Swarm and Evolutionary Computation 57 (2020) 100715

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Autonomous detection of collective behaviours in swarms

Md Mohiuddin Khan *, Kathryn Kasmarik, Michael Barlow

UNSW Canberra, Canberra, Australia

ARTICLE INFO

Keywords:
Collective behaviour
Artificial swarming
Evolutionary framework
Boids model
Computational value systems

ABSTRACT

Collective behaviours such as swarm formations of autonomous agents offer the advantages of efficient movement, redundancy, and potential for human guidance of a single swarm organism. This paper proposes a developmental approach to evolving collective behaviours whereby the evolutionary process is guided by a novel value system. A self-organising map is used at the core of this value system and motion properties of the swarm entities are used as input. Unlike traditional approaches, this value system does not need in advance the precise characteristics of the intended behaviours. We examine the performance of this value system in a series of controlled experiments. Our results demonstrate that the value system can recognise multiple "interesting" structured collective behaviours and distinguish them from random movement patterns. Results show that our value system is most effective distinguishing structured behaviours from random behaviours when using motion properties of individual agents as input. Further variations and modifications to input data such as normalisation and aggregation were also investigated, and it was shown that certain configurations provide better results in distinguishing collective behaviours from random ones.

1. Introduction

Swarm intelligence is inspired by behaviours occurring in insects, birds, fish and other organisms in nature. These biological systems have been there for millennia, and they have been efficiently solving complex problems through apparently simple rules. The current approach to swarm intelligence is a culmination of observations and findings from both biologists and computer scientists [1], contributing towards analysis of the behaviours in nature and applying them in artificial systems.

In artificial systems, swarm formations offer the advantages of efficient movement, redundancy, and potential for human guidance of the single swarm organism [2]. As a result, the evolution of collective behaviours in artificial swarms is currently receiving attention. This includes applying factors such as novelty search [3] and minimizing surprise [4]. Underlying factors such as the social characteristics of the evolved swarms were analysed [5] and evolutionary techniques to automatically generate rules for emergent behaviours from their atomic components were presented [6]. Evolutionary approaches have an advantage over reactive or rule-based approaches to swarming, as they can generate solutions for which rules are difficult to hand-craft [4]. However, in evolutionary approaches, typically the behaviour is guided towards a fixed or 'static' behavioural objective [3,7]. The experimenter designs a fitness function that estimates the quality of candidate solutions with respect to a give objective, and this fitness function is used to 'score' the behaviour of individuals in the population.

Our focus on this paper is to propose a value system that can work as the 'fitness function' in the context of an evolutionary approach but will not be limited to static fitness. Rather it will be able to guide the swarm towards interesting behaviours automatically. Our motivation behind designing such a system comes from the fact that such a value system can eventually be used to create a swarm of artificial agents that can automatically detect the status of their behaviour and evolve towards more meaningful, collective behaviours.

This paper proposes a developmental approach to evolving collective behaviours whereby the evolutionary process is guided by a fitness function that can recognise multiple "interesting" structured collective behaviours, without the need to specify in advance the precise characteristics of those behaviours. We first present a developmental evolutionary framework for swarm robotics. We then propose a value system that can recognise multiple interesting collective behaviours and distinguish them from random movements. This value system is based on Self-Organising Map (SOM), as it utilises the movement patterns of the swarms to generate metrics representing behaviour patterns. We examine the performance of this value system in a series of controlled experiments to establish the most effective inputs and the kinds of behaviours it can distinguish. Results show that our value function is most effective

* Corresponding author.
E-mail address: mohiuddin.khan@adfa.edu.au (M.M. Khan).

https://doi.org/10.1016/j.swevo.2020.100715
Received 15 October 2019; Received in revised form 6 February 2020; Accepted 13 May 2020
Available online 18 May 2020

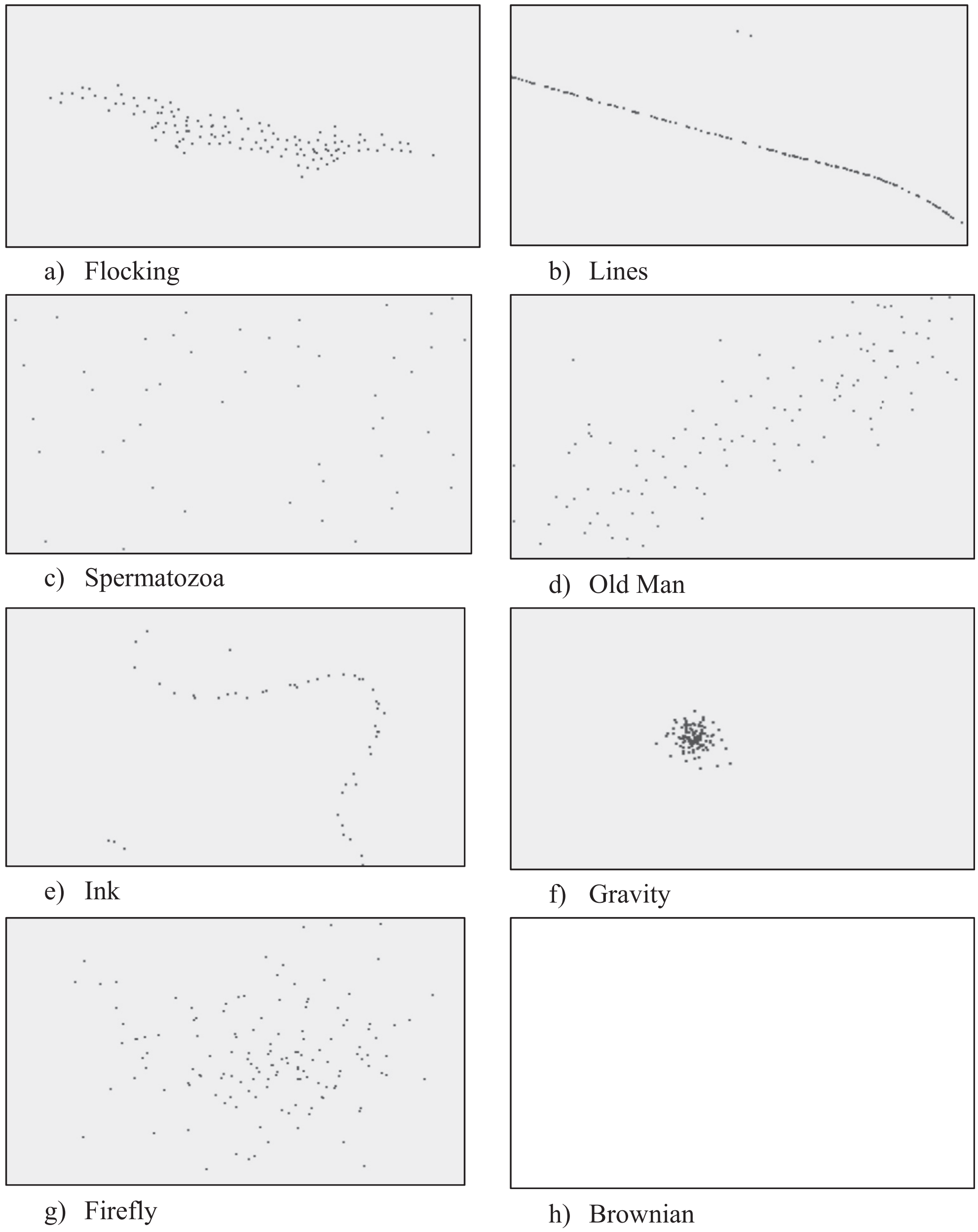


Fig. 1. 'Snapshots' of the artificial swarms exhibiting visually different behaviours. Videos of each of these are available in the supplemental material with this paper.

distinguishing structured behaviours from random behaviours when using motion properties of individual agents as input.

The contributions of this paper are:

1. An evolutionary developmental framework designed for sharing value across multiple agents.
2. A value system capable of identification of collective behaviours
3. Recommendations on the most effective inputs and configuration decisions for the value systems, supported by empirical evidence.

The remainder of this paper is organised as follows: Section 2 overviews literature relevant to this work in the areas of swarm intelligence, computational value systems and collective behaviour detection. Section 3 introduces our broad framework for embodied evolution using a value system. It then introduces the specific value system we propose in this paper. Section 4 conducts a series of experiments to establish the best parameters and settings to use in this value system in a swarm setting. Section 5 concludes the paper and examines directions for future work.

2. Background and related work

The work in this paper brings together two bodies of literature from swarm intelligence and from the study of computational value systems. In this paper, swarms form the underlying system for group motion. We thus review this literature in Section 2.1. The literature on computational value systems reviewed in Section 2.2 informs our approach to valuing "interesting" collective behaviour exhibited by the swarm, including an introduction to SOMs. Section 2.3 reviews the works most closely related to the models in this paper.

2.1. Artificial swarms

The term "Swarm intelligence" was first coined by Beni and Wang [8]. Defining 'intelligence' is a difficult proposition. As explained in Ref. [9], a machine can be defined intelligent if it demonstrates behaviour which is "neither random nor predictable". Following this definition, an intelligent swarm was defined as a group of non-intelligent agents that can collectively demonstrate intelligent behaviours. Swarm intelligence is inspired by the collective behaviours demonstrated in nature and translated to artificial swarms in various physical and abstract models, algorithms and frameworks.

Computer-based swarm systems replicating the flocking behaviour of birds, were first introduced by Reynolds [10]. Reynolds' "boids", short for "bird androids," were based on the emergent behaviour resulting from the interaction of three simple rules: cohesion, alignment and separation. Later extensions of these models called boid guidance algorithms (BGAs) [11] have been designed for unmanned ground and aerial vehicles. These models include further rules to ensure vehicles stay within the boundaries of their operating conditions, such as speed and turn rate.

While the dynamics of boid and BGA behaviour is recognisable as a representation of the behaviour of a natural flock [10,12], the behaviour is difficult to define or quantify accurately. Toner and Tu [13] define flocking as "*the collective, coherent motion of large numbers of self-propelled organisms*". Flocking itself is one of many terms used to describe a form of behaviour referred to as collective motion [14]. Harvey et al. [12] note that collective motion includes the behaviours known as flocking, swarming, schooling, shoaling, and herding. The term flocking generally applies to birds, shoaling and schooling to fish, and herding to land animals. The term "swarming" is applied in a broader sense, to collections of elements other than just animals, including unmanned aerial vehicles and ground robots. Fig. 1 show snapshots of a range of different formations that could be attributed as 'life-like' swarming. Videos of each of these are also available in the supplemental material with this paper.

There have been extensive studies and surveys [14] of many examples of swarming behaviour. Das et al. [15], for example, have demonstrated chaotic system dynamics in a social foraging swarm model. They further identified the range of parameters for which chaos exists in the dynamics. Mecholsky et al. [16] investigated a broad range of control parameters to determine the stability of models of flocks. Harvey et al. [17] applied measures used to characterize chaotic systems to a simplified version of Reynolds' boid model. They were able to identify the range of parameter values for which one particular type of swarming behaviour would occur. This paper seeks to extend this work to identify parameter values autonomously that result in swarming behaviour with various characteristics, such as those shown in the illustrations in Fig. 1.

Our proposed approach will use an artificial value system embedded in an evolutionary developmental framework. While evolutionary swarm robotics is a topic of current research [3–6], existing work has most commonly focused on fitness functions with a fixed or "static" objective [5]. A subset of work has examined the use of artificial value systems such as novelty [3] and surprise [4] to augment this static fitness function. The novelty and surprise models push the system to continuously seek new ways to meet the static objective. However, much of this work has been based on the use of global information about the behavioural characteristics of swarms to form the evolutionary phenotype. This includes embedding measures of agent cohesion and alignment in the fitness function, which require global level data to calculate. This paper poses a new approach to fitness generation: to value behaviour from local level experience data only. We will do this by drawing on the literature of computational value systems introduced in the next section.

2.2. Computational value systems

An artificial value system for autonomous agents is a mechanism that can reflect the effect of the past experience in future behaviours [18]. The word 'value' has been used loosely to refer to a range of ways in which this can be done, including to concepts such as expected reward [19–21], motivation value [19,22,23] and activation potential [24–27]. It is generally agreed that the notion of value can be incorporated into different types of computational systems [28] to meet different needs. This includes the need to seek novel solutions to problems [19,20], the need for knowledge [21], learning progress [29,30] and competency (skill) development, survival and self-preservation [31].

Researchers have posed a number of properties for artificial value systems [18,32]. In particular, it is proposed that they must be predictive, task non-specific, and unsupervised, making them ideal for open-ended learning. In the context of an approach to designing a value system for the evolution of collective behaviour in swarms, this implies three goals for the value system:

- *Predictive*: The value system should strive to be able to infer the consequences of its own action before actually executing it;
- *Task non-specific*: This means that the system should be transferable between one environment and another; or between robots with different physical forms; and
- *Unsupervised*: The value system should enable robots to learn by themselves from the consequence of their actions without help from a human instructor.

A large class of existing computational value systems achieve these goals by drawing on theoretical foundations in motivation psychology. Computational models of motivation have been separated in categories such as knowledge and competence-based models [33,34]. Examples of knowledge-based models include models of novelty [19,20], and curiosity [20,35–38]. These models incorporate prediction mechanisms and interleave periods of seeking predictability with seeking new experiences. This permits switching between periods of unsupervised learning and exploration to seek new tasks to learn. In contrast, competence-based models [29,30] are based on measures of learning progress and achievement/mastery of a skill. While competence based models have shown success in reinforcement learning systems in individual robots [23], knowledge based models such as curiosity and novelty have shown promise in evolutionary systems. We propose to use such models as the starting point for designing a value system for swarming behaviour in this paper.

