Adaptive phototaxis of a swarm of mobile robots using positive and negative feedback self-alignment

Yoones Mirhosseini, Matan Yah Ben Zion, Olivier Dauchot, Nicolas Bredeche

▶ **To cite this version:**

Yoones Mirhosseini, Matan Yah Ben Zion, Olivier Dauchot, Nicolas Bredeche. Adaptive phototaxis of a swarm of mobile robots using positive and negative feedback self-alignment. GECCO '22: Genetic and Evolutionary Computation Conference, 2022, Boston, United States. pp.104-112, 10.1145/3512290.3528816 . hal-03842204

HAL Id: hal-03842204

https://hal.sorbonne-universite.fr/hal-03842204v1

Submitted on 8 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Phototaxis of a Swarm of Mobile Robots using Positive and Negative Feedback Self-Alignment

Yoones Mirhosseini
Sorbonne Université, CNRS, ISIR
Paris, France
yoones.mirhosseini@sorbonne-universite.fr

Matan Yah Ben Zion
Tel Aviv University
Tel Aviv, Israel
matanbz@gmail.com

Olivier Dauchot
ESPCI, PSL Research University, CNRS, Gulliver lab
Paris, France
olivier.dauchot@espci.fr

Nicolas Bredeche
Sorbonne Université, CNRS, ISIR
Paris, France
nicolas.bredeche@sorbonne-universite.fr

ABSTRACT

In this paper, we explore how robots in a swarm can individually exploit collisions to produce self-organizing behaviours at the macroscopic scale. We propose to focus on two behaviours that modify the orientation of a robot during a collision, which are inspired by positive and negative feedback observed in Nature. These two behaviours differ in the nature of the feedback produced after a collision by favouring either (1) the alignment or (2) the anti-alignment of the robot with an external force, whether it is an obstacle or another robot. We describe a social learning algorithm using evolutionary operators to learn individual policies that exploit these behaviours in an online and distributed fashion. This algorithm is validated both in simulation and with real robots to solve two tasks involving phototaxis, one of which requires self-organized aggregation to be completed.

CCS CONCEPTS

• **Computing methodologies** → **Evolutionary robotics**; • **Theory of computation** → **Multi-agent reinforcement learning**.

KEYWORDS

swarm robotics, evolutionary robotics, active matter, phototaxis, online distributed evolutionary reinforcement learning

ACM Reference Format:
Yoones Mirhosseini, Matan Yah Ben Zion, Olivier Dauchot, and Nicolas Bredeche. 2022. Adaptive Phototaxis of a Swarm of Mobile Robots using Positive and Negative Feedback Self-Alignment. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528816

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '22, July 9–13, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528816

1 INTRODUCTION

Nature abounds with examples where self-organizing active units can be observed, whether it is at the microscopic scale with artificial colloids or bacteria [14], or at the macroscopic scale with animals [9], including humans [17]. One notable self-organizing behaviour is that of self-alignment, where individual units orient themselves with respect to one another to collectively move in the same direction [4], which at the microscopic scale is produced though collision-based interaction rather than sensing.

In this paper, we consider swarm robotics as one such kind of system composed of active units. In swarm robotics, each robot interacts with its immediate surrounding, and it is expected that self-organization at swarm level can be attained to address a user-defined task [3, 5, 13, 16, 32] and previous works have indeed taken inspiration from active matter for multi-robot morphogenesis [18, 24, 28], coordination [21, 22] and clustering [10].

We propose to take advantage of physical interactions between the robot and its surrounding. We exploit two simple behaviours triggered when a collision occur, with both behaviours displaying an opposite reaction when confronted to an external force:

- alignment, where a robot orients to align with the external force experienced. In practical, the robot will turn **away** from the point of collision until no external force is experienced anymore. In case of collision with a wall, the robot will escape by following the wall, as no more collision occur anymore;
- anti-alignment, where a robot orients to align with the *opposite* of the external force experienced. The robot will turn *towards* the point of collision, until a stable equilibrium is reached. A robot colliding with a wall will thus turns towards the wall up to the point the self-propulsive force and the external force cancel each other. I.e. the robot will be stuck facing the wall.

In the following, we simulate collision-based alignment and anti-alignment on both simulated and real robots. However, we have already observed such behaviours to independently emerge naturally from physical interactions between robots using dedicated exoskeletons (see [31] for details). To harness such morphological computation mechanisms, we present a social learning algorithm that is distributed over the robots and runs in an online fashion, accounting for the robots' limited communication capabilities. This algorithm allows to learn efficient policies while having already

deployed the robot swarm, which is an essential feature whenever the environment and/or the exact outcome of physical interactions are unknown before deployment.

2 METHOD

2.1 Active Matter Physical Model

The specificity of an active unit is to be propelled in a direction of its own, defined by design. In the absence of obstacle or noise, the active unit then naturally moves along this direction. This description is however too simple. On one hand, one must include the existence of various sources of noise, which primarily act on the orientation of the self-propulsion. On the other hand, experiments with model systems have demonstrated the existence of a coupling between the orientation $\boldsymbol{n}$ of the self-propulsion and the actual velocity $\boldsymbol{v}$ of the active unit [12, 31]. This coupling takes the form of a torque, which aligns or anti-aligns $\boldsymbol{n}$ with $\boldsymbol{v}$. The minimal equations describing the above behaviour read:

$$m\frac{d\boldsymbol{v}}{dt} = F_0\boldsymbol{n} - \gamma\boldsymbol{v} + \boldsymbol{F}_w, \tag{1}$$

$$\tau\frac{d\boldsymbol{n}}{dt} = \zeta(\boldsymbol{n}\times\boldsymbol{v})\times\boldsymbol{n} + \sqrt{2\alpha}\xi\boldsymbol{n}_\perp. \tag{2}$$

