Discovery and Exploration of Novel Swarm Behaviors Given Limited Robot Capabilities

Chapter · March 2018

DOI: 10.1007/978-3-319-73008-0_31

Daniel Brown
University of Texas at Austin

Discovery and Exploration of Novel Swarm Behaviors given Limited Robot Capabilities

Daniel S. Brown, Ryan Turner, Oliver Hennigh, and Steven Loscalzo

Abstract Emergent collective behaviors have long interested researchers. These behaviors often result from complex interactions between many individuals following simple rules. However, knowing what collective behaviors are possible given a limited set of capabilities is difficult. Many emergent behaviors are counter-intuitive and unexpected even if the rules each agent follows are carefully constructed. While much work in swarm robotics has studied the problem of designing sets of rules and capabilities that result in a specific collective behavior, little work has examined the problem of exploring and describing the entire set of collective behaviors that can result from a limited set of capabilities. We take what we believe is the first approach to address this problem by presenting a general framework for discovering collective emergent behaviors that result from a specific capability model. Our approach uses novelty search to explore the space of possible behaviors in an objective-agnostic manner. Given this set of explored behaviors we use dimensionality reduction and clustering techniques to discover a finite set of behaviors that form a taxonomy over the behavior space. We apply our methodology to a single, binary-sensor capability model. Using our approach we are able to re-discover cyclic pursuit and aggregation, as well as discover several behaviors previously unknown to be possible with only a single binary sensor: wall following, dispersal, and a milling behavior often displayed by ants and fish.

D.S. Brown
Computer Science Department, University of Texas at Austin, TX, USA
e-mail: dsbrown@cs.utexas.edu

R. Turner, O. Hennigh, and S. Loscalzo
Air Force Research Laboratory, Rome, NY, USA
e-mail: ryan.turner.10@us.af.mil, oliver.hennigh@us.af.mil, steven.loscalzo@ecs-federal.com

Research performed by all authors while at AFRL, Rome NY.

1 Introduction

Biological swarms have long fascinated researchers and laymen alike. The ability of these swarms to perform complex tasks such as building temperature-controlled nests, comparing potential new nest sites, and coordinating and synchronizing flight patterns [4] have caused some observers to attribute these behaviors to supernatural abilities such as telepathy between flying birds [19] or centralized control from a queen. This notion has continued to persist in the popular media where swarm intelligence is often portrayed as many individuals controlled simultaneously by a single individual. However, despite humans' seemingly innate desire to attribute complex behaviors to higher-level, complex intelligence, researchers in robotics, biology, computer science, and physics continue to show that complex swarm behaviors are often a result of extremely simple local rules. Indeed, much of the research on swarms is focused on finding mappings between sets of specific rules and sets of specific behaviors and can be broken down into two questions: (1) Given a desired behavior, can we determine a set of rules that synthesize this behavior? and (2) Given a specific set of rules, can we determine what the resulting emergent collective behavior will be?

We propose to study a third fundamental, yet less well-defined question, that has received little attention: (3) Given a set of capabilities (i.e., computational power, number and type of sensors, communication range, etc.) what are the possible collective behaviors that can emerge?

Knowing what collective behaviors are possible given a limited set of capabilities is often quite difficult. Many emergent behaviors are counterintuitive and unexpected. Furthermore, even if the rules each agent follows are carefully constructed, it is difficult to predict what behavior will emerge. While much work has studied the problem of designing sets of rules and capabilities that result in a desired collective behavior, little work has examined the problem of characterizing the set of possible collective behaviors that can result from a limited set of rules or behaviors. We take what we believe is the first approach to address this problem.

In particular, we propose a general architecture for discovering a taxonomy of possible emergent swarm behaviors given a set of capabilities. Similar to Wolfram's work on characterizing the behaviors of simple cellular automata [23] we take the approach of a naturalist and seek a taxonomy of possible collective behaviors that can result from a set of capabilities. To form a taxonomy of the emergent behaviors that result from a given capability model, we use an approach based on novelty search [13] which allows us to explore the space of possible behaviors in an objective-agnostic manner. By optimizing novelty rather than any particular task or objective, and keeping track of novel behaviors in an archive, we are able to generate a large number of controllers that synthesize a wide variety of behaviors. Given this set of explored behaviors we use dimensionality reduction and clustering techniques to explore and categorize the space of possible behaviors.

