Evolving Herding Behaviour Diversity in Robot Swarms

Scott Hallauer
HLLSCO001@myuct.ac.za
Department of Computer Science
University of Cape Town
Cape Town, South Africa

Geoff Nitschke
gnitschke@cs.uct.za
Department of Computer Science
University of Cape Town
Cape Town, South Africa

Emma Hart
e.hart@napier.ac.uk
School of Computing
Edinburgh Napier University
Edinburgh, Scotland, UK

ABSTRACT

Behavioural diversity has been demonstrated as beneficial in biological social systems, such as insect colonies and human societies, as well as artificial systems such as large-scale software and swarm-robotics systems. Evolutionary swarm robotics is a popular experimental platform for demonstrating the emergence of various social phenomena and collective behaviour, including behavioural diversity and specialization. However, from an automated design perspective, the evolutionary conditions necessary to synthesize optimal collective behaviours (swarm-robotic controllers) that function across increasingly complex environments (difficult tasks), remains unclear. Thus, we introduce a comparative study of behavioural-diversity maintenance methods (swarm-controller extension of the MAP-Elites algorithm) versus those without behavioural diversity mechanisms (Steady-State Genetic Algorithm), as a means to evolve suitable degrees of behavioural diversity over increasingly difficult collective behaviour (sheep-dog herding) tasks. In support of previous work, experiment results demonstrate that behavioural diversity can be generated without specific speciation mechanisms or geographical isolation in the task environment, although the direct evolution of a functionally (behaviorally) diverse swarm does not yield high task performance.

CCS CONCEPTS

• **Computing methodologies** → **Evolutionary robotics**.

KEYWORDS

Swarm-Robotics, Quality-Diversity Methods, Behavioural Diversity

1 INTRODUCTION

Behavioural (or functional) diversity in natural populations has been shown to improve both their robustness and overall performance. For example, functional diversity in bee populations has been demonstrated to increase pollination rates [3]. Furthermore,

cultural diversity amongst humans in the workplace is related to better problem solving with different ways of approaching the same tasks [5]. A more general study using agent-based modelling also found that a diverse team of problem solvers is more likely to outperform a team of high-ability problem solvers [4].

Our aim, in this project, is to investigate methods for generating behavioural diversity in a collective herding task with the end goal of creating a heterogeneous robot swarm. It is envisioned that the practical benefits might involve sub-task specialisation amongst robots which improve overall swarm task performance (e.g. some robots which circle the target zone and others which search the environment periphery for more agents to herd). This is approached using two evolutionary algorithms: Steady-State Genetic Algorithm (SSGA) and Multi-dimensional Archive of Phenotypic Elites (MAP-Elites). The key difference between these algorithms is that MAP-Elites explicitly promotes diversity, whereas SSGA does not. We compare the effectiveness of these algorithms from the two perspectives of either evolving behaviours for a homogeneous swarm (that could be combined in future to make a heterogeneous swarm) or directly evolving a heterogeneous swarm.

2 METHODS

2.1 Simulation Task Environment

The collective herding task is simulated using an extended version of the Roborobo! multi-agent simulation framework [1]. A swarm of N robots, called "dogs", is assigned the objective of capturing a dispersed flock of M agents, called "sheep", inside a centrally-located target zone. Sheep actively avoid entering the target zone, unless pursued by a dog. Once they enter the target zone, they are considered "captured" and removed from the simulation. The 2D environment is bounded on all sides by walls. Figure 1 provides a visual snapshot of the environment during a simulation run.

2.2 Agent Representation

Two types of agent, dogs and sheep, are simulated in this task environment. Although both incorporate a similar body shape and sensory configuration, there are major differences in the controllers used (and thus behaviours elicited).

