Evolving Morphological Robustness for Collective Robotics

Ruben Putter
Department of Computer Science
University of Cape Town
Cape Town, South Africa
Email: PTTAND010@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 study evaluates a *Neuro-Evolution* (NE) method for controller evolution in simulated robot teams, where the goal is to evaluate the *morphological robustness* of evolved controllers. *Artificial Neural Network* (ANN) controllers are evolved for a specific sensory configuration (morphology) and then evaluated on a set of different morphologies. The morphological robustness of evolved controllers is evaluated according to team task performance given a collective construction task of increasing complexity. The overall objective was to ascertain an appropriate method for evolving ANN controllers that are readily transferable to robot teams with varied morphologies. Such controller transfer is necessary if task specifications change and different sensory configurations are required, or if robots are damaged and some sensors become disabled. In both cases it is ideal if teams continue to exhibit consistent behavior and a similar task performance. Results indicate that an indirect (developmental) encoding NE method consistently evolves controllers that fully function when transferred to teams with varied morphologies. That is, where comparable or higher task performances were yielded compared to controllers evolved specifically for the varied morphology.**

I. Introduction

An open problem in *collective* [1] and *swarm robotics* [2] is ascertaining appropriate sensory-motor configurations (morphologies) for robots comprising teams that must work collectively given automatically generated controllers [3]. In *evolutionary robotics* [4], a popular method is to co-evolve robot behaviors and morphologies [5], [6], [7], [8]. Within the purview of such *body-brain* co-evolution, indirect encoding (developmental) methods have been effectively demonstrated for various single-robot tasks [9], [10], [11]. However, with few exceptions [12], [13], [14] there has been relatively little work that evaluates developmental methods to evolve behavior and morphology for collective robotics tasks, excluding related research in self-assembling and reconfigurable collective robotics [15], [16]. Specifically, this study focuses on developmental methods to evolve controllers that exhibit consistently robust behaviors such that robot teams effectively adapt to the loss of sensors or new sensory configurations without significant degradation of collective task performance, that is, *morphologically robust behavior*.

This study contributes to an open problem in evolutionary robotics, that is to evolve robot controllers that can continue to appropriately function given unforseen morphological change to the robot such as sensor damage or malfunction [17], [18]. Such robust controller evolution would yield significant advantages in applications where robust autonomous behavior is continually required [19]. In this study, the specific focus is on the evolution of morphologically robust behavior for robotic teams that must accomplish collective behavior tasks.

Given the added computational complexity of co-evolving body-brain couplings for behaviorally and morphologically heterogenous robots, this study focuses on evolving collective behaviors for *homogenous* teams with a range of fixed morphologies. We apply the HyperNEAT [20] neuro-evolution method to evolve behavior for a range of morphologically homogenous teams that must solve various collective construction tasks. HyperNEAT was selected as it is a developmental encoding NE method with demonstrated benefits that include exploiting task geometry to evolve modular and regular ANN controllers with increased problem-solving capacity [21], [22].

The research contribution is to demonstrate the efficacy of developmental neuro-evolution (HyperNEAT) for addressing the evolution of morphologically robust controllers in the context of collective robotics. To address this objective, this study tested and evaluated five different robot team sensory configurations (morphologies) in company with three collective construction tasks of increasing complexity. The fittest *Artificial Neural Network* (ANN) controller evolved for a given morphology was then transferred and evaluated in each of the other team morphologies. Such transferred controllers were also evaluated in each of the three tasks. In all cases, evolved controller efficacy was evaluated in terms of collective construction task performance yielded by a robot team.

The *Collective construction* task [23] was selected since it benefits from fully automated robot teams that must exhibit robust behavior to handle changing task constraints, potential robot damage and sensor noise. For example, collective construction of functional structures and habitats in remote or hazardous environments [24].

As in related work [23], the task was for robots to search the environment for *building-block* resources, then move them such that are connected to other blocks. The task is solved if the team connects all blocks forming a structure during its *lifetime*, and this is equated with optimal task performance. However, the task is considered partially solved if only a subset of the blocks are connected by the team, though in such instances, team task performance is proportional to the number of blocks connected. In this study, task complexity was equated with the number of robots needed to collectively push blocks together (cooperation) to be connected as a structure and

whether the blocks must be connected in a specific sequence, that is, according to a *construction schema*, table III.

