Evolving Behavior Allocations in Robot Swarms

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

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

Emma Hart
School of Computing
Edinburgh Napier University
Edinburgh, United Kingdom
E.Hart@napier.ac.uk

***Abstract*—Behavioral diversity is known to benefit problem-solving in biological social systems such as insect colonies and human societies, as well as in artificial distributed systems including large-scale software and swarm-robotics systems. We investigate methods of evolving robot swarms in which individuals have heterogeneous behaviours. Two approaches are investigated to create swarm of size n. The first encodes a repertoire of n behaviours on a single individual, and hence evolves the swarm directly. The second approach uses two phases. First, a large repertoire of diverse behaviours is evolved and then another evolutionary algorithm is used to search for an optimal allocation of behaviours to the swarm. Results indicate that the two phase approach of generate then allocate produces significantly more effective collective behaviors (in terms of task accomplishment) than the direct evolution of behaviorally heterogeneous swarms.**

***Index Terms*—Swarm-Robotics, Behavioral Quality-Diversity**

I. Introduction

Ecological hypotheses suggest that behavioral diversity within and between species is positively related to overall population robustness to changes in the environment [1], [2]. It has been proposed this same principle is applicable to swarm-robotic system design, providing similar benefits to system performance and adaptability, where recent work has highlighted the potential of evolving behaviorally diverse robotic swarms which demonstrate robustness across various task environments [3], [4]. It has also been shown that diverse groups of individuals tend to perform better at solving problems and collective tasks than homogeneous groups [5]–[7]. As a result of these purported benefits, research interest has grown surrounding different approaches for evolving behavioral (or functional) diversity in robot swarms.

In particular, the use of Quality-Diversity methods such as *Multi-dimensional Archive of Phenotypic Elites* (MAP-Elites) [8] to create repertoires of behaviourally diverse controllers has rapidly increased in recent years. These methods return an archive of solutions that are diverse with respect to a set of characteristics defined by a user. Engebråten *et. al.* demonstrate that a diverse repertoire of behavior primitives can be created using MAP-Elites, that can be combined to enable more effective control of a large group or swarm of unmanned system [9]. Gomes *et. al.* [10] go further in evolving a repertoire of *task-agnostic* primitives: the evolutionary process is driven towards the generation of arbitrary but distinct swarm behaviours, thus departing from the vast majority of previous works where automatic design methods are used as means to solve a specific task or achieve a specific swarm behaviour. Most recently, Montague *et. al.* combined MAP-Elites with Genetic Programming to evolve diverse fragments of behaviour-trees for a foraging task [11]. However, all of these methods produce behavioral primitives which can be combined into a single task appropriate behaviour allocated to all robots in the swarm.

In contrast, this study investigates methods to evolve a repertoire of diverse neural controllers that result in a swarm that is heterogeneous with respect to its controllers. That is, each robot in the swarm may have a different controller. This study's objective is to understand how to best evolve a behaviourally-heterogeneous swarm for a given specific task. To address this objective, we evaluate two approaches:

- Evolving an individual that encodes n different controllers to directly define a behaviourally- heterogeneous swarm (referred to as the *direct* approach)
- A two-phase approach that first evolves a large repertoire of diverse controllers, and then evolves an optimised allocation of behaviours from the repertoire to the swarm (referred to as the *allocation* approach).

For the direct approach, we compare two evolutionary methods: the first uses a steady-state genetic algorithm (SSGA) [12] while the second uses MAP-Elites [8]. For the allocation approach, an evolutionary algorithm (EA) is used to allocate behaviours from a previously obtained repertoire to a swarm. Repertoires are created through (1) repeatedly applying an SSGA to evolve a single controller, and combining the results into a single repertoire, and (2) by using of MAP-Elites. Experiments are conducted on three variants of a herding task with varying levels of difficulty. Our results show that the allocation methods achieve significantly improved task performance in a collective behavior task, when compared with the direct evolution of behaviorally heterogeneous swarms. Furthermore, the results demonstrate that behavioral diversity is beneficial to collective behavior task performance and can be generated without speciation mechanisms or geographical isolation in the environment, extending previous related work [3], [4], [13].

II. Methods

This section describes the methods (section II-C) used to evolve behaviorally heterogeneous robotic swarms (section II-B) in a collective herding task environment (section II-A).

