Coevolutionary Learning of Swarm Behaviors Without Metrics

Wei Li
Department of Automatic Control and Systems Engineering
The University of Sheffield, UK
wei.li11@sheffield.ac.uk

Melvin Gauci
Department of Automatic Control and Systems Engineering
The University of Sheffield, UK
m.gauci@sheffield.ac.uk

Roderich Groß
Department of Automatic Control and Systems Engineering
The University of Sheffield, UK
r.gross@sheffield.ac.uk

ABSTRACT

We propose a coevolutionary approach for learning the behavior of animals, or agents, in collective groups. The approach requires a replica that resembles the animal under investigation in terms of appearance and behavioral capabilities. It is able to identify the rules that govern the animals in an autonomous manner. A population of candidate models, to be executed on the replica, compete against a population of classifiers. The replica is mixed into the group of animals and all individuals are observed. The fitness of the classifiers depends solely on their ability to discriminate between the replica and the animals based on their motion over time. Conversely, the fitness of the models depends solely on their ability to 'trick' the classifiers into categorizing them as an animal. Our approach is metric-free in that it autonomously learns how to judge the resemblance of the models to the animals. It is shown in computer simulation that the system successfully learns the collective behaviors of aggregation and of object clustering. A quantitative analysis reveals that the evolved rules approximate those of the animals with a good precision.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.2.9 [**Artificial Intelligence**]: Robotics—*Autonomous vehicles*

Keywords

animal behavior; coevolution; swarm robotics; evolutionary robotics; system identification; artificial life

1. INTRODUCTION

The scientific study of animal behavior, or ethology [2], is pursued not only because it is a subject of interest in itself, but also because the knowledge gained from it has several practical applications. For instance, models of animal decision-making processes can be used to predict their behavior in novel environments, which can help in making ecological conservation policy [26]. Knowledge about animal behaviors has also been applied for solving computational problems [8], and for constructing biologically-inspired robotic agents [22]. Swarm behaviors are emergent behaviors that arise from the interactions of a number of animals in a group, whose individual behaviors tend to be relatively simple [5]. A wide variety of swarm behaviors can be observed in nature, such as the aggregation of cockroaches [14] and foraging by ants [6]. Learning about the behaviors exhibited in such animal collectives is particularly challenging, as the individuals do not act independently of each other, and the motion of each individual is hard to predict.

Recent developments in science automation have shown that machines can autonomously conduct scientific investigations [15, 25]. In this paper, we propose a method that allows a machine to infer the rules of interaction between a group of homogeneous animals in an autonomous manner. The approach requires a replica that resembles the animal under investigation in terms of appearance and behavioral capabilities. It is based on a coevolutionary algorithm that comprises two competitive populations: one of *models*, to be executed on the replica, and the other of *classifiers*. The fitness of the classifiers depends solely on their ability to discriminate between the replica and the animals based on their motion. Conversely, the fitness of the models depends solely on their ability to 'trick' the classifiers into categorizing them as an animal. The approach does not rely on a pre-defined metric for judging the resemblance of models to the animal. Rather, such metrics are implicitly defined by the classifiers, and hence incorporated into the evolutionary process.

There are a number of works on system identification using coevolutionary algorithms [17, 16, 20]. Bongard and Lipson [4] proposed the *estimation-exploration algorithm*, a nonlinear system identification method to coevolve tests and models in a way that minimizes the amount of tests that have to be carried out on the system. They applied this method to evolve the parameters of models that approximate the morphology of a mobile robot after it undergoes physical damage. In [3], they reported that "in many cases the simulated robot would exhibit wildly different behaviors even when it very closely approximated the damaged 'physical' robot. This result is not surprising due to the

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2576768.2598349.

fact that the robot is a highly coupled, non-linear system: thus similar initial conditions [...] are expected to rapidly diverge in behavior over time". Bongard and Lipson addressed this problem by using a more refined comparison metric reported in [3]. In [20], an alternative way of designing fitness functions based on Shannon information theory was used for model inference. All the works mentioned above rely on pre-defined metrics for gauging the difference between the target system and the model.

In experimental biology, researchers have used robots that interact with animals (e.g. ducks [27], cockroaches [12] and chicks [11]), in order to understand and model their behaviors. Robots can be created and systematically controlled in such a way that they are accepted as con- or heterospecifics by the animals in the group, which allows the researchers to learn the behavior of the animals [18, 24, 13]. We believe that this framework can be enhanced through the use of metric-free learning algorithms that infer the collective behaviors in an autonomous manner. In this paper, we present for the first time such an algorithm.

Li et al. [19] presented a coevolutionary approach for learning the behavior of a single animal that moved in a 1-D world. The system uses classifiers to distinguish between the models and the animal. Our system extends the framework in [19] to learn about the collective behavior of groups of animals. The classifiers decide whether an agent is an animal or the replica solely based on the motion data of this particular agent. We present two case studies in computer simulation showing that this information is sufficient to guide the learning of the collective behaviors of self-organized aggregation and object clustering.