Hence, we report a preliminary investigation into developmental methods (HyperNEAT is tested in this case study) for evolving ANN controllers that are robust to morphological change in robotic teams or swarms that must operate in dynamic, noisy and hazardous environments.

II. Methods

HyperNEAT [20] is an extension of NEAT (*Neuro-Evolution of Augmented Topologies*) [25], where ANNs are indirectly encoded using a CPPN (*Compositional Pattern Producing Network*) [26]. HyperNEAT was selected as it has a number of benefits demonstrated in previous work [22], [13]. This includes its capability to exploit geometric features such as symmetry, regularity and modularity in robot morphology and the task environment during controller evolution.

In this study, HyperNEAT evolves the connection weights between each robot's sensory input layer, hidden layer and motor output layer, where each robot used the same controller, making teams homogenous. Controller evolution experiments were initialized with a given morphology (table I). However, in one experiment set, each robot's sensor configuration of team morphology could be *co-adapted* via HyperNEAT activating and deactivating sensory input node connections over the evolutionary process. For these experiments (section III), *add connection* and *remove connection* mutation operators (table II) from previous work [14] were applied every generation to a sensory input node chosen with uniform random selection. The mutation operator applied depended on whether the chosen input node was connected or not. The construction zone sensor (table I) was permanently activated for all morphologies and could not be disconnected, as this enabled robots to detect *construction zones* (section III-A).

Table I presents a list of *morphology identification* (ID) numbers and the number and type of sensors that correspond to each morphology. For example, morphology 2 has four proximity sensors, one ultrasonic sensor, one colour-ranged sensor, and one low-resolution camera. Note that all morphologies have a construction sensor as this is necessary to complete the collective construction task.

A. Robot Team Controller

Each robot in the team used an ANN controller with *N* sensory input nodes, determined by the given morphology being evaluated (table I). Each robot's controller mapped sensory inputs, via a fully connected hidden layer, to two motor outputs, the robot's left and right wheels (figure 1).

Figure 1 illustrates the sensory configuration for $N = 11$ (morphology 1), and the associated substrate and CPPN used by HyperNEAT. For each robot morphology (table I), the sensors corresponding to the input layer of the controller was a circle *N* nodes distributed about a robot's periphery, where the exact geometric configuration corresponded to the morphology being evaluated (figure 1 illustrates morphology 1)[1]. The intermediate ANN hidden layer reflects the configuration of the input layer, preserving the geometry of the sensory input layer, that is, the direction of each sensor's FOV (figure 1). The ANN was initialized with random weights normalized to the range [-1.0, 1.0], with full connectivity between adjacent layers, however, partial connectivity was evolvable via the CPPN generating a zero weight. Collectively all sensors approximated up to a 360 degree *Field of View* (FOV).

The nodes comprising each robot's ANN controller, connected by the CPPN, were placed in the substrate illustrated in figure 1. Each node in the substrate was placed at specific (x, y) locations in the two-dimensional geometric space of the substrate (x, y axes were in the range: [-1, 1]). Connection weights in the controller were evolved via querying the CPPN for the weight of any connection between two points (x_1, y_1) and (x_2, y_2) by inputting (x_1, y_1, x_2, y_2) into the CPPN, which subsequently output the associated weight. During HyperNEAT's evolutionary process, the CPPN was evolved via having nodes and connections added and removed, as well as connection weight values mutated [20].

Thus, the CPPN evolved a connectivity pattern across the geometry of the ANN via querying all the potential connections for their weights. This connectivity pattern was effectively a function of the task and ANN geometry, which enabled HyperNEAT to exploit the structure (regularity, repetition and symmetry) of the task and robot morphology. For example, there was symmetry in the robot morphology in terms of the positioning of sensors about each robot's periphery (figure 1) and there was regularity and repetition in the collective construction task, in terms of repeating block types comprising modular and regular structures. In the collective construction task, *modularity* was defined as the composition of modular structures (buildings in construction zones) from a sequence of connected blocks and *regularity* was defined as the same sequence of blocks repeated in a building.