A. Simulation Task Environment

The collective herding task is simulated using an extended version of the Roborobo! multi-agent simulation framework [14]. A swarm of N robots (*dogs*), is assigned the objective of capturing a dispersed flock of M agents (*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. The collective herding task was selected as it represents a uniquely dynamic interaction between swarm and task environment [4]. This task not only encourages robots to adapt to a dynamic environment, but also to a collective behavior task with differing solutions. Since the herd's movement is randomly generated for each simulation, the task environment promotes the evolution of robots equipped with more general herding approaches rather than specialized robots attuned to a few specific herd configurations. Also, this task environment allows us to investigate the general applicability of our swarm behavior evolution methods (section II-C).

B. Robots: Dogs and Sheep

1) Dogs: Dog robots use fully connected feed-forward *Artificial Neural Network* (ANN) controllers (all hidden and output nodes use tanh activation functions) adapted via various evolutionary methods for the collective herding task. Dogs have a simple, circular morphology supporting an array of radar-type proximity sensors about the periphery. These sensors detect the nearest instance of each type of object (dog, sheep and wall) within a specific range and field of view (table I). Dog ANN controller topology comprises nine input nodes, ten hidden nodes and two output nodes. This thus corresponds to 110 evolvable connection weights values in each dog's genotype. Each sensory input node uses distance and angle values from three radar sensors (one for each object type), and distance and angle values from a target zone sensor. Distance values are normalized in the range [0.0, 1.0], where 0.0 denotes undetected and 1.0 denotes an object is as close as possible to the dog. Angle values are also normalized in the range [-1.0, 1.0], where -1.0 corresponds to -180 degrees and 1.0 corresponds to +180 degrees. The two motor output nodes are the dog's translation value normalized to [-1.0, 1.0], where -1.0 is maximum translation speed backwards and +1.0 is maximum forward translation speed, and the dog's rotation value normalized to [-1.0, 1.0], such that -1.0 denotes maximum rotation speed to the left and +1.0 denotes the maximum rotation speed to the right.

2) Sheep: Sheep robots use a pre-defined heuristic behavior causing them to collectively wander as a herd. Sheep use the same morphology and sensory configuration of the dog robots with the same radar-type proximity sensors. However, different range and field of view values are set (table I). A *Boids* [15] algorithm variant is used to control sheep collective (movement) behavior. This sheep controller is not adapted by evolutionary methods and controls sheep collective behavior using simple avoidance and flocking rules. Avoidance rules use proximity thresholds for each type of object, ordered by priority, so as sheep first avoid dogs and then avoid the target zone. Flocking rules use coherence and alignment parameters, where coherence controls the speed with which sheep move towards one other. Alignment is the degree to which sheep follow the average direction of neighboring sheep. While dogs adapt their speed, all sheep move at a constant speed (table I).

C. Behavior Evolution Methods

The ANN dog controllers are evolved as genotypes encoded as strings of floating point weights (normalized to the range [-1.0, 1.0]). These genotypes are evolved with the objective of optimizing collective (herding) behavior (section II-D). Evolutionary methods applied for dog controller adaptation (collective herding evolution) use the SSGA [16] or MAP-Elites [8] methods (detailed in sections II-C1 − II-C6).

1) SHOM: SSGA Homogeneous: A genotype population is randomly initialized, where each genotype corresponds to a dog controller, and the same controller is used by all dogs in the group (homogeneous swarm). Thereafter, per generation, genotypes are selected using tournament selection (tournament size of 3). These genotypes undergo two-point crossover and Gaussian mutation [17], with a given probability (table I), prior to evaluation in the task. These offspring genotypes then replace the previous population to become the next generation.

2) SHET: SSGA Heterogeneous: SHET is similar to SHOM, except each genotype comprises floating point values (weights) for N dog controllers. Each genotype is evaluated as a heterogeneous swarm and each dog uses a unique controller subset (ANN weights) from the swarm genotype.

