Aggregation through adaptive random walks in a minimalist robot swarm

Luigi Feola*
feola@diag.uniroma1.it
DIAG, Sapienza University of Rome
ISTC, National Research Council
Rome, Italy

Antoine Sion*
antoine.sion@unamur.be
Université de Namur
Namur, Belgium

Andreagiovanni Reina
andreagiovanni.reina@gmail.com
IRIDIA, Université Libre de Bruxelles
Bruxelles, Belgium

Vito Trianni
vito.trianni@istc.cnr.it
ISTC, National Research Council
Rome, Italy

Elio Tuci
elio.tuci@unamur.be
Université de Namur
Namur, Belgium

ABSTRACT

In swarm robotics, random walks have proven to be efficient behaviours to explore unknown environments. By adapting the parameters of the random walk to environmental and social contingencies, it is possible to obtain interesting collective behaviours. In this paper, we introduce two novel aggregation behaviours based on different parameterisations of random walks tuned through numerical optimisation. Cue-based aggregation allows the swarm to reach the centre of an arena relying only on local discrete sampling, but does not guarantee the formation of a dense cluster. Neighbour-based aggregation instead allows the swarm to cluster in a single location based on the local detection of neighbours, but ignores the environmental cue. We then investigate a heterogeneous swarm made up of the two robot types. Results show that a trade-off can be found in terms of robot proportions to achieve cue-based aggregation while keeping the majority of the swarm in a single dense cluster.

CCS CONCEPTS

• **Computing methodologies → Multi-agent systems**; *Mobile agents*; Cooperation and coordination.

KEYWORDS

Random walks, aggregation, heterogeneous swarm, swarm robotics, minimal computing, iterated racing

ACM Reference Format:
Luigi Feola, Antoine Sion, Andreagiovanni Reina, Vito Trianni, and Elio Tuci. 2023. Aggregation through adaptive random walks in a minimalist robot swarm. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583131.3590485

*Both authors contributed equally to this research.

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590485

1 INTRODUCTION

Aggregation is a recurring collective behaviour observed in many biological systems [2] such as young honeybees clustering near the optimum inside a temperature field [29], or cockroaches forming dense groups [15]. It is a fundamental building block on which more complex forms of cooperation between individuals can emerge thanks to its grouping effect. Aggregation can occur based on cues found in the environment like humidity gradients or shelters (cue-based aggregation), but it can also appear purely from the interactions between the individuals in the absence of such cues (neighbour-based aggregation). These collective behaviours have been implemented and studied in swarm robotics [30], making agents able to gather information through local communication and local sensing, hence resulting in a heterogeneous spatial distribution of the swarm.

Previous work on neighbour-based and cue-based aggregation in swarm robotics focused on simple mechanisms to form large clusters, but the proposed solutions do not always lend themselves to minimalist swarms (e.g., swarms formed by micro and nano-robots with limited computational abilities). In [6], aggregation has been implemented by exploiting long-range signals among agents, and the robots' ability to navigate towards areas of high signal intensity. However, these abilities may be not trivial to obtain at very small scales. In [12], aggregation mechanisms have been designed using probabilistic behaviours, based on the modulation of the time agents spend in a cluster as a function of the cluster size. In some studies, the emergence of clusters of agents are facilitated by the presence of specific areas for aggregation such as shelters or other types of environmental cues [1, 11, 14, 27]. Although probabilistic behaviours are relatively simple to design, some of them require some learning to fine tune the parameters that underpin the robots adaptability to local contingencies. These learning capabilities may be difficult to be implemented in minimalist robots. Moreover, even in systems that support learning, some probabilistic behaviours have already shown limitations in coping with dynamic environments [3].

Minimalist robots at very small scales often lack the ability to precisely sense the environment. They show limited accuracy in movement and in reaching specific locations. Their behaviour cannot be finely modulated. In spite of these limitations, interesting collective behaviours can still be attained by regulating the individual responses to easily perceivable physical and social cues, such as changes of the diffusion properties [4, 10, 17]. Inspired by these approaches, we study to what extent it is possible to achieve aggregation in a minimalist robot swarm solely as the result of an adaptive random walk, whereby the robots change the motion pattern according to environmental or social cues.

Random walks are basic behaviours, extensively used in swarm robotics [5], since they require very little complexity to be implemented. They allow the exploration of unknown environments without requiring complex functions such as localisation or mapping. Most studies using random walks in swarm robotics are about search and coverage problems [7, 22], where the swarm is either required to find a point of interest in the environment or explore as much as possible a specific area. Duncan et al. [7] compare Brownian and Lévy walks through a continuum model and robotic simulations. Their results show that the long displacements made with the Lévy walks induce a faster exploration of the environment by the swarm. Lévy walks turned out to be particularly suitable also in a target detection problem, where robots must first locate the target and then communicate its position through intermediate relay robots to a base station [18, 19]. In [5], the authors investigated the efficiency of different parameterisations of the random walks in searching tasks in bounded and unbounded environments using both simulated and physical robots. The study shows that the random walk efficiency depends on the type of environment, with a correlated random walk being the best strategy in the bounded environment and the Lévy walk in the unbounded environment. Random walks have also been used for indirectly mapping the environment, with robots exploring and creating a map based on their local experience and sharing their maps offline at the end of the process [20].

The goal of this study is to generate aggregation dynamics through minimalist behaviours leveraging different random walk patterns. Our robots are programmed to dynamically transition between an exploratory behaviour associated with a persistent random walk to a local search in response to their perceptual states. This adaptive walk has been used in swarms to underpin division of labour, biasing the distribution of robots towards areas in which more work is required [9]. Following a similar approach, we study aggregation behaviours in three different conditions. First, we study cue-based aggregation where cues are provided by an environmental feature distributed as a continuous quantity covering the entire environment, with the minimum quantity in the centre of the arena. Since we are considering minimalist robots, we evaluate the ability of the swarm to aggregate in the range where the environmental feature reaches its minimum using a low-resolution sensor. Second, we study neighbour-based aggregation, i.e., an aggregation behaviour triggered by the perception of spatially proximal robots. This neighbour-based aggregation aims at the formation of robot clusters of progressively higher density, possibly reaching a single cluster including all the robots of the swarm. Finally, we merge the two types of aggregation in a heterogeneous swarm, in which the proportion of robots implementing the cue-based aggregation and the neighbour-based aggregation are systematically varied to study which ratio of the two behaviours results in the formation of the largest possible single cluster located in the target area (i.e., where the environmental cue reaches its minimal quantity). Both aggregation behaviours are fine-tuned through a series of parameters which bear upon the robot motion quality. The parameters' space is explored with an iterative racing algorithm [21] that enables us to optimise the performance metrics for cue-based and neighbour-based aggregation. Recently, a similar research work studied a swarm of robots required to follow a gradient based on a local continuous-value environmental feature. These robots can use knowledge about distance and the bearing of spatially-proximal neighbours to develop navigation strategies [31]. Differently from this and other previous studies, we focus on more minimalist robots, as our agents can only rely on one type of sensor—either to perceive the environmental cue or to detect the current number of neighbours. Furthermore, the innovative contribution of this paper lies in the evaluation of a heterogeneous swarm in which robots are equipped with different controllers and sensing abilities, enabling to obtain cue-based and neighbour-based aggregation at the same time and with minimal complexity.