We evaluate our architecture on a simple agent capability model that assumes only a single line-of-sight sensor that has only two possible values. Despite this parsimonious agent model, we show that there is a surprising variety of interesting collective behaviors. This approach allows us to "re-discover" previously studied computation-free circling and aggregation [7] as well as identify several behaviors previously unknown to be possible given a swarm of memory-less, single-sensor agents. These new behaviors include wall following, dispersal, and coordinated milling often found in ants and schools of fish [18, 21].

Our main contributions are summarized as follows:

- We propose the first general architecture to explore and form a taxonomy of the space of possible behaviors given a limited-capability robot model.
- We demonstrate the feasibility of using novelty search and archive clustering to generate a set of representative behaviors for a simple single-sensor robot capability model.
- We validate our approach by showing that it discovers previously known behaviors, as well as discovering several behaviors previously unknown to be possible for swarms of single-sensor robots.

2 Problem Statement

The main question this research seeks to answer can be stated as follows:

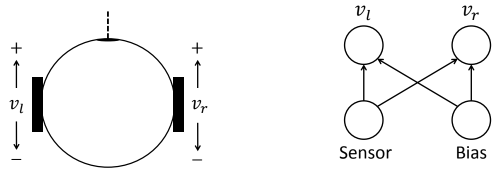
What is the set of possible emergent behaviors in a swarm of robots possessing a specific set of individual capabilities?

To formalize this problem we provide the following definitions. We define a *capability model* as a three-tuple $\langle S, M, A \rangle$ composed of sensors S, memory and computational processing resources M, and actuators A. The capability model captures what information a robot can collect from the world, how it can process that information, and how it can change its state and the state of its environment. Given a set of N agents each with capability model $c_i = \langle S_i, M_i, A_i \rangle$ for $i = 1, \ldots, N$, we define the capability model of the swarm as $\mathscr{C} = \{c_i : i = 1, \ldots, N\}$. [1]

We define an *emergent behavior* as a global pattern or structure resulting from local interactions between a collection of agents. We denote the *set of possible emergent behaviors* as $\mathscr{B}$.

We also define an *environment* $\mathscr{E}$. Given a capability model $\mathscr{C}$ and an environment $\mathscr{E}$ we desire to find a mapping from capabilities and environments to behaviors. Note that we have made no mention of how the capabilities are used by an agent with capability $c_i \in \mathscr{C}$ in environment $\mathscr{E}$. Rather than specifying the controller, we desire to find the image of a function Φ that maps from all possible controllers that can be instantiated on capability model $\mathscr{C}$ for environment $\mathscr{E}$ to the set of possible emergent behaviors $\mathscr{B}$. Thus we desire to find $\mathscr{B}$ where

[1] This formalism also captures homogeneous swarms which can be modeled by letting $c_i = c$, $\forall i$.



(a) Robot (b) Controller

Fig. 1 Simple neural network controller for a single binary line-of-sight differential drive robot. The sensor value is input to the network which outputs the left and right wheel velocities.

$$\Phi : \mathscr{U}(\mathscr{C}) \times \mathscr{E} \rightarrow \mathscr{B} \tag{1}$$

where $\mathscr{U}(\mathscr{C})$ is the controller space resulting from the capability model $\mathscr{C}$.

It is worth noting that in general the controller space is enormous. For example, consider the popular, and very simple, Kilobot robot platform [17]. A Kilobot could be modeled as a three-tuple $\langle S_k, M_k, A_k \rangle$ where S_k includes the infrared receiver and the ambient light sensor, M_k represents all programs programmable on an Atmega328 microprocessor with 32K of memory, and A_k includes the speed of the two vibrating motors, as well as the output of the infrared LED transmitter.