This paper is organized as follows. Section 2 describes the animal simulation platform, the two swarm behaviors investigated in this paper as well as the implementation of the proposed coevolutionary algorithm. Section 3 discusses the results obtained. Section 4 concludes the paper.

2. METHODOLOGY

2.1 Animal Simulation Platform

We use the open-source library Enki [21]. Enki is able to model the kinematics and dynamics of the e-puck, which is a miniature, differential wheeled mobile robot [23]. The robot is a disk with a diameter of 7.4 cm and a height of 5.5 cm. The inter-wheel distance is 5.1 cm. The velocities of the left and right wheels along the ground can be set independently in $[-12.8, 12.8]$ cm/s. Enki generates noise for each wheel through multiplying its left and right speeds by random numbers in the range of $(0.95, 1.05)$ following uniform distribution.

The sensor of the robot is a line-of-sight sensor of unlimited range. At any given time, it returns one of a finite number of readings depending on what it is pointing towards. Note that it does not provide distance information. The sensor is implemented by projecting a line from the robot's front and checking whether it intersects with anything in the environment.

The length of the control cycle was set to 0.1 s, and the physics was updated at a rate of 10 times per control cycle. In the following, we refer to the simulated robots as 'animals' and 'replicas'.

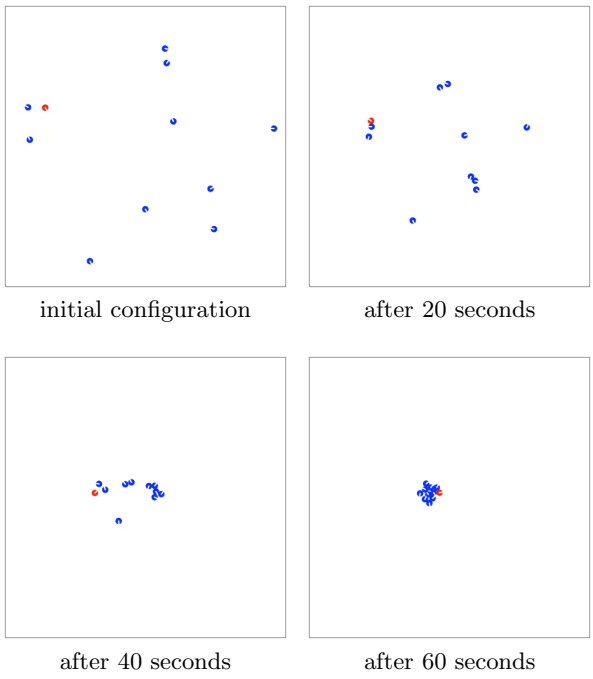


Figure 1: Snapshots of the aggregation behavior. A replica executing a model (red) is mixed into a group of ten animals (blue). All agents are observed by a population of classifiers, which co-evolve with the models. These classifiers attempt to tell apart the animals from the model.

2.2 Aggregation Behavior

2.2.1 *Rule of Aggregation Behavior*

We use the aggregation behavior reported in [10]. In this behavior, the line-of-sight sensor is binary: it gives a reading of $I = 1$ if there is an agent in the line of sight, and a reading of $I = 0$ otherwise. The environment is homogeneous, 2-D, and free of obstacles. The objective of the animals is to aggregate into a single, compact cluster as quick as possible. This behavior was validated with a swarm of 40 physical e-puck robots in [10].

Each animal decides how to move based on whether or not it can see another agent in its line of sight. Its rule is a mapping from each of the two possible sensor readings, $I = 0$ and $I = 1$, onto a pair of pre-defined velocities for the left and right wheels; it is an *if-then-else* rule. Let $\bar{v}_l, \bar{v}_r \in [-1, 1]$ represent the normalized left and right wheel velocities, respectively, where -1 (1) corresponds to the wheel rotating backwards (forwards) with maximum speed. The controller for the animal can be written as a quadruple,

$$\mathbf{p} = (\bar{v}_{l0}, \bar{v}_{r0}, \bar{v}_{l1}, \bar{v}_{r1}) \,, \; \mathbf{p} \in [-1, 1]^4 \,, \qquad (1)$$

where $\bar{v}_{l0}$ refers to the normalized velocity of the left wheel when $I = 0$, and so on.

The optimal (memoryless) controller for aggregation was found by performing a grid search over the entire space of possible controllers within a finite resolution [10]. The resolution used was 21 settings per parameter, with each pa-

rameter taking values in the set:

$$\mathbf{s} = \{-1, -0.9, \ldots, -0.1, 0, 0.1, \ldots, 0.9, 1\}\,. \qquad (2)$$

Therefore, $21^4 = 194481$ combinations were evaluated. The parameters of the optimal controller were given by

$$\mathbf{p}' = (-0.7, -1, 1, -1)\,. \qquad (3)$$

Now, we can describe the behavior of the animal as follows. When $I = 0$, it moves backwards along a clockwise circular trajectory. When $I = 1$, it rotates clockwise on the spot, with the maximum possible angular velocity. When employing this controller, a single robot provably aggregates with another robot, or cluster of robots [10].