The paper is organised as follows. In Section 2, we discuss the experimental setup of our experiments and we describe the implemented behaviours with a particular focus on the metrics employed to tune the different controllers. In Section 3, we present the controllers' tuned parameters for both behaviours and the swarm's performances for the cue-based aggregation, the neighbour-based aggregation and the heterogeneous swarm. In Section 4, we discuss our results and possible future work.

2 EXPERIMENTAL SETUP

We study aggregation through adaptive random walks in a 1 m×1 m squared bounded arena. We consider two different simulators: a simple simulator developed to speed up the optimisation process, and the ARGoS simulator [24] to validate the optimised solutions in an experimental setup closer to the physical robotic system.

Our swarm is composed of $S = 25$ simulated Kilobots [26]. These are small low cost robots that can move and locally exchange messages. The Kilobots are widely used by the swarm robotics community for testing collective behaviours. In the abstract simulator, they are approximated as circles, modelled as differential drive robots with just three possible actions: move straight, rotate left, rotate right, with robot rotation that takes place instantly on the spot. Each robot has a speed of 1 cm/s and a communication radius of 10 cm. To lower computation costs, we do not consider collisions among robots. Collisions with walls are managed by emulating the behaviour of a bouncing sphere on a flat surface.

In the ARGoS simulator [24], we use the Kilobot plugin [23] to model the robots. We employ the simulated version of the ARK system [25] to equip the Kilobots with a virtual sensor able to sample the environmental cue. The robots are also equipped with virtual wall proximity sensors allowing them to avoid the walls of the arena with a collision avoidance routine, similar to the one adopted in the abstract simulator. However, collision avoidance between the robots is not implemented, they are thus free to collide with each other. Kilobots are not designed to rotate on the spot, but pivot around the left or right legs at $\frac{\pi}{4}$ rad/s. To emulate the rotation on the spot featured in the abstract simulator, for each Kilobot we randomly define a fixed pivot rotation (i.e., clockwise or anticlockwise). For example, a clockwise rotation of an angle β will result in an anticlockwise rotation of $2\pi - \beta$ if the Kilobot was given a fixed anticlockwise pivot.

The adaptive random walk behaviours performed by the robots are based on a finite state machine (FSM) where each state corresponds to a different parameterisation of the random walk. State transitions are triggered by perceptual states, effectively modifying the parameters of the random walk. In any FSM state, robots perform a Lévy-modulated correlated random walk (LMCRW) [5], which is characterised by two parameters controlling the distribution of turning angles and step lengths. Specifically, the turning angle is drawn from a wrapped Cauchy distribution with the following probability density function:

$$f(\theta, \rho) = \frac{1}{2\pi} \frac{1-\rho^2}{1+\rho^2+2\rho\cos(\theta)} \tag{1}$$

where ρ determines the distribution width: for $\rho = 0$ the distribution becomes uniform and provides no correlation between consecutive movements, while for $\rho = 1$ a Dirac distribution is obtained, corresponding to ballistic motion. The step length δ follows a Lévy distribution characterised by a power law $P(\delta) \approx \delta^{-(\alpha+1)}$, with $0 < \alpha \leq 2$. For $\alpha = 2$ the distribution becomes Gaussian, while for $\alpha \to 0$ the random walk reduces to ballistic motion. Each FSM state i corresponds to specific values of the random walk parameters pair $\langle \alpha_i, \rho_i \rangle$. Each FSM controller is also characterised by a parameter σ influencing the scale of the step length distribution of the random walk, which is shared by all states of the controller.

2.1 Cue-based aggregation

In this study, we tested the ability of agents to aggregate autonomously in response to an environmental cue. The simulated environment contains a continuous scalar cue with its minimum in the centre of the arena as shown in Figure 1a. The cue increases linearly from the centre (where the scalar value is 0) up to the borders of the arena (where the scalar value is 1). The goal of the robots is to aggregate where the environmental cue has the lowest intensity. The aggregation behaviour has to be performed in spite of the robot limited sensing abilities. That is, the robots can only perceive the local value of the cue and do not perceive the gradient direction. Additionally, we consider a very low resolution sensor with $n_d \in \{2, 3, 4\}$ quantisation levels. As a consequence, the environment is perceived as formed by n_d uniform areas delimited by $n_d - 1$ concentric circles (referred to as area A_i for the corresponding level i, see Figures 1b–d). Finally, robots cannot sample the environment at high rate, but are limited to get a new sample only when they change motion direction (i.e., simulating a blind relocation).

We devise an adaptive random walk by associating a different parameterisation to each quantisation level. Specifically, when the robot has to define a new movement (rotation and straight displacement), it checks the quantisation level $i \in [1, n_d]$ and selects the corresponding parameterisation $\langle \alpha_i, \rho_i \rangle$. For instance, when $n_d = 2$, the robot perceives two uniform areas (A_1 and A_2, respectively represented in black and white in Figure 1b). As soon as a robot finds itself in the black area A_1 (characterised by the quantisation level 1) it switches to a random walk governed by the $\langle \alpha_1, \rho_1 \rangle$ parameter pair. Conversely, when the robot finds itself in the white area A_2 (characterised by the quantisation level 2), it changes its motion parameters to the $\langle \alpha_2, \rho_2 \rangle$ pair. Our objective is to find the set of n_d $\langle \alpha_i, \rho_i \rangle$ pairs (and the common σ value) so that robots aggregate as much as possible in areas with a low cue intensity. To this end, we tune the parameters of our controllers with the iterative racing tuning algorithm iRace [21]. We minimise a metric based on the cue values corresponding to the robot positions recorded during the whole experiment, which corresponds to maximising the aggregation of the swarm towards the centre of the arena in the minimum feasible time. The swarm performance is computed at the end of a trial with the following equation:

$$G = \frac{\sum_{t=0}^{T} \sum_{s=0}^{S} g_{s,t}}{S \cdot T} \tag{2}$$

where $G \in [0, 1]$ is the sum of the overall ground-truth value of the environmental cue $g_{s,t}$ normalised by the number of robots S and the experiment time T (in time steps).

