Automating Collective Robotic System Design

Alexander Furman
Department of Computer Science
University of Cape Town
Cape Town, South Africa
Email: FRMALE003@myuct.ac.za

Danielle Nagar
Department of Computer Science
University of Cape Town
Cape Town, South Africa
Email: NRGDAN001@myuct.ac.za

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

***Abstract*—This paper presents a study on methods for body-brain (behavior-morphology) co-evolution in a collective evolutionary robotics system. We investigate a neuro-evolution developmental encoding method designed for the co-evolution of robot behavior-morphology couplings. This behavior-morphology evolution method is evaluated across increasingly complex (difficult) collective behavior task environments. This is in comparison to controller evolution within *pre-engineered* robot morphologies (sensory configurations). Task-complexity is equated with the degree of cooperation required in collective robotics tasks. Results indicate that the developmental method produces significantly more effective behavior-morphology couplings, compared to those evolved with direct encoding methods and controllers evolved within fixed morphologies. These results suggest that such developmental encoding methods could serve as a general evolutionary simulation design tool for automating collective robotic designs. An end goal is for such collective robotic system designs to be rapidly prototyped and deployed in the physical task environments for which they were evolved.**

I. Introduction

Despite over two decades of research in body-brain (behavior-morphology) co-evolution in evolutionary robotics [1], designing methods to suitably automate the design of behavior-morphology couplings for increasingly complex tasks remains an open problem in evolutionary design. The problem of evolutionary robot behavior-morphology design is exacerbated in collective [2] and swarm [3] robotics.

Although collective [3] and more generally swarm [2] robotics are gaining significant research attention given their vast range of potential applications [4], [5], [6], [7], [8], general methods for automating the design of collective and swarm robotic system design are still lacking[1]. Specifically, evolutionary optimization methods for automating the design of robot controllers (collective behavior) as well as sensory-motor systems (morphologies) coupled to these controllers.

General evolutionary design methods to automate the design of collective robotic systems are especially important for two key reasons. First, robot morphology significantly impacts the types and complexity of behaviors that each robot's controller elicits [9]. There are numerous examples in body-brain co-evolution of how morphological evolution significantly impacts elicited behaviors [10], [11], [12], [13]. Second, collective behavior design remains a complex and challenging task, as collective robotic behavior is an emergent property resulting from interactions between individual robots and robots and the environment [2]. Such robot interactions cannot be explicitly pre-defined in controllers given the high uncertainty, noise, and complexity of interactions that are inherent in collective robotic behavior.

Methods for collective robotic design are also confounded by morphological design that must adapt to suit changing task constraints. Hence, if robot morphologies are also adaptable then pre-designing robot and environment interactions such that specific collective behaviors emerge in response to specific tasks becomes an intractably complex.

We advocate a general methodological design approach using evolutionary optimization for automating the design of collective robotic systems [14]. However, in this study we propose evolutionary optimization methods to co-adapt both controllers (behaviors) and morphologies (sensory configurations) of robots in homogenous groups. That is, the same controller-morphology adaptations are applied to each robot in the group at each iteration of the evolutionary design process.

In this study automated collective controller-morphology design is an evolutionary optimization problem that is solved off-line prior to collective robotic system construction and deployment in specific task environments. The evolutionary optimization algorithm thus searches a space of possible robot designs (controller-morphology couplings) with the goal of maximizing given task performance metrics in simulation.

With notable exceptions [15], there has been relatively little work on evolutionary controller-morphology design methodologies for collective robotics, where task performance is dependent upon collective (cooperative) behavior. We anticipate that such design methodologies will become indispensable given the increasing relevance of collective robotics applications such as planetary exploration [16], oceanic monitoring [17] and autonomous drones [18].

Furthermore, while the benefits of developmental encoding (specialized mapping functions between genotypes representing controller-morphology couplings to complete robot designs) have been demonstrated in various evolutionary robotic body-brain co-evolution studies [10], [11], [13], [12], using developmental encodings to co-adapt controller-morphology couplings and thus automate the design of *collective robotic systems* has received little research attention [19], [20].

This study thus has two objectives. First, to demonstrate that developmental encoding is a suitable approach for evolving robot controllers and morphologies in collective robotic systems. We define a developmental encoding neuro-evolution

[1]This study focuses on collective robotic systems, simply defined as swarm-robotic systems with fewer individual robots (tens instead of hundreds).

method (HyperNEAT-M, extending HyperNEAT [21]), to account for robot controller-morphology evolution for given collective behavior tasks. The second objective is to use the experimental demonstration of HyperNEAT-M as an initial contribution to a general evolutionary optimization method for automating collective robotics system design, where such evolved designs could be rapidly prototyped and physically produced for their specific task environments [22], [23].

In this study, the efficacy of HyperNEAT-M as the evolutionary controller-morphology design method is evaluated across increasingly complex (difficult) *collective gathering* [24] tasks. Collective gathering is a well-established collective robotics benchmark task [3], and was thus deemed suitable as a collective behavior test case. Task complexity is equated with the degree of cooperation required for robots to gather (cooperatively push) resources from one area to another.