We have used Self-organising Map (SOM) as the value system in this paper. A SOM, proposed by Kohonen [39,40], is an artificial neural network that has been extensively used as an automatic data-analysis method. SOMs are used to perform clustering and visualization in exploratory data analysis. SOMs can be effectively used to transform complex, high dimensional input data into low dimensional outputs while preserving the relationships in the data. Structurally, SOMs are consisting of input and output layers, while every input node is connected to every output node. SOM is an unsupervised learning algorithm where competitive learning among the output nodes are used so that they can represent the patterns in the provided input. This process is comprised of three phases [41]:

i. Competition: An input vector is compared with the output vectors (which are initialised with random weights), and the vector having the most similar connection weight is defined as the winner or the best matching unit (BMU). Various distance function can be used to measure the distance. Euclidean distance is a common measure among these.
ii. Cooperation: To preserve the topographic organization, a neighbourhood is defined based on the winning node. The neighbourhood represents the closely related pieces of information in the input space.
ii. Adaptation: The weights of the winning node and the nodes in the neighbourhood are then updated according to the distance from the input node. With this step, nodes have an increased chance to respond to similar input data in future.

Practical application areas of SOMs include industrial instrumentation, pattern recognition, data mining and processing, robotics, optimisation etc. [42]. More specifically, Quantisation Error (QE) of SOMs, which is a measure of the quality of the map [43], is shown to be effective in detecting structural changes in geographic regions and medical images [44,45].

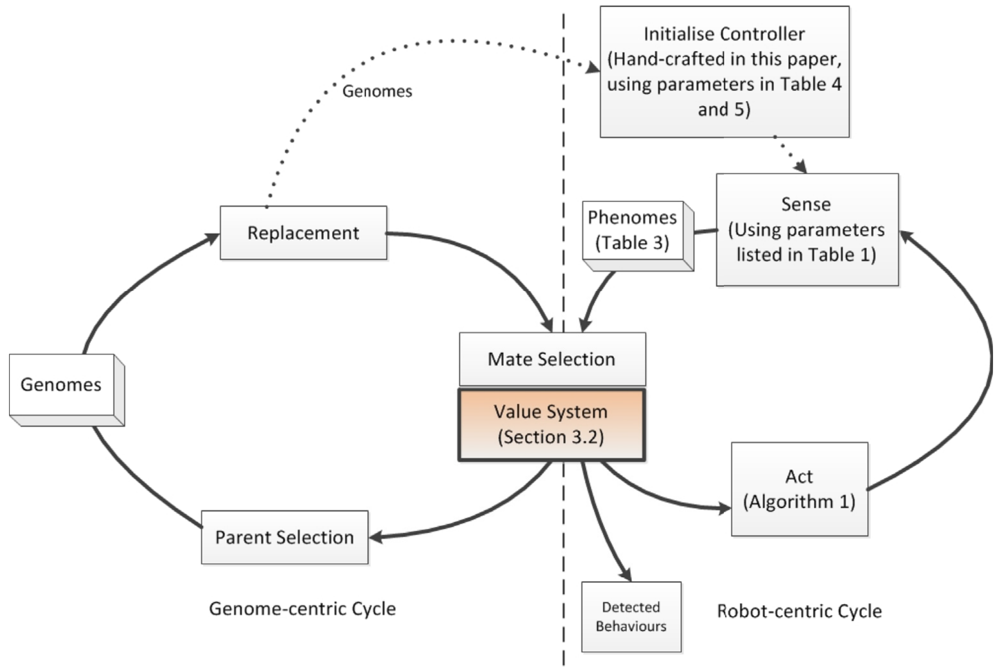


Fig. 2. Our conceptual framework assumes parallel genome-centric (left) and robot-centric (right) processes, adapted from the general architecture presented in Ref. [7]. Our focus on this paper is on the Value System.

2.3. Collective behaviour detection

There are some existing works that consider value systems in evolutionary frameworks. Hamann [46] presented an evolutionary approach to generate collective behaviours by minimizing surprise. Their fundamental concept is based on the 'free-energy principle' [47], or the idea that *'a minimal prediction error is interpreted as an evolutionary advantage'*. Their work shows that basic collective behaviours can be evolved as a by-product of minimizing prediction errors. As the authors pointed out, their approach is more generic than concepts such as novelty search [48], as no measure of behavioural distance is needed. Their experiments are conducted in 1-dimensional setting, where agents can move clockwise or anticlockwise inside a ring. In this paper we consider a 2-dimensional environment.

Gomes et al. [49] uses novelty search for evolution of a swarm of robots. They have used aggregation and resource sharing as representatives of simple and complex swarm robotics tasks. Their work demonstrates that novelty search is effective in both simple and deceptive scenarios. They provide detailed insights on how novelty search finds a broader diversity of solutions compared to fitness-based evolution. Our current paper differs from this work by considering the design of fitness functions as value systems that reward only generic structured properties of behaviour, rather than specific behavioural configurations.

Dahl et al. [50] presented a statistical framework to detect emergent, collective behaviour of simulated agents. They recorded agent trajectories and then computed statistics and similarities of the agent behaviours. Past behaviour data was used to create behaviour primitives for future behaviour detection. Their work was able to detect transitions between flocking and swarming. In contrast, this paper uses real-time behaviour data to predict patterns, and differentiates among various 'interesting' and 'random' patterns.

Miner [51] demonstrated an approach to learn emergent properties of multi-agent systems. They define the 'forward-mapping problem' as developing a functional (many-to-one) mapping from agent-level parameters onto a system-level property. This mapping can be used to determine system behaviour for specific sets of agent parameters. The 'reverse-mapping problem' is defined as mapping some system-level parameters onto a configuration vector. This mapping can be used to determine the agent parameters that would result in a given system wide property. Their work sheds light on the relation of boid parameters and resulting velocity. In contrast, the focus of our current paper is on detecting structured patterns in behaviour.

Croitoru [52] outlined an approach to derive low-level agent steering behaviours in flocks by analysing observed trajectories of groups. Their approach is based on the reconstruction of the weighted steering forces necessary to move the group of agents between consecutive time steps. They show that a reconstruction can be formulated as an optimisation problem and proposed a particle swarm optimisation algorithm to reconstruct the steering forces. The focus of their work is on explaining the relation between emergent patterns and local steering interactions between agents. Our current paper considers the direct detection of high-level behaviours.

Laube et al. [53] used mobile wireless sensor networks to detect flocking in sensor nodes from low-level trajectory data. Their proposed approach is 'decentralised' in the sense that individual processing units only have access to local information about other individuals. Their approach is specific to wireless sensor networks and the primary focus is on demonstrating the effectiveness of the decentralised detection approach as opposed to a central one.

Odonkor et al. [54] proposed an algorithm for mapping of oil spills with UAVs. Their approach is based on Particle Swarm Optimisation (PSO). Their approach relies on the emergent capabilities of swarms and utilises agent interaction in local and global neighbourhoods. Scalability and fault-tolerance was demonstrated using test cases where UAV team size is increased or team members are lost. Essentially, this is a contribution towards anomaly detection where agents share their knowledge to complete a task in efficient and time-sensitive manner.

Nitschke et al. [55] proposes Collective Neuro-Evolution (CONE) to solve a gathering and collective construction tasks. These tasks require agents to cooperate or coordinate their behaviour for building structures in the environment. The agents do this by placing building blocks in a 'construction zone' in a specific sequence. Their work provides insight on evolving multiple robot controllers to increase collective behaviour task performance. While this work focuses on a particular task, our proposal is more generic, as we focus on detecting interesting patterns as collective behaviour emerges in swarms.

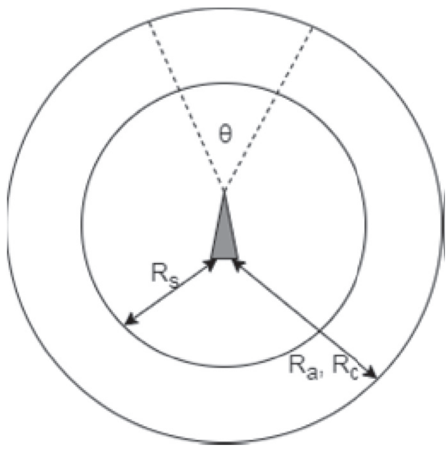


Fig. 3. A boid having vision angle θ and radii R for separation, cohesion and alignment neighbourhoods.

Innocente et al. [56] proposed a swarm robotics approach that enables drones to self-organise according to changing wildfire conditions. Their method includes utilising physics-based fire propagation model and leverage particle swarm optimisation (PSO) based approach to self-coordinate the drones so that they are addressing the areas which has higher temperature. Their work represents the potentials usage of Unmanned Aerial Vehicle (UAV) in various conditions. Similar to the previous work [55], their method is based on a particular task, which in this case is to react to the generated physic-based model.

3. Evolutionary developmental framework to support autonomous detection of interesting collective behaviour in swarms

This section first proposes a conceptual framework that will use a value system for dynamically assessing the fitness of changing swarm dynamics in an evolutionary setting in Section 3.1. We then propose a value system for assessing fitness within this framework in Section 3.2.

3.1. Evolutionary developmental framework

The essence of Evolutionary Computation (EC) is rooted in the Darwinian notion of evolution in biological systems. In evolution, the behaviours of individuals are shaped over many generations by random variations and natural selection, resulting in a 'fitter' population. Computer scientists have used the elements of the natural evolution to devise a computational procedure so solve various problems [57]. In EC, a set of solutions are initialised first, and then offspring are generated from these parents by means of random variations. These generated solutions are then tested for their 'fitness' by evaluating how well they fare in solving the problem. Implementing the notion of 'survival of the fittest', the least performing solutions are then discarded, the best performing ones are selected and the whole process gets repeated over generations. Compared to other approaches, EC provides flexibility, adaptability to changing conditions, and robust performance [58]. Historically, algorithms in EC were inspired by approaches such as Evolutionary Programming (EP), Evolution Strategies (ES) and Genetic Algorithm (GA). A recent review of EC and Bio-inspired computation in general can be found in Ref. [59]. We do not consider the specifics of evolutionary algorithms in this paper. We focus instead on the value system that will sit within this framework.

Our framework synthesises three areas of work: evolutionary computation, artificial swarming and computational value systems. These components of the conceptual framework are illustrated in Fig. 2. The components of Fig. 2 are discussed in the following paragraphs.

The role of the evolutionary developmental framework, shown on the left of Fig. 2, is to generate sets of parameter values to change the behaviour of the swarm. An evolutionary developmental framework requires representations of the genome and phenome to permit evolution, and agent experiences to permit the calculation of value to drive the developmental process. Various approaches have been made to embodied evolution including Markov Brains [5], neuro-evolutionary algorithms [60] and traditional genetic algorithms. The genome representation underlying evolution will be based on the parameters of a swarm model. For example, Reynolds' [10] boids model uses three rules, generally implemented as forces for cohesion, alignment and separation. These rules are governed by parameters controlling properties such as the weights of the rules and sensory radius of the agents. More complex variants of the boids model introduce further parameters governing properties such as the vision angle of the agents, frequency of rule application and so on [61]. BGAs [11] introduce further rules and parameters to ensure specific vehicles stay within the boundaries of their operating conditions. This paper will focus on an extended boid model [61], to establish the parameters that best support computation of value. This framework includes additional parameters that affect the "situational awareness" (SA) of the boids. Fig. 3 illustrates a simple boid with these relevant properties.

The basic *boids* model can be viewed as a type of rule-based reasoning. The three fundamental rules are:

- ***Cohesion:*** An agent should move towards the average position of its neighbours;
- ***Alignment:*** An agent should steer to align itself with the average heading of its neighbours;
- ***Separation:*** An agents should move to avoid collision with its neighbours.

The rules are generally implemented as forces that act on agents when a certain condition holds. Suppose we have a group of N agents $A^1, A^2, A^3 \ldots. A^N$. At time t each agent A^i has a position, x_t^i, and a velocity, v_t^i. x_t^i is a point and v_t^i is a vector. At each time step t, the velocity of each agent is updated as follows:

$$\boldsymbol{v}_{t+1}^i = \mathrm{v}_t^i + \mathrm{W_c}\, \boldsymbol{c}_t^i + \mathrm{W_a}\, \boldsymbol{a}_t^i + \mathrm{W_s} \boldsymbol{s}_t^i \tag{1}$$

$\mathbf{c}_t^i$ is a vector in the direction of the average position of agents within a certain range of A^i (called the neighbours of A^i); $\boldsymbol{a}_t^i$ is a vector in the average direction of agents within a certain range of A^i; and $\boldsymbol{s}_t^i$ is a vector in the direction away from of the average position of agents within a certain range of A^i. These vectors are the result of cohesive, alignment and separation forces corresponding to the rules outlined above. Weights W_c, W_a and W_s strengthen or weaken the corresponding force. W_d strengthens or weakens the perceived importance of the boid's existing velocity. Once a new velocity has been computed, the position of each agent is updated by:

$$x_{t+1}^i = x_t^i + \boldsymbol{v}_{t+1}^i \tag{2}$$

Formally, we can define a subset $\boldsymbol{N}^i$ of agents within a certain range R of A^i as follows:

$$\boldsymbol{N}^i = \{A^k | A^k \neq A^i \wedge dist(A^k, A^i) < R\}, \tag{3}$$

where $dist(A^k, A^i)$ is generally the Euclidean distance between two agents. Different ranges may be used to calculate cohesion, alignment and separation forces, or other factors such as the communication range of a *boid*. The average position $\vec{\mathbf{c}}_t^i$ of agents within range R_c of A^i is calculated as:

$$\vec{\mathbf{c}}_t^i = \frac{\sum_k \mathbf{x}_t^k}{|(\mathbf{N}_c)_t^i|} \tag{4}$$

The vector in the direction of this average position is calculated as:

Table 1
Basic boid parameters and situational awareness parameters.