3) MHOM: MAP-Elites Homogeneous: We use the standard version of MAP-Elites [8]. This evolves a repertoire of solutions in a single run that are diverse with respect to a desriptor containing d user-defined characteristics that define a d-dimensional archive discretised into b bins. Each solution is mapped to a bin according to its descriptor. Each bin retains a single solution which has the best fitness found so far for that bin. We define 3 characteristics: (1) the average distance between each dog and the nearest dog; (2) the average distance between each dog and the nearest sheep; (3) the average distance between each dog and the target zone. Each characteristics is normalized to the range [0.0, 1.0], where 0.0 is the minimum average distance and 1.0 is the maximum average distance observed. A genotype

Neuro-Evolution Parameters	
Replications per experiment (runs)	20
Generations per experiment run	200
Trial evaluations per individual	3
Initial population size	100
ANN nodes: Input / Hidden / Output	9 / 10 / 2
MAP-Elites archive: Dimensions / Bins	3 / 729
Operator probability: Crossover / Mutation	0.5 / 0.2
Simulation Parameters	
Time steps per trial evaluation	800
Initial agent positions	Random
Dog swarm size: Easy/Medium/Difficult	20 / 15 / 10
Sheep swarm size: Easy/Medium/Difficult	10 / 15 / 20
Dog speed: Easy/Medium/Difficult	1 / 0.75 / 0.5
Sheep speed: Easy/Medium/Difficult	0.5 / 0.75 / 1
Arena size (width × height)	600px × 600px
Target zone size (radius)	100px
Dog radar proximity sensor: Range/FOV	(0px,100px]/[-90°,90°]
Sheep radar proximity sensor: Range/FOV	(0px,50px]/[-180°,180°]
Sheep object avoidance: Wall/Dog/Sheep	15px / 50px / 5px
Sheep zone avoidance: Radius/Strength	50px / 0.25

TABLE I: **Neuro-evolution and simulation parameters.** For all experiments in the simulated collective herding task.

population is randomly initialized. As with SHOM, genotypes are evaluated as a homogeneous swarm and adapted using the same crossover and mutation operators used by the SSGA-based methods. Individuals are selected from the archive using tournament selection (size 3).

4) MHET: MAP-Elites Heterogeneous: MHET is the same as MHOM, except genotypes comprise floating point values (ANN weights) for *N* dog controllers. Thus, each genotype is evaluated as a heterogeneous swarm and each dog uses a unique subset of ANN weights from the swarm genotype.

5) ASHET: Allocate SSGA Heterogeneous: In this approach, the objective is to optimize a swarm allocation of ANN controllers previously evolved by SHOM. Since SHOM is based on SSGA, its final population contains the same number of genotypes as the initial population (100). These genotypes are not guaranteed to be unique and are likely to demonstrate controllers with behavioral similarities given the SSGA tendency to converge on variations of high-performing solutions. Thus, before commencing allocation evolution, the final SHOM populations (from multiple experimental runs) are projected into MAP-Elites archives based on the tracked behavioral characteristics (see section II-C3) for each genotype. These archives are aggregated into a single reference archive containing only the elite solutions across all populations. The M genotypes in this reference archive are each assigned a unique index. Using this reference archive of ANN controllers, a population (100) of random allocation genotypes is initialized. Each genotype consists of N indices selected, with replacement, from the range [0, $M-1$]. The number of dogs in the swarm, N, depends on the task environment difficulty being simulated. These genotypes are evaluated as heterogeneous swarms in which each dog is allocated an ANN controller from the reference archive based on its index. Thereafter, per generation, genotypes are selected using tournament selection (tournament size of 3), keeping the population size constant (as per SSGA). These genotypes undergo two-point crossover and uniform integer mutation [17], each with a given probability (table I), before being evaluated in the task. These offspring genotypes then constitute the next generation of genotypes.

6) AMHET: Allocate MAP-Elites Heterogeneous: As with ASHET, this method optimizes a swarm allocation of pre-evolved ANN controllers. However, in this case, the controllers being allocated are pre-evolved by MHOM. Since MHOM is based on MAP-Elites, its final population is already contained in a MAP-Elites archive of behaviorally unique individuals. To produce a reference archive for AMHET, the final MHOM population archives from multiple experimental runs are aggregated together. The M genotypes in this reference archive are each assigned a unique index. Using this reference archive of ANN controllers, a population (100) of random allocation genotypes is initialized. Thereafter, evolution of these allocation genotypes continues as for ASHET (using SSGA) until the maximum number of generations (table I).

D. Dog Fitness Evaluation

Dog genotypes are evaluated according to the number of sheep captured, c, out of all sheep, t, over the course of the simulation. Hence, a *zero* evaluation score corresponds to no sheep captured and a score equal to *one* means that all sheep are captured. Given that the task environment is stochastic (sheep positions are randomly initialised), genotype fitness is averaged across n evaluation trials (equation 1).