The collective gathering task is also a suitable surrogate for a range of current and speculative collective robotics applications [25], such as exploration and mapping [16], surveillance and environment monitoring [18], and collective construction and repair [26]. In such tasks it is assumed that the environments are remote and hazardous where tasks must be accomplished autonomously given dynamically changing task constraints. For example, changing mission specifications and resource constraints for autonomous construction of landing sites in preparation for human expeditions to other planets [27].

For simplicity and experimental tractability in this study, robot designs are homogenous, a robot's morphology is specified as its sensory configuration from a pre-specified sensor set, and controller-morphology couplings are evolved across increasingly task complexity for collective gathering.

II. Methods

To evaluate controller (behavior) and morphology (sensory configuration) evolution in collective robotic systems, an extension to *Hypercube-based NEAT* (HyperNEAT) [21] (*HyperNEAT-M*) was developed. We chose to extend the HyperNEAT method as it has already been successfully applied as a controller evolution method in various collective evolutionary robotic tasks [28], [19], [20]. For experiments that evaluated controller evolution in robots with static morphologies (predefined sensory configurations), the HyperNEAT method was used (section II-B), otherwise HyperNEAT-M (section II-C) was applied to evolve robot controllers and morphologies.

A. NEAT and NEAT-M: Benchmark Methods for Controller and Controller-Morphology Evolution

In order to conduct a benchmark comparison with direct-encoding neuro-evolution methods, we employed two methods (NEAT and NEAT-M) from related work [29]. *Neuro-Evolution of Augmenting Topologies* (NEAT) [30], was used for controller evolution in robots with fixed morphologies (section II-D). To evaluate controller and morphology evolution in robot groups, the *Neuro-Evolution of Augmenting Topologies and Morphologies* (NEAT-M) was used. The application of both NEAT and NEAT-M to controller and controller-morphology evolution of robot groups given the collective gathering task is fully described in related work [29].

B. HyperNEAT: Controller Evolution

HyperNEAT [21] is an indirect (developmental) encoding neuro-evolution method that extends NEAT [30] and uses two networks, a *Composite Pattern Producing Network* (CPPN) [31] and a *substrate* (neural network controller). The CPPN is the generative (developmental) encoding mechanism that indirectly maps evolved genotypes to controllers and encodes pattern regularities, symmetries and smoothness of the geometry of a given task in the form of the substrate. This mapping functions via having coordinates of each pair of nodes connected in the substrate fed to the CPPN as inputs.

The CPPN then outputs a value assigned as the synaptic weight of that connection and a value indicating whether that connection can be expressed or not. HyperNEAT uses the evolutionary process of NEAT to evolve the CPPN and determine controller fitness values. The main benefit of HyperNEAT is scalability as it exploits task geometry and thus effectively represents complex solutions with minimal genotype structure [21]. This makes HyperNEAT suitable for evolving complex collective behaviors for a range of tasks [32], [33].

C. HyperNEAT-M: Controller-Morphology Evolution

While a CPPN in HyperNEAT computes only a connection weight value as a function of two Cartesian points (that is, two connected nodes), a HyperNEAT-M CPPN computes a *sensory configuration* constituted by five additional values: *range, bearing, Field of View (FOV), orientation* and *type* (figure 1, left). This is achieved by adding additional output nodes to the CPPN, in similar fashion to that demonstrated in related work, where neural controller parameters were encoded directly into the CPPN [34].

The sensory configuration (robot morphology) is computed by converting numerical CPPN output ($[-1.0, 1.0]$) to a sensory parameter. For categorical parameters (*sensor type*), where each categorical value maps to a number in the CPPN's output range $[-1.0, 1.0]$ (for example, 0.2 for an *UltraSonic sensor* and 0.5 for a *Low Res Camera*. Numerical sensory parameters (*range*, *sensor position*, *FOV*, *orientation*) are computed by normalizing CPPN output given the maximum parameter ranges pre-defined for a given sensor type.

For example, if the CPPN outputs 0.2 for a *sensor position* value and the maximum bearing range pre-defined for that specific sensor type is $[-2.0, 2.0]$, then the *bearing* parameter will be set to 0.4 (0.2 normalized to the range $[-2.0, 2.0]$). The pre-defined maximum parameter ranges for each sensor type are provided in table II and based on Khepera III robot specifications [35].

Subsequently, each connection in the resulting network (neural controller) encoded by the CPPN contains a sensory configuration (*type*, *range*, *sensor position*, *FOV*, *orientation*) and connection weights. The *sensor position* parameter, that is the location of a sensor on the chassis of a robot, corresponds to the Cartesian coordinate of an input node on the substrate which receives environmental input from that sensor (figure 1, center). The final configuration for each sensory input node is established by traversing the outgoing connections from each CPPN input node and selecting the sensory configuration from the connection with the highest weight.