where m is the mass of the active unit, $F_0 n$ is the self-propulsive force, γ is a friction coefficient and $\boldsymbol{F}_w$ is an external force, such as the one exerted by a wall. τ is the characteristic time of the orientational dynamics and α is the amplitude of the noise $\xi(t)$, assumed to be Gaussian, with correlations $\langle\xi(t)\xi(t')\rangle = \delta(t-t')$. Finally, the orientation dynamics (Eq. (2)) contains the key ingredient, specific to the model, namely the presence of an aligning (when $\zeta > 0$), respectively anti-aligning (when $\zeta < 0$), torque of the orientation $\boldsymbol{n}$ towards the velocity $\boldsymbol{v}$. In free space, the active units performs a persistent random walk, with a short time ballistic dynamics characterized by a velocity $v_0 = F_0/\gamma$ and a long time diffusive dynamics with a diffusion constant $D = \alpha/\tau^2$. The effect of the sign of ζ on the dynamics of an active unit when it encounters a wall is illustrated on Figure 1. When $\zeta > 0$, the active unit follows the wall; when $\zeta < 0$, it points into the wall and remains stuck. A positive alignment was also shown to be at the root of the emergence of collective motion in a system of vibrated polar discs [20, 30].

2.2 Robotic Implementation

We implemented two behaviours, which we call *alignment* and *anti-alignment*. Both are defined with respect to the robot rotation in response to the relative position of a collision (with another robot, or with an obstacle such as a wall). The detection of collision is achieved by using short-range proximity sensors or bumpers on the front-side of a round-shaped robot. Action is taken by setting a rotational velocity (using a 2-wheel differential setup) resulting in the robot turning either towards or away from the collision point, depending on the behaviour considered.

- Implementation of alignment: using front sensors, a detected collision triggers *evasion* by rotating in the opposite direction.
- Implementation of anti-alignment: similar to alignment, front sensors are used to detect collision. However, rotation is now triggered to *face* the collision event. In case of symmetry, no rotation action is triggered.

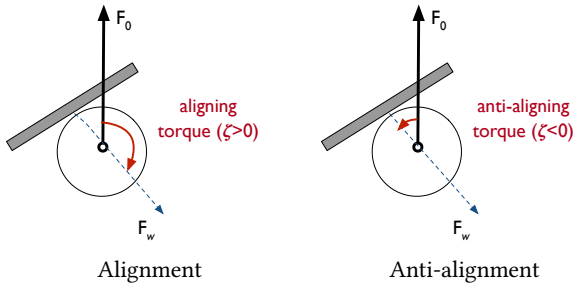

Figure 1: Illustration of the two variations of the active unit model, with F_0 the self-propulsive force and F_w the external force (applied from the contact point with the wall). Left: the unit's self-propulsive force aligns with the external force resulting from the collision (e.g. the unit escapes as soon as it no longer experience collision). Right: the unit's self-propulsive force anti-aligns with the external force, ultimately ending up with the unit facing the external force (e.g. the unit get stuck facing a the wall). See text for the full explanation.

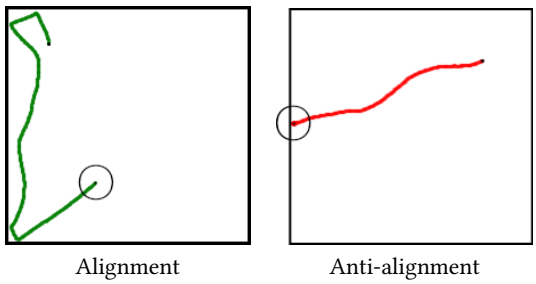

Figure 2: Trajectory obtained with either behaviour. The circle shows the location of the robot at the end of its trajectory. Left: the robot goes forward (with diffusion) and aligns with walls to escape. Right: the robot goes forward (with diffusion) and orients itself to face the wall.

With either behaviour, the robot keeps going forward even during rotation. This means that a robot may exert a force on the obstacle in front (i.e. it can theoretically push small objects or robots).

As a default behaviour, which is applicable in the absence of a collision event, the robot simply goes straight-forward to which a small diffusion is added with a limited amount of Gaussian noise in the orientation velocity. It should be noted that while we explicitly implement diffusion, this is usually given for free when using real robots due to motor and/or wheel wear, as well as (if applicable) deviation of odometry or IMU through time.

Figure 2 illustrates our implementation in a pseudo-realistic robotic simulation [8] of both straight-forward translation with diffusion (left and right images), alignment (left) and anti-alignment (right) behaviours. Diffusion during translation is achieved by adding Gaussian noise of mean 0 and standard deviation of 1.0, with maximum value in $[-3°, +3°]$ degrees. Maximal rotation speed during

alignment or anti-alignment is within $[-30°, +30°]$ (i.e. the robot may rotate for 30° in-between calls to the robot controller), Maximal speed is $1/6^{th}$ the diameter of the robot in-between calls to the robot controller. Assuming 10 control updates per second, this corresponds to a robot with a translation speed of maximum $x1.7$ diameter per second and a rotation of maximum 300° per second, which is the same order of magnitude of typical autonomous robots such as the Kilobots [23] or Mona Robots [1]. For example, the Mona robot ($\varnothing \approx 8.2cm$, version with an ESP32 microcontroller) rotates at a maximum of 180 deg per second and translates forward at a maximum of $x1.6$ diameter per second.

2.3 Task

In this work, we evaluate the efficiency of the proposed behaviours for addressing phototaxis with a group of robots. The objective is for robots to maximize the light received, which implies standing in the region which is closer to the light source. Previous works in swarm and collective robotics have addressed phototaxis using *ad-hoc* bio-inspired algorithms [2, 16, 19, 21, 25] as well as evolutionary swarm robotics algorithms [11, 26, 27].

Phototaxis can be instantiated in various fashions, depending on the structure of the environment, the number, location and nature of light sources, etc. In particular, the light source can diffuse and gradually loses intensity through space, or project a spot with uniform light intensity. In the case of a diffusive light source, the question of competitive access to the best spot may arise when multiple individuals are considered.