As a first step towards discovering novel collective behaviors, we examine an even simpler capability model based on the e-puck robot [16], where S is a single, on/off, binary sensor, A consists of two differential drive wheels, and M is a fully connected 2 layer neural network connecting sensor inputs to wheel velocities. This capability model is depicted in Figure 1. Even for this simple example, the controller space is $\mathbb{R}^4$, the space of all neural network weights.

Given the size of the controller-space, brute force or analytical methods for determining the mapping Φ would be incredibly difficult, if not impossible. Thus, we resort to a genetic search methodology outlined in the next section.

3 Behavior Discovery Architecture

Our proposed architecture relies heavily on novelty search as a means for exploring the space of emergent behaviors. Lehman and Stanley [13] proposed novelty search as a way to avoid getting stuck in local minima and to overcome deceptive fitness landscapes in genetic algorithms. Rather than using an objective function that rewards fitness, they show that simply trying to maximize the novelty of an evolved behavior will often generate a solution to the original problem more quickly than using pure fitness.

Rather than measuring similarity on the actual genotype, novelty measures similarity on the phenotype—the actual behavior resulting from executing the evolved controller. There are many potential ways for a user to define a behavior space; however, behavior spaces are typically represented by a vector with components that contain statistics over the state of the simulation collected periodically [13, 10]. Given a representation of a learned behavior in d dimensional behavior space, the typical measure of novelty as used in [13] is the sparseness of a point $b \in \mathbb{R}^d$ in behavior space, defined as $\texttt{Novelty}(b) = \frac{1}{k}\sum_{i=0}^{k} dist(b, \beta_i)$,where β_i is the ith nearest neighbor of b with respect to the distance metric $dist$. The nearest neighbor calculations take into consideration both individuals in the current population as well as previous members of the population that are stored in an archive that is updated each generation.

Novelty search has been used successfully to evolve many different types of single agent [6] and swarm behaviors [10]. The success of novelty search is attributed to its success in exploring the behavior space and discovering successively more complex behaviors [13]; however, to the best of our knowledge, our approach is the first to use novelty search purely for exploration without a specific task in mind.

Our approach proceeds as follows, we first start with a random population of controllers. Each controller is evaluated in our environment and a feature vector describing the resulting behavior is calculated. Given these behavior features, each policy is evaluated for novelty. Based on some archiving scheme, some or all of the policies are stored in an archive. Then, artificial evolution and mutation is used to create the next generation of controllers, where novelty is used as the fitness score. This process is repeated until it reaches some stopping criterion, at which point the discovered behaviors in the archive are clustered and representatives of each cluster are used to form an approximate taxonomy of possible emergent behaviors. The basic algorithmic outline is given in Algorithm 1.

Algorithm 1 `NovelBehaviorDiscovery`

Require: environment $\mathscr{E}$, capability model $\mathscr{C}$, and controller model $\mathscr{U}$
  $P \leftarrow \texttt{InitializePolicies}(\mathscr{U}(\mathscr{C}))$ ▷ Generate initial population P_0
  archive $\leftarrow \texttt{InitializeArchive}(P)$
  while stopping criterion not met **do**
    for each policy p_i in population P **do**
      $f_i \leftarrow \texttt{ExtractFeatures}(p_i, \mathscr{E})$ ▷ extract features by evaluating policy
      $n_i(t) \leftarrow \texttt{Novelty}(f_i, \textbf{archive})$ ▷ evaluate novelty
      if $\texttt{addToArchive}(\langle p_i, f_i(t)\rangle)$ **then**
        archive$.\texttt{add}(\langle p_i, f_i\rangle)$ ▷ store individual in novelty archive
      end if
    end for
    $P \leftarrow \texttt{Update}(P, f)$ ▷ update population using a GA with fitness replaced by novelty
  end while
  $K \leftarrow \texttt{Cluster}(\textbf{archive})$ ▷ Cluster on archive and return K representative behaviors
  return K ▷ Return cluster representatives as taxonomy

4 Implementation

4.1 Simple Capability Model

We use a homogeneous capability model based on Gauci et al.'s recently proposed single, binary-sensor, line-of-sight robots [7]. Each robot is equipped with a differential drive and a single line-of-sight sensor that provides it with one bit of information that lets the robot know whether it is facing another agent (see Figure 1(a)).