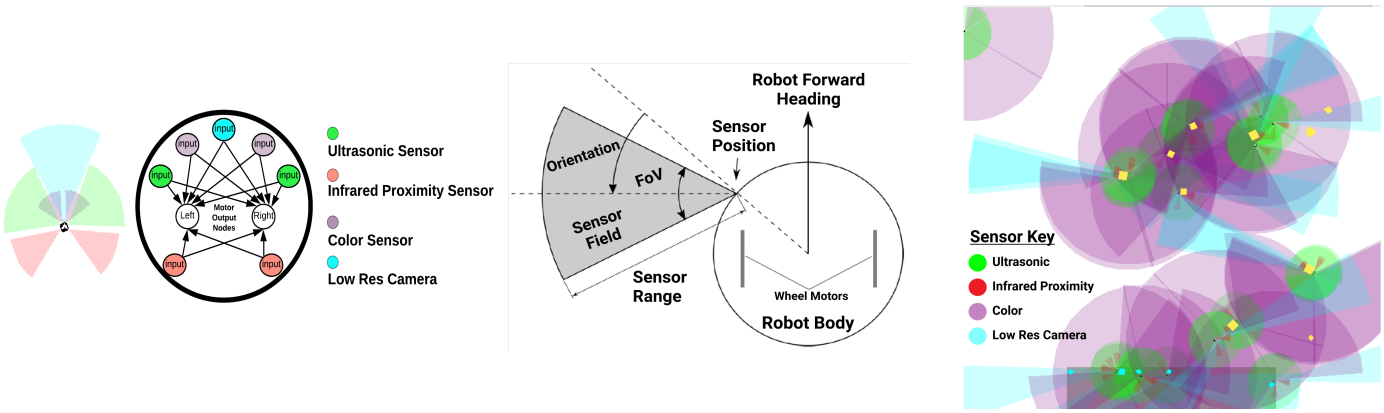

Fig. 1. LEFT: HyperNEAT-M Substrate (Neural controller). The CPPN computes the substrate connection weights between each sensory input node and motor output node, as well as a sensory parameter tuple for each connection: *range, bearing, Field Of View (FOV), orientation, type*. FOV image on far left: Initialized robot with sensory FOVs that correspond with both substrate topology and evolved sensory parameters. Robots were initialized, as per the substrate, with one low resolution camera, two ultrasonic sensors, two infrared proximity sensors and two color sensors (as well as a bottom-proximity gathering zone detector). Wheel motors were fixed throughout controller-morphology evolution. CENTER: Example robot with one sensor. Position determines sensor location on the robot's chassis with respect to the robot's heading. Orientation then determines the direction the sensor faces with respect to this position. RIGHT: Example simulation environment containing 20 robots and a distribution of different block types. The *gathering zone* containing gathered blocks (blue squares) is highlighted at the bottom. Varying sensory parameters (sensor type, position, orientation, field of view and range) are highlighted as shaded semi-circles.

Any given sensor (input node) of a robot is only enabled if it has at least one active outgoing connection.

D. Robot Controllers, Sensors and Motors

Experiments (section IV) tested only homogenous groups meaning robots used the same neural controller for a given generation of the HyperNEAT controller and HyperNEAT-M controller-morphology evolution processes.

Robots used either a fixed morphology (controllers evolved with HyperNEAT) based on the sensory-motor configuration of Khepera III robots or an adaptive morphology (controllers and sensory configuration evolved with HyperNEAT-M). In the latter case, the robot body is also modeled on the Khepera III, but HyperNEAT-M adapts the number and type of sensors and their placement on the robot chassis (section II-C).

The sensor types used were *ultrasonic*, *infrared proximity*, *color*, and *low-res camera* (table II). Sensor FOVs were modeled as conical fields emanating from the outer edge of a robot's body for a given range (figure 1, left). An additional ground facing sensor (not shown in figure 1) was included by default in both fixed and adaptive morphology robots, but was not subject to morphological adaptation in the latter case. This sensor detected when the robot was in the gathering zone.

In all experiments (section IV), both *HyperNEAT* and *HyperNEAT-M* evolved robot groups began with a pre-designed neural controller substrate of seven (7) sensory input nodes, coupled via a randomly initialized CPPN to two motor output nodes (wheels). Specifically, the initial sensory configuration was two (2) *ultrasonic sensors*, two (2) *color sensors*, one 1 *low-resolution* camera and two (2) *infrared proximity* sensors (figure 1, left).

That is, this initial substrate evaluated in company with a population of initially minimal CPPNs with random weight initializations [21]. These initial CPPN connections and functions were randomly initialized from a pre-specified set with connection weights initialized within a given range [21]. The CPPN mapping controller inputs to outputs was then subject to *complexification* during HyperNEAT adaptation.

To ensure that HyperNEAT-M controllers were initially able to execute actions and accomplish the collective gathering task with some degree of success, motor outputs were fixed throughout the HyperNEAT-M evolutionary adaptation process. All controller (NEAT, HyperNEAT) and controller-morphology adaptation methods (NEAT-M, HyperNEAT-M), used hidden and output sigmoidal [36] nodes. All controllers were initialized with a bias node connected to the motor outputs. The bias node had a constant weight value of -1.0 and was not subject to evolutionary adaptation.

Robot actuators (two wheel motors) controlled a robot's heading at a constant speed. Movement was calculated in terms of real valued vectors (*dx* and *dy*). Left and right wheel motors in figure 1 (left, center) needed to be explicitly activated by the robot's controller.

A robot's heading was determined by normalizing and scaling its motor output values by the maximum distance a robot could traverse in one iteration (table II). That is:

$$dx = d_{max}(o_1 - 0.5)$$

$$dy = d_{max}(o_2 - 0.5)$$

Where, o_1 and o_2 are the motor output values. To calculate the distance between this robot (v), other robots and blocks in the environment, the squared Euclidean norm, bounded by a minimum observation distance was used.