In the following, we endow each robot with the capability to perform a self-assessment of phototaxis score, which can be implemented on-board each robot. Individual performance accounts for the amount of light intensity perceived through a photosensor in the last N steps. The phototaxis score is computed as:

$$score_T = \sum_{t=T-N}^{T} lum_T \tag{3}$$

with lum_T, the amount of light measured at the current time t by the robot's onboard photosensor, normalized in $[0, 1]$. At any moment t, the current photosensor value can be queried as:

$$lum_T = photosensor.getSensorValue(),\ lum_T \in [0, 1] \tag{4}$$

2.4 Control

The control function is kept to the minimum, as a linear combination of sensory inputs which output sign is used to select the behaviour. It is actually a Perceptron with only two inputs (incl. one bias) and one output, which is computed as follows:

$$output = H(\theta_0 * PhotosensorValue + \theta_1) \tag{5}$$

With H the Heaviside step function and $PhotosensorValue$ the photosensor normalized value with 1.0 denoting a sensor saturated with light. The Heaviside step function outputs either 0 or 1, which value is used to select either the alignment behaviour or the anti-alignment behaviour. Remember that whatever the behaviour is used, the robot will always go forward (with diffusion) in the absence of collisions.

θ_0 and θ_1 are the policy parameters, with θ_1 acting as a bias value to enable a non-zero output in the absence of light. The search space is very small, with only two parameters, and the optimization problem is basically to tune both parameters with respect to the light intensity threshold relevant for phototaxis and aggregation close to the light source. These parameters can be set by hand in a controlled environment by measuring light intensity at the frontier of the region where the robots should find a way to stop, possibly by anti-aligning with an obstacle nearby.

However, if the environment is unknown, we face two problems. Firstly, the light intensity threshold must be discovered by the robots following a trial and error in order to maximize the efficiency of phototaxis. Secondly, and most importantly, we consider robots with limited communication capabilities. This has an important impact as the light intensity threshold optimization must be conducted in an online and distributed manner, which we will describe hereafter.

2.5 Learning

In order to learn the control parameter values, we rely on a distributed online evolutionary learning algorithm [7, 29] where each robot uses particular policy parameter values, which can be exchanged with other robots and selected with respect to their performance. In this class of problems, an algorithm is implemented on *each* robot. The whole robot swarm running the algorithm shares some similarities with an island model for parallel evolutionary optimization, only with a highly dynamic reconfiguration of the communication network that depends on robots' actions and proximity with one another.

This mimics the process of social learning where an individual's innovation diffuses to neighbours and possibly invades the whole population [6]. This makes it possible to deploy the robot swarm without the need to *a priori* know the exact characteristics of the environment (e.g.: minimum and maximum light intensities, photosensor dynamic range, robot motor dynamics, etc.).

The artificial social learning algorithm we use is built on the algorithm introduced in [15], referred to as the HIT algorithm (for *Horizontal Information Transfer*). It is distributed over the robots, and evolutionary optimization is conducted by passing messages in-between neighbours. It uses local mutation and robot-to-robot transfer of policy parameter values. The use of mutation allows individuals to innovate by exploring new values for the parameter. Parameter values that improve performance can then be copied by robots within reach of the onboard communication device.

We extend the HIT algorithm by enabling each robot to store candidate policy parameters from several other better-performing robots before selecting and possibly mutating new policy parameters (rather than selecting the first better-performing set of policy parameters received, as in the original algorithm). Our algorithm is termed HIT-Res, which uses a "reservoir" where incoming policy parameters are stored. HIT-Res is described in algorithm 1, which important features are the following:

- *performance self-assessment* is performed on-board (in the current paper, we use equation 3). In order to ensure a reliable evaluation, a *maturation period* is defined during which

the robot does not send nor receive information. The maturation period is used to compute a reliable average score with respect to the task compared to instantaneous noisy evaluations (see Line 10);
- innovation is achieved through *mutation*: whenever policy parameters are updated, a mutation event may occur that can possibly introduce new values in the policy parameters. Mutation has both a probability to occur and an amplitude in terms of the number of parameters concerned (Line 20);
- diffusion is ensured by the *transfer* operator, which uses the robot's communication apparatus to (continuously) transmit part or all of the policy parameters (and the current performance self-assessment) to other robots within reach, if any. It is important to note that robot-to-robot transfer is the *only* way for robots to run the learning algorithm (i.e. there is no central computer to conduct learning) (Line 11 – 14);
- policy parameters received from other robots are stored in a *reservoir* (i.e. a list), from which the *selection* operator picks a candidate set of policy values for updating some (or all) the robot's policy parameters. Selection only occurs for candidate solutions whose performance dominates that of the current policy (if none is better, the reservoir is emptied, and no selection occurs). This feature differs from the original HIT algorithm by sampling several possible candidate solutions, rather than just one (Line 16);

In this work, the choice of a lightweight social learning algorithm and the low-dimensional policy search space (though continuous) is motivated by a frugal approach to distributed online reinforcement learning, which is nonetheless easily deployable in a swarm of robots with limited computation and communication capabilities.

Algorithm 1: The HIT-Res algorithm (*Horizontal Information Transfer with Reservoir*).

