with four different object quantities: 10, 15, 20, and 25.

The results are presented in Table 1, which shows the average performance metrics of the system under different configurations. In the table, N_O represents the number of objects, N_C indicates the number of clusters formed at the end of the simulation, T_A corresponds to the total number of pick-up and drop actions performed by the robot, and T_C refers to the total number of object detections, Time indicates the total simulation time in seconds. Finally, σ_{T_A}, σ_{T_C}, and σ_{Time} denote the standard deviation associated with each respective metric, reflecting the variability observed across multiple simulation runs.

Table 1. Results of Case Study 1: Impact of the number of objects

N_O	N_C	T_A	σ_{T_A}	T_C	σ_{T_C}	Time (s)	σ_{Time} (s)
10	2	16.2	2.5	47.0	5.1	8780.2	1384.17
15	3	30.0	16.5	81.2	47.3	14238.2	9854.48
20	4	23.4	2.1	90.0	20.8	13604.2	4371.05
25	5	37.6	9.8	148.6	50.9	19388.0	8347.84

It can be observed that as the number of objects increases, there is a rise in the average values of object detections and execution time. This is due to the larger number of interactions required by the agent to reorganize the dispersed objects in the environment. On the other hand, the average number of clusters formed also tends to increase, indicating that with more objects available, the robot tends to form additional groups throughout the simulation. Furthermore, even with only one robot and no explicit communication, the system demonstrated the ability to perform selective aggregation efficiently. The results indicate that the robot's emergent behavior, driven solely by local perceptions and probabilistic decisions based on the density of similar items, is sufficient to promote coherent item organization.

5.2 Impact of Object Type Diversity

The second case study analyzed the impact of object type diversity on the system's performance. A single robot operated in an arena containing objects of three distinct types (red, blue, and green), with total quantities ranging from 10 to 25, evenly distributed whenever possible. Unlike Case Study 1, this scenario required type discrimination, adding complexity to the decision-making process.

Table 2. Results regarding the impact of the object type diversity

N_O	N_C	T_A	σ_{T_A}	T_C	σ_{T_C}	Time (s)	σ_{Time} (s)
10	3	11.0	2.9	66.0	23.1	10227.0	4734.8
15	3	20.5	5.2	170.5	70.9	22258.5	9225.1
20	3	27.6	12.5	215.0	75.0	24343.3	10846.3
25	5	35.3	7.2	300.3	79.4	29710.3	7261.1

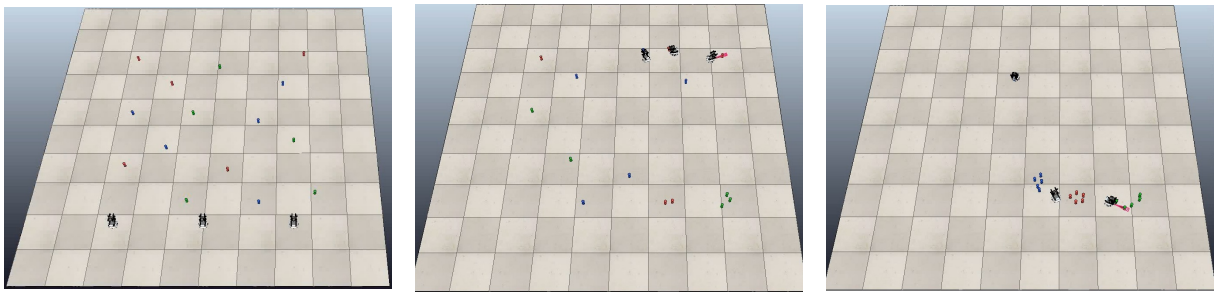
The results, summarized in Table 2, show the average values across multiple runs. The columns present: the number of objects (N_O), number of clusters formed (N_C), total actions of pick-up/drop (T_A) and its standard

deviation (σ_{T_A}), number of object detections (T_C) and its standard deviation (σ_{T_C}), as well as the execution time in seconds (Time) and its respective standard deviation (σ_{Time}).

The data indicate that selective aggregation was achieved even with object diversity. However, as the number of items increased, there was a notable rise in detections and execution time, reflecting the added complexity of identifying and handling multiple object types. The number of clusters also increased slightly in the 25-object configuration, suggesting a tendency to form more distinct groups for each category.

5.3 Impact of the Number of Robots in the Swarm

The third case study evaluated the impact of varying the number of robots on the performance of the selective aggregation system. An arena containing 15 objects divided into three types, randomly distributed, was used. Simulations were performed with 1, 2, and 3 robots, while keeping perception, decision-making, and behavior parameters constant. Figure 1 illustrates an example of the simulation conducted with three robots.



(a) (b) (c)

Figure 1. Snapshots of the arena during a simulation with three robots: (a) Initial arena. (b) Intermediate stage during the aggregation process. (c) Final arena configuration.

The results presented in the top part of Table 3 show the average performance values of the selective aggregation system in different scenarios with varying numbers of robots. The table uses the following abbreviations: N_R represents the number of robots used; N_C indicates the number of clusters formed; T_{A1}, T_{A2}, and T_{A3} correspond to the number of pick-up and drop-off actions performed by robots 1, 2, and 3, respectively; T_{C1}, T_{C2}, and T_{C3} indicate the number of object detections made by each robot; and Time refers to the total execution time of the simulation, measured in seconds. These results enable a comparative analysis of the system's performance as the number of agents increases.