2.2.1 Dogs. These are the robot agents that undergo neuro-evolution for the herding task. They incorporate a simple, circular morphology similar to Khepera or e-puck robots. In terms of sensory configuration, a radar-type proximity sensor is used which detects the nearest instance of each type of object (dog, sheep and wall) within a specific range and field of view, where dog radar sensors detect objects at 15-degree intervals.
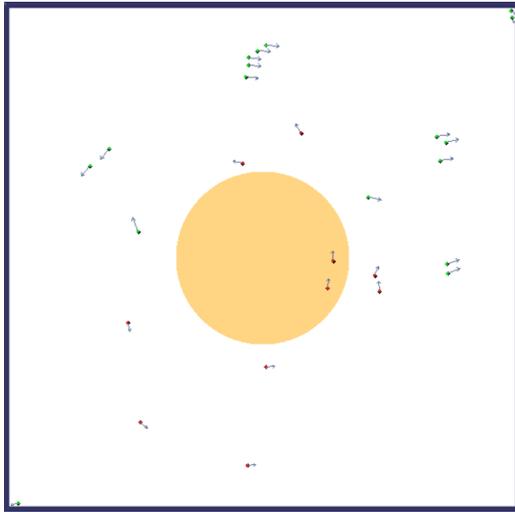
Figure 1: Simulation environment for the collective herding task. Red agents are dogs, green agents are sheep and the yellow circle represents the target zone.

Neuro-evolution Parameters	
Replications per experiment (runs)	20
Generations per experiment run	200
Trial evaluations per generation	3
Initial population size	100
ANN dimensions (nodes): input / hidden / output	9 / 10 / 2
MAP-Elites archive: dimensions / bins	3 / 729
Crossover probability	0.5
Mutation probability	0.2
Simulation Parameters	
Time steps per trial evaluation	800
Initial agent positions	Random (outside target zone)
Dog team size: easy / medium / difficult	20 / 15 / 10
Sheep flock size: easy / medium / difficult	10 / 15 / 20
Dog translation speed: easy / medium / difficult	1 / 0.75 / 0.5
Sheep translation speed: easy / medium / difficult	0.5 / 0.75 / 1
Arena size (width × height)	600×600
Target zone size (radius)	100
Dog radar proximity sensor: range / FOV	(0, 100] / [-90°, 90°]
Sheep radar proximity sensor: range / FOV	(0, 50] / [-180°, 180°]
Sheep object avoidance (radius): wall / dog / sheep	15 / 50 / 5
Sheep target zone avoidance: radius / strength	50 / 0.25

Table 1: Neuro-evolution and simulation parameters. Configuration options used for the neuro-evolution process and all experimental simulations of the collective herding task.

A fully-connected artificial neural network (ANN) is implemented for each dog's controller. The topology consists of 9 input nodes, 10 hidden nodes and 2 output nodes, resulting in a total of 110 connection weights for the genome. The 9 input nodes include distance and angle values from 3 radar sensors (one for each object type), distance and angle values from a target zone sensor, and a bias input which is set to a constant value of 1. Distance values are normalised in the range [0, 1], where 0 is undetected and 1 is as close as possible. Angle values are normalised in the range [-1, 1], where -1 is -180 degrees and 1 is +180 degrees. The 2 output nodes include the dog's translation value in the range [-1, 1] (where -1 is maximum translation speed backwards and +1 is maximum translation speed forwards) and the dog's rotation value in the range [-1, 1] (where -1 is maximum rotation speed to the left and +1 is maximum rotation speed to the right). The tanh activation function is used between network layers.

2.2.2 Sheep. These are the heuristic agents that wander around the arena and should be herded into the target zone. The same circular morphology employed by the dog agents is also used by the sheep agents. Additionally, the same radar-type proximity sensor is used, although different range and field of view values may be set.

A variation of the "boids" algorithm for flocking behaviour [8] is implemented for each sheep's controller. This controller remains static throughout the evolutionary process and guides the movement of the sheep using simple avoidance and flocking rules. Avoidance rules are based on proximity thresholds for each type of object, ordered by priority (i.e., avoiding dogs is more important than avoiding the target zone). Flocking rules are configured with the coherence and alignment parameters. Coherence controls the rate at which sheep steer towards each other, while alignment controls the rate at which sheep match the average direction of other surrounding sheep. Unlike the dogs which can vary their translation speed, sheep move at a constant speed throughout their lifetime.