Data:
π : Policy function,
θ : Policy parameters (random initial values), $\dim(\theta) = m$,
α : transfer rate $\in [0, 1]$,
σ : mutation rate $\in [0, 1]$,
V_m : mutation volume $\in [1, \dim(\boldsymbol{\theta})]$,
T : evaluation time,
$R[T]$: Empty reward buffer of size T,
r : Null reward scalar,
G : Null score value ($= \sum_{k=0}^{T-1} R[k]$),
$\mathbf{a}$: Null action vector,
$\mathbf{o}$: Null observation vector,
ϑ : a subset of θ ($vartheta \subset \theta$)
$reservoir$: messages received from other robots
K: maximum number of messages stored in the reservoir

```
1  begin
2      t = 0
3      while True do
4          o, r = sense()
5          R[t mod T] = r
6          a = π(o|θ)
7          act(a)
8          t = t + 1
9          if t > T then
10             G = Σ_{k=0}^{T-1} R[k]
11             ϑ = get_random_subset(θ, α)
12             broadcast(ϑ, G)
13             if new_message then
14                 reservoir.add( {ϑ_incoming, G_incoming} )
15                 if reservoir.length() ≥ K then
16                     ϑ_update, G_update = reservoir.select_best()
17                     θ.update_with( ϑ_update ) if
                        σ < random() then
18                         ϑ.mutate(V_m)
19                     end
20                     t = 0 R.empty()
21                 end
22             end
23         end
24     end
25 end
```

3 RESULTS

3.1 Learning in Simulation

We use a multi-robot simulation tool [8] to implement both the alignment and anti-alignment behaviours, as well as the HIT-Res social learning algorithm. Learning and policies are implemented in Python, the simulation core is implemented in C++. Table 1 summarizes the technical details of the robots and arena, and the source code is available at https://github.com/YoonesMir/gecco2022-sourcecode.

The phototaxis environment is implemented by placing a diffusive light that can be perceived from anywhere in the arena, but with decreasing intensity depending on the distance to the source. With respect to the objective function described in Equation 3, we introduce a proxy value for computing the light intensity lum_t^{proxy} as the normalized distance between the robot x and the light source l at time t. This is written as:

$$lum_t^{proxy} = 1 - \frac{1}{distance_{max}} \sqrt{(x_t - x_l)^2 + (y_t - y_l)^2} \quad (6)$$

With (x_t, y_t) the position of the robot at time t and (x_l, y_l) the position of the light. $distance_{max}$ gives the maximum possible distance between the robot and the light source in a closed arena, which ensures that $0 \leq d_t \leq 1$. We set $distance_{max}$ to be equal to the diagonal length of the arena.

Two experimental setups are devised according to the position of a diffusive light:

- **changing-light** setup: the light source is placed outside the arena, and its position alternates between two opposite walls every 100000 iterations, which is the time it takes for a robot to move from one wall to the opposite approx.25 times. It is expected robots will converge towards the light source, and follow it when moved;

arena	$1000pix \times 1000pix$
robot diameter	$32pix$
initial positions	uniform random
max. translation speed	4px/step
max. angular speed	±30 deg/step
translation diffusion	$3 \times N(\mu = 0, \sigma = 1)$ deg/step

Table 1: Technical details for the robot simulation

Perceptron	
Inputs (light+bias)	2
Output	1
Search space	$\mathbb{R}^2$
Initialization range	$[-1, 1]$
HIT-Res	
Evaluation time (T)	400
Transfer rate (α)	0.5
Mutation rate (σ)	0.01
Mutation volume (V_m)	0.5
Reservoir size	3

Table 2: Parameters for policy and learning.

- **center-light** setup: the light source is placed in the center of the arena. The challenge, in that case, is that a single robot will *never* remain close to the light source *only by itself* as there is no such stopping behaviour.

In the following, we present results with 60 robots running the HIT-Res algorithm with a reservoir size of 3 (see Table 2 for all parameters). Learning in each setup is evaluated with 32 independent runs. The robots' initial positions and orientation are randomly drawn, as well as policy parameter values (uniform distribution in both cases).

Fig.3-top shows results in the changing-light setup. Results show that performance increases over time for all runs, reaching a plateau even before the light is moved to the opposite wall for the first time (100000^{th} iteration). After the light is moved, performance decreases immediately as most robots are now positioned at the opposite end of the arena with respect to the light. Performance drops and then recovers, confirming a phototaxis behaviour. Note that each boxplot compiles results from 400 iterations, capturing both the actual drop after the light is moved and the first steps of recovery. Performance going up afterwards indicates that learning is still ongoing at least for some robots. During the next occurrence of light relocation, the performance drop is very small which is due to the time taken by the now efficient robots to follow the light.

This is confirmed by looking at the trajectories, displayed in Fig.4. In the second snapshot (last 400 iterations before relocating the light), robots are shown to remain close to the light source by anti-aligning to collide with any nearby obstacle (wall or robots). Just after the light is moved (third snapshot), robots switch to alignment to explore the environment by aligning with the force exerted by obstacles and other robots and then back to anti-alignment when close enough to the light source.

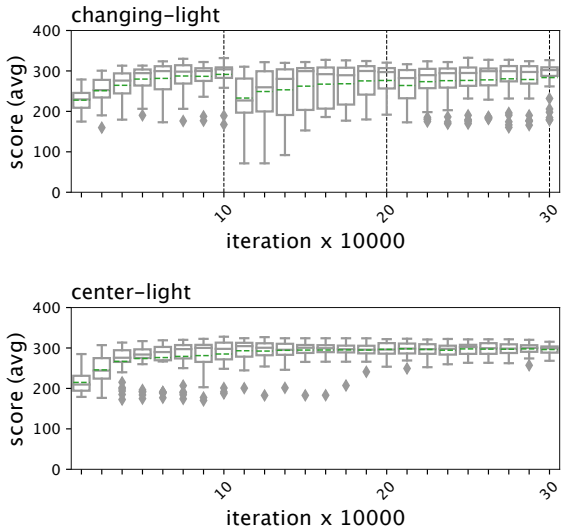

Figure 3: Results for the changing-light (top) and center-light (bottom) setups. Boxplots aggregate data from 32 independent runs, considering the average score for each run during 400 simulation steps. Outliers are shown as losanges. In the changing-light setup, the light moves from one wall to the opposite every 100000 iterations (indicated by vertical lines). X-axis: iteration of the robot simulation. Y-axis: compilation of the scores of all robots of the 32 runs. See Section 2.3 for the score metric.