Using this simple robot capability model, Gauci et al. optimized controllers to perform aggregation [7] demonstrating that highly robust aggregation was possible despite extremely limited capabilities. Subsequent research has shown that increasing the robot capability to include trinary sensors allows specific controllers to be evolved to accomplish tasks such as collecting pucks [8] and forming a perimeter, aggregating to a specific location, and foraging [11]. Our work extends previous work on simple, single-sensor swarms by examining the entire space of collective behaviors that are possible given a swarm of robots whose input is limited to a single, binary, line-of-sight sensor.

4.2 Simulation Environment

Due to the infeasibility of evaluating thousands of controllers on physical robots, we follow the common practice of using a simulator [7, 10] to allow rapid exploration of the behavior space. Following recent work on novelty search for swarms [10], we used the MASON multi-agent simulation environment [14] to simulate the physics of simple differential drive robots modeled after the e-puck robot [16]. Agent movement is simulated within a frictionless walled region of 50 by 50 units, where one unit equals one robot diameter. Each agent has two differential drive wheels. The controller for each robot is a simple neural network with one input node for the binary sensor and one output node for each wheel. The output is fixed in the range [-1,1] by a tanh function. The actual robot velocity on each wheel is then the output multiplied by the maximum speed. Figure 1 shows a representation of the robot and the controller architecture.

Following the approach used by [10], we use NeuroEvolution of Augmenting Topologies (NEAT) [20] with novelty search to optimize the weights on the neural network controller shown in Figure 1(b). We computed novelty using the 15 nearest neighbors in the archive, consistent with best practices found by Gomes et al. [9]. In our work we are interested in the full space of behaviors so we keep all individuals from each generation and add them to the novelty archive.

Table 1 Behavior vector feature descriptions

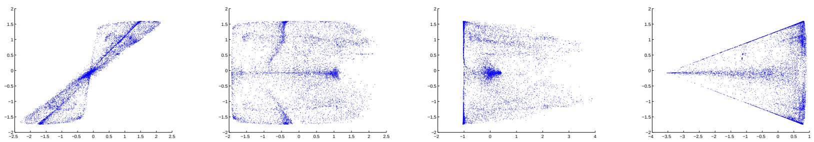
Name	Equation	Name	Equation
average speed	$\frac{1}{N}\sum_{i=1}^{N}\lVert v_i\rVert_2$	scatter	$\frac{1}{R^2\cdot N}\sum_{i=1}^{N}\lVert x_i-\mu\rVert^2$
ang. momentum	$\frac{1}{R\cdot N}\sum_{i=1}^{N}(v_i\times(x_i-\mu))$	group rotation	$\frac{1}{N}\sum_{i=1}^{N}\left(v_i\times\frac{x_i-\mu}{\lVert x_i-\mu\rVert}\right)$
radial variance	$\frac{1}{R^2\cdot N}\sum_{i=1}^{N}\left(\lVert x_i-\mu\rVert-\frac{1}{N}\sum_{i=1}^{N}\lVert x_i-\mu\rVert\right)^2$		

4.3 Behavior vector

To explore the impacts of different behavior features on the discovered behaviors, we used a five element behavior vector. The five element behavior vector measures the average speed, scatter, radial scatter, angular momentum, and group rotation. *Average speed* measures the average speed of the agents in the swarm. *Scatter* [7] measures the average squared distance of the agents to the center of mass μ where $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$. *Radial variance* measures the variance of the distance of the agents to the center of mass μ. *Angular momentum* measures the true angular momentum about the center of mass of the swarm. Finally, *group rotation* measures a normalized angular momentum, ignoring the length of the moment arm [21]. R is the world radius (distance from center of world to corner in the case of a square world). The value R is used to normalize several of the features to be invariant to the size of the world. To create our final behavior vector, we used a sliding window average of each feature over the last 100 time steps. The details of these behavior vectors are shown in Table 1.