The distinguishing aspect of this experiment was the analysis of workload distribution among the robots, measured by the number of pick/drop actions and detections performed by each agent. The goal was to investigate service balance, an important feature in decentralized swarm robotic systems. The results show a significant reduction in execution time as the number of robots increases, indicating that the natural parallelism of the system enhances the efficiency of the aggregation process. Moreover, the distribution of pick/drop actions and detections became more balanced with three robots, with all agents contributing similarly to the task.

Table 3. Results regarding the impact of the number of robots

N_R	N_C	T_{A1}	T_{C1}	T_{A2}	T_{C2}	T_{A3}	T_{C3}	Time (s)
1	3	20.5	170.5	-	-	-	-	22258.5
2	3	6.3	57.0	5.7	72.3	-	-	6447.0
3	3	4.2	35.6	4.0	35.4	4.2	28.8	3881.4
N_R	N_C	$\sigma_{T_{A1}}$	$\sigma_{T_{C1}}$	$\sigma_{T_{A2}}$	$\sigma_{T_{C2}}$	$\sigma_{T_{A3}}$	$\sigma_{T_{C3}}$	σ_{Time} (s)
1	3	5.2	70.9	-	-	-	-	9225.1
2	3	1.5	14.7	1.2	24.8	-	-	1636.6
3	3	1.1	8.8	1.0	9.6	1.3	6.3	1083.4

The bottom part of Table 3 the standard deviations corresponding to the results of Case Study 3, reflecting the variability across simulation runs for each robot configuration. Columns N_R and N_C indicate the number of robots and clusters formed. The values $\sigma_{T_{Ai}}$ and $\sigma_{T_{Ci}}$ represent the standard deviations of actions and detections performed by robots $i = 1$ to 3, and σ_{Time} corresponds to the standard deviation of the total execution time. The reduction in variability as the number of robots increases suggests more stable and consistent swarm behavior.

6 Conclusions

This work presented an adaptation of the Lumer and Faieta's algorithm for distributed control od a set of multiple robots, focusing on the selective aggregation of objects by type. The experiments conducted in a simulated environment using Khepera III robots demonstrated the effectiveness of the proposed approach, even in scenarios with limited perception and no inter-agent communication.

From the three case studies developed, it was possible to analyze the impact of different factors on the system's performance. In the first study, it was observed that increasing the number of objects led to longer execution times and more interactions, without compromising the formation of selective clusters. The second study, which introduced object type diversity, showed that the algorithm is robust enough to handle multiple categories, although this increases the agent's perceptual complexity and the time required. The third study highlighted that increasing the number of robots in the swarm results in a significant reduction in execution time and promotes a more balanced distribution of tasks among agents, improving collective efficiency. The results confirm the feasibility of the proposed approach based on local perception and probabilistic decisions, reinforcing the potential of swarm robotics for applications in dynamic and unstructured environments. Furthermore, the experiments showed that behaviors inspired by natural systems, even under perception and control constraints, can lead to effective and scalable emergent solutions.

In future work, we propose to expand the evaluation beyond obstacle-based scenarios by validating the proposed approach with real Khepera III robots. We intend to consider challenges that are absent in simulation, such as sensor noise, physical misalignments during manipulation, communication delays, and environmental variability, such as lighting conditions and surface irregularities. Additionally, we plan to carry out comparative studies with similar approaches in the literature, aiming to contextualize the performance and limitations of our method under both simulated and real-world conditions. This validation will allow us to assess the robustness, adaptability, and scalability of the system under non-ideal and unpredictable circumstances.

References

[1] E. D. Lumer and B. Faieta. Diversity and Adaptation in Populations of Clustering Ants. In *August 8-12, 1994, Brighton, England. From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adpative Behavior*. The MIT Press, 1994.

[2] H. C. Abdelouahab Moussaoui. Messaoud Hameurlaine. Antmeans: A new hybrid algorithm based on ant colonies for complex data mining. *International Journal of Computer Applications*, vol. 60, n. 17, pp. 6–12, 2012.

[3] S. Salima and M. Batouche. An efficient ant algorithm for swarm-based image clustering. *Journal of Computer Science*, vol. 3, pp. 162–167, 2007.

[4] S. C. Tan, K. M. Ting, and S. W. Teng. Simplifying and improving ant-based clustering. *Procedia Computer Science*, vol. 4, pp. 46–55. Proceedings of the International Conference on Computational Science, ICCS, 2011.

[5] J. Chavarría-Molina, J. J. Fallas-Monge, and J. Trejos-Zelaya. *Clustering via Ant Colonies: Parameter Analysis and Improvement of the Algorithm*. Springer Singapore, Singapore, 2020.

[6] A. Krischanski, Y. K. Lopes, A. B. Leal, R. F. Martins, and R. S. Rosso. Multi-instance task in swarm robotics: Sorting groups of robots or objects into clusters with minimalist controllers. pp. 8827–8832, Detroit, MI, USA. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023.

[7] O. Misir and L. GÖkrem. Dynamic interactive self organizing aggregation method in swarm robots. *Biosystems*, vol. 207, pp. 104451, 2021.

[8] Z. Firat, E. Ferrante, N. Cambier, and E. Tuci. Self-organised aggregation in swarms of robots with informed robots. In D. Fagan, C. Martín-Vide, M. O'Neill, and M. A. Vega-Rodríguez, eds, *Theory and Practice of Natural Computing*, pp. 49–60, Cham. Springer International Publishing, 2018.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.