We perform a similar experimental study with the center-light setup. Fig.3-bottom shows that performance increases steadily to reach a plateau when phototaxis has been learned by most robots. The behavioural strategy is similar to what was observed in the previous setup with alignment used for exploration and anti-alignment for aggregating with others, as shown in Fig.5. One important point here is that in the absence of a wall near the light source, robots rely on a self-organizing aggregation behaviour to perform phototaxis: it is mandatory for *at least* two robots to collide near the light source to start forming an aggregate of robots at the right spot.

By looking closer at the robot's behaviour, self-organized aggregation is actually observed in both the changing-light and center-light setups. As robots are physical entities, it is necessary that a consensus is reached that benefits *all* robots with respect to the value of the light intensity threshold from which anti-alignment is triggered. A too large threshold would lead to sparse aggregation, while a too small one would end up excluding robots that cannot aggregate in an already overcrowded spot. This is illustrated by looking at the reaction maps of robots from collective that have successfully learned to self-aggregate. Reaction maps from Fig.6 shows the behavioural pattern adopted by the majority of robots in both setups. They are obtained during a *post-mortem* study by placing a robot at every location in the environment and monitoring the selected behaviour. As seen for both setups, the light intensity threshold at which the anti-alignment behaviour is triggered is far from the actual light source so as to enable all robots to aggregate.

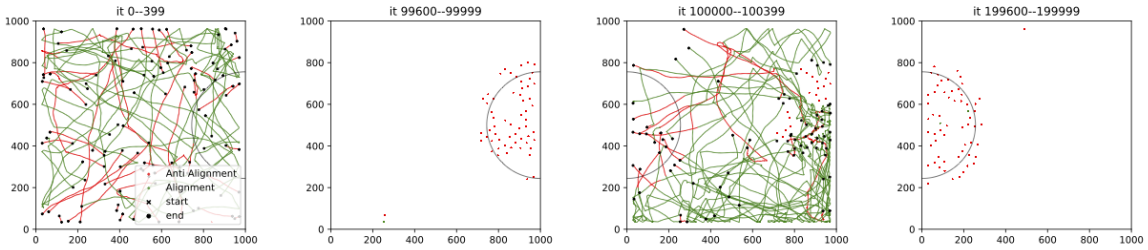

Figure 4: Typical robot trajectories for the changing-light setup throughout learning. The light source is initially positioned on the left. It changes position after 100000 iterations. Each snapshot shows monitoring for 400 simulation steps. (1) trajectories with initial random policies, light on the right; (2) trajectories of policies learned so far, just before the light is moved; (3) trajectories just after the light is moved; (4) trajectories at the end of the run.

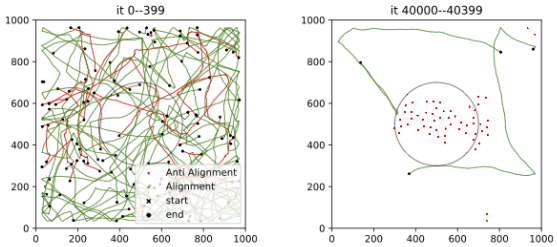

Figure 5: Typical robot trajectories for the center-light setup throughout learning. The light source is positioned in the center. Each snapshot shows monitoring for 400 simulation steps. Left: trajectories with random initial policies. Right: trajectories at the end of the run.

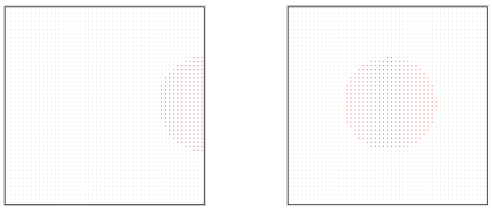

Figure 6: Reaction Maps for the most common behavioural strategy, built by measuring a robot's behaviour selected (small green dots for alignment, bigger red dots for anti-alignment) depending on the location with respect to the light source. Left: Changing-light setup (with light on the left). Right: center-light setup. Reaction maps are built by manually placing one robot at each location in the arena and monitoring its output to determine behaviour activity depending on the location. This is performed for all robots of the last iteration for each run and setup. The most common behavioural pattern is shown here for each setup.

It should also be noted that a minor proportion of robots follow less common behavioural patterns (less than $1/4^{th}$ of the robots, not shown here) where only one behaviour is always triggered. While some of these robots may fail to aggregate at the right spot, they are more often to be trapped within an already formed aggregate (see details in Fig.4 and 5, where aligners are sometimes trapped within groups of anti-aligners).

We performed two sets of additional experiments to evaluate the impact of using (1) a reservoir of candidate solutions, and (2) the Neural Network control architecture. Fig. 7-top shows that using a reservoir size greater than 1 significantly improves performance. Increasing the sample size of candidate solutions from others is beneficial, even though it implies less frequent parameter updates. Fig. 7-bottom confirms that more complex neural network architecture including hidden layers is not required in the setups at hand, possibly due to the alignment and anti-alignment behaviours simplifying the control problem.

3.2 Validation on Real Robots

We previously used simulation to allow for the large number of replications necessary to evaluate the social learning algorithm and the proposed behaviours. However, the question remains open as to how the alignment and anti-alignment are actually exploitable in a real-world setting. In this Section, we turn our attention to the study of the dynamics of the behavioural dynamics itself (both individual and collective), first by implementing an *ad hoc* phototaxis strategy, and then by learning phototaxis, as before.

We use a small group of Mona robots [1], an open-source robotic platform using C++. The Mona robot is a round-shaped robot of $\varnothing \approx 8.2cm$ with two motorized wheels and a front passive ball wheel. It carries 5 infrared sensors distributed on the front of the robot which are used to measure the distance to nearby obstacles (left, front-left, front, front-right and right sensors). We use the ESP-32 microcontroller version (ver.2021) and we manually soldered an additional photosensor on top of the robot in our lab.

We start by focusing on the validation of the behaviours, which means we upload the policy described in the previous Section that selects either behaviour depending on a light intensity threshold. The light intensity input is no longer simulated but uses the raw value provided by the photosensor (cf. Equation 3 and Equation 4).