Previous work has demonstrated that the indirect encoding of an evolved CPPN facilitates the evolution of robot controllers with increased task performance enabled by a compact representation of task and robot geometry [27], [13]. Table II presents the HyperNEAT parameters used in this study, where *delta* was angle between the (x_1, y_1, x_2, y_2) positions of nodes in the substrate. These parameter values were determined experimentally. All other HyperNEAT parameters not listed in table II, were set as in previous work [27].

1) Detection Sensors: Each robot was equipped with various sensor types, where the exact sensor complement, including the relative position and direction on the robot depends upon the given experiment (section III) and morphology being evaluated (table I). Each robot had *N* sensors corresponding to the *N* inputs comprising the robot's ANN sensory input layer (figure 1), each with a range of *r* (portion of the environment's length). A robot's sensory FOV was split into *N* sensor quadrants, where all sensors were constantly active for the duration of the robot's lifetime. The *nth* sensor returned a value in the normalized range [0.0, 1.0], in the corresponding *nth* sensor quadrant. A value of 0.0 indicated that no blocks were detected and a value of 1.0 indicated that an object was detected at the closest possible distance to the given sensor.

Table II presents the different sensor types used in this study, where the functional properties of each sensor (range

[1] Illustrations of all robot morphologies tested can be found at: https://github.com/not-my-name/SSCI_Paper_Appendix

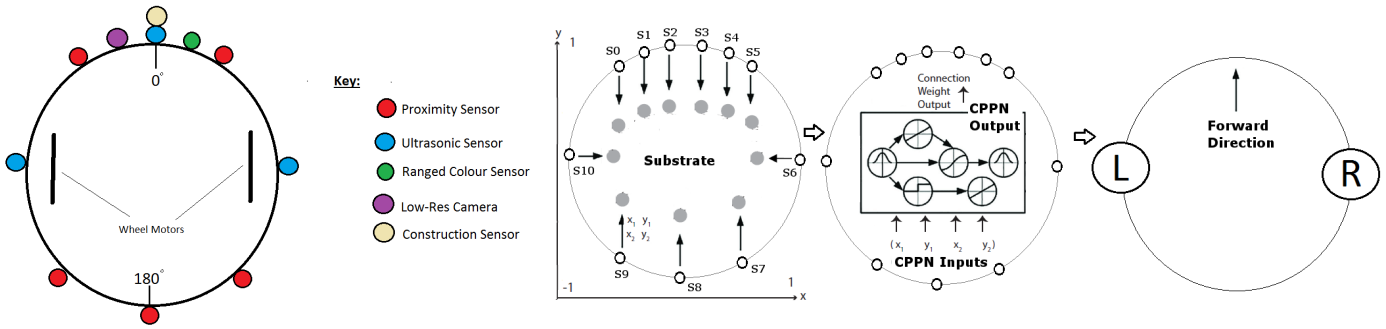

Fig. 1. *Left (color):* Robot morphology 1, with relative positions of various sensors on the robot. *Right (gray-scale):* ANN Topology as it relates to robot morphology 1: 11 Sensory inputs [S0, S10]. Sensory inputs connect to a hidden layer (left). Connection weight values between two nodes (x_1, y_1, x_2, y_2) are evolved by querying the CPPN (center) with x, y values in the range [-1, 1] (axis shown). The hidden layer is fully connected to all inputs and outputs (connectivity not depicted). Motor outputs (right) *L* and *R* determine the speed of the left and right wheels, respectively, and thus a robot's speed and direction.

and FOV) were abstractions of corresponding physical sensors typically used on the Khepera III robots [28]. In table II, range values are units defined in relation to the environment size (20 x 20) and FOV values are in radians. Each morphology also included a special construction zone detection sensor that activated with a value in the range [0.0, 1.0] whenever a robot came into contact with a block that must be connected with other already connected blocks.

The construction zone sensor calculated the squared Euclidean norm, bounded by a minimum observation distance, as an inversely proportional distance between *this* robot and the closest construction zone, where a value of 1.0 indicated the robot (pushing a block) was in contact with the construction zone and a value of 0.0 indicated that the robot (pushing a block) was the maximum possible distance from the closest construction zone. Robots were unable to detect each other, thus all cooperative interactions were *stigmergic* [29] where robots interacted via pushing blocks into the environment's construction zone. Furthermore, robots had no *a priori* knowledge of the construction schema, but rather must discover the construction schema rules by trial and error. Also, once at least two blocks had been pushed and connected together this formed a construction zone (section III-A), that was then visible to each robot's construction zone sensor.