2.2.2 *Experimental Setup*

In our setup, we use 11 agents (consisting of 10 animals and 1 replica that executes a model). They are initially placed in arbitrary positions, and facing in arbitrary directions in a (virtual) square of size 331.66 cm, such that the area per agent is, on average, 10000 cm^2. Fig. 1 shows snapshots of an example trial.

2.3 Object Clustering Behavior

2.3.1 *Rule of Object Clustering Behavior*

We use the object clustering behavior reported in [9]. The sensor of the animals used here is an extension of that in the aggregation behavior. Each animal has a line-of-sight sensor I, which indicates to the animal what it is pointing at: $I = 0$ if it is pointing at nothing (or the walls of the environment, if this is bounded), $I = 1$ if it is pointing at an object, and $I = 2$ if it is pointing at another agent. The aim of the animals is to push these objects into a single cluster as fast as possible. The behavior was validated using 5 physical e-puck robots and 20 cylindrical objects in [9].

The rule of the object clustering behavior is represented as a mapping from each of the three possible values onto a pair of velocities for the two wheels. The controller can thus be represented as a six-tuple:

$$\mathbf{q} = (\bar{v}_{l,0}, \bar{v}_{r,0}, \bar{v}_{l,1}, \bar{v}_{r,1}, \bar{v}_{l,2}, \bar{v}_{r,2})\,, \quad \mathbf{q} \in [-1, 1]^6\,, \qquad (4)$$

where $\bar{v}_{l,0}$ denotes the normalized angular velocity of the left wheel when $I = 0$, etc. The controller was found using an evolutionary algorithm [9], and was given by:

$$\mathbf{q}' = (0.5, 1.0, 1.0, 0.5, 0.1, 0.5)\,. \qquad (5)$$

The behavior can be described as follows. When $I = 0$ and $I = 2$, the animal moves forward along an anti-clockwise circular trajectory, but with different linear and angular speeds. When $I = 1$, it moves forward along a clockwise circular trajectory.

2.3.2 *Experimental Setup*

In our setup, we use 10 cylindrical objects and 5 agents (consisting of 4 animals and 1 replica that executes a model). They are randomly placed in a (virtual) square of size 100 cm, so that the area per object is, on average, 1000 cm^2. Fig. 2 shows snapshots of an example trial.

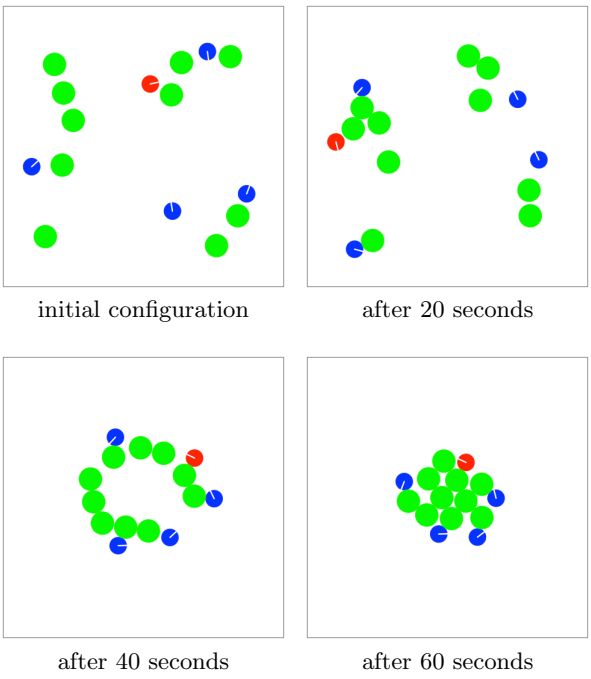


Figure 2: Snapshots of the object clustering behavior. There are 4 animals (blue), 1 replica executing a model (red), and 10 objects (green) to be pushed by them.

2.4 Coevolutionary Algorithm and Implementation

The coevolutionary algorithm is comprised of two populations: one of models, and one of classifiers, which coevolve with each other competitively.

The fitness of the classifiers depends solely on their ability to distinguish the behavior of the model from the behavior of the animals. The fitness of the models depends solely on their ability to mislead the classifiers into making the wrong judgment, that is, classifying them as an animal.

2.4.1 *Model Structure*

The model is executed on the replica, which resembles the animal under investigation in term of appearance and behavioral abilities. In particular, we assume that the replica has a disk body, two wheels and a line-of-sight sensor and that these are identical to the ones for the animal. The task is thus to estimate the parameters ($\mathbf{p}'$ and $\mathbf{q}'$), in the two behaviors. Note that different from Eqs. 1 and 4, the search space is unbounded. The fitness of the models depends solely on the performance of the classifiers, and that the latter do not have any knowledge about the model structure and parameters.