III. Experiments

Experiments[1] evaluated the direct behavior versus allocation evolution (section II-C) methods for heterogeneous robot group (swarm) behavior across three task environments (table II), where results are averaged over 20 runs per experiment. First, we conducted direct *behavior evolution* experiments, using the SHET and MHET methods, where both evolved behaviorally heterogeneous swarms. Second, we conducted *allocation evolution* experiments, using the ASHET and AMHET methods. Both ASHET and AMHET use SSGA to evolve optimal swarm controller allocations, but use different reference archives of pre-evolved controllers. ASHET uses aggregate archives produced by SHOM using SSGA [12] (section II-C1) whereas AMHET uses aggregate archives produced by MHOM using MAP-Elites [8] (section II-C3). Experiments also recorded the number of unique behaviors

[1]Source code: https://anonymous.4open.science/r/ssci23-sheepdogai

Evolution Type	Algorithm	Task Environment		
		Easy	**Medium**	**Difficult**
Direct	**SSGA**	SHET	SHET	SHET
	MAP-Elites	MHET	MHET	MHET
Allocation	**SSGA**	ASHET	ASHET	ASHET
		AMHET	AMHET	AMHET

TABLE II: **Experiment Setup.** Experiments are based on evolution type (direct or allocation), controller evolution method (SSGA or MAP-Elites) and variant (SHET, MHET, ASHET, AMHET) and task environment (easy, medium and difficult).

in evolved swarm allocations to measure behavioral diversity.

Task difficulty is varied by altering the ratio of dogs to sheep and their relative maximum translation speeds. The easy task had more dogs moving faster than sheep, while the difficult task had more sheep moving faster than dogs. The three behavioral characteristics (see section II-C3) were recorded for individuals across all experiments. Although not used in SSGA, these values enabled post-processing of evolved populations and projecting them into 3D solution archives so that they could directly be compared with those produced by MAP-Elites for behavioral diversity. Table I presents experiment and method parameter values.

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

IV. Results and Discussion

Given this study's research objective (section I), we focus our discussion on comparing behavior and allocation evolution experiment (section III) results. Specifically, this section reports *archive size*, *maximum fitness*, *QD scores*[2], and pairwise t-tests to determine any statistical significance between comparative method results. Figures 1 and 2 present average archive size, maximum fitness and QD scores, for the direct behavior and allocation evolution methods, respectively. Figure 2(d) also presents the average number of unique behaviors in swarms over evolutionary time for each allocation evolution method in each task environment.

First, we assess statistical differences between average results of the heterogeneous behavior evolution (SHET, MHET) and behavior allocation evolution (ASHET, AMHET). Pairwise t-tests indicate direct behavior evolution methods (SHET, MHET) produce significantly greater ($p < 0.05$) archive sizes than allocation evolution methods (ASHET, AMHET) across all task environments. Conversely, for maximum fitness, ASHET and AMHET significantly outperform ($p < 0.05$) the direct behavior evolution methods for all task environments. For QD scores, ASHET achieves significantly higher ($p < 0.05$) scores than SHET across all environments. However, AMHET only achieves significantly higher ($p < 0.05$) QD scores than MHET in the easy task environment and MHET achieves significantly higher ($p < 0.05$) scores in the medium and difficult environments.

Across all environments, results indicate that the allocation evolution methods (ASHET, AMHET) produce the highest quality solutions for the collective herding task. In terms of behavioral diversity, MAP-Elites behavior evolution (MHET) produces significantly more diverse solution populations than those based on SSGA, where QD score results reflect this pattern (figures 1 and 2). For example, MHET achieves the highest QD scores in all environments and also produces the highest archive size for all environments (figure 1). This is a result of MAP-Elites archives retaining behaviorally unique solutions while SSGA does not but rather converges on similar solutions. Though, observing maximum fitness, allocation evolution (ASHET, AMHET) outperforms direct behavior evolution, and SHET and MHET yield the worst-performing solutions for all environments (figures 1 and 2).

Results comparing behavior allocation (ASHET, AMHET) versus direct behavior evolution (SHET, MHET) methods, indicate that ASHET and AMHET achieve significantly higher average task performance than SHET and MHET, across all environments (figures 1 and 2). Despite the higher average solution quality of ASHET and AMHET, the final solution archives (figure 3) indicate these allocation evolution methods explore a lower proportion of the behaviour space than the behaviour evolution methods (SHET, MHET). The final populations generated by each method (over 20 runs) were aggregated to produce these solution archives. Since both ASHET and AMHET evolve solutions using SSGA as the underlying algorithm, they do not produce MAP-Elites archives. Therefore, the final populations are projected into archives using the task behavioural characteristics for solutions (section II-C3) before being aggregated (figure 3).