2.2 Neighbour-based aggregation

Similarly to the cue-based aggregation behaviour, in the neighbour-based aggregation the robots switch between different types of random walks at the end of each forward movement. Differently from the cue-based, the neighbour-based aggregation is triggered by the perception of spatially proximal group-mates, which is possible thanks to short-range communication between robots [26]. In particular, the robots dynamically change the parameters of the random walk in response to the perceived number of neighbours.

We tested two different FSM controllers for this behaviour. The first one has three different possible random walk states and two thresholds (t_i) to switch between them (Figure 3a). The rationale for this FSM is the following: a robot adopts an exploratory behaviour

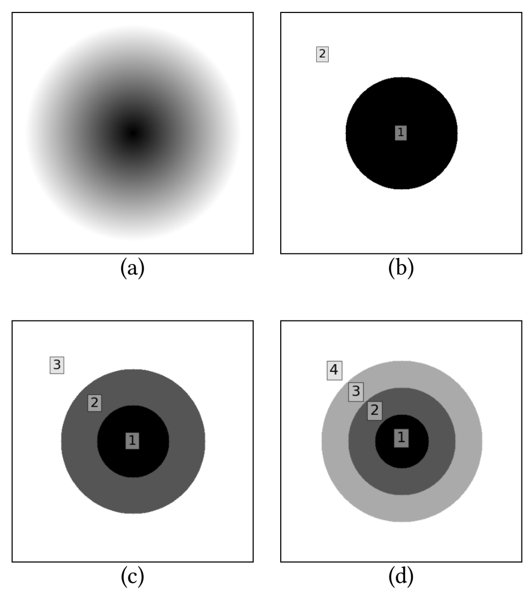


Figure 1: (a) The environmental cue forms a continuous field that increases radially from the arena centre. (b–d) Homogeneous areas corresponding to the $n_d \in \{2, 3, 4\}$ quantisation levels. The quantisation levels are numbered in ascending order from the innermost to the outermost area.

in search of higher densities of neighbours whenever their number is relatively low. Seemingly, a robot gradually adopts a more static behaviour whenever the number of its neighbours rises. The second controller has only two different states and two thresholds to switch between them (Figure 3b). Ideally, a robot transitions to a more static behaviour once the number of neighbours exceeds a threshold t_0, and remains in this state unless the number of neighbours decreases under a smaller threshold t_1.

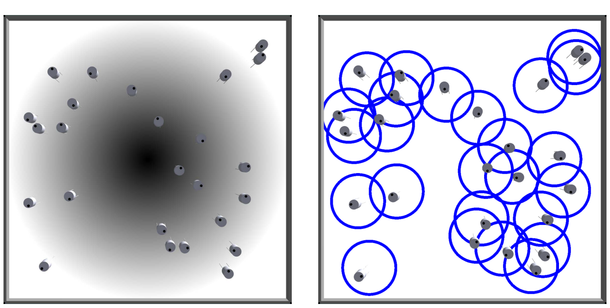
As for the cue-based aggregation, also in the neighbour-based aggregation the controller parameters are tuned with iRace [21]. To obtain an aggregation process such that a single cluster is formed by the majority of the swarm at the end of the experiment, we maximise the cluster metric from [13] computed from the robots' positions at the end of a trial. This metric is based on the graph defined by the robots' positions, in which two robots are connected if they are within their communication range. A cluster is defined as a connected subgraph with a dimension larger than one (i.e., at least two robots are connected). Thus, the cluster metric can be defined as the size of the biggest cluster (S_M) divided by the swarm size (S):

$$C = \frac{S_M}{S} \tag{3}$$

When the swarm is aggregated in a single cluster, $C = 1$. When no cluster is formed and all robots are isolated, $C = 0$.

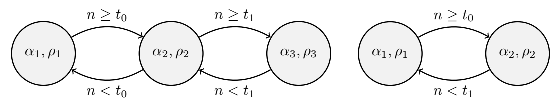
2.3 Heterogeneous swarm

We also study a heterogeneous swarm composed of robots with the two previously described behaviours. We use the same experimental setup detailed before for the ARGoS simulator, and the size of the swarm remains fixed at $S = 25$ robots. We systematically vary the number $S_s = SH$, with H being the percentage of robots that sense the number of neighbours. As a consequence, $S_c = S - S_s$ corresponds to the number of robots that sense the environmental cue. Our objective is to find the best proportion of the two types of robots to have a swarm that forms a single aggregate on the area in which the environmental cue has its minimum value. In order to allow the robots that sense the cue to signal their presence to spatially proximal robots that sense neighbours, the former robots, when included in the heterogeneous swarm, have been provided with the additional capability to broadcast messages signalling their presence.



(a) (b)

Figure 2: Representation of the two experimental setups in ARGoS. The setup used for the cue-based aggregation is represented in (a) and the setup with an empty arena for the neighbour-based aggregation is represented in (b).



(a) (b)

Figure 3: First (a) and second (b) FSM of the neighbour-based aggregation behaviour. α_i and ρ_i are the random walk parameters for each state, t_i are the thresholds used in the transitions and n is the number of neighbours sensed by the robot at the end of a step.

Table 1: Parameters bounds for the iRace algorithm

Parameters	Type	Bounds
α_i	Real	[1.0, 2.0]
ρ_i	Real	[0.0, 1.0[
σ	Integer	[1, 20]
t_i	Integer	[1, 10]

3 RESULTS

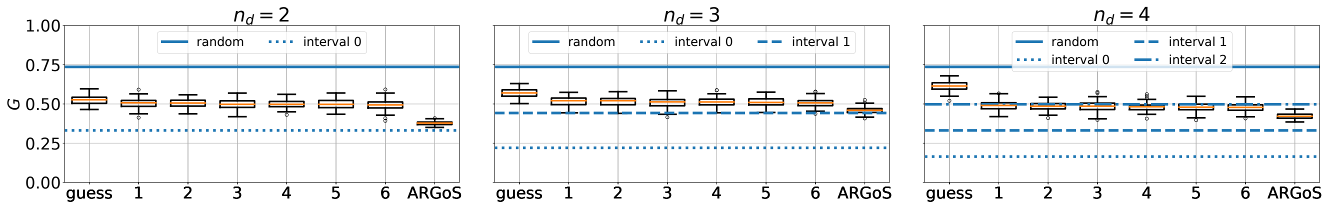
3.1 Cue-based aggregation

We tune the FSM controller parameters for each quantisation level $n_d \in \{2, 3, 4\}$. For each controller, we have a total of $2n_d + 1$ parameters, bounded as detailed in Table 1, with a precision of 2 decimal digits for ρ and α. Six independent runs of iRace are performed for each n_d level, to evaluate the robustness of the optimisation method. Each run has a budget of 10^5 evaluations and a total evaluation time of $T = 10^4$ time steps. The large evaluation budget derives from the observation during preliminary tests that performance is rather noisy, and many evaluations are required to ensure that iRace finds differences among the evaluated solutions.