2.4.2 *Classifier Structure*

The structure of the classifiers is a recurrent Elman neural network [7] with two inputs, five hidden neurons, and one output neuron. Each neuron of the hidden and output layers has a bias. The network has a total of 46 parameters, which all assume values in $\mathbb{R}$. The activation function used in the hidden and the output neurons is the logistic

sigmoid function, which has the range $(0, 1)$ and is defined as $\text{sig}(x) = 1/(1 + \exp(-(x)))\ \forall x \in \mathbb{R}$.

One of the inputs to the network is the linear speed (v) of the agent, and the other input is its angular speed (ω). In order to make the setup more feasible to implement on a physical system, it is assumed that the system cannot directly measure the linear and angular speeds of the agent, but rather its position and orientation at time t. The absolute value of the animal's linear speed is obtained by calculating the relative distance of the previous measured position and the current measured position, and then dividing the resulting number by the time interval between two measurements. When the angle between the animal's orientation and its direction of motion is smaller than 90 degrees (i.e. the animal is moving forward), we define the sign of the linear speed to be positive; otherwise, the linear speed is negative. The angular speed is calculated by subtracting the previous measured orientation from the current orientation, and dividing the resulting number by the time interval between two measurements.

In order to make a judgment between a model and an animal, the classifier observes the behavior (motion) of the agent over a period of time—here 10 s at 0.1 s intervals for a total of $T = 100$ time steps. After having iterated through all the time steps (a single trial), the final value of output neuron O is used to make a judgment: the classifier decides on a model if $O < 0.5$, and on an animal if $O \geq 0.5$. The memory of the classifier is reset after making a judgment. All the classifiers use the same data for each trial to make their own judgments. The initial positions of the animals and the replica are randomly generated in each trial.

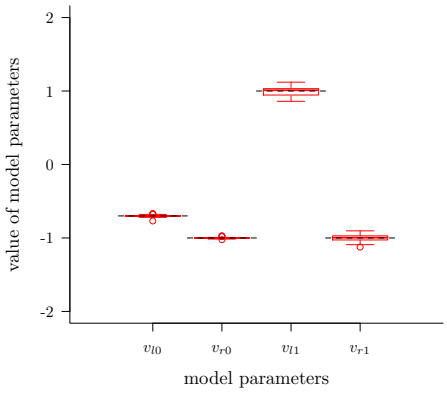
2.4.3 *Optimization Algorithm*

The algorithm used here is based on a $(\mu + \lambda)$ evolution strategy with self-adaptive mutation strengths [1], and can be thought of as consisting of two sub-algorithms: one for the models, and another for the classifiers. The populations are treated identically, and they do not interact with each other except for the fitness calculation step (described in Sec. 2.4.4). The details of the implementation of the evolutionary algorithm can be found in [19].

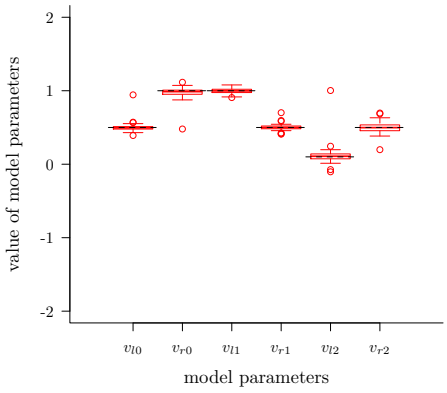
2.4.4 *Fitness Calculation*

The populations in the coevolution comprise of 100 models and 100 classifiers. The fitness of each model is obtained by evaluating it with each of the classifiers in the competing population (100 in total). For every classifier that wrongly judges the model as being the animal, the model's fitness increases by 1. Therefore, the fitness takes a value in $\{0, 1, 2, \ldots, 100\}$. The fitness value is then normalized to the interval $[0, 1]$.

The fitness calculation of the classifiers depends on the behavior investigated. For the aggregation behavior, in each trial, the fitness of each classifier is obtained by using it to evaluate 1 model and 10 animals. For the correct judgment of the model, the classifier's fitness increases by 0.5; for each correct judgment of the animal, the classifier's fitness increases by 0.05. Therefore, the fitness of each classifier in each trial is in the set $\{0, 0.05, 0.1, \ldots, 1\}$. This evaluation is performed 100 times, and the fitness value of each classifier is then normalized to the interval $[0, 1]$. For the object clustering behavior, the method of calculation is analogous.



(a) Aggregation



(b) Object Clustering

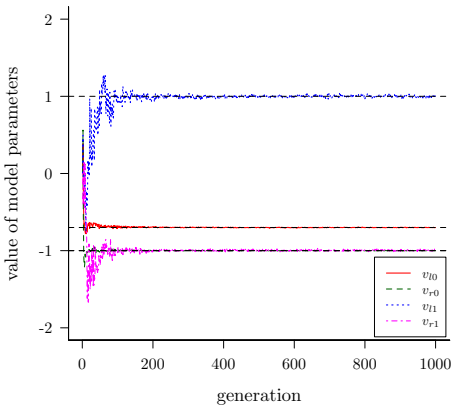
Figure 3: This box-and-whisker plot shows the parameters of the evolved models with the highest subjective fitness in the 1000^{th} generation in the coevolutions for (a) the aggregation behavior and (b) the object clustering behavior. Each box corresponds to 30 coevolutionary runs. The dotted lines correspond to the values of the parameters that the system is expected to learn (i.e. those of the animal).