Parameter	Description	Range
W_s	Weight for separation rule	0–4
W_a	Weight for alignment rule	0–4
W_c	Weight for cohesion rule	0–4
V_{max}	Maximum speed: Maximum distance a boid will cover per tick	5–20
V_{min}	Minimum speed: Minimum distance a boid will cover per tick	2–10
R_s	Separation radius: Boids under this distance qualify for separation rule	2–62
R_c	Cohesion radius: Boids under this distance qualify for cohesion rule	5–1005
R_a	Alignment radius: Boids under this distance qualify for alignment rule	5–1005
θ_v	Vision angle: Agents within this angle relative to the agent's heading may be visible, if they are in range	17°–360°
P_{sa}	SA likelihood: Agents will update SA each tick with this probability	0–1
F_{sa}	SA frequency: Number of simulation ticks between agents updating situational awareness	1–10
$P_{fullscan}$	Full scan likelihood: The probability with which agents will look all around and do a 360-degree scan	0–1
F_{rule}	Rule frequency: Number of ticks between successive applications of rules to calculate change of velocity	1–10
P_{rule}	Rule likelihood: Probability agent will update velocity each tick	0–1
F_s	Separation frequency: Number of ticks between successive applications of the separation rule	1–10
P_s	Separation likelihood: Probability agent will apply separation rule each tick	0–1
F_a	Alignment frequency: Number of ticks between successive applications of the alignment rule	1–10
P_a	Alignment likelihood: Probability agent will apply alignment rule each tick	0–1
F_c	Cohesion frequency: Number of ticks between successive applications of the cohesion rule	1–10
P_c	Cohesion likelihood: Probability agent will apply cohesion rule each tick	0–1

$$c_t^i = \overrightarrow{c}_t^i - x_t^i \tag{5}$$

Similarly, we can calculate the average position $\overrightarrow{\mathbf{s}}_t^i$ of agents within range R_s of A^i as:

$$\overrightarrow{\mathbf{s}}_t^i = \frac{\sum_k \mathbf{x}_t^k}{\left|(\mathbf{N}_s)_t^i\right|} \tag{6}$$

The vector away from this position is calculated as:

$$\mathbf{s}_t^i = \mathbf{x}_t^i - \overrightarrow{\mathbf{s}}_t^i \tag{7}$$

Finally, the vector $\mathbf{a}_t^i$ in the average direction of agents within range R_a of A^i, is calculated by the sum:

$$\boldsymbol{a}_t^i = \frac{\sum_k \mathbf{v}_t^k}{\left|(\boldsymbol{N}_a)_t^i\right|} \tag{8}$$

These vectors are then normalised and multiplied by the corresponding weights, before calculating the new velocity. The newly calculated velocity is further normalised and scaled by a speed value chosen between a maximum and minimum range (V_{max} and V_{min} respectively). See Table 1 for a summary of boid parameters.

Several variations of this basic scheme are possible, with each variation resulting in a slightly different characteristic behaviour of the swarm. In this paper we use vision angle, likelihood and frequency parameters [61], so we can modify the 'situation awareness' of the swarm. Specifically, the update of velocity does not happen in every clock tick. Rather, we check if the conditions for the frequency and/or likelihood parameters are met and only then update the velocity and position accordingly. To apply these rules, we keep track of the timestamps when rules were last applied. This is shown in Algorithm 1. In lines 1 to 5, we update the situational awareness values of the swarm. This includes checking the frequency and probability variables for updating situational awareness, checking full scan likelihood and setting the value of θ accordingly, and then calculate the number of boids in respective neighbourhoods. In lines 6 to 12, we apply the three boid rules and calculate the three force vectors. Each of these steps are performed after checking the respective frequency and likelihood requirements. In lines 13 to 17, we use these updated vectors to calculate new velocities and eventually update the positions of all boids.

Table 2
Recorded behaviour data.

Notation	Description
$V_x^i tV_y^i t$	Elements of the velocity of the boid
$W_s \overrightarrow{s}_t^i = (S_{x_t}^i, S_{y_t}^i)$	Elements of separation vector
$W_a \overrightarrow{a}_t^i = (A_{x_t}^i, A_{y_t}^i)$	Elements of alignment vector
$W_c \overrightarrow{c}_t^i = (C_{x_t}^i, C_{y_t}^i)$	Elements of cohesion vector
$\lvert N_s \rvert$	Number of boids in separation radius
$\lvert N_a \rvert$	Number of boids in alignment radius
$\lvert N_c \rvert$	Number of boids in cohesion radius

Algorithm 1
Velocity and Position update for boids at each step

1. If $t - T_{saupdates} \geq F_{sa}$ AND $P_{sa} \geq Rand$
2. $\theta = \theta_v$
3.   If $P_{fullscan} \geq Rand$
4. $\theta = 360°$
5.   Calculate N_s, N_a, N_c
6. If $t - t_{ruleappliedlast} \geq F_{rule}$ AND $P_{rule} \geq Rand$
7.   If $t - t_s \geq F_s$ AND $P_s \geq Rand$
8.   Calculate $\boldsymbol{s}_t^i$ as per Equation 7
9.   If $t - t_a \geq F_a$ AND $P_a \geq Rand$
10.   Calculate $\boldsymbol{a}_t^i$ as per Equation 8
11.   If $t - t_c \geq F_c$ AND $P_c \geq Rand$
12.   Calculate $\boldsymbol{c}_t^i$ as per Equation 5
13.   Apply Equation 1
14.   Normalise $\boldsymbol{v}_{t+1}^i$: $\boldsymbol{v}_{t+1}^i = \boldsymbol{v}_{t+1}^i / \boldsymbol{v}_{t+1}^i$
15.   Pick a random speed value, v such that $V_{max} \leq v \leq V_{min}$
16.   Scale velocity by speed: $\boldsymbol{v}_{t+1}^i = \boldsymbol{v}_{t+1}^i * v$
17.   Update position using Equation 2

As the boids move in the simulated environment, we collect phenotypic data from each of the boids in each time step. The list of the recorded data is summarised in Table 2. These phenotypic behaviour data are passed to the value system for behaviour detection.

3.2. Value system

The role of the value system within the framework in Fig. 2 is to generate a fitness measure for the emergent behaviour resulting from particular boid parameter values. We represent experiences from local state information, such as information about the velocity of the agent and relative positions and velocities of its neighbours. As an initial approach, we adopt a computational model of curiosity that uses a SOM as the underlying data structure [35]. We used the SOM toolbox [62,63] for performing our experiments. A heuristic formula was used to determine the number of neurons for the map: $No.\ of\ neurons = 5 * \sqrt{M}$, where M is the number of observations. The learning rate was fixed to 0.25 and the neighbourhood size was fixed to 1. The fixed values capture the property of a value system that it is always learning.

Each time-step, one or more observations $O_t = (o^1, o^2 \ldots)$ of the swarm are constructed, (optionally) normalised, and presented to the SOM. The most similar neuron K_t to the observation is selecting with the minimum distance d to the input stimulus. We use Euclidean distance in this paper. We experiment with three approaches to normalisation:

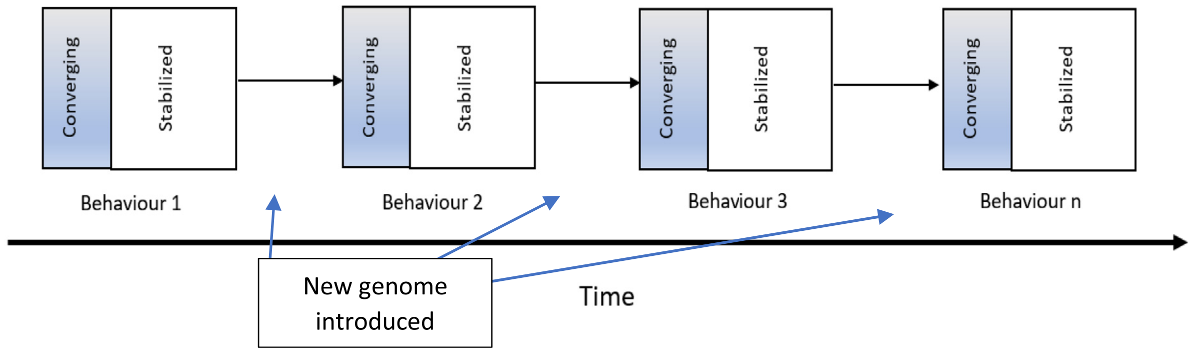


Fig. 4. Synthetic data generated for experiments. 16 genomes were introduced over 24,000 time-steps.

1. No normalisation: raw attribute values are used as input. This approach can be used in an online manner, with each data point processed by the SOM immediately as a boid experiences it.

2. Z-score normalisation/Standardisation: This is a linear transformation of the input attributes so that their variance is 1 and mean is 0. The following formula is used to perform z-score normalisation

$$o' = \frac{o - \mu}{\sigma} \tag{9}$$

where o is the original input, μ is the mean of the input feature and σ is the standard deviation. This is applied to each column of the recorded behaviour data. For example, for a component of an observation V_x(such as that shown in Equation (11)), the mean would be taken over all values of V_x over all observations. To calculate μ and σ, we would need a batch of data. To achieve this in runtime, we can use a window providing us with batches of data which will be used to calculate the mean and standard deviation.

3. Min-max normalisation: In this case, we normalise the attributes to have data in the range [0,1]. The following formula is applied to achieve this:

$$o' = \frac{o - o_{\min}}{o_{\max} - o_{\min}} \tag{10}$$

where o_{max} and o_{min} are the maximum and minimum possible values for a particular input attribute that have occurred during a given batch. For example, for the first component, $o_{min} = \min(V_x)$ where $V\_x = |V^i_{x_t}|$. Similar to z-score normalisation, we would need to work on a batch of data and a window can be used to calculate the min and max values in runtime.

All neurons in the neighbourhood of the winning neuron K are then moved closer to the input stimulus by adjusting each of their weights k_L using the SOM update equation.

In existing literature, various definitions of O_t have been proposed. These include representations of the state of the world, and representations of changes that occur in the world. In the case of swarm robotics, multiple agents are sensing the world at any given time. There are thus further alternatives that can be considered for computing curiosity in an evolutionary framework. Specific alternatives that we will examine in Section 4 are:

- Taking data from one agent at a time and inputting these separately to the SOM. This means that at any time-step t in the boid simulation, N inputs O^i_t are given to the SOM – one for each of N boids. It has the form:

$$O^i_t = \left[V^i_{x_t},\ V^i_{y_t},\ S^i_{x_t},\ S^i_{y_t},\ A^i_{x_t},\ A^i_{y_t},\ C^i_{x_t},\ C^i_{y_t} N^i_{ca_t},\ N^i_{s_t}\right] \tag{11}$$

- Aggregating data from the experiences of multiple agents in the swarm. This means that at any time-step t in the boid simulation there is one input to the SOM of the form

$$O_t = \left[O^1_t, O^2_t, \ldots, O^3_t, \cdots O^N_t\right] \tag{12}$$

The curiosity value output by the SOM in response to each input is the quantisation error (QE) computed at that timestep. In the case that input data has the form in Equation (11), QE is computed by calculating the average distance between the input vectors from each boid and their corresponding best matching unit. The formula for average quantisation error is as follows:

$$QE_t = \sum_{i=1}^{N} O^i_t - K^i_t \tag{13}$$

where K^i_t is the winning neuron for the corresponding input O^i_t.

In the case that input data has the form in Equation (12), QE is computed by calculating the average distance between the single input vectors generated for the whole swarm at each timestep, and the best matching unit. The formula for quantisation error in this case is:

$$QE_t = O_t - K_t \tag{14}$$

4. Investigating effective configurations for identifying collective behaviour

This section describes three experiments to investigate the following questions:

1. What is the most effective representation of O_t?
2. What are the most effective design decisions for the curiosity model?
3. Can the value system differentiate between random and structured behaviour?
4. Can the value system differentiate between different structured behaviours?

In this section we examine these questions under controlled conditions. Specifically, we work with a series of simulated robot-centric cycles, and a limited set of 16 hand crafted genomes. These genomes are for eight structured swarm behaviours (those shown in Fig. 1) and eight random behaviours. We simulate each of these behaviours by introducing a new genome to control each of 200 boids every 1500 time-steps. This produced a synthetic dataset that progressively converges on each of the behaviours then destabilise and reconverges on the next behaviour and so on as shown in Fig. 4.

We distinguish between the 'converging' data and the 'stabilised' data in Fig. 4, to permit us to investigate whether this has an impact on the performance of the value system. We observed that each behaviour was

Table 3
Settings of the boid simulation environment.