The performances of the best obtained parameters configurations on 100 trials are shown in Figure 4. As a baseline, we consider an initial guess with the $\langle\alpha_i,\rho_i\rangle$ pairs linearly varying from a persistent walk ($\alpha_{n_d} = 1.0$, $\rho_{n_d} = 0.99$) when the robot is in the outermost area A_{n_d} to a Brownian motion ($\alpha_1 = 2.0$, $\rho_1 = 0.0$) when the robot finds itself in area A_1, and σ set to 1. The parameters configuration with the lowest median is also tested in the ARGoS simulator over 100 trials with an evaluation time of $T = 5000$ seconds for each controller (see Table 2).

Table 2: Parameters corresponding to the run with the lowest median G metric.

n_d	$\langle\alpha_1,\rho_1\rangle$	$\langle\alpha_2,\rho_2\rangle$	$\langle\alpha_3,\rho_3\rangle$	$\langle\alpha_4,\rho_4\rangle$	σ
2	⟨1.01, 0.98⟩	⟨1.99, 0.0⟩			1
3	⟨1.0, 0.96⟩	⟨1.95, 0.02⟩	⟨1.96, 0.29⟩		1
4	⟨1.01, 0.96⟩	⟨1.02, 0.05⟩	⟨1.95, 0.0⟩	⟨1.99, 0.31⟩	1



(a) (b) (c)

Figure 4: Box plots of the cue-based metric at the end of 100 trials for different n_d: (a) $n_d = 2$; (b) $n_d = 3$; (c) $n_d = 4$. The box plot labelled *guess* shows the performance of the swarm considering the controller with parameters set from the initial guess. The box plots labelled [1,6] represent the performance of the controller corresponding to the six best configurations obtained using iRace. The last box plot shows the performances of the lowest median controller among the previous six validated with the ARGoS simulator. The blue solid line represents the mean G value expected when robots are randomly distributed in the environment, and represents an upper bound for a blind random walker. The other blue lines represents the mean G value corresponding to the expected performance when robots are in area A_1 (dotted line), A_1 or A_2 (dashed line) and A_1 or A_2 or A_3 (dash-dotted line).

With $n_d = 2$, the performance of the controllers tuned by iRace is similar to the initial guess, corresponding to a behaviour in which robots perform a persistent walk in order to find the low-intensity area; once in that area, robots switch to a Brownian motion. With $n_d = 3$, the tuned controllers perform slightly better than the initial guess. In this case, robots perform a persistent walk near the borders of the arena (area A_3, see Figure 1c) and a Brownian motion in the centre. The parameters tuned for the intermediate area A_2 are also very close to a Brownian motion, indicating that the best controller does not limit local movements in the centre, but also in a wider area. Similarly, the best controller with $n_d = 4$ presents a Lévy walk in the outer part of the arena (areas A_3 and A_4, see Figure 1d), while a Brownian motion in the inner part (areas A_1 and A_2). Hence, for both $n_d = 3$ and $n_d = 4$, Brownian motion is not limited to the smaller area A_1. This corresponds to a performance not better than with $n_d = 2$ (see Figure 4). Indeed, having a higher sensor resolution does not lead to higher performance. This is because the likelihood of leaving the inner area increases when the radius gets smaller. We estimated the residence time in each area recording the time intervals a robot spends in each area, and using a Kaplan-Meier estimator [16] to obtain the cumulative distribution of the residence times (data not shown). Our results indicate that, with $n_d = 2$, the average time required to enter the area A_1 is smaller than the exit time, hence supporting aggregation. Conversely, with $n_d > 2$ the average time required to enter A_1 is higher than the exit time, suggesting that it is harder to remain in the centre of the arena when the radius of A_1 is too small. Hence, the optimal strategy is to perform a Brownian walk in the wider region including area A_2.

When the best controllers are tested in the realistic simulations with ARGoS, the swarm displays better aggregation performance than in the abstract simulations (see Figure 4). This is due to the embodiment of the robots and the collisions that occur between them. Indeed, robots aggregating in the centre of the arena will tend to stay in that area due to other neighbours colliding with them. Moreover, rotation is not instantaneous in ARGoS, and such an additional time spent turning helps the robot to stay in place when it is performing Brownian motion, lowering the probability to leave the inner area. For $n_d = 2$, the aggregation behaviour in ARGoS is good enough to approach the performance bound corresponding to robots always residing in area A_1. For larger values of n_d, the best controller corresponds to robots always residing in area A_1 and A_2.

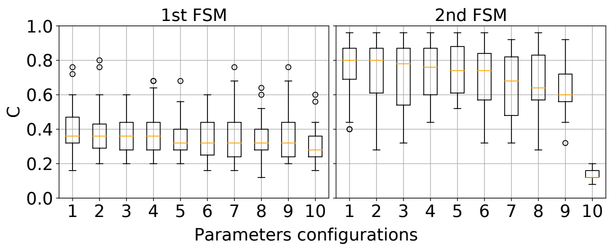
3.2 Neighbour-based aggregation

We tuned the two controllers presented in Section 2.2 for a swarm of S robots. Our first controller has a total of 9 parameters and the second one has only 7 (see Figure 3). For each one, we performed 10 runs of iRace with the parameters bounds found in Table 1 and a budget of 10^5 evaluations over a total evaluation time $T = 10^4$ time steps. Forbidden parameters configurations with the following conditions were excluded: $t0 \geq t1$ for the first FSM, $t0 < t1$ for the second FSM.