3. RESULTS

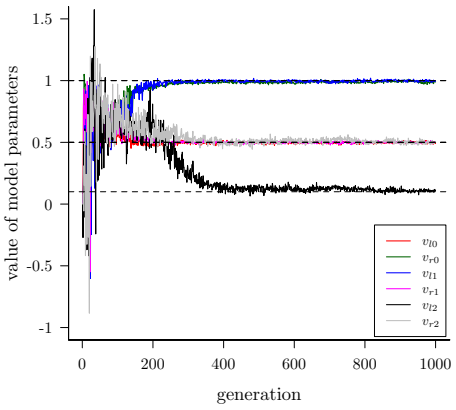
3.1 Analysis of the Evolved Models

We performed 30 coevolutionary runs for each of the two setups described in Sec. 2. Each run lasted for 1000 generations. Fig. 3 shows a box plot[1] with the parameters of the evolved models with the highest subjective fitness in the final generation. As can be seen, the system identified the

[1]The box plots presented here are all as follows. The line inside the box represents the median of the data. The edges of the box represent the lower and the upper quartiles (25-th and 75-th percentiles) of the data, whereas the whiskers represent the lowest and the highest data points that are within 1.5 times the inter-quartile range from the lower and the upper quartiles, respectively. Circles represent outliers.

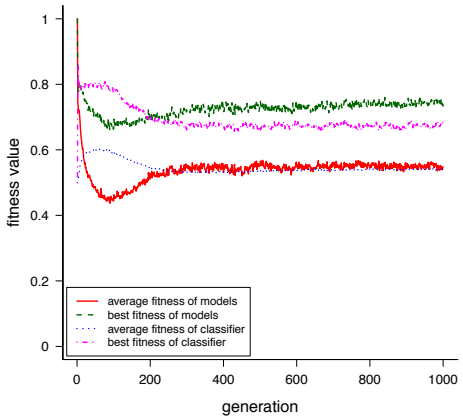


(a) Aggregation

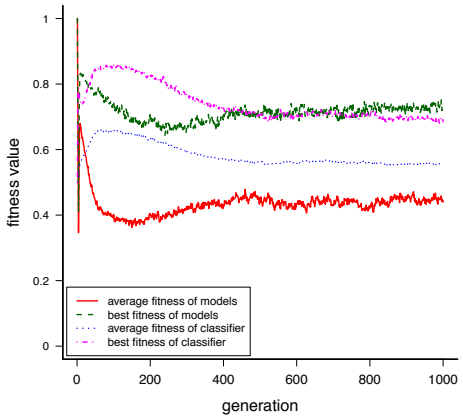


(b) Object Clustering

Figure 4: This plot shows how the model parameters evolve during the course of coevolution for (a) the aggregation behavior and (b) the object clustering behavior. The curves correspond to the median values from 30 coevolutionary runs.



(a) Aggregation



(b) Clustering Objects

Figure 5: This plot shows the subjective fitness of the classifiers and the models for (a) the aggregation behavior and (b) the clustering objects behavior. The curves show the average fitness across 30 coevolutionary runs.

parameters of these two behaviors with a good accuracy. For the case of aggregation, the relative error of the median of the four evolved parameters with respect to those of the animal is 0.23%, 0.13%, 1.03%, and 0.37%, respectively. For the case of object clustering, the relative error of each evolved parameters is 1.13%, 1.18%, 0.24%, 0.04%, 8.41%, and 0.36%, respectively.

We also investigate how the model parameters are learned during the course of coevolution for the two behaviors. Fig. 4 shows the convergence of the parameters over generations. In the aggregation behavior, the rule corresponding to $I = 0$ is first to be learned, as after about the 60^{th} generation, both v_{l0} and v_{r0} closely approximated their true values (-0.7 and -1.0). For $I = 1$, it took about 140 generations for v_{l1} and v_{r1} to converge. This is due to the fact that the chance of the agents not seeing each other is higher than that of the agents seeing each other.

In the object clustering behavior, the parameters which correspond to the cases of $I = 0$ and $I = 1$ are learned faster than the other two parameters, as shown in the Fig. 4(b). After about 200 generations, v_{l0}, v_{r0}, v_{l1} and v_{r1} started to converge; however it took about 360 generations for v_{l2} and v_{r2} to approximate their true values. This phenomenon can be explained as follows. Since there are fewer robots than objects, the chance of the agents seeing each other is lower than that of the agents seeing objects or nothing.

A video showing both the learning process and a swarm of replicas (executing an evolved model of the final generation) is available in the supplementary material.