2) Movement Actuators: Two wheel motors control a robot's heading at constant speed. Movement is calculated in terms of real valued vectors (*dx* and *dy*). Wheel motors (*L* and *R* in figure 1) need to be explicitly activated. A robot's heading is determined by normalizing and scaling its motor output values by the maximum distance a robot can 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, corresponding to the left and right wheels, respectively, producing an output in the range: [-1.0, 1.0]. These output values indicate how fast each respective wheel must turn. Equal output equates to straight forward motion and unequal output results in the robot rotating about its own axis. The d_{max} value indicates the maximum distance a robot can move in one simulation iteration (normalized to 1.0, table II).

III. Experiments

Experiments[2] tested 15 robots in a bounded two dimensional continuous environment (20 x 20 units) with randomly distributed type *A*, *B* and *C* blocks (table II). Robots and blocks were initialized with random orientations and positions throughout the environment. A construction schema (table III) dictated the sequence of block types that must be connected together in order that a specific structure be built [30]. Figure 2 presents an example of the team of 15 robots working to solve the collective construction task in the simulation environment containing a distribution of five of each block type (*A*, *B* and *C*), colored blue, green and red, respectively. Other colored blocks in the environment indicate those already connected in construction zones (three illustrated). The purple, blue and green semi-circles emanating from each robot represent the FOV of active sensors, where the different colors correspond to different sensor types (table II).

As the purpose this study was to demonstrate the morphological robustness of HyperNEAT evolved controllers for a collective behavior task of increasing complexity, the first two versions of the collective construction task required no cooperation and some degree of cooperation, respectively, though any block could be connected to any other block. Where as, the most complex version of the task required cooperation and block types to be connected according to a construction schema (table III).

A. Collective Construction Task

This task required the robot team to search the environment for building-blocks and cooperatively push them together in order that they connected to form a structure, where connected blocks then formed a construction zone. Task complexity was equated with the degree of cooperation required to collectively push blocks and connect them together in the construction zone and whether or not a construction schema was required. In this construction task, there were three block types, *A*, *B* and *C* requiring one, two and three robots to push, respectively. Cooperation occurred when at least two robots simultaneously pushed a type *B* block, or at least three robots pushed a type *C* block.

[2]Source code for all experiments is online at: https://github.com/not-my-name/ExperimentsRerun

TABLE I. SENSORY CONFIGURATION (NUMBER OF SENSORS) FOR EACH MORPHOLOGY.

Morphology ID	Proximity Sensors	Ultrasonic Sensors	Color Ranged Sensors	Low-Resolution Camera	Construction Zone Sensors
1	5	3	1	1	1
2	4	1	1	1	1
3	0	0	1	1	1
4	2	0	1	1	1
5	2	2	1	1	1

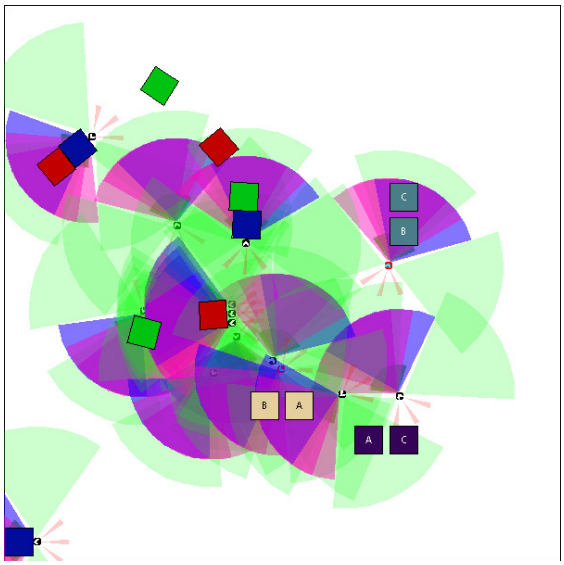

Fig. 2. Example of the simulation environment. Robots search for randomly distributed type A, B, and C blocks (blue, green and red, respectively). Other colored and labeled blocks indicate those already connected in construction zones. Different coloured semi-circles emanating from each robot represent the field of view of currently active different sensor types (table II).