Parameter	Value
World size	1400 × 1000 units with a wrap-around
Number of boids, N	200
Boid speed, v	1–20 units per tick

able to stabilise inside 500 time-steps (called the converging period). We then allowed each behaviour to run for a further 1000 time-steps (call the stabilised period). We leave the genome-centric cycle for future work.

4.1. Experiment 1: learning structure in collective behaviour using a curiosity model

4.1.1. Aim

The aim of this experiment is to determine whether a curiosity model can identify structure in the experiences of boids.

4.1.2. Hypothesis

A curiosity model should achieve a low QE in response to experiences of structured behaviours and higher QE in response to experiences of random behaviours.

4.1.3. Method

4.1.3.1. Environment. We simulated a robot-centric cycle with $N = 200$ boids in a 1400X1000 arena with the wrap around property at the boundaries. These experimental parameters are summarised in Table 3.

The boid simulation software was written in Java. Java Swing was used to build the user-interface using which the boid parameters (as stated in Table 1) can be modified. The corresponding swarm behaviour is observable in real-time as one changes the parameters. The simulation software outputs the behaviour data (as stated in Table 2) in text files, which is analysed and processed in MATLAB R2019a. A modified version of the SOM Toolbox [62] was used to implement the SOM and generate the QE values. MATLAB parallel pool was used when possible to expedite the processing. HP Z420 workstations with Intel Xeon processors were used to run the simulations and conduct corresponding post-processing and SOM implementation.

4.1.3.2. Dataset: hand crafted collective behaviours. As introduced above, our dataset included 16 behaviours, hand crafted to facilitate a controlled experiment. Here we describe the characteristics of each of these behaviours, explaining in detail the parameters and emergent behaviours. We start with the structured behaviours and then continue with the random behaviours.

Flocking: Flocking is a group behaviour displayed by many species in nature such as birds, insects and fish. Flocking is an example of collective motion where many individuals follow some simple rules to exhibit complex group behaviour. Following the ‘Boids’ example demonstrated by Reynolds, we set the baseline by letting the boids move in flocks, creating smooth group motions. This is achieved by setting a separation weight of 1.5 while the alignment and cohesion weights are set to 1.0. The boids can be moving in multiple groups and as groups came within the vision range, they will join and make bigger groups.

Lines: In this behaviour, the boids tightly follow each other in a line which has typical width of one boid. All the behavioural parameters of this scenario are the same as flocking, with the only exception of separation weight, which is changed to 0 from 1.5. This resulted in the similar group movement as flocking, but as the separation force is reduce to zero, the boids can overlap with each other’s position. As a result, the boids move by preserving the flocking motion. However, as the separation force is null, the flocking happens in a line.

Spermatozoa: In this behaviour, the separation weight is increased to 4.0, while alignment weight is 1.04 and cohesion weight is reduced to 0. This results in the boids moving in the same direction, but no flocking happens as cohesion is set to zero. Moreover, the avoid distance is increased 2.5 times than the flocking scenario, and the vision range is increased more than three times. This resulted in a more spread out group of boids moving in the same direction (due to the alignment force) but keeping a large distance among themselves (due to large avoid distance) on all sides (due to 360° vision angle).

Old man river: In this behaviour, the boids have a larger avoid distance than flocking scenario, and vision range is almost five times increased compared to the flocking scenario. Separation, alignment and cohesion weight is set to 1.88.0.56 and 0.32 respectively. Rule frequency and Situational Awareness (SA) frequency are set to higher values, resulting in more number of ticks between position and velocity updates of the boids and the neighbours. As a result, the boids move like flocks, but with larger distance in between them. The minimum speed is decreased as well, so the boids move slower compared to the aforementioned scenarios.

Ink in water: This behaviour is similar to *Lines* in terms of separation, alignment, cohesion weights and vision range, vision angle and avoid distance. The difference between this and *Lines* lies in rule frequency, rule likelihood, situational awareness (SA) frequency and situational awareness likelihood. In this scenario, the time tick between the updates are increased, and the likelihoods are decreased. That means the velocity and position of the boids themselves as well as the neighbouring boids’ are updated less frequently and with a lesser probability. Overall, the resultant motion is similar to lines, i.e. boids closely moving together with flock-like motion but not as rigidly. In this case, the boids are more free-flowing due to lesser probability of applying the rules, and shows a wiggly movement pattern.

Gravity wells: In this behaviour, separation weight is set to 0.44 while alignment weight is 0 and cohesion weight is 1. Moreover, vision angle is set to 360° and vision range is highly increased to 755, which is more than 7 times than that of flocking. As a result, the boids exhibiting this behaviour have a large neighbourhood (given the world size is 2000 × 2000 units) to look at. With zero alignment and standard cohesion, the boids are tightly grouped together as if gravity has pulled them into a well. Typically two or three such groups are generated, because of the ratio of the world size and neighbourhood size.

Firefly: In this behaviour, separation weight is increased to 1.88, alignment weight is kept to 1.0 and cohesion weight is set to 2.68. Vision range is increased to 345, vision angle is set to 360°, and avoid distance is increased as well. Moreover, maximum speed is increased to 18.65, compared to 10 in flocking scenario. The SA likelihood is decreased to 0.28. As a result, the boids move much faster in a seemingly erratic manner. However, a closer inspection shows that boids are forming groups and consistently following a pattern. This is due to relatively high cohesion weight, and lower situational awareness likelihood. In essence, the behaviour is similar that of groups of fireflies with a centre while a few of them repeatedly drifts away from the centre and returns to it.

Brownian: In this behaviour, separation weight is set to 1.0 while alignment and cohesion weight is set to zero. The vision range is set to 205, with a vision angle of 180°. Compared to the baseline flocking scenario, this provides each of the boids with the freedom of having their own velocity and speed, without being affected by the group dynamics. The separation weight keeps them from bumping into each other. As a result, the boids do not form any group rather continue moving freely in random directions. As a group, this is similar to the random Brownian motion of particles.

The random behaviours result from a combination of behaviour parameters that generate random movement patterns of the boids. To elaborate, our focus was on generating seemingly random, unstructured behaviour for the boids. To achieve this, we have to deliberately set the behaviour parameters in such a way that the resulting swarm behaviour seems random to human perception. The randomness of these generated behaviours were confirmed by human study. The description of the parameters that comprised these random behaviours are listed below.

Table 4
Parameters for the structured swarm behaviours.

Parameter	Flocking	Lines	Sperm	Old Man	Ink	Gravity	Fire fly	Brown
W_s	1.5	0	4	1.88	0	0.44	1.88	1
W_a	1.0	1	1.04	0.56	1	0	1	0
W_c	1.0	1	0	0.32	1	1	2.68	0
V_{max}	10	10	12.95	10	10	5	18.65	10
V_{min}	7	7	6.96	6.08	7	2	7	7
R_s	25	25	60.8	37.4	25	24.8	62	37.4
R_c, R_a	100	100	335	475	100	755	345	205
θ (degree)	180	180	360	103	180	360	360	180
P_{sa}	1	1	0.46	0.6	0.51	1	0.28	1
F_{sa}	1	1	1	3.07	2.26	1	1	1
$P_{fullscan}$	0	0	0	0.3	0	0	0	0
F_{rule}	1	1	1	3.16	1.99	1	1	1
P_{rule}	1	1	0.28	1.0	0.59	1	1	1
F_s	1	1	1	1.0	1	1	1	1
P_s	1	1	1	1.0	1	1	0.49	1
F_a	1	1	1	1.0	1	1	1	1
P_a	1	1	1	1.0	1	1	1	1
F_c	1	1	1	1.0	1	1	1	1
P_c	1	1	1	1.0	1	1	1	1

Random 0: In this scenario, the separation weight and alignment weights are set to 1.8 and 1.16 respectively, while the cohesion weight is 0. These weights, couple with a high velocity range, result in group of boids showing alignment but moving in variable speeds in a stop-start manner. Due to not having any cohesion, some boids keep moving in radically different directions compared to the group of boids moving in similar direction.

Random 1: In this scenario, the alignment weight is 0, while separation weight is 0.56 and cohesion weight is 0.32. It results in multiple groups of boids moving together but they have varied directions. Also, each boid has a start-stop movement pattern.

Random 2: In this scenario, the boids have low alignment and cohesion weights (0.32) and a separation weight of 1.36. There is little presence of group behaviour. Looking at individual boids, they maintain a steady velocity for a certain period, and then speeds up slows down suddenly with a slight variation in direction.

Random 3: This scenario has a separation weight of 1.4, an alignment weight of 0.92 and a low cohesion weight of 0.28. The alignment/cohesion radius is quite high as well. As a result, there is a hint of alignment in the group behaviour but is overshadowed by the randomness in the movement. For individual boids, the magnitude and direction of speed both vary a lot.

Random 4: In this scenario, weights for all the three vectors are zero, and the range between maximum speed and minimum speed is relatively high. As a result, there is no presence of group behaviour, and each boid keeps moving in a straight line. However, the change in velocity can be sudden as the range of possible speed is relatively high.

Random 5: In this scenario, boids have zero alignment weight and zero cohesion weight and a moderate separation weight. As a result, the only change is velocity is due to the separation force. There is no presence of group behaviour as the boids have no coherence and alignment force. The velocity of individual boids remains constant for a time period and then changes slightly when affected by the separation force.

Random 6: This scenario has a separation weight of 1.8 and zero alignment and cohesion weights. The possible range of velocity is high as well. As a result, the final velocity of each boid is affected only by the separation force. Individual boids mostly stay on a straight line, with sudden increase and decrease in velocity.

Random 7: In this scenario, the separation weight is quite high (4.0), with moderate alignment weight (1.12) and zero cohesion weight. The velocity range is high as well. This results in boids showing erratic behaviour both in individual and group level. The velocity and direction of individual boids change abruptly as they come close to each other and the strong separation force is applied.

The parameters for these behaviours are listed in Table 4 (structured

Table 5
Parameters for the random behaviours.

Parameter	Random 0	Random 1	Random 2	Random 3	Random 4	Random 5	Random 6	Random 7
W_s	1.8	0.56	1.36	1.4	0	1.24	1.8	4
W_a	1.16	0	0.32	0.92	0	0	0	1.12
W_c	0	0.32	0.32	0.28	0	0	0	0
V_{max}	18.65	12.95	12.35	14.6	20	10.4	18.65	18.65
V_{min}	4.8	6.96	3.12	3.36	2.0	3.12	2.4	4.8
R_s	62	37.4	62	62	48.8	35.6	44	62
R_c, R_a	795	425	215	1005	665	95	795	465
θ (degree)	86	178	148	360	338	178	24	178
P_{sa}	0.28	1	0.34	0.43	1	1	0.28	0.48
F_{sa}	3.70	1	1.99	1	1	1	3.61	3.61
$P_{fullscan}$	0.33	0	0	0	0	1	0.23	0.33
F_{rule}	7.75	6.4	6.22	10	7.03	4.96	4.51	7.75
P_{rule}	0.29	0.26	0.14	0.12	0.48	1.0	1.0	0.28
F_s	4.06	1	8.74	6.31	1	1	4.06	10
P_s	0.49	1	0.45	0.26	0.26	1	0.49	0.49
F_a	3.79	1	9.45	3.25	1	1	3.79	3.79
P_a	1	1	0.54	0.24	0	1	1	1
F_c	3.07	1	6.76	1	1	1	3.07	3.07
P_c	0.45	1	0.33	0.48	0	1	0.45	0

Table 6
Quantisation error recorded for different behaviours.

Scenario	Mean QE
Flocking	0.76
Lines	0.64
Spermatozoa	1.39
Old Man	1.03
Ink	0.97
Gravity	2.25
Firefly	10.24
Brownian	0.78
Random 0	2.49
Random 1	18.76
Random 2	1.19
Random 3	4.76
Random 4	2.06
Random 5	0.72
Random 6	2.03
Random 7	98,968.15

behaviours) and Table 5 (random behaviours).

We first examine the performance of the curiosity model (in terms of QE) using raw input data from both the converging and stabilised periods (i.e. no normalisation), and the representation in Equation (11) (that is the sensory data from each boid is input to the SOM separately). This means that 4,800,000 data instances are input to a SOM. We repeated the experiments 30 times, randomising the order in which the data for the 16 behaviours are presented to the SOM (but maintaining the ordering of the data for any single behaviour). We report on average results in this section.

4.1.4. Results

Table 6 shows the mean QE for each behaviour, and Fig. 5 visualises a subset of these results. We can see that the QE for the random behaviours tends to be higher than that for the structured behaviours, although there are exceptions. It is possible to draw in a threshold (blue dashed line in Fig. 4) where most (6/8) of the structured behaviours have QE below the threshold and most (6/8) of the random behaviours have QE above the threshold. There are two structured behaviours that resulted in relatively high QEs: gravity and firefly.

The Receiver Operating Characteristics (ROC) curve in Fig. 6 shows the corresponding classification performance. To calculate the ROC, the mean QE values were sorted and then threshold points were determined by averaging consecutive values. True Positive Rate (TPR) and False Positive Rate (FPR) were calculated for these corresponding threshold values and plotted in y and x axis accordingly.