While archive sizes are significantly lower for ASHET and AMHET than SHET and MHET (for all environments, figures 1 and 2), the QD score results (providing a combined measure of solution fitness and diversity) favor the allocation evolution over the behaviour evolution methods. That is, the ASHET and AMHET methods achieve significantly higher QD scores across all task environments, highlighting that the greater solution quality yielded by these behavior allocation methods outweighs their lower solution diversity, thus outperforming the behaviour evolution methods overall (figures 1 and 2).

In terms of the number of unique behaviours allocated to a swarm for the behavior allocation evolution experiments, results indicated AMHET on average allocated more unique behaviours to swarms than ASHET (figure 2). This metric was constrained by swarm size, where swarms of 20, 15 and

[2]This metric captures both quality and diversity and is obtained by summing the highest fitness values found in each grid bin Q_i, i.e. as $\sum_{i=1}^{i=m} Q_i$, [18]

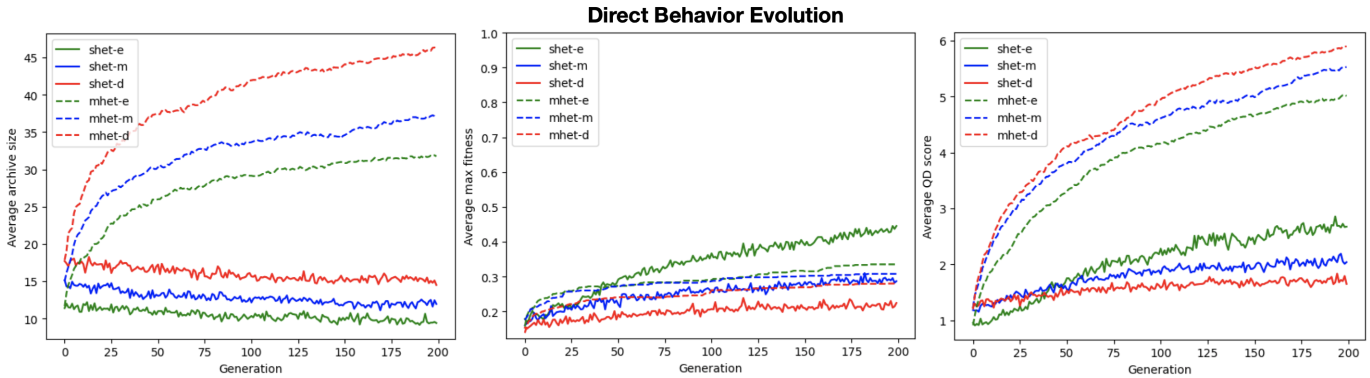

Fig. 1: **Direct behavior evolution.** Archive size (left), max fitness (middle) and QD score (right) results (averaged over 20 runs), for evolved heterogeneous swarms (SHET and MHET) in easy (green), medium (blue) and difficult (red) tasks.