For an appropriate experimental results comparison (section IV), initial controller configurations of HyperNEAT and HyperNEAT-M evolved robots also corresponded to that used by controller (NEAT) and controller-morphology (NEAT-M) methods in related work [29].

E. Controller Heuristics

Given that the research focus was on evolving collective gathering behavior in robots with respect to fixed or adaptive morphologies, behavioral control heuristics were included to speed up the evolution of collective gathering behaviors. If a robot was within *gripping distance* (table II) of a block it would automatically attach itself and attempt to push the block. If the robot was unable to push the block it would *wait for help* (table II), for another robot to attach to the same block. If another robot did not attach itself to the block in this time, *this* robot would detach itself and continue to search the environment. If a robot was pushing a block and the *gathering zone* was detected then it would *detach* from the block. Blocks dropped in the gathering zone could not be picked up again.

III. Collective Gathering Task

Collective gathering requires robots to locate distributed resources (blocks) in a bounded environment and transport them, via cooperative pushing, to a *gathering zone* [37].

Cooperation was defined as the number of robots required to push a given block type. *Task difficulty* (environment complexity) was defined as a function of the *number of blocks* and *degree of cooperation* mandated for task accomplishment.

Block types were: *small*, *medium*, or *large*, which could be pushed by at least one, two and three robots, respectively. Thus, task difficulty was calibrated via initializing environments 1, 2, and 3 (*simple*, *medium*, *difficult*, respectively) with varying combinations of block types (table II). For example, the *simple* environment contained 10 small and 5 medium sized blocks, so robots could work concurrently with minimal cooperation to move all blocks to the gathering zone.

Collective gathering task performance (*fitness*) was the average number of blocks pushed into the gathering zone by robots over five simulated task trials (*lifetimes*) in a given generation of evolutionary adaptation (table I). We defined v_c as total value of resources (blocks) in the gathering zone, v_t as total value of all resources (blocks) in the environment, s_e as the number of simulation time-steps in the robots' lifetime and s_t as number of trial evaluations per genotype (representing a given controller-morphology configuration). As such, task performance T was maximized according to equation 1:

$$T = 100 \times \frac{v_c}{v_t} + 20 \times (1.0 - \frac{s_e}{s_t}) \quad (1)$$

In equation 1, 100 was the maximum number of blocks that could be gathered during an experiment run, and 20 was an experimentally determined weighting (boosting fitness for efficient individual and cooperative gatherers).

IV. Experiments

Collective gathering experiments measured the impact of controller evolution in robots with fixed morphologies (NEAT, HyperNEAT) versus robots where both controller and morphology were adapted for each robot (NEAT-M, HyperNEAT-M). Experiments applying the direct-encoding neuro-evolution methods: NEAT and NEAT-M, for robot controller and controller-morphology evolution in the collective gathering task, are described in related work [29]. These experiments use the same neuro-evolution (table I), task simulation and experiment (table II) parameters, and as such the application of NEAT and NEAT-M is not further discussed here.

Experiments executed simulations of 20 robots (section II-D) in a bounded two dimensional continuous environment[2] containing a distribution of *small, medium and large* blocks (table II). Blocks were randomly distributed throughout the environment, excluding the gathering zone. Robots were initialized and randomly placed in the gathering zone. Figure 1 (right) illustrates an example simulation environment containing 20 robots (with specific evolved sensory configurations) gathering a distribution of small, medium, and large blocks.

The three block type distributions given in table II correspond to three environments testing the impact of collective gathering tasks requiring *low, medium and high* cooperation (environments 1, 2, 3: *simple*, *medium*, and *difficult*, respectively) for all blocks to be gathered. These block type distributions were selected given that previous research indicated that specific block type distributions facilitate emergent cooperative behavior during collective robotic evolution [37], [29].

Experiments applied HyperNEAT (controller evolution in fixed morphology robots) or HyperNEAT-M (controller-morphology evolution in robots) to evolve collective behavior for 200 generations. A generation comprised five trial runs of each genotype in the population, where one trial run was 10000 simulation iterations, representing one *lifetime* for all robots in their environment. At each generation, average group task performance (fitness) was taken over the five trial runs of each genotype and used for HyperNEAT and HyperNEAT-M genotype selection. The best group fitness was then taken at the end of each run and an average calculated over 20 runs.

Only homogenous robot groups were tested, meaning that at each HyperNEAT and HyperNEAT-M generation, the selected controller was copied 20 times to represent the group. HyperNEAT-M evolved groups were thus also morphologically homogenous per generation, meaning that robot morphology was adapted at each generation but all 20 robot morphologies were the same (for a given generation).

Table I presents the simulation and neuro-evolution (HyperNEAT and HyperNEAT-M) parameter settings. These parameter values were determined experimentally. Minor changes to these values produced similar results for both HyperNEAT and HyperNEAT-M evolved groups. All other parameters used the same settings as in previous work [30], [29].

V. Results and Discussion

Table III and figure 2 present the average best fitness (collective gathering task performance), calculated over 20 runs, for groups evolved by controller evolution (NEAT, HyperNEAT) versus controller-morphology evolution (NEAT-M, HyperNEAT-M) methods in environments: 1, 2, and 3.