If we examine the behaviour of some individual boids in Fig. 7, we can obtain some insight into why some of the random behaviours have

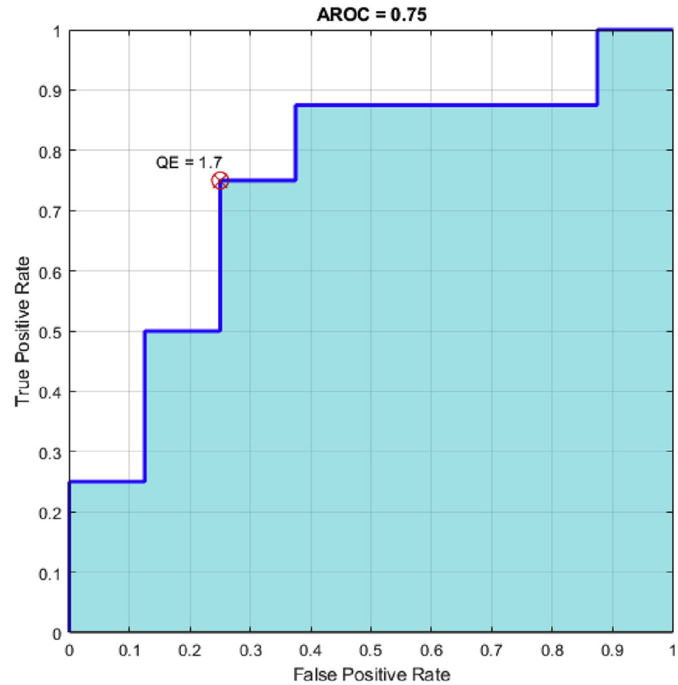


Fig. 6. ROC curve for converging and stabilised period data; phenome representation in Equation (11), no normalisation.

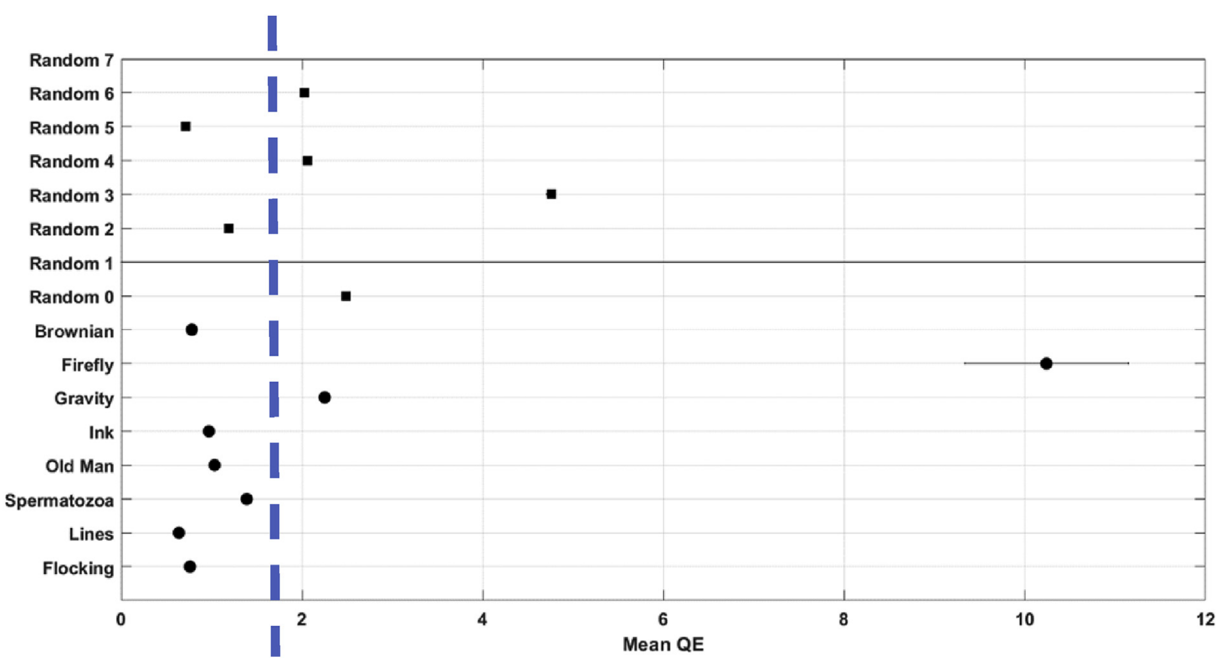


Fig. 5. Mean QE over converging and stabilised period data; phenome representation in Equation (11), no normalisation. We can see it is possible to select a threshold (blue line) where most (6/8) structured behaviours have QE below the threshold, and 6/8 random behaviours have QE above the threshold.

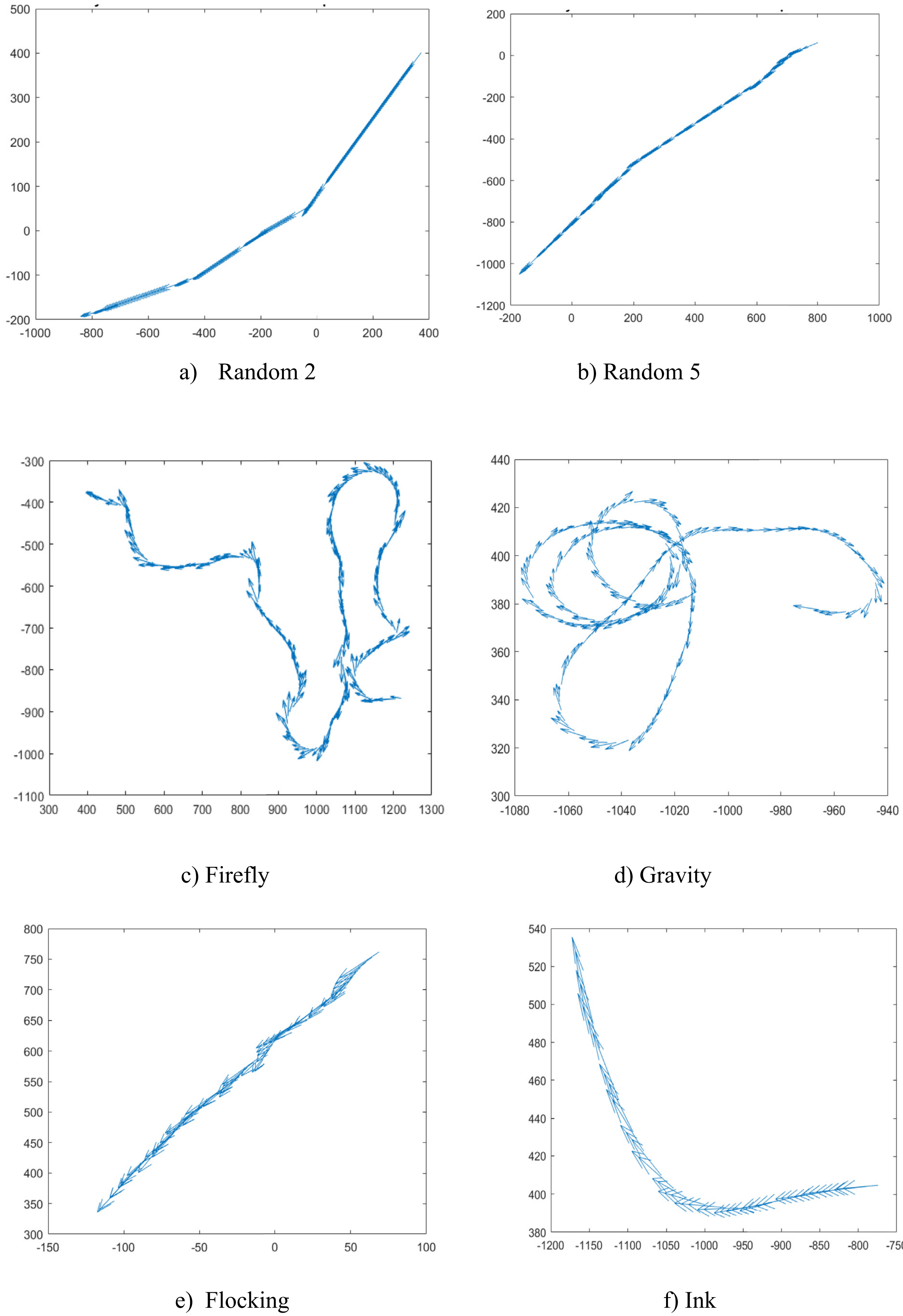
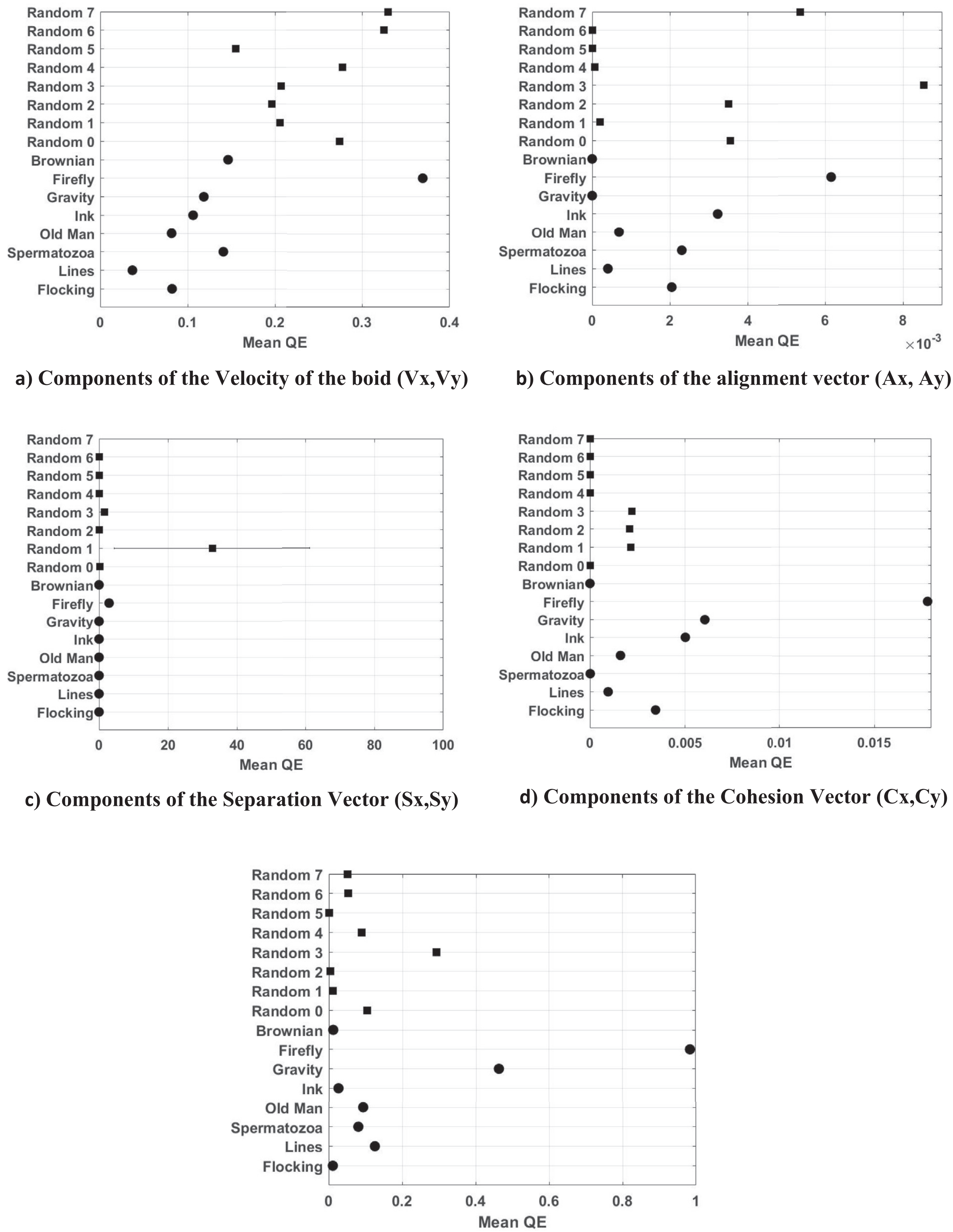


Fig. 7. 200 timesteps position and velocity data for (a) a single 'Random 2' boid; (b) a single 'Random 5' boid; (c) a single 'Firefly' boid; (d) a single 'Gravity' boid; (e) a single 'Flocking' boid; (f) a single 'Ink' boid.



a) Components of the Velocity of the boid (Vx,Vy)

b) Components of the alignment vector (Ax, Ay)

c) Components of the Separation Vector (Sx,Sy)

d) Components of the Cohesion Vector (Cx,Cy)

e) Number of boids in separation and alignment/cohesion radius (|Ns|,|Na/c|)

Fig. 8. Mean QE values for Experiment 2, showing QE values for the individual components of the input. This was conducted with converging and stabilised period data, using phenome representation in Equation (11), no normalisation, and Euclidian distance for computing QE. Note that x-axis scales are different in each subfigure.