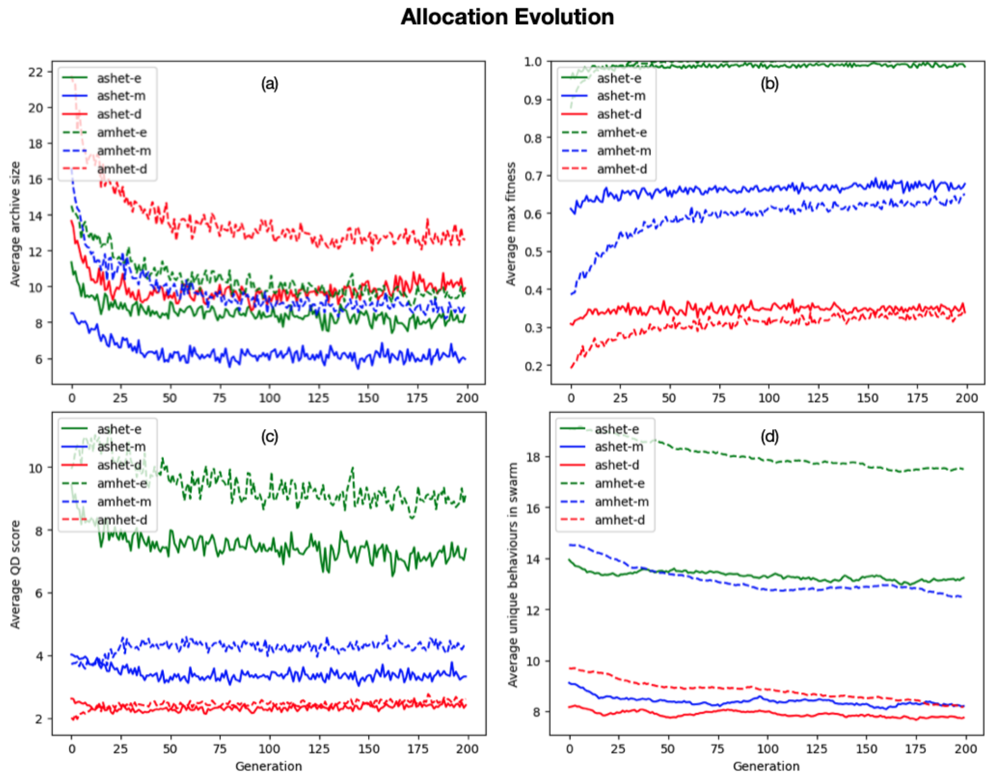

Fig. 2: **Behavior allocation evolution.** Archive size (a), max fitness (b), QD score (c) and unique behaviors in swarm (d) results (averaged over 20 runs), for the easy (green), medium (blue) and difficult (red) task environments (ASHET and AMHET).

10 dogs were evolved for the easy, medium, and difficult task environments, respectively. For all environments, AMHET allocated a number of unique behaviors close to the swarm size, further supporting the benefits of Map-Elites (section II-C6) for enabling behavioral diversity in evolved controller allocation of swarms (figure 2). Despite using different reference archives for evolving swarm controller allocations, ASHET and AMHET both produce solutions occupying similar regions of the behaviour search space (figure 3). Most of this behaviour space remains unexplored by both methods, which results from SSGA optimising for solution quality (when evolving the controllers to be later allocated) rather than diversity. Thus, comparing ASHET and AMHET with SHOM and MHOM (evolution of homogeneous swarm behavior) [4], we observed that despite ASHET and AMHET not modifying (further evolving) the controllers generated by SHOM and MHOM, both ASHET and AMHET (especially) produce behaviorally heterogeneous swarms with significantly greater task performance than SHOM and MHOM. This further supports this study's approach of optimising allocations of

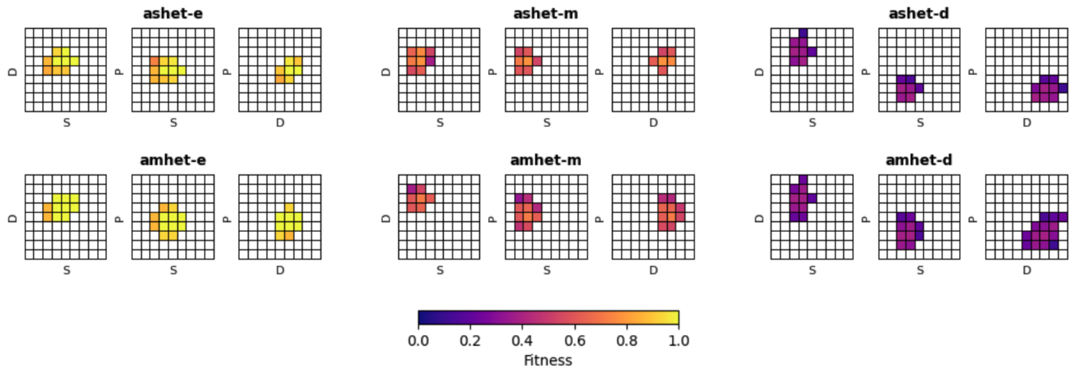

Fig. 3: **Allocation evolution solution archives.** Projected MAP-Elites solution archives for the ASHET and AMHET methods in the easy (e), medium (m) and difficult (d) task environments (generation 200, aggregated over 20 runs). Behavioural characteristics: dog-dog distance (D), dog-sheep distance (S) and dog-pen distance (P). The 3D archives are flattened on each plane (taking the best solution along the hidden axis at each position) and presented as 2D grids (D×S, P×S and P×D).

pre-generated behaviors in a swarm, compared to direct swarm behavior evolution, as a means to achieve behavioral diversity necessary for suitably high swarm task performance.