Environment 1 contained mostly *small* blocks and required a low degree of cooperation to optimally solve (for all blocks to be gathered). *Environment 2* contained an equal distribution

[2]Simulator, controller and controller-morphology evolution methods are online at: https://github.com/costofcomplexity/SSCI2019

TABLE I. NEURO-EVOLUTION PARAMETERS: USED FOR HYPERNEAT AND HYPERNEAT-M

Crossover rate		0.32
Probability to apply a mutation operator		0.34
Mutation Operators : Selection rate	Sensor weight perturbation	0.08
	Add / Remove sensor	0.07
	Sensor position / Orientation perturbation	0.10
	Sensor FOV / Range perturbation	0.07
	Add / Remove hidden node	0.05
	Add / Remove connection weight	0.05
	Connection weight perturbation	0.335
Generations per experiment / Experiment replications (runs)		200 / 20
Trial (robot lifetime) evaluations per generation		5
Population size		150
Controller connection weight range		$[-1.0, 1.0]$
Controller Hidden, output nodes		Sigmoidal
Controller Input nodes		Sensor input: $[0.0, 1.0]$
Initial Connection Density		0.5
Initial Sensory Input Nodes / Output Nodes		5 / 2
Output Nodes		2
Minimum sensor placement distance (Portion of chassis circumference)		0.01

TABLE II. EXPERIMENT AND COLLECTIVE GATHERING TASK SIMULATION PARAMETERS

Block size	Small	0.01 x 0.01
	Medium	0.015 x 0.015
	Large	0.02 x 0.02
Sensor types : Range / FOV	Ultrasonic	$(0.0, 1.0]$ / $(0.0, \pi)$
	Infrared Proximity	$(0.0, 0.4]$ / $(\pi/6, 5\pi/6)$
	Color	$(0.0, 0.4]$ / $(\pi/6, 5\pi/6)$
	Low Res Camera	$(0.0, 0.8]$ / $(\pi/9, 8\pi/9)$
	Gathering Zone Detection	Bottom facing
Sensor bearing range	$[-\pi, \pi]$ Radians	
Sensor orientation range	$[-\pi/2, \pi/2]$ Radians	
Robot *lifetime* (Time-steps per simulation task trial)	10000	
Robot group size	20	
Wait for help (cooperation) time	Remaining *lifetime*	
Robot size (Diameter) / Gripping distance	0.004 / 0.002 (As portion of environment size)	
Initial robot / block positions	Random (Outside gathering zone)	
Environment width x height / Gathering zone size	1.0 x 1.0 / 0.5 x 0.2	
Minimum / Maximum number of sensors	1 / 10	
Task environments (Blocks: small, medium, large)	1: Simple	10, 5, 0
	2: Medium	5, 5, 5
	3: Difficult	0, 5, 10
Cooperation needed for block pushing	Small	1 Robot
	Medium	2 Robots
	Large	3 Robots

TABLE III. AVERAGE TASK PERFORMANCE FOR GROUPS EVOLVED BY EACH METHOD: NEAT AND NEAT-M [29], HYPERNEAT AND HYPERNEAT-M VALUES IN PARENTHESES ARE STANDARD DEVIATIONS. ENVIRONMENTS 1, 2, AND 3, ARE LABELED *simple*, *medium* AND *difficult*, AS COLLECTIVE GATHERING IN THESE ENVIRONMENTS REQUIRES MINIMAL, MEDIUM AND MAXIMAL DEGREES OF COOPERATION (RESPECTIVELY).

	NEAT (Static Morphology)	NEAT-M (Co-adapting Morphology)	HyperNEAT (Static Morphology)	HyperNEAT-M (Co-adapting Morphology)
Environment 1 (Simple)	0.76 (0.082)	0.92 (0.071)	0.91 (0.031)	0.96 (0.01)
Environment 2 (Medium)	0.45 (0.065)	0.73 (0.087)	0.58 (0.053)	0.79 (0.074)
Environment 3 (Difficult)	0.32 (0.049)	0.52 (0.070)	0.31 (0.082)	0.50 (0.094)