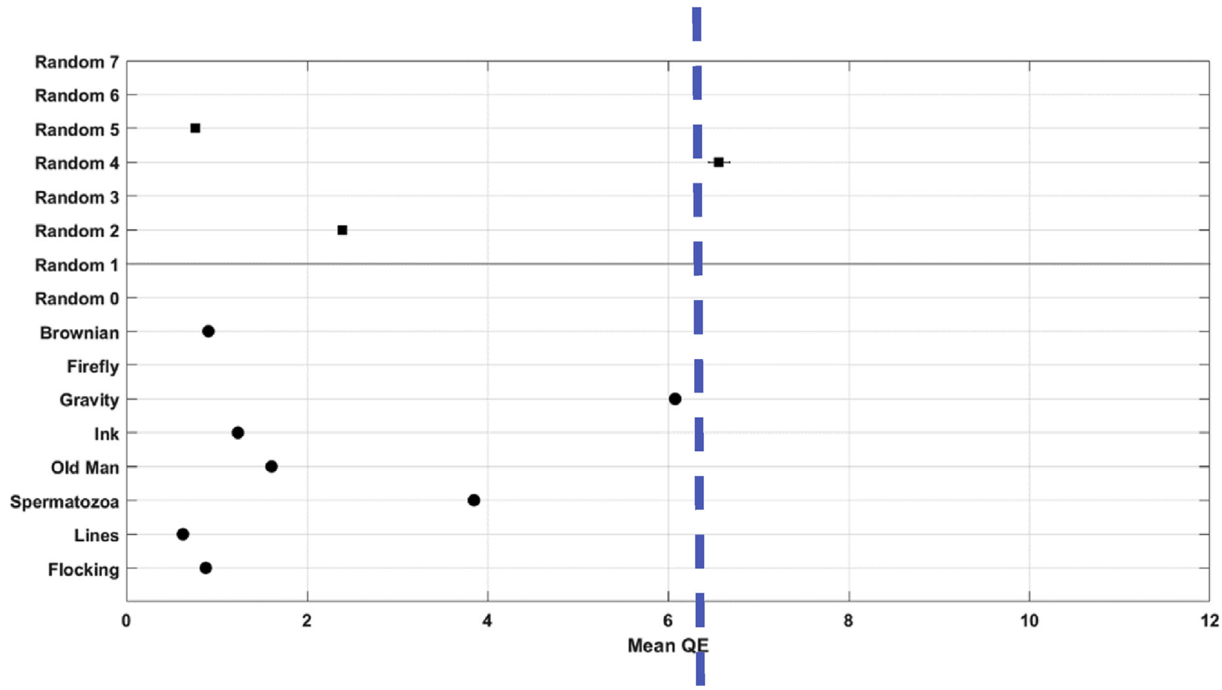


Fig. 9. Mean QE over converging and stabilised period data; phenome representation in Equation (11), no normalisation and squared Euclidean distance for computing QE. We can see more distinction between QE values for random and structured behaviours.

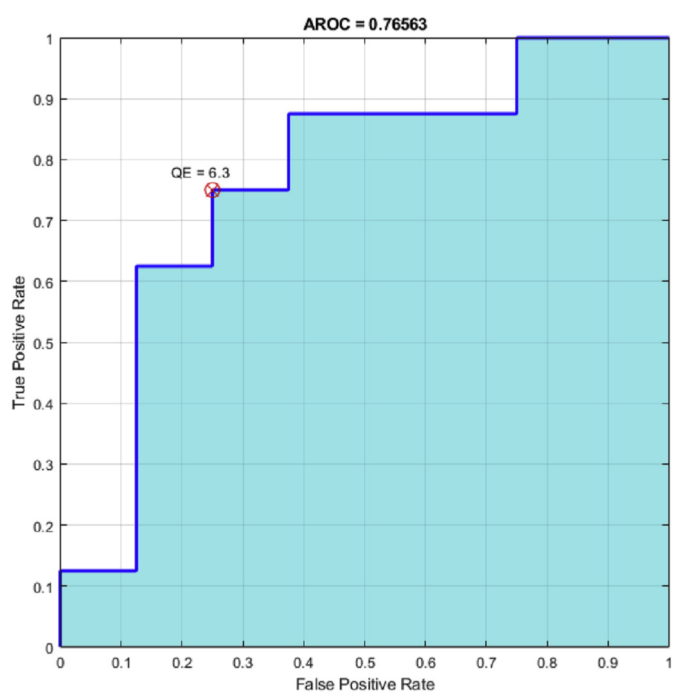


Fig. 10. ROC curve for converging and stabilised period data; phenome representation in Equation (11), no normalisation and squared Euclidian distance for computing QE.

low QE and why some of the structured behaviours have high QE. For example, Fig. 7(a) and (b) show the trajectories of single 'random 2' and 'random 5' boids for sample periods of 200 timesteps. We can see that these trajectories have periods of constant velocity. Thus, although all the boids in random 5 and random 2 have different velocities, these periods of constant velocities by individual boids are enough to result in low QE.

For a similar reason, Brownian motion, which we might expect to behave like a random motion, also has a low QE. This is because individual particles will continue in straight lines until they encounter another particle.

In contrast, in the case of firefly motion (Fig. 7(c)), although all boids are grouped, any individual boid will travel in many different directions over just a 200 timestep sample. This results in a relatively high QE for this behaviour. In case of Gravity Wells (Fig. 7(d)), the boids are more tightly grouped than Firefly boids but individual boids change their direction quite frequently. This results in a comparatively higher QE.

These results provide initial evidence that it may be plausible to use a curiosity based value system to distinguish between certain types of structured and random behaviours. However, it is clear that further tuning is needed to better filter some types of random behaviour, and better identify some types of structured behaviour. The next experiments examine some alternative design choices for the curiosity model to attempt to address these issues.

4.2. *Experiment 2: study of the input space*

4.2.1. *Aim*

The aim of this experiment is to determine the effects of the individual elements of the input space on the performance of the value system.

4.2.2. *Method*

To conduct this experiment, we take the input values (as in Equation (11)) and divide them to five components – velocity, alignment vector, separation vector, cohesion vector, and neighbouring boids. We then used these individual components as inputs and observe how the SOM behaves. We conducted this experiment by taking data from both converging and stabilised periods, using phenome representation in Equation (11), without normalisation and using Euclidian distance for calculating QE. Experiments were run 30 times, and results were obtained by averaging the values.

4.2.3. *Results*

Fig. 8 shows the mean QE for each set of parameters. Comparing these

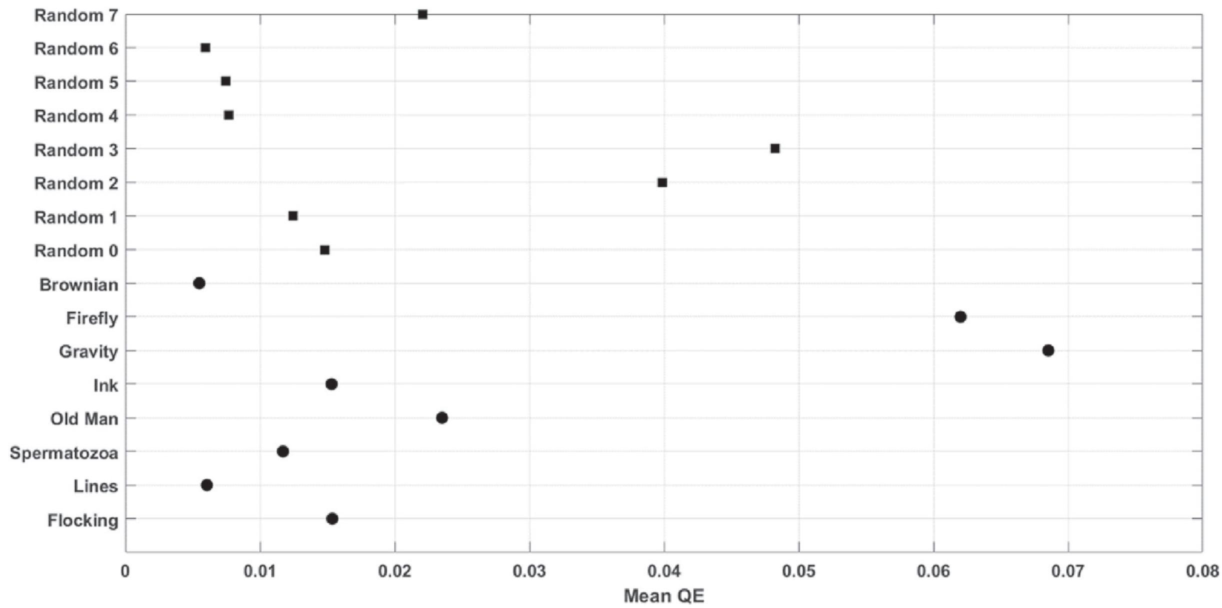


Fig. 11. Mean QE over converging and stabilised period data; phenome representation in Equation (11), min-max normalisation and squared Euclidean distance.

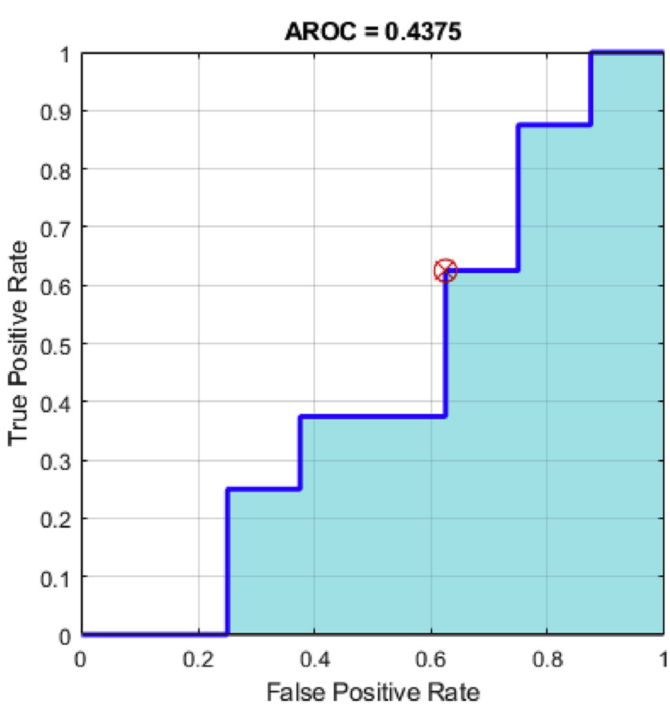


Fig. 12. ROC curve for converging and stabilised period data; phenome representation in Equation (11), min-max normalisation and squared Euclidian distance.

results with the previously presented ones, we can observe that input components such as alignment vector and separation vector provide less distinction between structured and random behaviours than the velocity component. As shown in Fig. 8(a), using the velocity components as input to the value system provides a clearer distinction between most of the structured and random behaviours. However, it still cannot detect Firefly as a structured behaviour (note that it sits far to the right of the x-axis having the highest QE values). Conversely the cohesion and alignment inputs cannot distinguish between random and structured behaviours on their own, but are able to contribute to ordering or the structured behaviours. We conclude from this experiment that while the velocity component seems to be important in distinguishing random and structured behaviours, further studies are necessary to determine the ideal set of input parameters. We leave this as future work and continue the rest of this section describing experiments involving the effects of distance measure, normalisation, and structure of the phenome on the combined input parameters.

4.3. Experiment 3: distinguishing between different collective behaviours using a curiosity model

4.3.1. Aim

The aim of this experiment is to determine whether it is possible to increase the distinction between different behaviours.

4.3.2. Hypothesis

Using Euclidean squared distance to calculate QE should result in a better spread of QE values.

4.3.3. Method

This experiment uses the same method as the previous experiment, but uses the squared Euclidean distance to compute QE in Equations (11) and (12).

4.3.4. Results

Fig. 9 shows that use of the squared Euclidean distance tends to increase the QE of behaviours where QE was above 1, thus, providing a greater distinction between QE values for random and structured behaviours. We thus conduct our remaining experiments using squared Euclidean distance. Fig. 10 shows the ROC curve for this combination.

This experiment and the previous experiment highlight an issue when not normalising the input data, that a single attribute that has a wider range than other attributes will have a greater influence on choice of the best matching unit in the SOM. In the case of our boid data, the velocity attribute had the greatest range in the un-normalised data and thus dominates the distance calculation in the SOM. We now consider two alternative approaches to normalisation to permit other features to have more influence.