3.2 Coevolutionary Dynamics

In order to analyze how the classifiers and the models interact with each other during the course of the coevolution, we investigate the dynamics of the subjective fitness of the classifiers and the models as shown in Fig. 5.

In the aggregation behavior, at the beginning, the fitness of the classifiers is 0.5 as they output 1 for all the animals and models, which means the classifiers make uninformed decisions[2]. Therefore, the fitness of the models starts from 1.0, as all the classifiers judge them as the animal. Then, the fitness of the classifiers quickly increases, corresponding to the decline of model fitness. As the models learn to adapt, the fitness of the classifiers only increases slightly until about the 100$^{\text{th}}$ generation. After that, the fitness of the models starts to increase. The fitness of the models surpasses that of the classifiers after about the 200$^{\text{th}}$ generation; at this point, the fitness of the best models is around 0.7. This means that the models are able to mislead 70% of the classifiers into judging it as the animal. From the 200$^{\text{th}}$ generation onwards, the fitness of the classifiers and models remains "balanced" until the last generation.

The coevolutionary dynamics of the object clustering behavior is similar. However, the average fitness of classifiers remains higher than that of the models after the initial peak. This could be explained by the higher number of parameters and the higher complexity of the behavior to be evolved.

3.3 Post-Evaluation of the Evolved Classifiers

For the sake of simplicity, in the following sections, we only analyze the aggregation behavior. We analyze the best evolved classifiers from the last generation of each coevolutionary run; therefore, there are 30 classifiers to be evaluated in total. We evaluate the performance of each classifier when the parameters of the animals are disturbed. Here, we change one parameter at a time while keeping the other three parameters identical to those of the animal. We consider the parameter values shown in Eq. 2. Therefore, there are $21 \cdot 4 = 84$ combinations to be tested by each classifier. For each combination, we performed 100 trials and calculated the average fitness of the classifier.

In order to evaluate the performance of the classifiers, we defined the following metric:

$$W = \frac{1}{\Omega} \sum_{j=1}^{4} \sum_{\substack{i=1, \\ x_i \neq p_j}}^{20} \frac{\bar{f}_{ij}}{|x_i - p_j|}, \tag{6}$$

where, p_j is one of the parameters of the animal from Eq. 3, x_i is one of the parameter values from Eq. 2, and $\bar{f}_{ij}$ is the average fitness of the evaluated classifier over 100 trials where we assign a value x_i to the parameter p_j. $\Omega = 160.66$ is the maximum value that the double sum can achieve (i.e. if all the $\bar{f}_{ij}$'s are equal to 1). This metric penalizes small values of $\bar{f}_{ij}$ for each combination, and gives more emphasis to small distances of $|x_i - p_j|$. As all the classifiers have a low fitness value in the case of $x_i = p_j$, this is not considered in the metric.

The best classifier according to the metric (Eq. 6) had a value of $W = 0.81$. The average fitness of this classifier over 100 trials per combination is shown in Fig. 6. The classifier is most sensitive to changes in v_{r1}, followed by v_{r0}, v_{l0} and v_{l1}. There is a sudden jump when v_{r0} and v_{r1} deviate from the target value (of the animal), suggesting that the classifier can discriminate very well between the animal and models

[2]Note that in the implementation of the coevolutionary algorithm, the parameters of the models and classifiers are initialized to 0, which means all the classifiers are identical at the 1st generation.

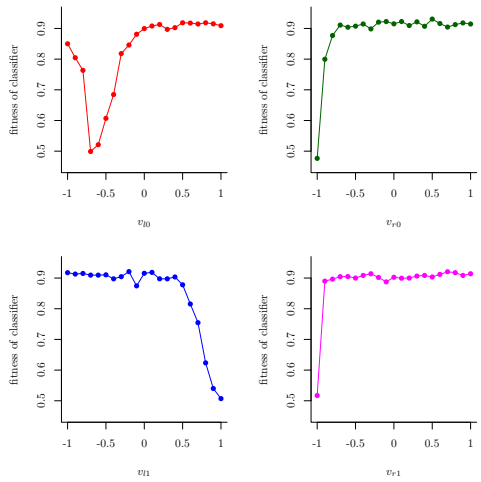


Figure 6: This plot shows the average fitness of the best classifier when the parameters of the animal are disturbed in the aggregation behavior. There is only one parameter changing at a time. See the text for details.

that do not approximate particular well these parameters. For the disturbance of the left wheel speed, v_{l0} and v_{l1}, the fitness curve of the classifier changes more smoothly. In all the four cases, when the parameter is close to its target value (i.e. the replica is behaving like the animals), the classifier has a fitness value of around 0.5. We have verified from the data that it judges almost every agent as an animal, which is correct when the model is *exactly* like the animal.

We also analyze the scalability of the best classifier with respect to the number of agents in the mixed group. Fig. 7 shows the average fitness of the best classifier over 100 trials when changing the number of animals. In each trial, the model parameters for a single replica are randomly generated in $[-1.0, 1.0]$. The average fitness of the classifier is not affected by the variation of the number of animals except for the cases that there are few animals in the group. For instance, in the case of 1 animal and 1 model, the fitness of the classifier declined to 0.82. Since there are only two agents in the environment, they do not interact with each other too often to reveal their full behavior. This influences the judgment of the classifier.