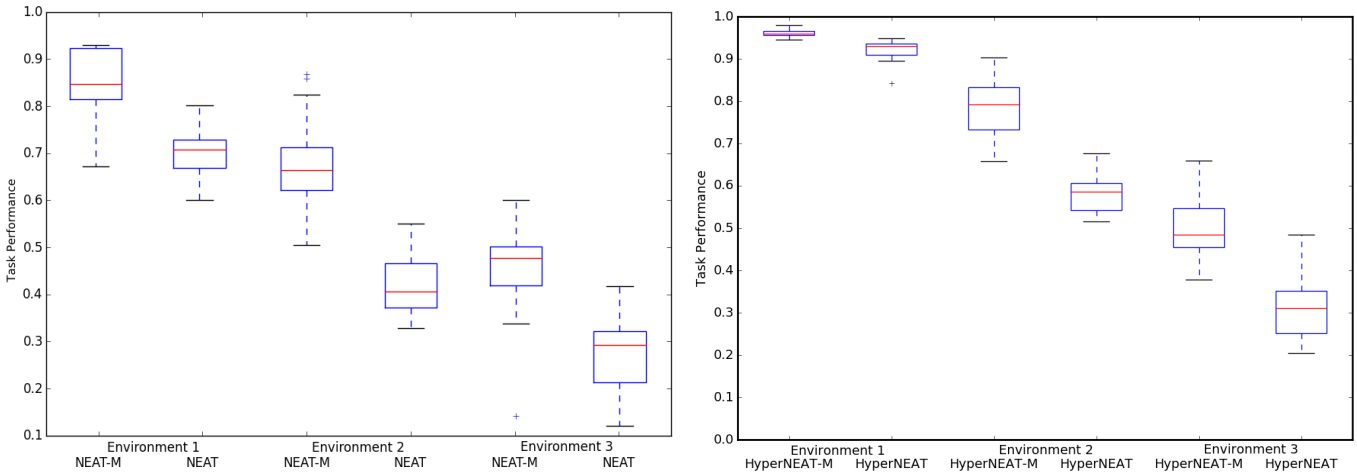

Fig. 2. Box plots of *average maximum* fitness for robot groups evolved by NEAT, NEAT-M (left, figure from related work [29]) and HyperNEAT, HyperNEAT-M (right), for environments 1 (simple), 2 (medium) and 3 (difficult). Environment 1: Required *low cooperation* and contained mostly small blocks. Environment 2: Required *medium cooperation* and contained mostly medium sized blocks. Environment 3: Required *high cooperation* and contained mostly large blocks.

of *small*, *medium* and *large* sized blocks and required a medium degree of cooperation for robots to optimally solve. *Environment 3* contained mostly *large* blocks and required high cooperation to optimally solve (table III).

Results indicated that for increasing environment complexity (collective gathering task difficulty, section III), groups with adaptive morphology and controller evolution out-performed groups using a fixed morphology and evolving controllers. This was the case between all methods that evolved controllers for fixed morphology robots (NEAT and NEAT-M [29]) versus methods that evolved robot controllers and morphologies (HyperNEAT and HyperNEAT-M). Note that, results for NEAT and NEAT-M evolved groups are taken from previous work [29], where the same simulation task environments and experiment parameters as in this study (table II, I) were used.

This fitness gain of controller-morphology evolution over controller evolution is supported by a statistically significant difference ($p < 0.05$, pair-wise t-tests [38] with Bonferroni correction [39]). That is, a statistically significant difference between the *average best* collective gathering task performance (*fitness*) of NEAT, NEAT-M, HyperNEAT and HyperNEAT-M evolved groups. Statistical tests thus indicated that there was a significant difference between groups evolved with static versus co-adapting morphologies for all environments.

The average fitness of HyperNEAT-M evolved groups was significantly higher than HyperNEAT evolved groups, and similarly the average fitness of NEAT-M evolved groups was significantly higher than that of NEAT evolved groups. This average best fitness gain also held for HyperNEAT-M over NEAT-M evolved groups (in environments 1 and 2), where HyperNEAT-M evolved groups yielded the highest average fitness overall for all task environments (figure 2, table III).

Robots adapted with HyperNEAT-M (evolving controllers and morphologies) yielded the highest average fitness for all task environments, however there was no significant difference between NEAT-M and HyperNEAT-M evolved groups in the most difficult version of the task (environment 3). In other comparisons, HyperNEAT-M yielded an average fitness gain of approximately 9%, 20% and 12% over the comparative methods (NEAT, NEAT-M and HyperNEAT) in environments 1 (simple), 2 (medium) and 3 (difficult), respectively.

Thus, for increasing task complexity, HyperNEAT-M yielded significant average fitness benefits over both controller evolution (NEAT, HyperNEAT) and comparative controller-morphology (NEAT-M) evolution methods. This was especially the case in medium difficulty task (environment 2), where there was an average 20% fitness gain. In this environment, the success of HyperNEAT-M is theorized to be due to the demonstrated benefits of controller-morphology adaptation in such collective gathering tasks [29], [20] and the nature of task environment 2.

Consider that in environment 2 there was an equal distribution of block types, where a third of the blocks could be moved by individual robots, a third required at least two robots to cooperatively push and the final third required at least three robots to cooperatively push. Comparisons with the related controller-morphology evolution method (NEAT-M, table III, figure 2) demonstrates that such an approach is suitable for evolving controller-morphology couplings that enable cooperation. In environment 2, the fitness of groups evolved with these methods is further boosted by the relative ease of individually gathering a third of the blocks.

In environments 1 and 2, the average fitness gains of HyperNEAT are further supported by its capability to exploit geometric features such as symmetry, regularity and modularity in robot morphology and controller evolution in collective behavior task environments [40], [20].

In the simplest task (environment 1), there was the least difference between controller and controller-morphology evolution methods. This was expected as achieving optimal fitness in environment 1 did not require any cooperation and there were enough robots (20 in the group, table II), to ensure sufficient concurrency in the gathering behavior so as the group could achieve near optimal task performance (table III). However, in the most difficult task environment (environment 3), HyperNEAT-M evolved groups still yielded more of an average fitness gain (12%) over other methods compared to an average gain of 9% in the simplest task (environment 1).

Thus, importantly, these results also indicate that the role of indirect (developmental) encoding has less of an impact in these task environments than that of robot controller-morphology evolution. This result is especially salient for the most difficult version of the task (environment 3), where both NEAT-M and HyperNEAT-M evolved groups yield statistically comparable average fitness. The lack of any significant fitness gain of HyperNEAT-M over NEAT-M in environment 3 is attributed to the difficult nature of the task, which mandated the cooperation of at least three robots to push two thirds of the blocks into the gathering zone. However, the impact of more complex task environments on the efficacy of these controller-morphology evolution methods is the subject of ongoing experiments and future work.