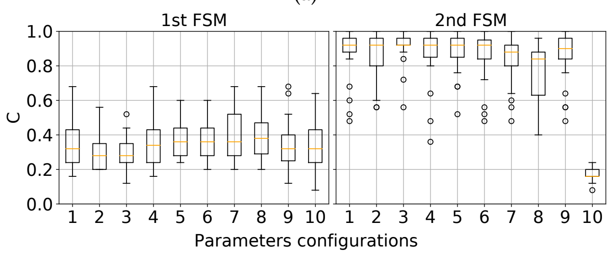
The cluster metrics of the best obtained configurations over 30 trials are shown in Figure 5a. Poor performance is obtained for the first FSM in all the configurations, indicating that the swarm struggles to form a single cluster. The second FSM has overall better aggregation performance with a cluster containing the majority of the robots in most cases.

We also tested these configurations in the ARGoS simulator with a total evaluation time $T = 3600$ seconds (Figure 5b). Similarly, the first FSM performs poorly overall. For the second FSM, the third configuration performed the best with a median around 0.9. Even for this type of aggregation, the swarm displays better performance in the ARGoS simulator compared to the abstract simulator. This is due to the embodiment of the robots, which has a stabilising effect on larger clusters. In the abstract simulator, since collisions between robots are not considered, clusters disband more easily and the performance degrades.

The parameters values for the best configuration of the second controller are the following: $t_0 = 4$, $t_1 = 1$, $\langle \alpha_1, \rho_1 \rangle = \langle 1.13, 0.99 \rangle$, $\langle \alpha_2, \rho_2 \rangle = \langle 1.93, 0.02 \rangle$, and $\sigma = 1$. With these parameters, a robot will first perform a Lévy walk in order to find other neighbours to aggregate. When it finds a group of at least $t_0 = 4$ other robots, it will transition to a Brownian motion. If, due to the constant motion of the robots, the neighbour count reduces below $t_1 = 1$ (that is, no



(a)



(b)

Figure 5: Box plots of the cluster metric C computed at the end of 30 trials for different parameters configurations obtained with (a) the abstract simulator and (b) the ARGoS simulator. The left panels correspond to the 10 best parameters configurations obtained with the first FSM, while the right panels correspond to the best parameters configurations obtained with the second FSM (see Figure 3).

neighbour is perceived), the robot will fall back to the initial Lévy walk. In this way, clusters will originate in a stochastic manner due to the close proximity of multiple robots and will grow in size to the detriment of smaller clusters, eventually reaching a single dense cluster containing the majority of the swarm. This is the controller that we have selected for further tests in heterogeneous swarms (see below).

The evolution of the cluster metric over time with the ARGoS simulator is shown on Figure 6 for this particular configuration. We obtain a reasonable convergence time of 1500 seconds and the cluster metric stabilises around 0.9 but never fully attains a perfect value of 1. This is due to the fact that robots never stop moving since they constantly perform a random walk, in opposition to other classic approaches in which robots do not move when aggregated [3, 28].

3.3 Heterogeneous swarm

In this section, we study a swarm made up of S_c robots adopting the cue-based aggregation behaviour and S_s robots adopting the neighbour-based aggregation. The rationale is that robots performing neighbour-based aggregation can form a dense cluster, while robots performing the cue-based aggregation can guide the group towards areas with low cue intensity. The goal is to observe the impact of the different proportions of robot types on the formation of a dense cluster near the minimum of the environmental cue. We tested multiple configurations of the swarm starting from a homogeneous swarm of neighbour-based robots ($S_c = 0$, $S_s = 25$) and increasing S_c until obtaining a swarm composed exclusively of robots performing the cue-based aggregation behaviour ($S_c = 25$, $S_s = 0$). All experiments are performed with the ARGoS simulator exploiting, for each aggregation behaviour, the configuration with the best median obtained from iRace.

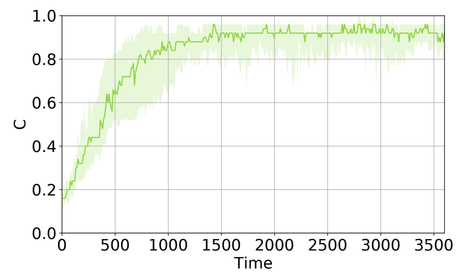


Figure 6: Evolution of the median of the cluster metric C with the best tuned controller over 3600 seconds and 30 trials in the ARGoS simulator.

Figure 7 shows the evolution over time of the cue-based metric G and the cluster metric C with varying n_d. As expected, for all the resolution levels, the worst performance in tracking the environmental cue is displayed by the swarm composed of only neighbour-based robots since no robot can sense the local value of the cue (top row in Figure 7). Despite this fact, the final value of the cue-based metric at the end of the experiment is a little bit lower than the initial one when all the robots are randomly distributed in the arena, due to the clustering of the robots. We can also observe that with as few as $S_c = 5$ robots performing cue-based aggregation, the G metric significantly decreases in all the plots. Increasing the number of the S_c robots increases the performances but the gain becomes gradually smaller.

The evolution of the cluster metric over time is presented in the bottom row of Figure 7. Clearly, the best performance is displayed by a swarm made up exclusively of neighbour-based robots ($S_s = 25$), with the cluster metric stabilising just over 0.9. By adding cue-based robots into the swarm, the cluster metric progressively decreases. However, it is possible to notice differences when varying the sensor resolution. With $n_d = 2$, all configurations of the swarm keep a good performance with the majority of the swarm forming a single dense cluster. With $n_d = 4$, the performance degrades but at least 60% of the swarm remains inside the biggest cluster. The worst performance is exhibited with $n_d = 3$, with only small clusters forming when $S_s < 15$. Indeed, in this case the cue-based aggregation is not very efficient, with robots scattered in a wide area (A_1 and A_2, see Figure 1c). In these conditions, the neighbour-based aggregation is not efficiently performed, and a single tight cluster seldom emerges.