2.3 Evolutionary Algorithms

The ANN controllers governing dog behaviour are directly encoded as genomes of floating point weights, each in the range [-1, 1]. These genomes are optimised for high task performance (see section 2.4) using two alternative evolutionary algorithms, based on either the Steady-State Genetic Algorithm (SSGA) [9] or the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [6] algorithm. The algorithms are applied to the evolution of both homogeneous and heterogeneous swarms as detailed in the following.

2.3.1 SHOM: SSGA Homogeneous. In this approach, a population of genomes is randomly initialised and evaluated for the first generation. Thereafter, for each generation, individuals are selected from this population by tournament selection with a tournament size of 3. These individuals undergo two-point crossover and Gaussian mutation, each with a specific probability, before being evaluated. Each individual genome is evaluated as a homogeneous team of dogs (i.e., every dog in the simulation task environment has the same ANN weights applied to it). The evaluated individual genomes then become the offspring population for the next generation.

2.3.2 SHET: SSGA Heterogeneous. This method is similar to SHOM (see section 2.3.1), except that each genome consists of floating point weights for *N* dog ANN controllers. Therefore, each individual genome is evaluated as a heterogeneous team of dogs, with each dog using a unique subset of ANN weights from the genome.

2.3.3 MHOM: MAP-Elites Homogeneous. Similar to the SSGA-based algorithms (see sections 2.3.1 and 2.3.2), a population of genomes is randomly initialised and evaluated for the first generation. However, during evaluation, a set of three behavioural characteristics is also measured: (1) average distance between each dog and its nearest neighbouring dog, (2) average distance between each dog and its nearest neighbouring sheep, and (3) average distance between
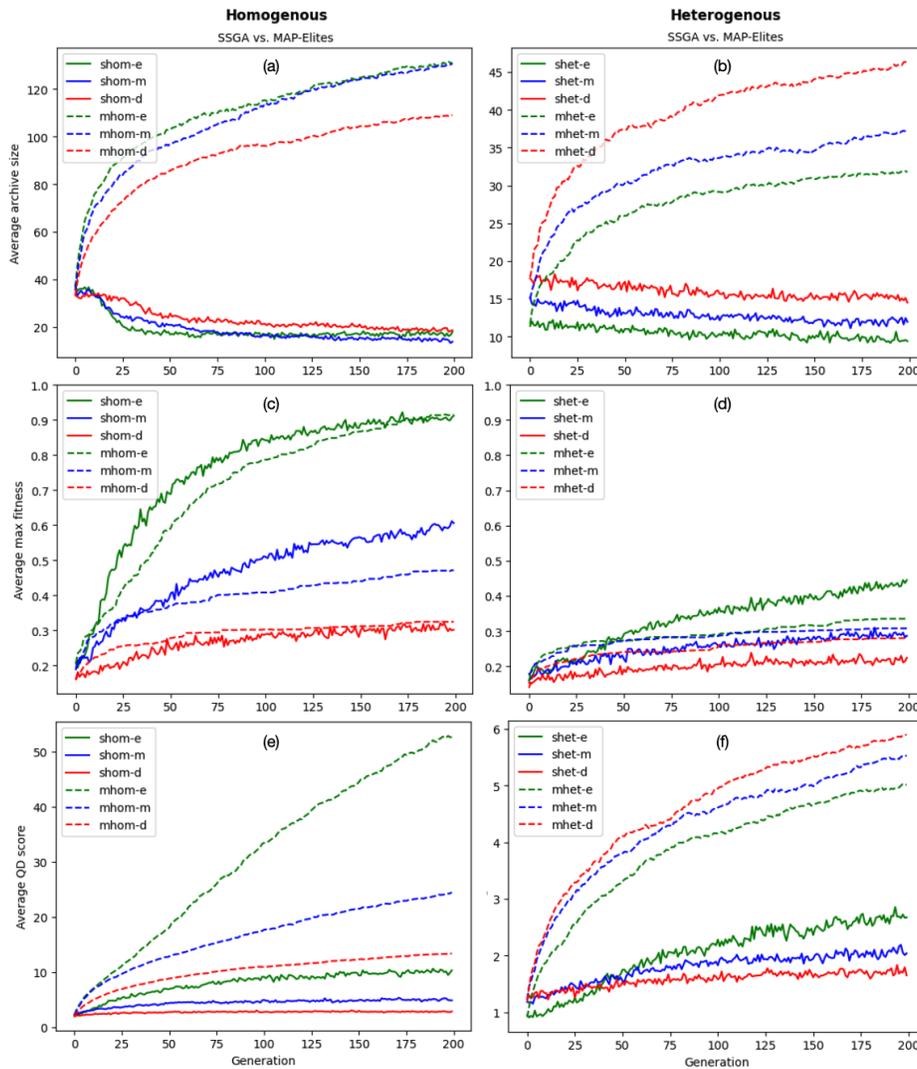
Figure 2: Archive size, maximum fitness and QD score results for experiments. The algorithms which evolve homogeneous swarms (SHOM and MHOM) are on the left, while the algorithms which evolve heterogeneous swarms (SHET and MHET) are on the right. Results from the easy, medium and difficult task environments are presented for each algorithm, averaged over 20 runs.