Overall, results support the hypothesis that evolving behaviorally heterogeneous swarms via optimising allocation of pre-generated controllers produces significantly greater swarm task performance than direct controller evolution. This holds even with lower population diversity, where the combined QD score for allocation evolution algorithms still exceeds that of behavior evolution algorithms. Results also support existing work indicating that behavioral diversity is evolvable without speciation mechanisms or geographical isolation [3], [4].

V. Conclusions and Future Work

This study investigated the impact of varying methods for collective herding behavior evolution in a swarm-robotic system given increasing task difficulty. Heterogeneous swarm behavior was optimized using direct behavior evolution methods (SHET, MHET), compared with methods for evolving allocations of pre-evolved controllers (ASHET, AMHET). Results indicated significantly improved task performance for heterogeneous swarms generated by evolved behavior allocation (ASHET, AMHET). This supports our hypothesis that evolving behavioral allocation, compared to direct behavior evolution, is more suitable for adapting behaviorally heterogeneous swarms across increasing task difficulty. As such, the introduction of our multi-step approach for evolving swarm-controller allocations (ASHET, AMHET) represents the main contribution of this work. Future work will focus on evaluating the approach in a wider range of other collective behavior tasks. We also plan to use the QED method [13] to generate solution archives based on environmental characteristics rather than explicitly-defined behavioral characteristics.

References

[1] M. Cadotte *et al.*, "Beyond species: functional diversity and the maintenance of ecological processes and services," *Journal of Applied Ecology*, vol. 48, no. 5, pp. 1079–1087, 2011.

[2] S. Villéger *et al.*, "New multidimensional functional diversity indices for a multifaceted framework in functional ecology," *Ecology*, vol. 89, no. 8, pp. 2290–2301, 2008.

[3] E. Hart *et al.*, "Evolution of a functionally diverse swarm via a novel decentralised quality-diversity algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 101–108, 2018.

[4] S. Hallauer *et al.*, "Evolving herding behaviour diversity in robot swarms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, Lisbon, Portugal, 2023.

[5] P. Hoehn *et al.*, "Functional group diversity of bee pollinators increases crop yield," in *Proceedings of the Royal Society B: Biological Sciences*, vol. 275, pp. 2283–2291, The Royal Society London, 2008.

[6] L. Hong and S. Page in *Proceedings of the National Academy of Sciences*, vol. 101, pp. 16385–16389, 2004.

[7] G. Martin *et al.*, "The effects of cultural diversity in the workplace," *Journal of diversity management*, vol. 9, no. 2, pp. 89–92, 2014.

[8] J. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.

[9] S. Engebråten *et al.*, "Evolving a repertoire of controllers for a multi-function swarm," in *International Conference on the Applications of Evolutionary Computation*, pp. 734–749, Springer, 2018.

[10] J. Gomes and A. Christensen, "Task-agnostic evolution of diverse repertoires of swarm behaviours," in *Swarm Intelligence* (M. Dorigo *et al.*, eds.), pp. 225–238, Springer, 2018.

[11] K. Montague *et al.*, "A quality-diversity approach to evolving a repertoire of diverse behaviour-trees in robot swarms," in *International Conference on the Applications of Evolutionary Computation*, pp. 145–160, Springer, 2023.

[12] D. Whitley and J. Kauth, "GENITOR: A different genetic algorithm," in *Proceedings of the 4th Rocky Mountain Conference on Artificial Intelligence*, Springer, Boulder, USA, 1988.

[13] D. Bossens and D. Tarapore, "QED: using Quality-Environment-Diversity to evolve resilient robot swarms," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 346–357, 2020.

[14] N. Bredeche *et al.*, "Roborobo! a fast robot simulator for swarm and collective robotics," *arXiv:1304.2888*, 2013.

[15] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 25–34, 1987.

[16] G. Syswerda, "A study of reproduction in generational and steady-state genetic algorithms," in *Foundations of genetic algorithms*, vol. 1, pp. 94–101, Elsevier, 1991.

[17] A. Eiben and J. Smith, *Evolutionary Computing*. Berlin, Heidelberg: Springer, 2nd ed., 2015.

[18] J. Pugh *et al.*, "Confronting the challenge of quality diversity," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, pp. 967–974, 2015.