These results also support previous work demonstrating that while fixed morphology and controller evolution approaches are sufficient for relatively simple tasks, adapting behavior and morphology is advantageous as collective behavior task complexity increases [15], [19], [20].

Furthermore, this study's results support the efficacy of simple extensions to well-established neuro-evolution methods (such as NEAT [30] and HyperNEAT [21]) for the purposes of robot controller-morphology evolution. As in related work [29], this study demonstrated that such simple extensions significantly boost the quality of evolved collective behaviors given increasing task complexity, while also addressing the larger goal of defining a general evolutionary optimization methodology for automating collective robotic design.

An added value of controller-morphology evolution methods such as NEAT-M and HyperNEAT-M is that they require significantly less computational expense compared to traditional cooperative co-evolution methods that have been applied to robot controller-morphology evolution [10], [11], [13]. It is for this reason that there are relatively few examples of cooperative co-evolution of robot controller-morphology in collective robotic systems [4].

Thus, this study contributes to the larger research objective of defining a suitable robot controller-morphology evolution method for the efficient and effective design of collective robotic systems [14]. Where as previous work has focused on automating collective robotic design as a controller (collective behavior) optimization algorithm [3], [2], [4], this study also focused on demonstrating the efficacy of controller-morphology evolution methods in collective behavior tasks.

These results hold important implications for future applications in the artificial evolution of physical problem solving products [22] such as collective robotics systems [41] that have been specially designed to autonomously accomplish specific tasks in specific environments. Such methods for automating the design of robot controller-morphology couplings is thus envisaged as the computational component of rapid prototyping technologies that could be used to build physical robotic systems on demand to accomplish challenging tasks in isolated and perilous environments [16], [26], [25].

Future work will focus on evaluating the efficacy of HyperNEAT-M for evolutionary optimization of significantly larger collective robot groups (robot swarms) as well as for a diverse range of complex collective behavior tasks that require or benefit from heterogenous groups [25]. That is, the efficacy of HyperNEAT-M will be tested for evolving controller-morphology couplings for behaviorally and morphologically heterogenous groups that are significantly larger and must accomplish complex collective behavior tasks with potential real-world applications such as planetary exploration and terra-forming [16], collective construction and repair [26] and surveillance and environment monitoring [18].

VI. Conclusions

This research investigated a developmental encoding neuro-evolution method (HyperNEAT-M) that applied to the evolutionary (automated) design of robot controllers and morphologies. HyperNEAT-M was applied to automate the design of collective robotics systems such that robots were suitably designed (in terms of controller-morphology couplings) for increasingly complex collective gathering tasks (requiring increasing degrees of cooperation).

Results indicated that HyperNEAT-M produced significantly more effective behavior-morphology couplings, compared to those evolved with direct encoding methods and controllers evolved within fixed morphologies. Results thus elucidated that such developmental encoding methods could serve as a general evolutionary simulation design tool for automating collective robotic designs. An end goal of this research is the off-line evolutionary design of collective robotic systems (comprising behaviorally and morphologically heterogenous robots) for specific collective behavior tasks, and the subsequent rapid prototyping, construction and deployment of collective robotic systems in corresponding physical task environments.

REFERENCES

[1] S. Doncieux, N. Bredeche, J.-B. Mouret, and A. Eiben, "Evolutionary robotics: what, why, and where to," *Frontiers in Robotics and AI — Evolutionary Robotics*, vol. 2(4), pp. 1–18, 2015.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7(1), pp. 1–41, 2013.

[3] S. Kernbach, *Handbook of Collective Robotics: Fundamentals and Challenges*. Singapore: Pan Stanford Publishing, 2013.

[4] G. Yang, J. Bellingham, P. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. Nelson, B. Scassellati, M. Taddeo, R. Taylor, and M. Veloso, "The grand challenges of science robotics," *Science Robotics*, vol. 7650, pp. 1–15, 2018.

[5] I. Slavkov, D. Carrillo-Zapata, N. Carranza, X. Diego, F. Jansson, J. Kaandorp, S. Hauert, and J. Sharpe, "Morphogenesis in robot swarms," *Science Robotics*, vol. 3(25), 2018.

[6] J. Yu, B. Wang, X. Du, Q. Wang, and L. Zhang, "Ultraextensible ribbon-like magnetic microswarm," *Science Robotics*, vol. 9(3260), 2018.

[7] S. Li, R. Batra, D. Brown, H.-D. Chang, N. Ranganathan, and C. Hoberman, "Particle robotics based on statistical mechanics of loosely coupled components," *Nature*, vol. 567(1), pp. 361–365, 2019.

[8] H. Xie, M. Sun, X. Fan, Z. Lin, L. Chen, and L. Wang, "Reconfigurable magnetic microrobot swarm: multimode transformation, locomotion, and manipulation," *Science Robotics*, vol. 4(1), 2019.

[9] S. Kriegman, N. Cheney, and J. Bongard, "How morphological development can guide evolution," *Nature Scientific Reports*, vol. 8, 2018.

[10] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic life forms," *Nature*, vol. 406(1), pp. 974–978, 2000.