We notice also that there is a trade-off between the two metrics considered: optimising one metric leads to the degradation of the other. Such trade-off is evident in the Pareto diagram shown in Figure 8. The ideal case to reach is the point located in the bottom right corner ($C = 1$, $G = 0$) representing a swarm forming a single cluster exactly in the centre of the arena. First of all, we can appreciate that the best trade-offs are reached when $n_d = 2$, which dominate the other configurations. We can also notice a trade-off between

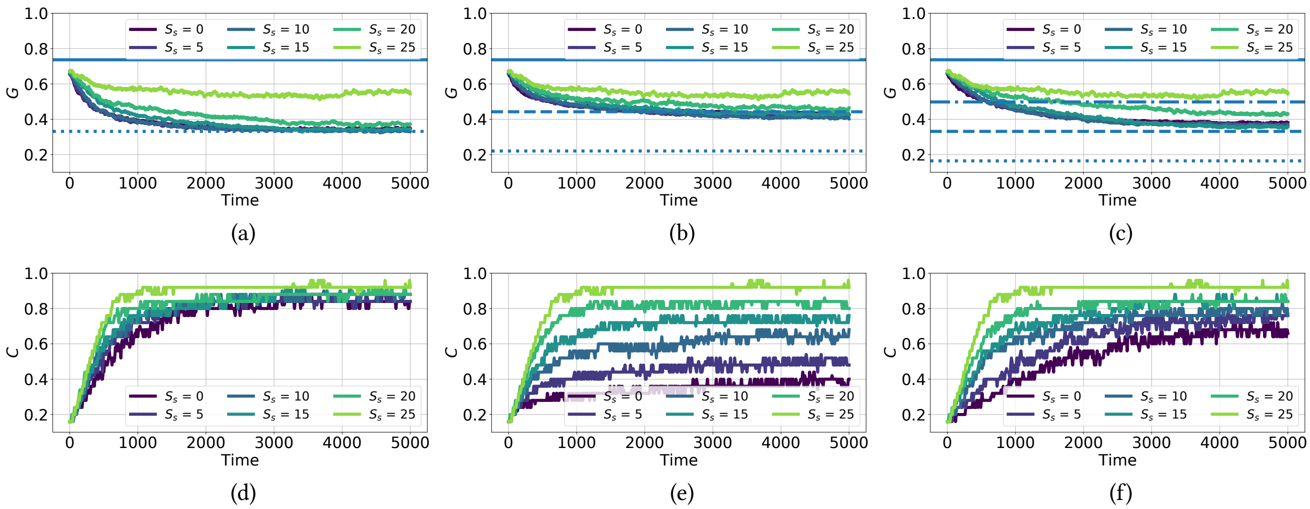


Figure 7: Top row: evolution of the median of the cue-based metric G over time with different configurations of the heterogeneous swarm in the ARGoS simulator over 100 trials with (a) two, (b) three and (c) four levels quantisation of the environmental feature. Bottom row: evolution of the median of the cluster metric over time with different configurations of the heterogeneous swarm in the ARGoS simulator over 100 trials with (a) two, (b) three and (c) four levels quantisation of the environmental feature. The number of neighbour-based robots (S_s) in the swarm is varied in the range [0,25].

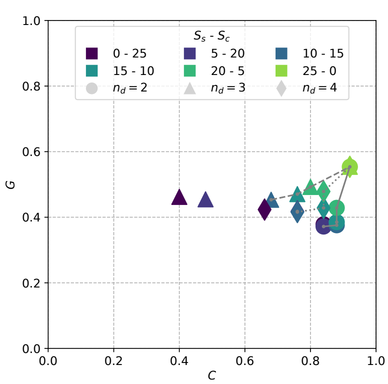


Figure 8: Pareto diagram for different configurations of the heterogeneous swarm. The median of the cue-based metric over the total experiment time is represented on the vertical axis and the median of the cluster metric found at the end of the experiment on the horizontal axis.

the composition of the swarm and its effect on the two metrics. Starting from a swarm made up of only neighbour-based robots, we obtain the best value of the cluster metric but also the worst value of the cue-based metric. By introducing progressively more cue-based robots, the cue-based metric improves but the cluster metric decreases. Looking at the three Pareto fronts represented by the lines in Figure 8 for each value of n_d, specific configurations of the swarm can be chosen by the designer in order to attain reasonable performances in both metrics. Within our experimental setup, it appears that the best compromise is a swarm made up of $S_s = 10$ or 15 cue-based robots.

Overall, our results indicate that a swarm composed exclusively of one of the two robot types will perform well in one metric but poorly in the other. Neighbour-based robots form a big cluster containing the majority of the swarm but do so at a random location; cue-based robots aggregate near the centre of the arena but fail to form a dense cluster. A heterogeneous swarm combining both robot types offers a compromise between the two situations. Importantly, a minority of cue-based robots is sufficient to obtain a single cluster in correspondence to areas with low cue intensity. In other words, the addition of a minority of cue-based robots is enough to steer the whole swarm towards the desired areas while keeping a reasonably dense cluster (see the video in the supplementary material [8]).

3.4 Scalability analysis

We conducted a scalability analysis for different arena and swarm sizes, keeping fixed the environmental cue shape and extension, to evaluate the impact of swarm density on performance. We choose the case with $n_d = 2$ and a heterogeneous swarm featuring a fraction of $H = 0.4$ social robots, as these provided the best trade-off between the G and C metrics (see Figure 9). Additional plots for different configurations are available as supplementary material [8]. Results indicate that the cue-based metric gets slightly worse with increasing swarm size. We observed that, for larger swarm sizes, robots tend to form a large connected cluster in the centre of the arena. However, neighbour-based robots tend to remain outside of the central black area because they still perceive enough neighbours to perform a local walk. Moreover, the black area becomes more crowded with robots that are more likely to move out or being pushed outside by other robots. This effect is consistent across all studied arena dimensions. The G metric also degrades when increasing the arena size. Due to a lower chance of finding

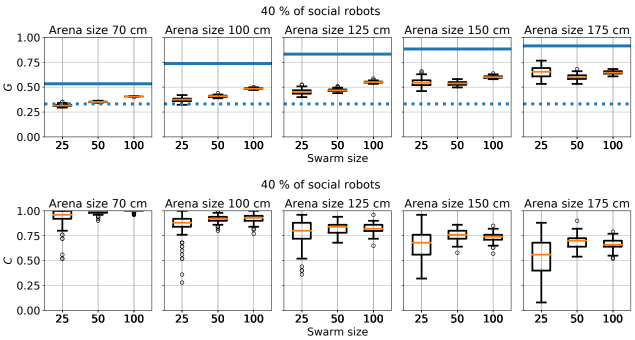


Figure 9: Scalability tested over 100 trials of $T = 5000$ s for different arena and swarm sizes, with a fraction of $H = 0.4$ social robots and $n_d = 2$.

the black area, robots remain outside for longer periods. Overall, the fraction of aggregated robots degrades but a single cluster is very often formed (see the videos in the supplementary material [8]).

The cluster metric displays a very good performance for small arena sizes and degrades progressively when increasing the size of the arena. This is due to cue-based robots not finding the centre of the arena and neighbour-based robots struggling to find each other to form clusters. One large cluster is always formed in the centre without containing the totality of the swarm. A higher variability also appears for a swarm size of 25 robots due to the low density: even when the robots are clustered in the black area, they are not always forming a single cluster (see the videos in the supplementary material [8]).