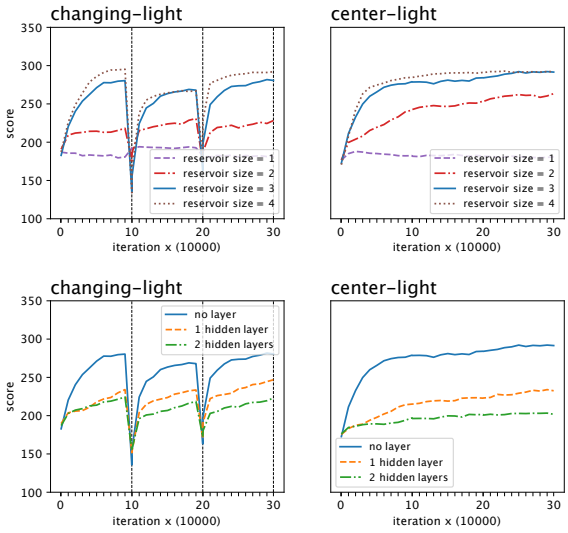

Figure 7: Analysis of the impact of reservoir size (top) and neural networks architecture (bottom) on performance. Only average are shown, and compile data from 32 independent runs per variant, for the changing light setup (left column) and the center light setup (right column). Top: using a reservoir size of 1, 2, 3 (same data from Fig.3) and 4. Bottom: using a neural networks without hidden layer (same data from Fig.3, 2 parameters), with one hidden layer of two neurons (7 parameters) and two hidden layers of two neurones each (13 parameters).

To account for sensor noise, light intensity is smoothed over one second (10Hz sampling rate).

Experiments are performed in two setups which are close to the previously cited, with a few differences to account for two types of light projections (diffusion or light cone):

- border-light with diffusive light: a light source is positioned close to a wall, similar to the changing-light setup described before, but without changing its position during the experiment. We use a softbox light system placed on one side of the arena (approx. 60 *cm* height, pointing to the nearest wall).
- center-light with a light cone: similar to the center-light setup presented earlier, the light source is positioned in the center of the arena. However, the light source is concentrated within a circular area without diffusion (the robot's photosensor is either strongly stimulated, or not at all). An overhead video projector is placed 110 *cm* above the arena that projects a light cone of $\varnothing \approx 40$ *cm* in the center.

Figure 8 shows the experimental setup. The arena has a black non-reflective flat ground, delimited by metallic walls of higher height than the robots. The softbox for diffusive light is on the left and the overhead projector is on top of the image, next to a Pixelink camera system for visual monitoring. Technical details are provided in Table 3.

We performed 10 experiments for the border-light setup using 7 robots with a hand-written policy that selects between alignment

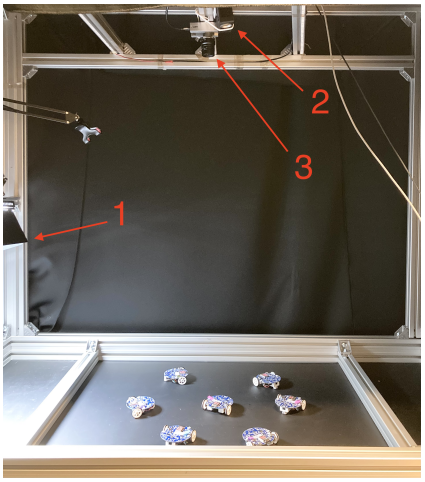

Figure 8: Arena with 7 Mona robots, equipped with a softbox light for diffusive light (no.1, left in the picture), overhead projector for light cone projection (no.2, up), overhead Pixelink camera for tracking and monitoring (no.3) and black occulting sheet for blocking external light interference (front part is temporarily removed for the picture).

Arena	
arena	$89cm \times 72cm$
diameter of spotlight cone	$40cm$
robot	
robot diameter	$8.2cm$
max. translation speed	$6.6cm/sec$
max. rotation speed	$90°/sec$
policy update	$10Hz$
$P_{diffusion}$ (per policy update)	0.05
$amplitude_{diffusion}$	$+/-9°$

Table 3: Technical details for the real robotic setup

and anti-alignment depending on the current light intensity being below or above a fixed threshold. Robots start in a circle in the center of the arena, except for one which is in the center. All robots start with random orientation, and positions on the circle vary from one run to the other. A typical run (run no.2) is shown in Fig.9 through a series of snapshots. The first snapshot shows the initial conditions for this run, and the following snapshots each show the time when one robot has converged to a stationary position. Note that any robot that remains in the same location is actually *still* trying to move and to anti-align with the obstacle it faces (whether it is a wall or another robot).

Fig.11-left tracks the number of robots that remain stationary near the light source ($\leq 3 \times \varnothing_{robot}$ away from the wall) throughout the duration of the experiment, for each of the experiments. As expected, the number of robots increases through time and a growth that is close to linear and dampen near the end. This is expected as phototaxis can be achieved by each robot independently from

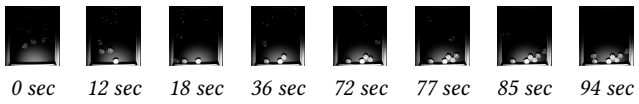

Figure 9: Phototaxis with diffusive light from the side using 7 Mona robots. The light is visible at the bottom (i.e. it is positioned above the left wall in Fig.8, video capture is rotated). Robots are visible as white circles when close enough to the light.

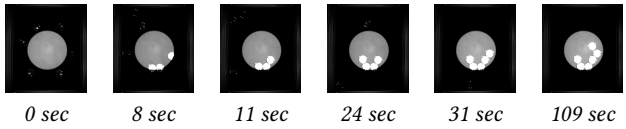

Figure 10: Phototaxis with a cone of light in the center using 6 Mona robots. Robots are visible as white circles when standing in the light cone.

the other since it only requires to anti-align with the wall which is closest to the light source.