[11] G. Hornby and J. Pollack, "Creating high-level components with a generative representation for body-brain evolution," *Artificial Life*, vol. 8(3), pp. 1–10, 2002.

[12] N. Cheney, J. Bongard, and V. S. H. Lipson, "Scalable co-optimization of morphology and control in embodied machines," *Journal of the Royal Society Interface*, vol. 15, 2018.

[13] J. Auerbach and J. Bongard, "Environmental influence on the evolution of morphological complexity in machines," *PLoS Computational Biology*, vol. 10(1), 2014.

[14] M. Birattari, A. Ligot, D. Bozhinoski, M. Brambilla, G.Francesca, L. Garattoni, D. Garzon, K. Hasselmann, M. Kegeleirs, J. Kuckling, F. Pagnozzi, A. Roli, M. Salman, and T. Stutzle, "Automatic off-line design of robot swarms: A manifesto," *Frontiers of Robotics and AI*, vol. 6(59), pp. 1–6, 2019.

[15] G. Buason, N. Bergfeldt, and T. Ziemke, "Brains, bodies, and beyond: Competitive co-evolution of robot controllers, morphologies and environments," *Genetic Programming and Evolvable Machines*, vol. 6(1), pp. 25–51, 2005.

[16] M. Sabatini and G. Palmerini, "Collective control of spacecraft swarms for space exploration," *Celestial Mechanics and Dynamical Astronomy*, vol. 105(1), pp. 229–244, 2009.

[17] J. Jaffe, P. Franks, P. Roberts, D. Mirza, C. Schurgers, R. Kastner, and A. Boch, "A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics," *Nature Communications*, vol. 8(14189), pp. 1–8, 2017.

[18] G. Vasarhelyi, C. Viragh, G. Somorjai, T. Nepusz, A. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, pp. 1–13, 2018.

[19] J. Watson and G. Nitschke, "Evolving robust robot team morphologies for collective construction," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Cape Town, South Africa: IEEE Press, 2015, pp. 1039–1046.

[20] R. Putter and G. Nitschke, "Evolving morphological robustness for collective robotics," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Honolulu, USA: IEEE Press, 2017, pp. 1104–1111.

[21] K. Stanley, D'Ambrosio, and J. Gauci, "Hypercube-based indirect encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15(1), pp. 185–212, 2009.

[22] A. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, pp. 476–482, 2015.

[23] D. Howard, A., D. K. Eiben, J.-B. Mouret, P. Valencia, and D. Winkler, "Evolving embodied intelligence from materials to machines," *Nature Machine Intelligence*, vol. 1(1), pp. 12–19, 2013.

[24] D. Strombom and A. King, "Robot collection and transport of objects: A biomimetic process," *Frontiers in Robotics and AI*, vol. 5(48), 2018.

[25] L. Garattoni and M. Birattari, "Autonomous task sequencing in a robot swarm," *Science Robotics*, vol. 3(20), 2018.

[26] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343(6172), pp. 754–758, 2014.

[27] C. Parker and H. Zhang, "Collective robotic site preparation," *Adaptive Behavior*, vol. 14(1), pp. 5–19, 2006.

[28] D. D'Ambrosio and K. Stanley, "Generative encoding for multiagent learning," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Atlanta, USA: ACM Press, 2008, pp. 819–826.

[29] J. Hewland and G. Nitschke, "Evolving robust robot team morphologies for collective construction," in *The Benefits of Adaptive Behavior and Morphology for Cooperation in Robot Teams*. Cape Town, South Africa: IEEE, 2015, pp. 1047–1054.

[30] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[31] K. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines: Special Issue on Developmental Systems*, vol. 8, no. 2, pp. 131–162, 2007.

[32] D. D'Ambrosio and K. Stanley, "Scalable multiagent learning through indirect encoding of policy geometry," *Evolutionary Intelligence*, vol. 6(1), pp. 1–26, 2013.

[33] S. Didi and G. Nitschke, "Multi-agent behavior-based policy transfer," in *Proceedings of the European Conference on the Applications of Evolutionary Computation*. Porto, Portugal: Springer, 2016, pp. 181–197.

[34] S. Risi and K. Stanley, "A unified approach to evolving plasticity and neural geometry," in *Proceedings of the International Joint Conference on Neural Networks*. Amsterdam, Netherlands: IEEE, 2012, pp. 1–8.

[35] F. Lambercy and J. Tharin, *Khepera III User Manual: Version 3.5*. Lausanne, Switzerland: K-Team Corporation, 2013.

[36] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*. Redwood City: Addison-Wesley, 1991.

[37] G. Nitschke, M. Schut, and A. Eiben, "Evolving behavioral specialization in robot teams to solve a collective construction task," *Swarm and Evolutionary Computation*, vol. 2, no. 1, pp. 25–38, 2012.

[38] B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge University Press.

[39] O. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56(293), pp. 52–64, 1961.

[40] J. Watson and G. Nitschke, "Evolving robust robot team morphologies for collective construction," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Cape Town, South Africa: IEEE, 2015, pp. 1039–1046.

[41] M. Jelisavcic, M. D. Carlo, E. Hupkes, P. Eustratiadis, J. Orlowski, E. Haasdijk, J. Auerbach, and A. Eiben, "Real-world evolution of robot morphologies: A proof of concept," *Artificial life*, vol. 23(2), pp. 206–235, 2017.