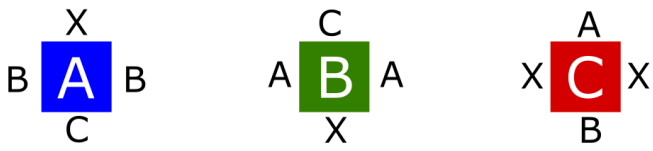

Fig. 3. Task level 3 construction schema: *A*, *B*, and *C* are the block types. The label on each side of each block type indicates what block type can be connected to this side. An *X* label indicates that no block can be connected.

Table III presents the three levels of task complexity for the collective construction task. Level 1 was the least complex as it did not require any cooperation, given that it in this case there were only type *A* blocks in the environment. Level 2 was of medium complexity as there are equal numbers of type *A*, *B*, and *C* blocks in the environment, where block types *B* and *C* required at least two and three robots to push, respectively. Level 3 was the most complex, as it required the same degree of cooperation as task level 2, though blocks had to be connected according to a construction schema. Figure 3 illustrates this construction schema, where the label on each of the four sides of each block type indicates what other block type can be connected to the given side. The *X* label indicates that no block can be connected to a given side.

The construction zone was formed via at least two blocks pushed together and was thus any structure being built in the environment. Once a construction zone was created, all blocks attached to it were fixed in position and could not be disconnected. The task mandated a maximum of three construction zones and unconnected blocks had to be pushed and connected to one of these construction zones. For task levels 1 and 2, any block could be connected to any other block, meaning that when two blocks were pushed together they automatically connected. For task level 3, blocks had to be pushed together such that they were connected on specific sides according to the construction schema (figure 3).

Team task performance was calculated as the number of blocks connected in construction zones during a team's lifetime (equation 1), where average task performance was calculated as the highest task performance selected at the end of each run (100 generations) and averaged over 20 runs (table II). The fitness function to direct controller evolution was a weighted sum of the number of times *type A* blocks were pushed by *one robot* and connected (a in equation 1), the number of times *type B* blocks were pushed by *two robots* and connected (b in equation 1), and the number of times *type C* blocks were pushed by *three robots* and connected (c in equation 1).

$$f = r_a a + r_b b + r_c c \tag{1}$$

Parameter tuning experiments found that setting the weights (reward values r_a, r_b and r_c) in equation 1 to 0.3, 0.6, and 1.0, respectively, resulted in functional controller evolution. Fitness was normalized to the range $[0.0, 1.0]$ using the maximum possible fitness yielded from all blocks being pushed and connected in construction zones.

B. Experiment Design

Experiments evaluated the *morphological robustness* of HyperNEAT evolved controllers for robot teams that must accomplish collective construction tasks of increasing complexity (section III-A). We measured the average comparative task performance of controllers evolved for a given team morphology and task complexity where such controllers were then transferred to and re-evaluated in other team morphologies. Thus, teams that achieved an average task performance that was not significantly lower across all *re-evaluated* morphologies were considered to be *morphologically robust*.

This study comprised five experiment sets, where the first four experiment sets evolved controllers given team morphologies $1-4$ (table I) for three levels of increasing task complexity (table III). The fifth experiment set investigated the *co-adaptation* of team morphology and behavior, where morphology 5 (table I) was used as the initial sensory configuration for all robots in the team. This fifth experiment set was included in order to gauge if co-adapting behavior and