4 CONCLUSIONS

We have shown that the use of adaptive random walks relying on the sensing of environmental cues or the presence of neighbours can generate aggregation behaviours in a swarm of minimalist robots. Our proposed method follows the principle of minimal computing, using agents with low or no processing power that are equipped with few simple sensors. This makes our approach especially relevant for swarms of micro or nano-robots lacking the ability to precisely sense or explore their environment. Based on the sensed inputs, the robots performed different types of random walk, displaying gregarious or exploratory behaviours. We devised two controllers based on different sensor types to generate aggregation dynamics within a swarm of Kilobots that we tuned with the use of the iRace optimisation algorithm [21].

The first controller achieved cue-based aggregation following a continuous environmental cue with a local minimum in the centre of the arena. Robots were only equipped with a sensor measuring the local scalar value of the cue at their position. Three different sensor resolutions were studied in order to compare the effect of the amount of information available to the robots on the aggregation performance. Results show that the controllers relying on high sensor resolution (i.e., $n_d > 2$) did not produce better performance in terms of proximity of the swarm to the minimum of the environmental cue.

Our second controller achieved a neighbour-based aggregation behaviour relying only on the sensing of other neighbours. In opposition to the previous cue-based aggregation behaviour aiming at regrouping the swarm towards a certain location, neighbour-based aggregation can occur anywhere in the environment with the only goal being the formation of a dense cluster containing the majority of the swarm. To do so, our chosen FSM is composed of two different random walk states where transitions happen based on two different thresholds: when the number of neighbours is high, the robot will follow a Brownian motion; when it is low, it will perform a persistent walk in search of higher densities. Our results show that our swarm self-aggregates with a reasonable convergence time and consistently forms a single cluster containing the majority of the swarm. It can also be noted that performances were significantly improved in the ARGoS simulator compared to the abstract one due to the embodiment of the robots and the collisions that occur between them, resulting in a better aggregation.

We also studied the effect of heterogeneity on the aggregation of the swarm by forming a heterogeneous swarm with the two different robot types, varying the proportions of cue-based and neighbour-based robots. Our results indicate that a homogeneous swarm made up of one of the two types will perform very well in their own metric but will have the worse performance regarding the other. Cue-based robots will group themselves near the centre of the environmental cue but will fail to do so in a dense cluster, while neighbour-based robots will form a single dense cluster but at a random location in the environment. We show that a trade-off is possible with a heterogeneous swarm composed of a minority of neighbour-based robots, the aggregation dynamics being the following: the cue-based robots will first gather towards the minimum of the environmental cue, generating a spatial distribution of the robots with a higher density at this location; the neighbour-based robots will then begin to cluster near them, generating a dense cluster with the majority of the robots. Reasonable values are thus attained for both studied metrics with this compromise.

With this paper, we argue that adaptive random walks can display interesting collective behaviours despite the low complexity they entail. Aggregation can be achieved even with robots that never stop moving. We believe that the dynamic nature of these aggregation behaviours can be relevant in case of complex, dynamic environments, in which the cue gradient can have irregular shapes and can change over time. Indeed, by remaining in motion, robots can continue to sample the environment and adapt to its changes. Future work will test this intuition. We will also implement our proposed controllers on physical Kilobots to see if performance is consistent with what was observed in the ARGoS simulator.

ACKNOWLEDGMENTS

The authors thank Jonas Kuckling for the technical support on running simulations on the HPC. This work was partially supported by Service Public de Wallonie Recherche under grant n° 2010235-ARIAC by DigitalWallonia4.AI. A. Reina also acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Chargé de Recherches.

REFERENCES

[1] Farshad Arvin, Ali Emre Turgut, Farhad Bazyari, Kutluk Bilge Arikan, Nicola Bellotto, and Shigang Yue. 2014. Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. *Adaptive Behavior* 22, 3 (2014), 189–206. https://doi.org/10.1177/1059712314528009

[2] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. 2001. *Self-Organization in Biological Systems*. Princeton University Press, Princeton. https://doi.org/10.1515/9780691212920

[3] Nicolas Cambier, Dario Albani, Vincent Frémont, Vito Trianni, and Eliseo Ferrante. 2021. Cultural evolution of probabilistic aggregation in synthetic swarms. *Applied Soft Computing* 113 (2021), 108010. https://doi.org/10.1016/j.asoc.2021.108010

[4] Kilian Dietrich, Nick Jaensson, Ivo Buttinoni, Giorgio Volpe, and Lucio Isa. 2020. Microscale Marangoni Surfers. *Physical Review Letters* 125 (2020), 098001. Issue 9. https://doi.org/10.1103/PhysRevLett.125.098001

[5] Cristina Dimidov, Giuseppe Oriolo, and Vito Trianni. 2016. Random walks in swarm robotics: an experiment with Kilobots. In *Swarm Intelligence: 10th International Conference, ANTS 2016 (LNCS, Vol. 9882)*. Springer International Publishing, Cham, 185–196. https://doi.org/10.1007/978-3-319-44427-7_16

[6] Marco Dorigo, Vito Trianni, Erol Şahin, Roderich Gross, Thomas H Labella, Gianluca Baldassarre, Stefano Nolfi, Jean-Louis Deneubourg, Francesco Mondada, Dario Floreano, and Luca M Gambardella. 2004. Evolving Self-Organizing Behaviors for a Swarm-Bot. *Autonomous Robots* 17, 2 (2004), 223–245. https://doi.org/10.1023/B:AURO.0000033973.24945.f3

[7] S Duncan, G Estrada-Rodriguez, J Stocek, M Dragone, P A Vargas, and H Gimperlein. 2022. Efficient quantitative assessment of robot swarms: coverage and targeting Lévy strategies. *Bioinspiration & Biomimetics* 17, 3 (march 2022), 036006. https://doi.org/10.1088/1748-3190/ac57f0

[8] Luigi Feola, Antoine Sion, Andreagiovanni Reina, Vito Trianni, and Elio Tuci. 2023. Supplementary material. https://doi.org/10.5281/zenodo.7845595

[9] Luigi Feola and Vito Trianni. 2022. Adaptive Strategies for Team Formation in Minimalist Robot Swarms. *IEEE Robotics and Automation Letters* 7, 2 (2022), 4079–4085. https://doi.org/10.1109/LRA.2022.3150479