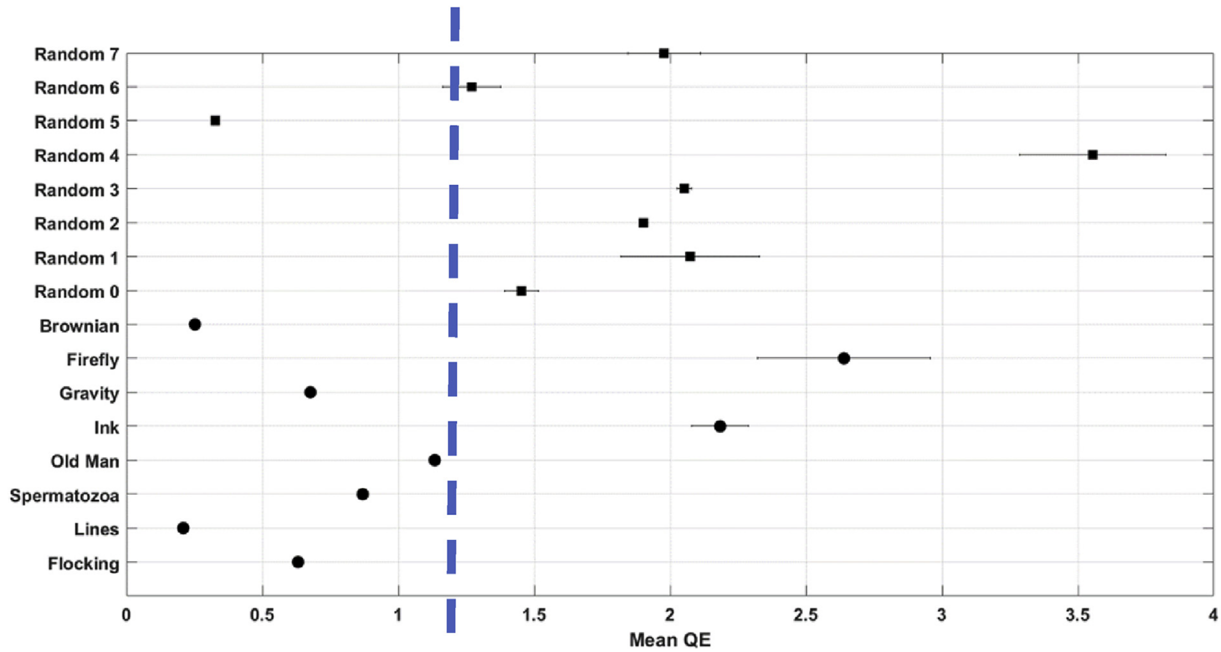


Fig. 13. Mean QE over converging and stabilised period data; phenome representation in Equation (11), z-score normalisation and squared Euclidean distance. We can see it is now possible to select a threshold (blue line) where most structured behaviours have QE below the threshold, and most random behaviours have QE above the threshold.

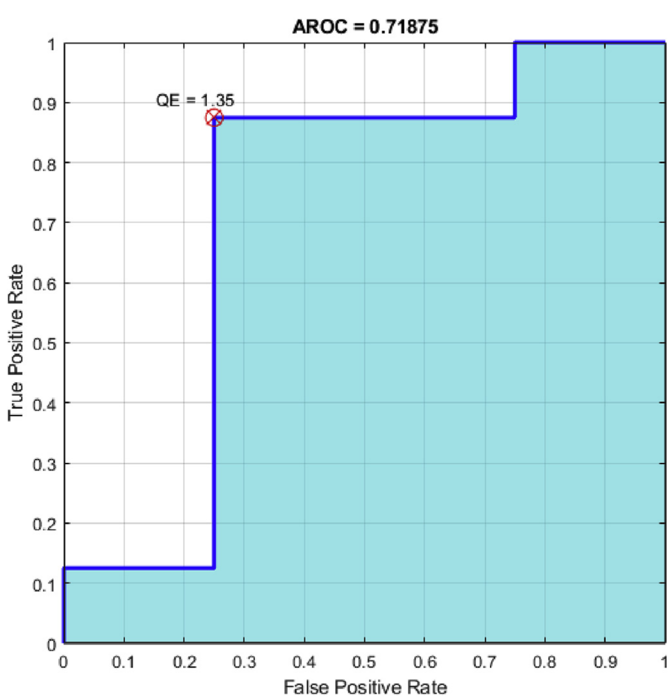


Fig. 14. ROC Curve for converging and stabilised period data; phenome representation in Equation (11), z-score normalisation and squared distance for calculating QE.

4.4. Experiment 4: curiosity model design decisions

4.4.1. Aim

The aim of this experiment is to determine whether it is possible to increase the QE distinction between different behaviours using different data normalisation approaches.

4.4.2. Hypothesis

Normalising data should permit more features to influence the QE results, resulting in better distinction between behaviours.

4.4.3. Method

This experiment uses the same method as the previous experiment, with squared Euclidean distance, but compares two data normalisation approaches.

4.4.4. Results

Fig. 11 shows the results using min-max normalisation and Fig. 13 shows z-score normalisation, while the corresponding ROC curves are showed in Fig. 12 and Fig. 14. We see that min-max of normalisation results in less distinction between random and structured behaviours, while z-score normalisation results in better distinction.

With z-score normalisation (Fig. 13), it becomes possible to select a threshold (blue line) where most (6/8) structured behaviours have QE below the threshold, and most (7/8) random behaviours have QE above the threshold. There are some exceptions, for example, firefly and ink still have higher QE than the other structured behaviours, due to the issues discussed above.

This experiment again confirms that it is plausible that we may be able to distinguish structured from random behaviours using a curiosity model. This result is slightly stronger than Experiment 1 as only one of the random behaviours is below the threshold, while two structured behaviours are above the threshold.

The next experiments examine some alternative variations on this experiment to see if there are more effective representations of O_t and whether it is possible to achieve a more discerning threshold. We use the z-score normalisation in the remaining experiments, as it is the most successful approach identified so far.

4.5. Experiment 5: determining the input of stabilised versus converging data

4.5.1. Aim

The aim of this experiment is to determine whether a curiosity model can better identify structure in the experiences of boids using only

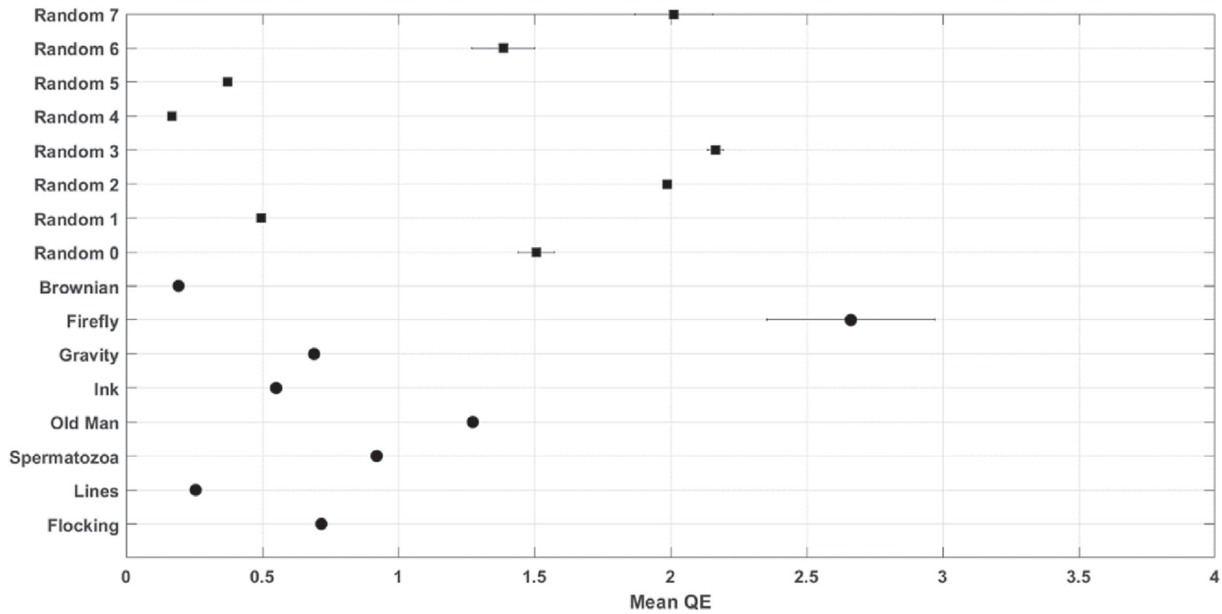


Fig. 15. Mean QE over stabilised period data only; phenome representation in Equation (11), z-score normalisation, squared Euclidean distance for QE. We can see it is no longer possible to select a clear threshold where most structured behaviours have QE below the threshold, and most random behaviours have QE above the threshold.

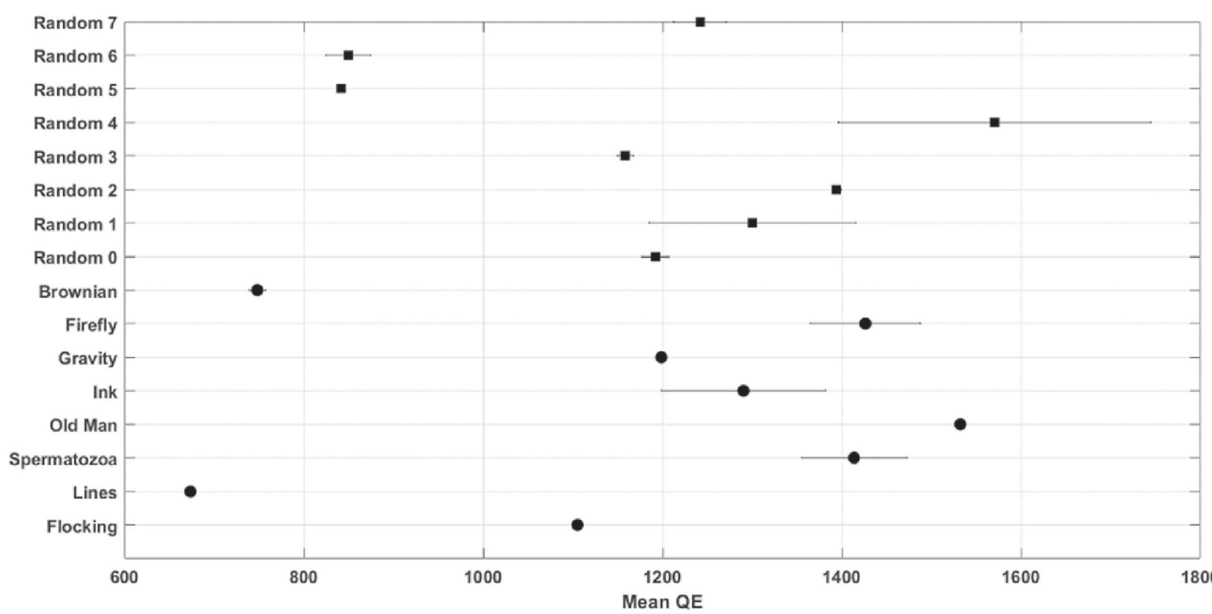


Fig. 16. Mean QE over converging and stabilised period data; phenome representation in Equation (12), z-score normalisation, squared Euclidean distance. We can see it is again not possible to select a clear threshold where most structured behaviours have QE below the threshold, and most random behaviours have QE above the threshold.

stabilised data as input.

4.5.2. Hypothesis

Using only stabilised data should remove random data from the converging period, and should thus result in lower QE for structured behaviours.

4.5.3. Method

The same method as the previous experiment was used, but this time, the first 500 time-steps of data after introduction of a new genome was removed from the experience trajectory. This means that only 200,000 data points were input to the SOM in this experiment. This experiment was repeated 30 times and results averaged.

4.5.4. Results

Fig. 15 shows the mean QE for the eight random and eight structured behaviours. We can see that it is no longer possible to identify a clear threshold where most structured behaviours have QE below the threshold, and most random behaviours have QE above the threshold. We conclude that the inclusion of the converging period data is important to allow the SOM to perturb before analysing another behaviour.

4.6. Experiment 6: Determining the structure of the phenome

4.6.1. Aim

The aim of this experiment is to determine the most effective structure of the phenome.

4.6.2. Method

The same method as the previous experiment was used, this time comparing the representation in Equation (11) to the representation in Equation (12) over both the stabilised and converging periods. Squared Euclidean distance and z-score normalisation are used.

4.6.3. Results

Fig. 16 shows the mean QE for the eight random (top) and eight structured (bottom) behaviours. We can see that it is again not possible to identify a clear threshold where most structured behaviours have QE below the threshold, and most random behaviours have QE above the threshold. In addition, QE is significantly higher (by two orders of magnitude) using the representation in Equation (12), compared to that using Equation (11).

We conclude that the representation in Equation (11) is the most efficient way to identify structure using a curiosity model.

5. Conclusions and future work

In this paper, we have presented a framework that can effectively imbue large groups of artificial agents with the ability to differentiate between certain kinds of 'interesting' structured behaviours and random movements. We have tested this framework under controlled conditions using simulated Reynold's boids as the group of artificial agents. We devised a range of handcrafted structured and unstructured behaviours and showed that it is possible to use the quantisation error of a self-organising map to distinguish between certain structured and unstructured collective behaviours.

We conclude that:

- Using Euclidean distance without normalisation is a plausible means of distinguishing between certain random and structured behaviours.
- The velocity component as input provide the best distinction between structured and random behaviours. However other components are able to order structured behaviours.
- Normalising behaviour data can lead to better distinction between behaviours, but there is a trade-off of in the requirement for lifelong learning as some batch processing to normalise data.
- Inputting individual boid data to the value function is more effective that inputting data from multiple boids in a concatenated format.
- Inputting all data from converging and stabilised periods of a simulation is more effective than using only the stabilised data.

5.1. Future work

As we described in the beginning of the paper, our grand vision is to build a group of artificial agents which will be able to detect as well generate interesting behaviours by themselves. In this paper, we have attempted the first part of this bigger picture. We have designed the behaviours, i.e., the scenarios, by hand-we used our human, outsider (to the boids) knowledge to set the simulation parameters. An immediate future work will be to take the insights gained in this paper and add the additional ability of selecting out behaviour patterns to the boids. With this feature, the group of agents will be able to react and adapt to external situations (such as environmental stimuli) completely by themselves. For example, a group of Unmanned Aerial Vehicle (UAV) can take cue from the territory and form a pattern by themselves. When this full loop works, we will in effect have a group of artificial agents that are capable of forming interesting, useful patterns by intrinsic motivation.