each dog and the target zone. These behavioural characteristics are normalised in the range [0, 1], where 0 is an average distance of 0 and 1 is the maximum distance observed for that characteristic in a calibration test run. Individual genomes are, as with SHOM, evaluated as a homogeneous team of dogs. Evaluated solution genomes are stored in a multi-dimensional archive, positioned in bins based on their behavioural characteristic values. If there already exists another solution genome at the assigned bin, the new solution is only inserted if it has a higher fitness score. For every subsequent generation, individuals are selected from this solution archive by tournament selection with a tournament size of 3. Crossover and mutation is the same as for the SSGA-based algorithms.

2.3.4 MHET: MAP-Elites Heterogeneous. This method is similar to MHOM (see section 2.3.3), except genomes consist of floating point weights for N dog ANN controllers. Therefore, each individual genome is evaluated as a heterogeneous team of dogs, with each dog using a unique subset of ANN weights from the genome.

2.4 Fitness Evaluation

Solution genomes are evaluated based on the number of sheep captured, c, out of the total number of sheep, t, during the simulation lifetime. Therefore, an evaluation score of 0 corresponds with none of the sheep captured and an evaluation score of 1 corresponds with all of the sheep captured. Due to the stochastic nature of the task environment, final genome fitness is averaged across n evaluation trials. Equation 1 summarises this fitness calculation.

$$F = \sum_{i=1}^{n} \left(\frac{c_i}{t_i} \right) \div n \qquad (1)$$

3 EXPERIMENTS

We conducted four sets of experiments with the simulation framework[1], each using a different evolutionary algorithm (SHOM, SHET, MHOM and MHET). Three difficulties of task environment (easy, medium and difficult) were tested and averaged over 20 runs.

Task difficulty is based on the ratio of dogs to sheep and their relative maximum translation speeds. The easy task has more dogs that move faster than sheep, while the difficult task has more sheep that move faster than dogs. The three behavioural characteristics (see section 2.3.3) are tracked and recorded for evaluated individuals across all experiments. Although not used in SSGA, these values allow for post-processing the evolved populations and projecting them into three-dimensional solution archives which can be directly compared with those produced by MAP-Elites for behavioural diversity. Parameter values for the neuro-evolution process, simulation environment and different task difficulties can be found in Table 1.

4 RESULTS AND DISCUSSION

Figure 2 presents the archive size, maximum fitness and QD score over evolutionary time for each algorithm across task environment difficulties, averaged over 20 simulation runs. Archive size (Figure 2a and 2b) refers to the number of solutions (or individuals) in the population displaying unique behavioural characteristics. Maximum fitness (Figure 2c and 2d) refers to the fitness score of the best-performing individual in the population, where a score of 0 means that no sheep were captured during trial evaluations and a score of 1 means that all sheep were captured (see section 2.4). QD score (Figure 2e and 2f), as introduced in [7], refers to the sum of fitness scores for all individuals in the archive, which is maximised by increasing both the diversity and fitness of solutions.