[10] Miguel Angel Fernandez-Rodriguez, Fabio Grillo, Laura Alvarez, Marco Rathlef, Ivo Buttinoni, Giovanni Volpe, and Lucio Isa. 2020. Feedback-controlled active brownian colloids with space-dependent rotational dynamics. *Nature Communications* 11, 1 (2020), 4223. https://doi.org/10.1038/s41467-020-17864-4

[11] Z. Firat, E. Ferrante, Y. Gillet, and E. Tuci. 2020. On self-organised aggregation dynamics in swarms of robots with informed robots. *Neural Computing and Applications* 32, 17 (2020), 13825–13841. https://doi.org/10.1007/s00521-020-04791-0

[12] S. Garnier, C. Jost, J. Gautrais, M. Asadpour, G. Caprari, R. Jeanson, A. Grimal, and G Theraulaz. 2008. The embodiment of cockroach aggregation behavior in a group of micro-robots. *Artificial Life* 14, 4 (2008), 387–408. https://doi.org/10.1162/artl.2008.14.4.14400

[13] Melvin Gauci, Jianing Chen, Wei Li, Tony J. Dodd, and Roderich Groß. 2014. Self-organized aggregation without computation. *The International Journal of Robotics Research* 33, 8 (2014), 1145–1161. https://doi.org/10.1177/0278364914525244

[14] Heiko Hamann, Thomas Schmickl, Heinz Wörn, and Karl Crailsheim. 2012. Analysis of emergent symmetry breaking in collective decision making. *Neural computing & applications* 21, 2 (2012), 207–218. https://doi.org/10.1007/s00521-010-0368-6

[15] Raphael Jeanson, Colette Rivault, Jean-Louis Deneubourg, Stephane Blanco, Richard Fournier, Christian Jost, and Guy Theraulaz. 2005. Self-organized aggregation in cockroaches. *Animal Behaviour* 69, 1 (2005), 169–180. https://doi.org/10.1016/j.anbehav.2004.02.009

[16] E. L. Kaplan and Paul Meier. 1958. Nonparametric Estimation from Incomplete Observations. *J. Amer. Statist. Assoc.* 53, 282 (1958), 457–481. https://doi.org/10.1080/01621459.1958.10501452

[17] Hamid Karani, Gerardo E. Pradillo, and Petia M. Vlahovska. 2019. Tuning the Random Walk of Active Colloids: From Individual Run-and-Tumble to Dynamic Clustering. *Physical Review Letters* 123 (2019), 208002. Issue 20. https://doi.org/10.1103/PhysRevLett.123.208002

[18] Yoshiaki Katada, Sho Hasegawa, Kaito Yamashita, Naoki Okazaki, and Kazuhiro Ohkura. 2022. Swarm Crawler Robots Using Lévy Flight for Targets Exploration in Large Environments. *Robotics* 11, 4 (2022), 76. https://doi.org/10.3390/robotics11040076

[19] Yoshiaki Katada, Akihiro Nishiguchi, Kazuya Moriwaki, and Ryosuke Watakabe. 2016. Swarm robotic network using Lévy flight in target detection problem. *Artificial Life and Robotics* 21, 3 (01 Sep 2016), 295–301. https://doi.org/10.1007/s10015-016-0298-1

[20] Miquel Kegeleirs, David Garzón Ramos, and Mauro Birattari. 2019. Random Walk Exploration for Swarm Mapping. In *Towards Autonomous Robotic Systems*, Kaspar Althoefer, Jelizaveta Konstantinova, and Ketao Zhang (Eds.). Springer International Publishing, Cham, 211–222. https://doi.org/10.1007/978-3-030-25332-5_19

[21] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Thomas Stützle, and Mauro Birattari. 2016. The irace package: Iterated Racing for Automatic Algorithm Configuration. *Operations Research Perspectives* 3 (2016), 43–58. https://doi.org/10.1016/j.orp.2016.09.002

[22] Bao Pang, Yong Song, Chengjin Zhang, and Runtao Yang. 2021. Effect of random walk methods on searching efficiency in swarm robots for area exploration. *Applied Intelligence* 51, 7 (01 Jul 2021), 5189–5199. https://doi.org/10.1007/s10489-020-02060-0

[23] Carlo Pinciroli, Mohamed S Talamali, Andreagiovanni Reina, James AR Marshall, and Vito Trianni. 2018. Simulating Kilobots within ARGoS: Models and experimental validation. In *Swarm Intelligence: 11th International Conference, ANTS 2018*. Springer, Springer International Publishing, Cham, 176–187. https://doi.org/10.1007/978-3-030-00533-7_14

[24] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. 2012. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence* 6, 4 (2012), 271–295. https://doi.org/10.1007/s11721-012-0072-5

[25] Andreagiovanni Reina, Alex J. Cope, Eleftherios Nikolaidis, James A. R. Marshall, and Chelsea Sabo. 2017. ARK: Augmented Reality for Kilobots. *IEEE Robotics and Automation Letters* 2, 3 (2017), 1755–1761. https://doi.org/10.1109/LRA.2017.2700059

[26] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. 2012. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., United States, 3293–3298. https://doi.org/10.1109/ICRA.2012.6224638

[27] A. Sion, A. Reina, M. Birattari, and E. Tuci. 2022. Controlling Robot Swarm Aggregation through a Minority of Informed Robots. In *Swarm Intelligence: 13th International Conference, ANTS 2022 (LNCS, Vol. 13491)*. Springer International Publishing, Cham, 91–103. https://doi.org/10.1007/978-3-031-20176-9_8

[28] Onur Soysal and Erol Sahin. 2005. Probabilistic aggregation strategies in swarm robotic systems. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.* IEEE, 325–332. https://doi.org/10.1109/SIS.2005.1501639

[29] Martina Szopek, Thomas Schmickl, Ronald Thenius, Gerald Radspieler, and Karl Crailsheim. 2013. Dynamics of Collective Decision Making of Honeybees in Complex Temperature Fields. *PLOS ONE* 8, 10 (10 2013), 1–11. https://doi.org/10.1371/journal.pone.0076250

[30] Vito Trianni and Alexandre Campo. 2015. *Fundamental Collective Behaviors in Swarm Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1377–1394. https://doi.org/10.1007/978-3-662-43505-2_71

[31] Fuda van Diggelen, Jie Luo, Tugay Alperen Karagüzel, Nicolas Cambier, Eliseo Ferrante, and A. E. Eiben. 2022. Environment Induced Emergence of Collective Behavior in Evolving Swarms with Limited Sensing. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 31–39. https://doi.org/10.1145/3512290.3528735