4.4 Dimensionality reduction and clustering

While it is possible to cluster in the high-dimensional behavioral space, interpreting the clusters becomes more difficult and single behaviors tend to be falsely split into multiple clusters. To reduce the dimensionality of our data we use t-distributed stochastic neighbor embedding (t-SNE) [22], a state-of-the-art dimensionality reduction technique shown to outperform other standard techniques such as PCA, Sammon mapping, and Isomap. t-SNE is especially suited for taking high-dimensional data that lies on several low-dimensional manifolds and mapping it to a 2-dimensional mapping that preserves and reveals this structure. We used t-SNE [22] to reduce the dimensionality and compare two types of clustering: k-means and hierarchical single-link (min distance) agglomerative clustering to partition the behaviors.



(a) group rotation vs. angular momentum (b) group rotation vs. scatter (c) group rotation vs. radial variance (d) group rotation vs. average speed

Fig. 2 Two-dimensional projections of the behavior space. The y-axis is group rotation.

5 Results and Analysis

Using novelty search we ran 100 generations of 100 populations using NEAT to obtain an archive of 8020 data points[2] in 5 dimensional behavior space. Each experiment used 30 simulated robots. Figure 2 shows several 2-dimensional projections of the 5-dimensional data. Based on these results we see that there is definite structure captured by these features. We also see that group rotation and angular momentum are highly correlated, as expected, but do capture different information.

We used the MATLAB implementation of t-SNE[3] to map our 5-dimensional data to 2 dimensions. The resulting 2-dimensional data has definite structure and visible clusters as seen in Figure 4, as opposed to the projected data shown in Figure 2. We performed the dimensionality reduction before clustering to both reduce the computation time required for clustering and to make the results easy to visualize.

5.1 k-Medoids

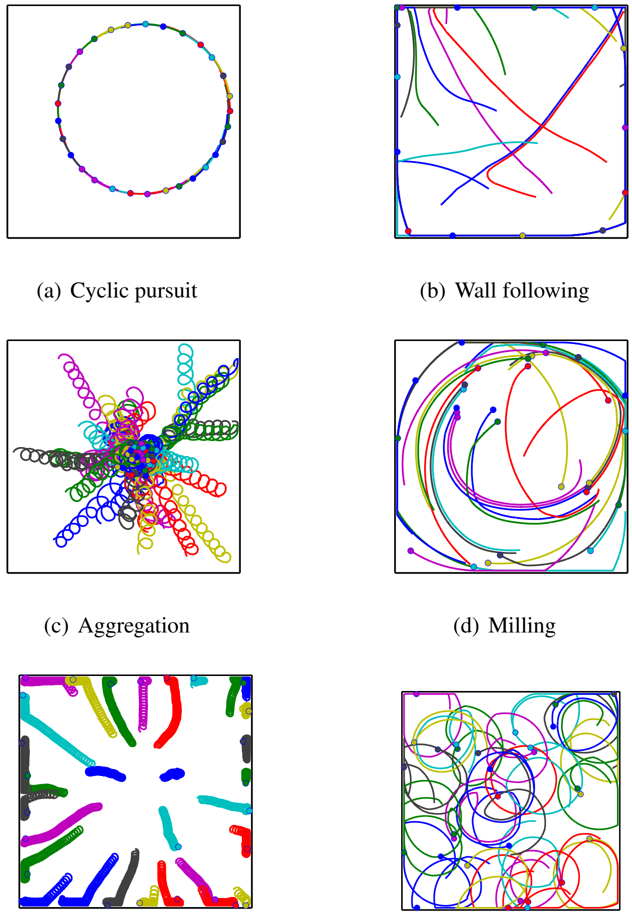
k-Means is the de facto clustering algorithm to begin data exploration. We use a related clustering algorithm called k-Medoids that returns k actual data points as cluster centers. We use the medoids as the representative behaviors.

Because our goal is to discover and categorize emergent behaviors, we do not have any way of knowing the number of clusters ahead of time. Thus, we explored the resulting clusters for values of k between 2 and 10 and visually inspected the behavior of the resulting medoids for each value of k. The results are shown in Table 3(a) using the abbreviations listed in Table 2. Sample trajectories of these behaviors are shown in Figure 3.