TABLE II. EXPERIMENT, NEURO-EVOLUTION AND SENSOR PARAMETERS

Generations	100	
Sensors per robot	11, 8, 4	
Evaluations per genotype	3	
Experiment runs	20	
Environment length, width	20	
Max Distance (Robot movement per iteration)	1.0	
Team size	15	
Team Lifetime (Simulation iterations)	1000	
Lifetimes per generation	3	
Type A blocks (1 robot to push)	15	
Type B blocks (2 robots to push)	15	
Type C blocks (3 robots to push)	15	
Mutation rate	Add neuron	0.25
	Add connection	0.008
	Remove connection	0.002
	Weight	0.1
Population size	150	
Survival rate	0.3	
Crossover proportion	0.5	
Elitism proportion	0.1	
CPPN topology	Feed-forward	
CPPN inputs	Position, delta, angle	
Sensor	Range	FOV
Proximity Sensor	1.0	0.2
Ultrasonic Sensor	4.0	1.2
Ranged Colour Sensor	3.0	1.5
Low-Res Camera	3.0	1.5
Colour Proximity Sensor	3.0	3.0

TABLE III. TASK COMPLEXITY. NOTE: TASK LEVEL 3 INCLUDES A CONSTRUCTION SCHEMA (FIGURE 3).

Construction Task Complexity	Level 1	Level 2	Level 3
Type A blocks (1 robot to push)	15	5	5
Type B blocks (2 robots to push)	0	5	5
Type C blocks (3 robots to push)	0	5	5
Construction schema	No	No	Yes

morphology yielded any benefits in this collective construction task as it did in related collective behavior tasks [14].

Each experiment set comprised a controller evolution stage and a re-evaluation stage (morphological robustness test). For controller evolution, each experiment applied HyperNEAT to evolve team behavior for 15 robots for 100 generations, where a generation comprised five team *lifetimes* (1000 simulation iterations). Each team lifetime tested different robot starting positions, orientations, and block locations in the simulation environment. The fittest controller evolved for each task level (yielding the highest absolute task performance) was then *re-evaluated* for morphological robustness in all other morphologies. For example, the fittest controller evolved for morphology 1 was re-evaluated in morphologies $2-4$ and the average task performance calculated across all re-evaluation runs.

Each re-evaluation run was *non-evolutionary*, where controllers were not further evolved, and each re-evaluation run was equivalent to one team lifetime. Re-evaluation runs were repeated 20 times for a given morphology, in order to account for random variations in robot and block starting positions and orientations. For each fittest controller, re-evaluated in a given morphology, an average task performance was calculated over these 20 runs, and then an overall average task performance was computed for all re-evaluated morphologies.

As per this study's objectives, these morphological re-evaluation runs tested how robust the fittest evolved controllers (for a given morphology) were to variations in that morphology. Thus, re-evaluating the fittest controllers on other morphologies emulated sensor loss due to damage or new robot morphologies introduced due to changing task constraints.

IV. RESULTS & DISCUSSION

For each controller evolution experiment, the average maximum task performance of controllers evolved for a given morphology and level of task complexity, was recorded. Specifically, this task performance was calculated by running the absolute fittest controller evolved after 20 evolutionary runs (for a given a morphology and task level), in the same morphology over 20 non-evolutionary runs. This is presented in figures 4, 5 and 6 (left column), where controllers were evolved given morphologies 1-5 (table I). In figures 4, 5 and 6 (left column), these results are presented from left to right. For example, average task performance results for morphology 1 are plotted on the left-most side and average task performance results for morphology 5 are plotted on the right-most side.

For each re-evaluation experiment, the fittest controller evolved for a given morphology and task complexity level was re-evaluated in all other morphologies (for the same level of task complexity), and an average task performance computed over 20 runs. These morphological re-evaluation results are presented in figures 4, 5 and 6 (right column). Each of the five plots (from left to right) in each figure corresponds to the fittest controller evolved for each of the five morphologies and re-evaluated in all other morphologies. For example, the left-most plot presents the average task performance of the fittest controller evolved in morphology 1 and re-evaluated on morphologies $2-5$. Where as, the right-most plot presents the average task performance of the fittest controller evolved given morphology 5 (as the initial sensory configuration) and re-evaluated on morphologies $1-4$.