3.4 Noise Study

We conducted a study to investigate how the performance of the system is affected by noise on the measurements of the agents' position and orientation. As the e-puck robot has a maximum linear speed of 12.8 cm/s, the maximum distance that it can travel in one control cycle (0.1 s) is 1.28 cm. For this reason, we define a disturbance of 1.28 cm to an agent's position as a measurement error of 100%. Similarly, we define a 100% measurement error on the agent's orientation as the maximum change in orientation that an e-puck can undergo in one control cycle. This corresponds to 0.5 rad.

We conducted 5 sets of 30 coevolutions for the noise values $M = \{0, 25, 50, 75, 100\}$%. In a coevolution with a noise value M, every measurement of an agent's position is perturbed in a random direction and by a random distance chosen uniformly in $\left[0, 1.28\frac{M}{100}\right]$ cm. Similarly, every orien-

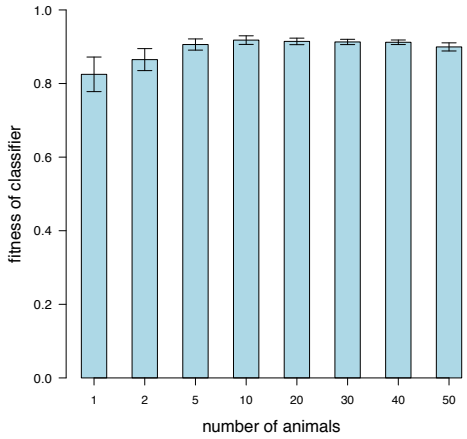


Figure 7: This plot shows the average fitness of the best classifier for different numbers of animals in the aggregation behavior. In each trial, a single replica was present and the model parameters were randomly generated in $[-1.0, 1.0]$.

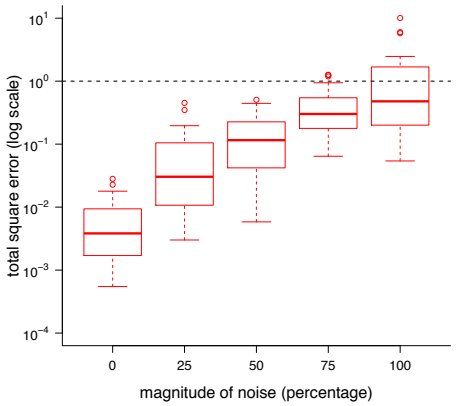


Figure 8: This plot shows the total square error of the evolved parameters when increasing the percentage of noise on the measurements of the agents' position and orientation in the aggregation behavior. Each box corresponds to 30 coevolutionary runs. See the text for details.

tation measurement is perturbed by a random angle chosen uniformly in $\left[-0.5\frac{M}{100}, 0.5\frac{M}{100}\right]$.

Fig. 8 shows the total square error of the evolved parameters in the final generations of the coevolutions. This plot reveals that the system still performs relatively well when a considerable amount of noise affects the agents' motion tracking system.

3.5 Scalability Study

We performed a further study on the aggregation behavior with 50 animals and 1 replica in the environment. However, instead of observing the trajectories of all the agents, the

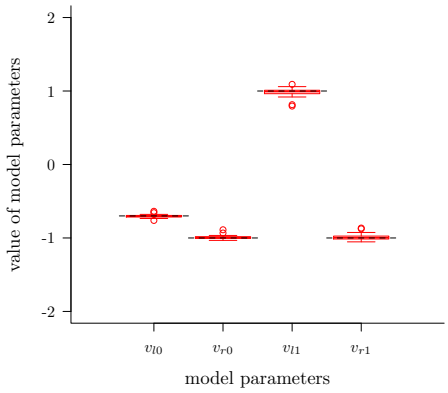


Figure 9: This plot shows the scalability of the coevolutionary approach. There are 1 replica and 50 animals in the group. The classifiers only observe two trajectories: one from the replica, the other from a randomly chosen animal.

classifiers only observed the trajectory of the replica and of a single, randomly-chosen animal in each trial. Fig. 9 shows the parameters of the evolved models with the highest subjective fitness in the 1000^{th} generation over 30 coevolutionary runs. All the four parameters are well approximated. The relative error of the median of the four evolved parameters with respect to those of the animal is 0.44%, 0.43%, 0.69%, and 0.38%, respectively.

4. CONCLUSIONS

This paper has presented a coevolutionary method that can autonomously learn the behavior of a group of animals, or other agents through observation. In principle, our approach does not require any prior information about the behavior. It does not rely on any pre-defined metric to gauge the difference between the animal and the model. Rather, such metrics are implicitly defined by the classifiers themselves, and hence incorporated into the evolutionary process. This eliminates the biasing effect that pre-defined metrics can have on the solutions obtained.