An example of the results is shown in Figure 4(a). We evaluated each cluster by comparing the medoids. While k-Medoids works well for forming equally sized

[2] Due to the elitism feature of NEAT, the best performing policies (most novel) in one generation are kept in the population for the next generation. Thus, the algorithm explores fewer than 10,000 unique controllers.

[3] https://lvdmaaten.github.io/drtoolbox/



(a) Cyclic pursuit (b) Wall following

(c) Aggregation (d) Milling

(e) Dispersal (f) Random

Fig. 3 Partial trajectories of swarm behaviors possible given a single, line-of-sight sensor. *Cyclic pursuit* forms a perfectly spaced, revolving circle. *Wall following* consists of agents spreading out to the boundary and then sliding along the walls. In *Aggregation*, the robots spiral into a single cluster. Robots in the *Milling* behavior constantly chase each other around in circles without ever forming a perfect circle. *Dispersal* is the opposite of the aggregation behavior, and results in agents spiraling away from each other. Finally, some behaviors were classified as *Random* due to agents never forming a coherent behavior.

Table 2 Abbreviations used to describe common behaviors.

abbreviation	description	abbreviation	description
cycp	cyclic pursuit	mill	milling
wall	wall slide	rand	individual circling w/out emergence
aggr	aggregation	cw	clockwise motion
disp	dispersal	ccw	counterclockwise motion

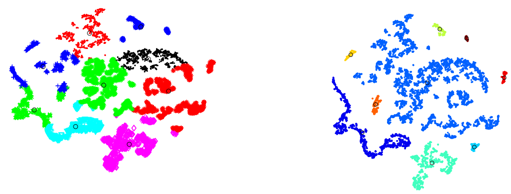
Table 3 Results of examining centers from k-Medoids and Hierarchical clustering on the t-SNE embedded behavioral data. #x denotes that # cluster medoids were of that type.

(a) k-Medoids

	cycp		wall		aggr	disp	mill		rand
k	cw	ccw	cw	ccw			cw	ccw	
2	x						x		
3				x			x	x	
4				x	x	x		x	
5						x	2x	2x	
6	x	x	x	x		x	x		
7	x	x	x		x	x	x	x	
8	x		x	2x	x	x		x	x
9	x	x	2x	x	x	x	x		
10	x	2x	2x	x	x	x		x	x

(b) Hierarchical Clustering

	cycp		wall		aggr	disp	mill		rand
k	cw	ccw	cw	ccw			cw	ccw	
2	x								x
3	x	x							x
4	x	2x							x
5	x	2x			x				x
6	x	3x	x		x				
7	x	4x				x			x
8	x	4x			2x	x			
9	2x	4x	x		x	x			
10	4x	4x			x	x			



(a) k-Medoids (b) Hierarchical Clustering

Fig. 4 An example clustering from k-medoids with k = 6. This approach partitions the t-SNE embedded behavior space into roughly equal partitions.

clusters, it also ignores much of the structure in the 2-dimensional embedding. This is a common downside to k-Means and k-Medoids clustering.

5.2 *Hierarchical Single-Link Clustering*

The previous section showed that just a simple k-Medoids approach allows us to partition the behavior space into roughly equal cells. We then examined the medoids returned from each cluster to determine how many distinct behaviors were discovered. However, even a superficial examination of the 2-dimensional t-SNE embedding and the clustering shown in Figure 4(a) shows that clusters in many cases do not fit well with the underlying structure. To try to remedy this we next examined hierarchical agglomerative single-link clustering.

As shown in Figure 4(b) hierarchical clustering sequentially picks out isolated islands in the embedded 2-d space. However, the resulting cluster centers do not exhibit the range of behaviors found through k-Medoids. We inspected the clusters and representative behaviors and found that many of the small clusters were simply different variations of cyclic pursuit with variations in radius and speed.