An immediate area of future work will be to further investigate the outliers of the detection mechanisms. As we have discussed earlier, some scenarios showed random behaviour but was detected as structured behaviour and vice versa. We have analysed the preliminary reasons, but further parameters such as the effect of SOM size on QE can be investigated. Another aspect that can be further looked into is the effect of individual parameters on bettering the proposed value system. As our preliminary results have showed, some input components provided better distinction than others. One can investigate the further implications along this line, e.g., using relative velocity of the swarming entities as an input.

Another area of future work originating from the current research would be to decentralize the learning mechanism. Currently, we assume that the learning framework has global access to behaviour data (e.g., position and velocity in the grid) from each of the artificial agents. Hence, the learning happens in a central location and the feedback gained from the learning is dispersed from there as well. However, one may want to revisit this theory and implement a decentralised framework. In that case, the learning may get delegated to individual agents. We will have to answer questions like if we can exploit the local knowledge and still obtain performances similar to the global access. One possible solution will be to divide the whole flock into neighbourhoods. However, that will require novel communication mechanisms.

We have showed that normalising the raw input data can improve the performance of the value system. However, this required access to all of the data. A possible future work to address this issue will be to experiment with setting a window so that we can gather data for a certain period and conduct the processing on that respective batch.

Our focus in this paper was on the robot-centric cycle and the performance of the proposed value system. The other half of the conceptual framework, i.e., the genome-centric cycle can be the focus of a closely related future work. This will involve looking at various evolutionary algorithms and determine their suitability in the given context.

In this paper, we have simulated the behaviours of the agents. Another pertinent work will be to implement this in real robots such as UAVs or robot vehicles moving around in a space. Factors such as physics of real vehicles, wireless connection, communication protocol, electronic latency will bring newer challenges to the proposition. It will be quite interesting to observe the effectiveness of the proposed framework in real life.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Australian Research Council (Grant number DP200100710).

References

[1] M. Beekman, G.A. Sword, S.J. Simpson, Biological Foundations of Swarm Intelligence," Swarm Intelligence, 2008, pp. 3–41.
[2] A. Kolling, P. Walker, N. Chakraborty, Human interaction with robot swarms: a survey, IEEE Transactions on Human-Machine Systems 64 (1) (2016) 9–26.
[3] J. Gomes, P. Urbano, A. Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intelligence 7 (203) (2013) 115–144.
[4] H. Hamann, Evolution of Collective Behaviours by Minimizing Surprise, 2014 presented at the ALIFE2014.
[5] D. Fischer, S. Mostaghim, L. Albantakis, How Swarm Size during Evolution Impacts the Behavior, Generalizability, and Brain Complexity of Animats Performing a Spatial Navigation Task, 2018 presented at the GECCO2018.
[6] D. Samarasinghe, E. Lakshika, M. Barlow, K. Kasmarik, Automatic Synthesis of Swarm Behavioural Rules from Their Atomic Components, 2018 presented at the GECCO2018.
[7] N. Bredeche, E. Haasdijk, A. Prieto, Embodied evolution in collective robotics: a review, (in English), Frontiers in Robotics and AI, Review 5 (12) (2018), https://doi.org/10.3389/frobt.2018.00012. February-22 2018.

[8] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in: Robots and Biological Systems: towards a New Bionics?, Springer, 1993, pp. 703–712.
[9] G. Beni, From swarm intelligence to swarm robotics, in: International Workshop on Swarm Robotics, Springer, 2004, pp. 1–9.
[10] C.W. Reynolds, Flocks, herds and schools: a distributed behavioral model, Comput. Graph. 21 (4) (1987) 25–34.
[11] J. Clark, D. Jacques, Flight test results for UAVs using boid guidance algorithms, in: Presented at the New Challenges in Systems Engineering and Architecting Conference on Systems Engineering Research, CSER, St. Louis, MO, 2012.
[12] J. Harvey, K. Merrick, H. Abbass, Assessing human judgement of computationally generated swarming behaviour, Frontiers in Robotics and AI 5 (13) (2018).
[13] J. Toner, Y. Tu, Flocks, herds and schools: a quantitative theory of flocking, Phys. Rev. 58 (1998), 4828.
[14] T. Vicsek, A. Zaferis, Collective motion, Phys. Rev. 517 (2012) 71–140.
[15] S. Das, U. Halder, D. Malty, Chaotic dynamics in social foraging swarms - an analysis, IEEE Transactions on Systems, Man and Cybernetics B 42 (2012) 1288–1293.
[16] N. Mecholsky, E. Ott, T. Antonsen, P. Guzdar, Continuum modelling of the equilibrium and stability of animal flocks, Phys. Nonlinear Phenom. 241 (2012) 472–480.
[17] J. Harvey, K. Merrick, H. Abbass, Application of chaos measures to a simplified boids flocking model, Swarm Intelligence 9 (2015) 23–41.
[18] M. Begum, F. Karray, Computational intelligence techniques in bio-inspired robotics, in: D. Liu, L. Wang, K.-T. Tang (Eds.), Design and Control of Intelligent Robotic Systems, Springer-Verlag, Berlin Heidelberg, 2009, pp. 1–28.
[19] X. Huang, J. Weng, Inherent value systems for autonomous mental development, Int. J. Humanoid Rob. 4 (2) (2007) 407–433.
[20] M. Frank, J. Leitner, M. Stollenga, A. Forster, J. Schmidhuber, Curiosity driven reinforcement learning for motion planning on humanoids, Front. Neurorob. 7 (25) (2014).
[21] V.R. Kompella, M. Stollenga, M. Luciw, J. Schmidhuber, Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots, in: Artificial Intelligence 247, 2017, pp. 313–335.
[22] G. Martius, R. Der, N. Ay, Information driven self-organization of complex robotic behaviors, PloS One 8 (5) (2013) 1–14.
[23] K. Merrick, A comparative study of value systems for self-motivated exploration and learning by robots, IEEE Transactions on Autonomous Mental Development, Special Issue on Active Learning and Intrinsically Motivated Exploration in Robots 2 (2) (2010) 119–131.
[24] V. Fiore, et al., Keep focussing: striatal dopamine multiple functions resolved in a single mechanism tested in a simulated humanoid robot, Front. Psychol. 5 (2014).
[25] R. Bolado-Gomez, K. Gurney, A biologically plausible embodied model of action discovery, Frontiers in Neurorobotics Special Issue on Value and Reward Based Learning in Neurobots 7 (4) (2013).
[26] O. Sporns, N. Almassy, G. Edelman, Plasticity in value systems and its role in adaptive behaviour, Adapt. Behav. 8 (2000) 129–148.
[27] T. Prescott, F.M. Montes Gonzales, K. Gurney, M. Humphries, P. Redgrave, A robot model of the basal ganglia: behavior and intrinsic processing, Neural Network. 19 (2006) 31–61.
[28] K. Friston, R. Adams, R. Montague, What is value - accumulated reward or evidence? Frontiers in Neurorobotics Special Issue on Value and Reward Based Learning in Neurobots 6 (11) (2012).
[29] L. Pape, et al., Learning tactile skills through curious exploration, Front. Neurorob. 6 (6) (2012).
[30] P.-Y. Oudeyer, F. Kaplan, V.V. Hafner, Intrinsic motivation systems for autonomous mental development, IEEE Trans. Evol. Comput. 11 (2) (April, 2007 2007) 265–286.
[31] K. Doya, E. Uchibe, The cyber rodent project: exploration of adaptive mechanisms for self-preservation and self-reproduction, Adapt. Behav. 13 (2) (2005) 149–160.
[32] K. Merrick, Value systems for developmental cognitive robotics: a survey, Cognit. Syst. Res. 41 (2017) 38–55.
[33] P.-Y. Oudeyer, F. Kaplan, What is intrinsic motivation? A typology of computational approaches, (in English), Frontiers in Neurorobotics, Original Research 1 (6) (2009), https://doi.org/10.3389/neuro.12.006.2007, 2009-November-02.
[34] M. Mirolli, G. Baldassare, Functions and mechanisms of intrinsic motivation, in: G. Baldassare, M. Mirolli (Eds.), Intrinsically Motivated Learning in Natural and Artificial Systems, Springer, Heidelberg, 2013.
[35] K. Merrick, M.L. Maher, in: R. Nugent (Ed.), Motivated Reinforcement Learning: Curious Characters for Multiuser Games, Springer, Berlin, 2009.
[36] R. Saunders, J.S. Gero, Curious agents and situated design evaluations, AI EDAM (Artif. Intell. Eng. Des. Anal. Manuf.) 18 (2) (2004) 153–161.
[37] W. Aguilar, R. Perez y Perez, E.-R. Dev, A computational model of early cogntive development as a creative process, Cognit. Syst. Res. 33 (2015) 17–41.
[38] D. Di Nocera, A. Finzi, S. Rossi, M. Staffa, The role of intrinsic motivations in attention allocation and shifting, Front. Psychol. 5 (2014).
[39] T. Kohonen, Essentials of the self-organizing map, Neural Network. 37 (2013) 52–65.
[40] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[41] S. Haykin, Neural Networks: a Comprehensive Foundation, Prentice Hall PTR, 1994.
[42] U. Asan, S. Ercan, An introduction to self-organizing maps, in: Computational Intelligence Systems in Industrial Engineering, Springer, 2012, pp. 295–315.
[43] Y. Sun, On quantization error of self-organizing map network, Neurocomputing 34 (1) (2000) 169–193, https://doi.org/10.1016/S0925-2312(00)00292-7, 2000/09/01/.
[44] J.M. Wandeto, H.O. Nyongesa, B. Dresp-Langley, Detection of structural change in geographic regions of interest by self organized mapping: las vegas city and lake mead across the years, arXiv e-prints. [Online]. Available: https://ui.adsabs.harvard.edu/#abs/2018arXiv180311125W.
[45] J.M. Wandeto, H. Nyongesa, Y. Rémond, B. Dresp-Langley, Detection of small changes in medical and random-dot images comparing self-organizing map performance to human detection, Informatics in Medicine Unlocked 7 (2017) 39–45, https://doi.org/10.1016/j.imu.2017.03.001. /01/01/2017.
[46] H. Hamann, Evolution of collective behaviors by minimizing surprisal and by micro-macro links, in: Presented at the Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015. Madrid, Spain.
[47] K. Friston, The free-energy principle: a unified brain theory? Nat. Rev. Neurosci. 11 (2) (2010) 127.
[48] J. Lehman, K.O. Stanley, Abandoning objectives: evolution through the search for novelty alone, Evol. Comput. 19 (2) (2011) 189–223.
[49] J. Gomes, P. Urbano, A.L. Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intelligence 7 (2–3) (2013) 115–144.
[50] S. Dahl, M. Cenek, Towards emergent design: analysis, fitness and heterogeneity of agent based models using geometry of behavioral spaces framework, in: Proceedings of the European Conference on Artificial Life, vol. 13, MIT Press, 2016, pp. 46–53.
[51] D. Miner, Predicting and controlling system-level parameters of multi-agent systems, in: 2009 AAAI Fall Symposium Series, 2009.
[52] A. Croitoru, Deriving low-level steering behaviors from trajectory data, IEEE International Conference on Data Mining Workshops (2009) 583–590, https://doi.org/10.1109/ICDMW.2009.76, 6-6 Dec. 2009 2009.
[53] P. Laube, M. Duckham, T. Wolle, Decentralized movement pattern detection amongst mobile geosensor nodes, in: International Conference on Geographic Information Science, Springer, 2008, pp. 199–216.
[54] P. Odonkor, Z. Ball, S. Chowdhury, Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping, Swarm and Evolutionary Computation 46 (2019) 52–68, https://doi.org/10.1016/j.swevo.2019.01.005, 2019/05/01/.
[55] G.S. Nitschke, M.C. Schut, A.E. Eiben, Evolving behavioral specialization in robot teams to solve a collective construction task, Swarm and Evolutionary Computation 2 (2012) 25–38, https://doi.org/10.1016/j.swevo.2011.08.002, 2012/02/01/.
[56] M.S. Innocente, P. Grasso, Self-organising swarms of firefighting drones: harnessing the power of collective intelligence in decentralised multi-robot systems, Journal of Computational Science 34 (2019) 80–101, https://doi.org/10.1016/j.jocs.2019.04.009. /05/01/2019.
[57] D.B. Fogel, What is evolutionary computation? IEEE spectrum 37 (2) (2000) 26–32.
[58] T. Back, U. Hammel, H.-P. Schwefel, Evolutionary computation: comments on the history and current state, IEEE Trans. Evol. Comput. 1 (1) (1997) 3–17.
[59] J. Del Ser, et al., Bio-inspired computation: where we stand and what's next, Swarm and Evolutionary Computation 48 (2019) 220–250.
[60] K. Stanley, R. Miikulainen, Evolving neural networks through augmented topologies, Evol. Comput. 10 (2) (2006) 99–127.
[61] M. Barlow, E. Lakshika, What cost teamwork: quantifying situational awareness and computational requirements in a proto-team via multi-objective evolution, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 24-29 July 2016, 2016, pp. 3525–3532, https://doi.org/10.1109/CEC.2016.7744236.
[62] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, Self-organizing map in matlab: the SOM toolbox, Proceedings of the Matlab DSP conference 99 (1999) 16–17.
[63] T. Vatanen, et al., Self-organization and missing values in SOM and GTM, Neurocomputing 147 (2015) 60–70.