To gauge the impact of a given team morphology (table I) in company with a given level of task complexity (table III), the *t-test* [31] ($p < 0.05$), was applied in pair-wise comparisons between sets of controller evolution results[3] (figures 4, 5 and 6, left column). Within each given level of task complexity, no statistically significant difference was found between controllers evolved given morphologies $1-5$ (controller evolution experiments $1-5$). Controller evolution experiments $1-4$ were those implementing controller evolution in fixed sensory configurations (morphologies $1-4$). Where as, controller evolution experiment 5 used morphology 5 as the initial sensory configuration and subsequently co-adapted behavior (controller) and morphology (complement of sensors). The lack of statistical difference between controllers evolved given morphologies $1-4$ (table I) indicates that these sensory configurations were not sufficiently different so as to result in significantly different average maximum task performances. Also, the lack of any significant difference between the average maximum

[3]Statistical test results for pair-wise comparisons for the fittest evolved controllers (for a given morphology) tested in each other morphology (individually) is online at: https://github.com/not-my-name/SSCI_Paper_Appendix

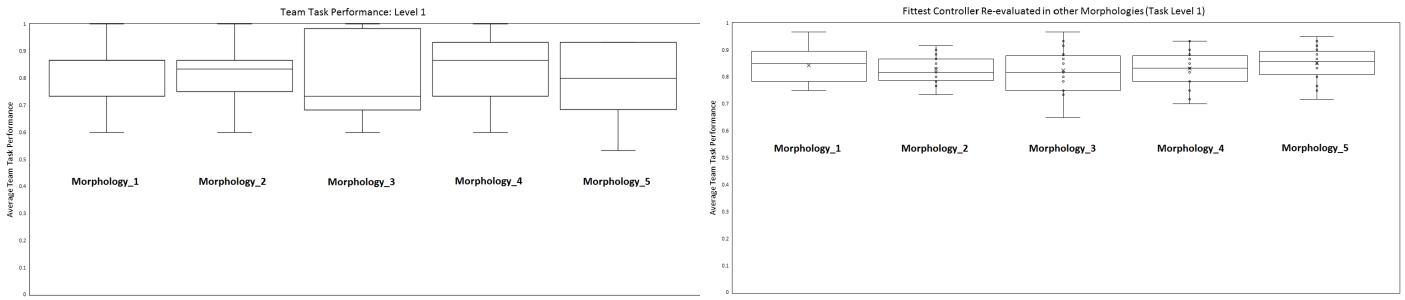

Fig. 4. *Left column:* Average team task performance for controller evolution (*task level 1*) given morphologies $1-5$ (depicted from left to right). *Right column:* Average task performance given the fittest controller evolved for each successive morphology ($1-5$, shown left to right) re-evaluated in all other morphologies. For example: Left-most plot is average task performance of fittest controller evolved for morphology 1, re-evaluated in morphologies $2-5$. Right-most plot is the average task performance of fittest controller evolved for morphology 5, re-evaluated in morphologies $1-4$.

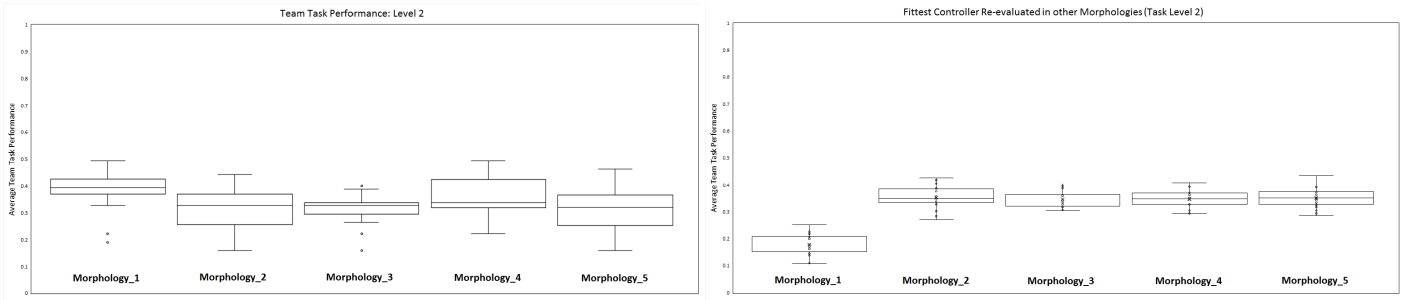

Fig. 5. *Left column:* Average team task performance for controller evolution (*task level 1*) given morphologies $1-5$ (depicted from left to right). *Right column:* Average task performance given the fittest controller evolved for each successive morphology ($1-5$, shown left to right) re-evaluated in all other morphologies. For example: Left-most plot is average task performance of fittest controller evolved for morphology 1, re-evaluated in morphologies $2-5$. Right-most plot is the average task performance of fittest controller evolved for morphology 5, re-evaluated in morphologies $1-4$.