We performed 10 experiments for the center-light setup using 6 robots (one robot had to be removed due to a technical problem). Robots start in a circle in the center of the arena, *outside* the light cone. All robots start with random orientation and positions on the circle vary from one run to the other.

A typical run (run no.1) is shown in Fig.10, with the first event involving two robots (2nd snapshot). As exposed in the previous Section, phototaxis with a centered light using alignment and anti-alignment behaviours can *never* be achieved by a single robot. Aggregation in an empty space necessarily requires (at least) two robots both performing anti-alignment so that each robot constitutes an obstacle to the other. Once two robots form a stationary group, other robots may join in. As the group grows bigger, the possibility to join in also increases (larger surface occupied).

Fig.11-middle tracks the number of robots that remain stationary in the light cone area for each of the experiments. Similar to the previous setup, the number of stationary robots grows over time, but with the first step towards successful group phototaxis always involving two robots, which naturally takes more time than when a robot only needs to come close to a wall. In the experiments, we actually observed the diffusive dynamics to be critical during moving without obstacles (see Sections 2.1 and 2.2, as well as noise due to the robots' mechanical design[1]. Such noise in the supposed straight-forward default behaviour that occurs in the absence of obstacle is what makes it possible to get away from walls and explore the environment.

Finally, we implemented the social learning algorithm from the previous Section on the Mona robots, with some adaptations to comply with hardware constraints. We implemented communication using the onboard WiFi device (The robots do not feature a local communication device). However, there is a significant time delay for establishing communication between robots ($\approx$ 1*sec*)

[1]motors are not completely synchronized nor aligned, no odometer was available for feedback control.

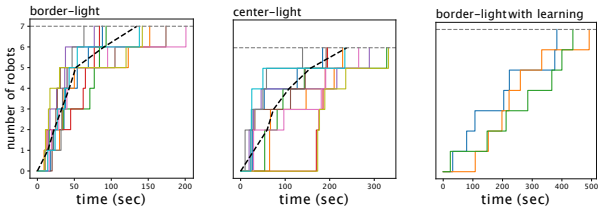

Figure 11: Number of robots standing in the light area throughout the experiment. Left, border-light setup: stationary robots within 3 robot diameters from the wall from where light diffuse (7 robots, 10 indep. runs). Middle, center-light setup: stationary robots within the spotlight (6 robots, 10 indep. runs). Right: border-light setup with learning (7 robots, 3 independent runs). Thick black curve shows average. A coloured discontinuous line corresponds to one run.

during which the robots cannot move. As this strongly limits the amount of information that can be transferred from one robot to another, a robot picks a random other robot to communicate with and selects the policy parameters for transfer *only* if they have a better score attached. The delay between each communication attempt is such that the robot's performance self-assessment can be completely refreshed.

Results with 7 robots running the social learning algorithm in the border-light setup are shown in Fig. 11-right. We use a transfer rate of 1.0 (which is more elitist than before) to speed up convergence and a mutation rate of 0.05. Learning is shown to make the robots converge close to the light, using a similar strategy as when using an *ad hoc* strategy. Learning takes of course significantly more time than before ($\approx$ $\times 2$ to $\times 10$) as the light intensity threshold must be optimized. In the final steps of the experiments, we observe that the learned behavioural strategy exploits the alignment and anti-alignment behaviours for a phototaxis task, similar to what was predicted in Section 3.1 for setups when the environment is initially unknown.

4 CONCLUSION

The work presented here is at the interface of swarm robotics and active matter, and provides two take-home messages. Firstly, simple collision-based aligning and anti-aligning behaviours can be used to perform macro-actions such as obstacle avoidance, self-aggregation and exploration, thus leaving the robot's policy with the task to arbitrate between seemingly simple behaviours. Secondly, social learning can be deployed in a group of simulated and real robots to perform online distributed evolutionary learning that can fine-tune the behavioural strategies of robots with limited communication capability deployed in unknown environments.

ACKNOWLEDGMENTS

We thank F. Caura for photosensor soldering. This work was supported by the MSR project funded by the Agence Nationale pour la Recherche under Grant No ANR-18-CE33-0006.

REFERENCES

[1] Farshad Arvin, Jose Espinosa, Benjamin Bird, Andrew West, Simon Watson, and Barry Lennox. 2019. Mona: an affordable open-source mobile robot for education and research. *Journal of Intelligent & Robotic Systems* 94, 3 (2019), 761–775. Publisher: Springer.

[2] Farshad Arvin, Khairulmizam Samsudin, Abdul Rahman Ramli, and Masoud Bekravi. 2011. Imitation of honeybee aggregation with collective behavior of swarm robots. *International Journal of Computational Intelligence Systems* 4, 4 (2011), 739–748. Publisher: Taylor & Francis.

[3] Levent Bayindir. 2016. A review of swarm robotics tasks. *Neurocomputing* 172 (2016), 292–321. https://doi.org/10.1016/j.neucom.2015.05.116 ISBN: 0925-2312.

[4] E Bonabeau and M Dorigo. 1999. M. and Theraulaz, G.(1999). Swarm Intelligence: From Natural to Artificial Systems. *Santa Fe Institute Studies in the Sciences of Complexity" series, Oxford University Press* (1999).

[5] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. 2013. Swarm robotics : A review from the swarm engineering perspective. *Swarm Intelligence* 7, 1 (2013), 1–41. https://doi.org/10.1007/s11721-012-0075-2

[6] Nicolas Bredeche and Nicolas Fontbonne. 2022. Social learning in swarm robotics. *Philosophical Transactions of the Royal Society B* 377, 1843 (2022), 20200309.