For both homogeneous and heterogeneous swarms, it is evident from the archive size (Figure 2a and 2b) that MAP-Elites generates a set of solutions with significantly greater ($p < 0.001$, in all cases) behavioural diversity that increases over time compared with those generated by SSGA that decreases over time. This is to be expected since SSGA tends to converge on and optimise a single high-performing solution, whereas MAP-Elites maintains and grows a population of functionally diverse solutions. It is notable that evolved homogeneous swarms generate a significantly greater ($p < 0.05$, in all cases) number of uniquely behaving solutions compared with heterogeneous swarms. This is likely due to the fact that heterogeneous solutions contain inherent behavioural diversity which overlap in the measurement of their behavioural characteristics. In other words, multiple heterogeneous allocations of behaviour can result in the same behavioural characteristics being measured at the swarm level, due to some individual behaviours cancelling out the effects of other individual behaviours.

Fitness results (Figure 2c and 2d) indicate significantly reduced ($p < 0.001$, in all cases) task performance for directly-evolved heterogeneous swarms compared with homogeneous swarms. This is most likely due to the significantly larger search space to explore

---

[1]Experiment source code: https://anonymous.4open.science/r/gecco23-sheepdogai

in the limited 200 simulation generations. As such, a longer evolutionary process may yield better results. Alternatively, a multi-step approach towards evolving heterogeneous swarms may be more appropriate, whereby an *allocation* of solutions pre-evolved for a homogeneous swarm is optimised for a heterogeneous swarm, as alluded to in our research objective. Although the maximum achieved fitness does not differ significantly ($p > 0.05$) between SSGA and MAP-Elites in homogeneous swarms (Figure 2c), with the exception of the medium task difficulty ($p < 0.001$), it is clear from the corresponding QD score results (Figure 2e) that MAP-Elites generates a significantly more ($p < 0.001$, in all cases) diverse set of high-performing individuals. The trend of significantly higher ($p < 0.001$, in all cases) QD score for MAP-Elites compared with SSGA also holds in heterogeneous swarms (Figure 2f). These results support previous work which demonstrates that behavioural diversity can be generated without specific speciation mechanisms or geographical isolation in the task environment [2].

Overall, these preliminary results demonstrate that behavioural diversity can be generated for high-performing solutions in homogeneous swarms. It also provides evidence that there may be more effective alternatives to the direct evolution of a heterogeneous swarm. Based on current results, future work will investigate the potential of evolving functionally diverse allocations for a heterogeneous swarm from an archive of existing behaviours.

5 CONCLUSION

In this paper, we investigated the direct evolution of behavioural diversity for homogeneous and heterogeneous swarms in a collective herding task environment. We found MAP-Elites successfully generates high-performing sets of diverse solutions for homogeneous swarms, although directly-evolved heterogeneous swarms do not achieve comparable fitness results. Future work will explore approaches for the evolution of heterogeneous swarm allocations.

REFERENCES

[1] N. Bredeche, J.M. Montanier, B. Weel, and E. Haasdijk. 2013. Roborobo! a fast robot simulator for swarm and collective robotics. *arXiv:1304.2888* (2013).
[2] E. Hart, A.S.W. Steyven, and B. Paechter. 2018. Evolution of a functionally diverse swarm via a novel decentralised quality-diversity algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 101–108.
[3] P. Hoehn, T. Tscharntke, J.M. Tylianakis, and I. Steffan-Dewenter. 2008. Functional group diversity of bee pollinators increases crop yield. In *Proceedings of the Royal Society B: Biological Sciences*, Vol. 275. The Royal Society London, 2283–2291.
[4] L. Hong and S.E. Page. 2004. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. In *Proceedings of the National Academy of Sciences*, Vol. 101. National Academy of Sciences, 16385–16389.
[5] G.C. Martin et al. 2014. The effects of cultural diversity in the workplace. *Journal of diversity management (JDM)* 9, 2 (2014), 89–92.
[6] J.B. Mouret and J. Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).
[7] J.K. Pugh, L.B. Soros, P.A. Szerlip, and K.O. Stanley. 2015. Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. 967–974.
[8] C.W. Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 25–34.
[9] G. Syswerda. 1991. A study of reproduction in generational and steady-state genetic algorithms. In *Foundations of genetic algorithms*. Vol. 1. Elsevier, 94–101.