Using this approach, the system successfully learned two swarm behaviors (self-organized aggregation and object clustering) with a good accuracy in the model parameters obtained. We also analyzed the performance of the best classifier in the aggregation behavior, and found that it could discriminate between models and animals based on a relatively small difference in only a single parameter. When changing the number of animals in the environment, the judgment of the classifier was only affected if the number of agents was very low, which is not typical in a swarm behavior. The method proved robust with respect to noise in the motion tracking system.

The results suggest that what constitutes collective behaviors can be fully characterised by the effects that the interactions have on the individual in the group. In other words, rather than analyzing how a group of animals interact with each other, it is sufficient to observe the trajectory of a single individual. The major advantage of this approach is

scalability. In principle, our learning technique would only have to processes a constant amount of information (e.g. data from 1 animal and 1 replica, as shown in this paper), regardless of the number of animals in the group.

In the future, we intend to perform experiments using physical robots and real animals, and apply our approach to other more complex social behaviors.

References

[1] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer, Berlin, Heidelberg, Germany, 2001.

[2] J. J. Bolhuis and L.-A. Giraldeau. *The Behavior of Animals: Mechanisms, Function, and Evolution*. Wiley, Hoboken, NJ, 2004.

[3] J. Bongard and H. Lipson. Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials. In *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware*, pages 169–176, Seattle, WA, June 2004.

[4] J. Bongard and H. Lipson. Nonlinear system identification using coevolution of models and tests. *IEEE Transactions on Evolutionary Computation*, 9(4):361–384, 2005.

[5] S. Camazine et al. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, 2001.

[6] C. R. Carroll and D. H. Janzen. Ecology of foraging by ants. *Annual Review of Ecology and Systematics*, 4:231–257, 1973.

[7] J. L. Elman. Finding structure in time. *Cognitive Sci.*, 14(2):179–211, 1990.

[8] D. Floreano and C. Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. MIT Press, Cambridge, MA, 2008.

[9] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Clustering objects with robots that do not compute. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, IFAAMAS, Richland, SC. In press.

[10] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Self-organized aggregation without computation. *International Journal of Robotics Research*. In press.

[11] A. Gribovskiy, J. Halloy, J.-L. Deneubourg, H. Bleuler, and F. Mondada. Towards mixed societies of chickens and robots. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4722–4728, Boston, MA, October 2010.

[12] J. Halloy et al. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853):1155–1158, 2007.

[13] J. Halloy, F. Mondada, S. Kernbach, and T. Schmickl. Towards bio-hybrid systems made of social animals and robots. In *Biomimetic and Biohybrid Systems*, volume 8064 of *Lecture Notes in Computer Science*, pages 384–386. Springer, Berlin, Heidelberg, Germany, 2013.

[14] R. Jeanson et al. Self-organized aggregation in cockroaches. *Animal Behaviour*, 69(1):169–180, 2005.

[15] R. D. King et al. The automation of science. *Science*, 324(5923):85–89, 2009.

[16] S. Koos, J. Mouret, and S. Doncieux. Automatic system identification based on coevolution of models and tests. In *Proceedings of 2009 IEEE Congress on Evolutionary Computation*, pages 560–567, Trondheim, Norway, May 2009.

[17] B. Kouchmeshky, W. Aquino, J. Bongard, and H. Lipson. Co-evolutionary algorithm for structural damage identification using minimal physical testing. *International Journal for Numerical Methods in Engeering*, 69(5):1085–1107, 2007.

[18] J. Krause, A. F. Winfield, and J.-L. Deneubourg. Interactive robots in experimental biology. *Trends in Ecology and Evolution*, 26(7):369–375, 2011.

[19] W. Li, M. Gauci, and R. Groß. A coevolutionary approach to learn animal behavior through controlled interaction. In *Proceedings of the 15th Annual Conference of Genetic and Evolutionary Computation*, pages 223–230, Amsterdam, Netherlands, July 2013.

[20] D. Ly and H. Lipson. Optimal experiment design for coevolutionary active learning. *IEEE Transactions on Evolutionary Computation*, PP(99):1–11, 2013.

[21] S. Magnenat, M. Waibel, and A. Beyeler. Enki: The fast 2D robot simulator, 2011.

[22] J.-A. Meyer and A. Guillot. Biologically inspired robots. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, Springer Handbooks, pages 1395–1422. Springer, Berlin, Heidelberg, Germany, 2008.

[23] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Canci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65, 2009.

[24] T. Schmickl et al. Assisi: Mixing animals with robots in a hybrid society. In *Biomimetic and Biohybrid Systems*, volume 8064 of *Lecture Notes in Computer Science*, pages 441–443. Springer, Berlin, Heidelberg, Germany, 2013.

[25] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[26] W. J. Sutherland. The importance of behavioural studies in conservation biology. *Animal Behaviour*, 56(4):801–809, 1998.

[27] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron. Experiments in automatic flock control. *Robotics and Autonomous Systems*, 31(1):109–117, 2000.