[7] Nicolas Bredeche, Evert Haasdijk, and Abraham Prieto. 2018. Embodied Evolution in Collective Robotics: A Review. *Frontiers in Robotics and AI* 5 (2018), 12. https://doi.org/10.3389/frobt.2018.00012

[8] Nicolas Bredeche, Jean-Marc Montanier, Berend Weel, and Evert Haasdijk. 2013. Roborobo! a Fast Robot Simulator for Swarm and Collective Robotics. *CoRR* abs/1304.2888 (2013). http://arxiv.org/abs/1304.2888 _eprint: 1304.2888.

[9] Scott Camazine, Jean-Louis Deneubourg, N Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. 2003. *Self-organization in biological systems*. Princeton University Press.

[10] Jianing Chen, Melvin Gauci, Michael J Price, and Roderich Groß. 2012. Segregation in swarms of e-puck robots based on the brazil nut effect. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 163–170.

[11] Anders Lyhne Christensen and Marco Dorigo. 2006. Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot. In *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems. Cambridge: MIT Press. A Bradford Book*. 248–254.

[12] Olivier Dauchot and Vincent Démery. 2019. Dynamics of a self-propelled particle in a harmonic trap. *Physical review letters* 122, 6 (2019), 068002.

[13] Marco Dorigo, Guy Theraulaz, and Vito Trianni. 2020. Reflections on the future of swarm robotics. *Science Robotics* 5, 49 (2020). Publisher: Science Robotics.

[14] Étienne Fodor and M. Cristina Marchetti. 2018. The statistical physics of active matter: From self-catalytic colloids to living cells. *Physica A: Statistical Mechanics and its Applications* 504 (aug 2018), 106–120. https://doi.org/10.1016/j.physa.2017.12.137 arXiv:1708.08652

[15] Nicolas Fontbonne, Olivier Dauchot, and Nicolas Bredeche. 2020. Distributed On-line Learning in Swarm Robotics with Limited Communication Bandwidth. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC48606.2020.9185697

[16] Heiko Hamann. 2018. *Swarm Robotics - A Formal Approach*. Springer. https://doi.org/10.1007/978-3-319-74528-2

[17] Dirk Helbing, Lubos Buzna, Anders Johansson, and Torsten Werner. 2005. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation science* 39, 1 (2005), 1–24.

[18] Noemí Carranza Xaver Diego Fredrik Jansson Jaap A. Kaandorp Sabine Hauert James Sharpe Ivica Slavkov, Daniel Carrillo-Zapata. 2018. Morphogenesis in robot swarms. *Science Robotics* 3, 25 (2018). https://doi.org/10.1126/scirobotics.aau9178 Publisher: Science Robotics.

[19] Serge Kernbach, Ronald Thenius, Olga Kernbach, and Thomas Schmickl. 2009. Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adaptive Behavior* 17, 3 (2009), 237–259. Publisher: Sage Publications Sage UK: London, England.

[20] Khanh-Dang Nguyen Thu Lam, Michael Schindler, and Olivier Dauchot. 2015. Self-propelled hard disks: implicit alignment and transition to collective motion. *New Journal of Physics* 17, 11 (2015), 113056.

[21] Shuguang Li, Richa Batra, David Brown, Hyun-Dong Chang, Nikhil Ranganathan, Chuck Hoberman, Daniela Rus, and Hod Lipson. 2019. Particle robotics based on statistical mechanics of loosely coupled components. *Nature* 567, 7748 (2019), 361–365. Publisher: Nature Publishing Group.

[22] Giorgio Oliveri, Lucas C van Laake, Cesare Carissimo, Clara Miette, and Johannes TB Overvelde. 2021. Continuous learning of emergent behavior in robotic matter. *Proceedings of the National Academy of Sciences* 118, 21 (2021). Publisher: National Acad Sciences.

[23] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. 2012. Kilobot: A Low Cost Scalable Robot System for Collective Behaviors. *Proceedings - IEEE International Conference on Robotics and Automation* (May 2012), 3293–3298. https://doi.org/10.1109/ICRA.2012.6224638

[24] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. 2014. Programmable self-assembly in a thousand-robot swarm. *Science* 345, 6198 (Aug. 2014), 795–799. https://doi.org/10.1126/science.1254295

[25] Thomas Schmickl, Ronald Thenius, Christoph Moeslinger, Gerald Radspieler, Serge Kernbach, Marc Szymanski, and Karl Crailsheim. 2009. Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems* 18, 1 (2009), 133–155. Publisher: Springer.

[26] Onur Soysal, Erkin Bahçeci, and Erol Şahin. 2007. Aggregation in swarm robotic systems: Evolution and probabilistic control. *Turkish Journal of Electrical Engineering & Computer Sciences* 15, 2 (2007), 199–225. Publisher: The Scientific and Technological Research Council of Turkey.

[27] Vito Trianni, Roderich Groß, Thomas H Labella, Erol Şahin, and Marco Dorigo. 2003. Evolving aggregation behaviors in a swarm of robots. In *European Conference on Artificial Life*. Springer, 865–874.

[28] Gao Wang, Trung V Phan, Shengkai Li, Michael Wombacher, Junle Qu, Yan Peng, Guo Chen, Daniel I Goldman, Simon A Levin, Robert H Austin, and others. 2021. Emergent field-driven robot swarm states. *Physical review letters* 126, 10 (2021), 108002. Publisher: APS.

[29] Richard A. Watson, Sevan G. Ficici, and Jordan B. Pollack. 2002. Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems* 39, 1 (April 2002), 1–18. https://doi.org/10.1016/S0921-8890(02)00170-7 Publisher: Elsevier Science.

[30] Christoph A Weber, Timo Hanke, J Deseigne, S Léonard, Olivier Dauchot, Erwin Frey, and Hugues Chaté. 2013. Long-range ordering of vibrated polar disks. *Physical review letters* 110, 20 (2013), 208001.

[31] Matan Yah Ben Zion, Nicolas Bredeche, and Olivier Dauchot. 2021. Distributed on-line reinforcement learning in a swarm of sterically interacting robots. *arXiv preprint arXiv:2111.06953* (2021).

[32] Erol Şahin. 2004. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*. Springer, 10–20.