6 Discussion

Our clustering results show that k-Medoids provides the most representative sampling of distinctly different behaviors, while hierarchical clustering tended towards finding different variations of cyclic pursuit while failing to find the milling behavior. One method is not clearly better than the other. If finding the largest number of clearly distinct behaviors is desired, then k-Medoids seems to perform the best. On the other hand, if a more nuanced definition of emergent behavior is desired, the hierarchical clustering seems better at uncovering the variations within behaviors.

Our clustering analysis found six possible behaviors. However, one of them, random circling, appears to not have any kind of collective behavior but is instead just a collection of robots moving in circles with no emergent properties. Thus, we focus on the five behaviors that we classify as emergent: cyclic pursuit, aggregation, wall following, dispersal, and milling. Cyclic pursuit resulting from robots with a single, binary sensor was first mentioned by Gauci et al. [7], but treated as a local minima in the search for an aggregation controller. Our method is able to "rediscover" this emergent behavior without an explicit objective. Cyclic pursuit is also well studied problem in control theory [15]; however, these problems are often solved using complex policies requiring positional and heading information, as opposed to the simple capability model we study here.

Aggregation is another behavior re-discovered by out method. Gauci et al. [7] first explored the problem of using a single-binary sensor to investigate whether they could evolve an aggregation algorithm that required no computation or memory.

Unlike cyclic pursuit or aggregation, the wall following behavior found by our algorithm is, to the best of our knowledge, a novel behavior for our capability model. While this behavior is a result of our specific environment, namely a walled environment without friction, it shows the power of our method in finding a novel behavior unknown to be possible in a swarm of memoryless single sensor robots. While it is possible to argue that wall following is simply a circle that is too big for the world size, this ignores the fact that the space of behaviors is inherently tied to the characteristics of the environment. As stated in our problem formalism, we are interested in discovering the different behaviors that are possible given both a capability model as well as an environment. Thus, for our specific environment, we argue that the wall following and cyclic pursuit are different behaviors due to their unique movement patterns and behavioral features.

The dispersal behavior is also a novel behavior that has not been previously shown to exist for single sensor swarms. Given that aggregation has previously been shown possible, it is not surprising that dispersal is also possible; however, the fact that our method finds both aggregation and dispersal shows the effectiveness of our approach.

The final emergent behavior that our method discovered in the milling, or torus behavior. The existence of this behavior is rather remarkable given the limited capability model we studied. It is well known that ants and fish form these types of milling patterns in the wild. However, we believe this is the first demonstration of these patterns shown to be possible with no memory and only a single bit of sensory information. This behavior is well-studied in the swarming community and is one of the four fundamental group types shown to emerge from the celebrated Couzin's model [5]. However, unlike Couzin's model, which makes strong assumptions about every agent being able to sense and respond to its neighbor's relative positions and velocities, we have discovered that a milling behavior is possible using only a single binary sensor.

7 Conclusions and Future Work

In this paper we formalized the problem of determining the emergent behaviors possible given a limited set of capabilities. Applying our method to a single binary sensor model, first proposed by Gauci et al. [7], we found that our method was able to rediscover a cyclic pursuit circling behavior, as well as aggregation. We also discovered three new behaviors not previously shown to be possible given our assumed capabilities: wall following, dispersal, and milling. We investigated both k-means clustering and hierarchical clustering after reducing the dimensionality of our data. We found that the centers of the k-Medoids clusters resulted in a wider variety of behaviors than the centers of clusters obtained from hierarchical clustering. Hierarchical clustering found fewer distinctly different behaviors, but was able to better select for variations within behaviors, such as speed, rotation direction, and radius.

While we believe that the problem we have studied is of fundamental and practical importance, we acknowledge the fundamental subjectivity in assigning boundaries between behavior types. Though this is an inherently subjective problem that may never admit an objective solution, we believe we have made some progress towards the goal of discriminating between qualitative behavioral groups in a principled way. While emergent behaviors will always, in some sense, be relative to the eye of the beholder, our approach allowed us to find a set of visually distinct behaviors, some of which were not previously known to exist for the single binary sensor capability model. While is it still difficult to know how well our approach will scale to more complex capability models, our proposed methodology could be useful to both scientists wanting to understand why some collective behaviors are present in a given animal species as well as engineers wishing to explore and design emergent behaviors to accomplish different tasks.

Future work should extend our method to investigate the space of possible emergent behaviors given more complex models, such as multiple sensors with more than two possible inputs, limited communication between agents, and more complex environments that include obstacles and movable items. It also remains to be seen how changes in the size and shape of the environment and number of robots affect the behaviors that are possible.

Future work should also investigate better techniques for determining what features are important for clustering. Our results have shown the difficulty in defining a behavior and in partitioning the explored space of behaviors without requiring a user to visually inspect the results and hand-tune parameters such as the number of clusters. There has been some work on using hand-crafted or learned features for classifying swarm behaviors [2, 1]; however, these methods are designed for already known behaviors, whereas we are interested in finding features that allow us to discover new behaviors. One possible avenue toward better disambiguation between behaviors would be to leverage crowd sourcing or machine learning. We hypothesize that human feedback combined with more advanced machine learning techniques such as deep convolutional neural networks [12] could allow us to better learn behavior features and similarities and improve the scalability of our approach.

Finally, we note that while discovering emergent behaviors is an interesting scientific question, there are also many open questions about how to interact with and use these behaviors. As more complex emergent behaviors are discovered, we hope there will also be research into how to use simple interactions with a swarm, either by changing the behavior of a subset of the agents [3], or even by changing the environment [11], to control and switch between different collective behaviors to accomplish interesting tasks.

References

1. M. Berger, L. M. Seversky, and D. S. Brown. Classifying swarm behavior via compressive subspace learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5328–5335. IEEE, 2016.
2. D. S. Brown and M. A. Goodrich. Limited bandwidth recognition of collective behaviors in bio-inspired swarms. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 405–412, 2014.
3. D. S. Brown, S. C. Kerman, and M. A. Goodrich. Human-swarm interactions based on managing attractors. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, pages 90–97. ACM, 2014.
4. I. D. Couzin and J. Krause. Self-organization and collective behavior in vertebrates. *Advances in the Study of Behavior*, 32(1), 2003.
5. I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of theoretical biology*, 218(1):1–11, 2002.
6. S. Doncieux and J.-B. Mouret. Behavioral diversity with multiple behavioral distances. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1427–1434. IEEE, 2013.

7. M. Gauci, J. Chen, T. J. Dodd, and R. Groß. Evolving aggregation behaviors in multi-robot systems with binary sensors. In *Distributed Autonomous Robotic Systems*, pages 355–367. Springer, 2014.
8. M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Clustering objects with robots that do not compute. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, pages 421–428, 2014.
9. J. Gomes, P. Mariano, and A. L. Christensen. Devising effective novelty search algorithms: A comprehensive empirical study. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, pages 943–950. ACM, 2015.
10. J. Gomes, P. Urbano, and A. L. Christensen. Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2-3):115–144, 2013.
11. M. Johnson and D. S. Brown. Evolving and controlling perimeter, rendezvous, and foraging behaviors in a computation-free robot swarm. In *International Conference on Bio-inspired Information and Communications Technologies*, 2015.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
13. J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.
14. S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.
15. J. A. Marshall, M. E. Broucke, and B. A. Francis. Formations of vehicles in cyclic pursuit. *Automatic Control, IEEE Transactions on*, 49(11):1963–1974, 2004.
16. F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, 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. IPCB: Instituto Politécnico de Castelo Branco, 2009.
17. M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3293–3298. IEEE, 2012.
18. T. C. Schneirla. *A unique case of circular milling in ants, considered in relation to trail following and the general problem of orientation*. Citeseer, 1944.
19. E. Selous. Thought transference (or what?) in birds, 1931.
20. K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
21. K. Tunstrøm, Y. Katz, C. C. Ioannou, C. Huepe, M. J. Lutz, and I. D. Couzin. Collective states, multistability and transitional behavior in schooling fish. *PLoS Comput Biol*, 9(2):e1002915, 2013.
22. L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.
23. S. Wolfram. *A new kind of science*, volume 5. Wolfram media Champaign, 2002.