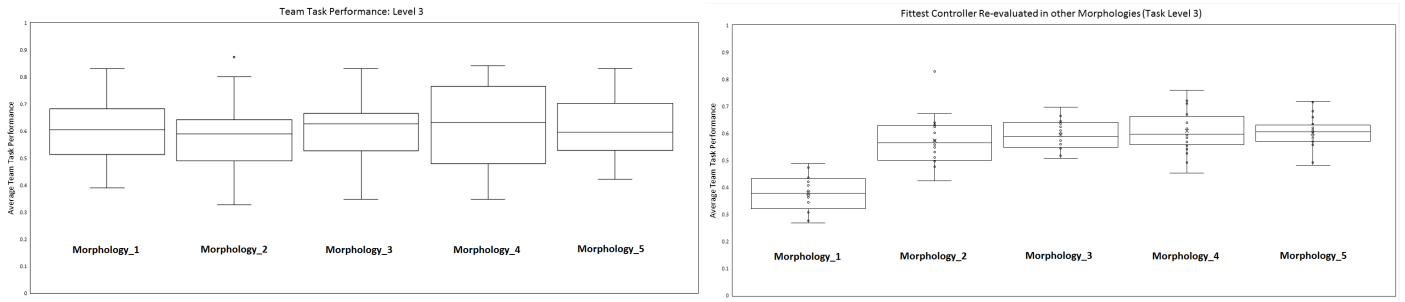

Fig. 6. *Left column:* Average team task performance for controller evolution (*task level 1*) given morphologies $1-5$ (depicted from left to right). *Right column:* Average task performance given the fittest controller evolved for each successive morphology ($1-5$, shown left to right) re-evaluated in all other morphologies. For example: Left-most plot is average task performance of fittest controller evolved for morphology 1, re-evaluated in morphologies $2-5$. Right-most plot is the average task performance of fittest controller evolved for morphology 5, re-evaluated in morphologies $1-4$.

task performance of controllers evolved given morphologies $1-4$ and behavior-morphology co-adaptation (starting with morphology 5), supports previous results demonstrating that behavior-morphology co-adaptation yields at least comparable task performance benefits (compared to fixed morphology controller evolution) in collective behavior tasks [14].

However, to address this study's main objective it was necessary to ascertain the morphological robustness of the fittest controller evolved in each morphology when re-evaluated in all other morphologies. To gauge the morphological robustness of the fittest controllers evolved for a given morphology ($1-5$), and a given task complexity, we applied the t-test in pairwise comparisons of two result data sets. First, the average maximum task performances yielded by controller evolution in morphologies $1-5$ and second, the average maximum task performances yielded from re-evaluating the fittest controller evolved for a given morphology in all other morphologies (section III-B).

Statistical test results indicated no significant difference (with one exception) between average task performance results yielded by controller evolution and morphological re-evaluation experiments for all task complexity levels. Specifically, the average maximum task performance yielded by controllers evolved given morphologies 2-5 (figures 4, 5, 6, left column) was not significantly lower than the average task performance yielded by the fittest controllers (evolved in morphologies $1-5$), and then re-evaluated on other morphologies (figures 4, 5, 6, right column). The exception was morphology 1 in task complexity levels 2 and 3. In these tasks, the fittest controller evolved for morphology 1, yielded a significantly higher average task performance than that yielded when this fittest controller was re-evaluated in morphologies $2-5$.

Hence, these results indicate that controllers evolved by HyperNEAT for a given morphology (table I), overall have the capacity to continue to effectively operate when transferred to other morphologies. This result was found to hold for all four of the five morphologies that controllers were evolved for, and for all levels of task complexity tested (table III).

The efficacy of HyperNEAT for evolving morphologically robust controllers is further supported by the controller evolution experiments that used morphology 5 (section III). In this case, the number of sensors was adapted meaning that team behavior and morphology were co-adapted. Specifically, these controller evolution experiments began with the sensory configuration of morphology 5 (table I) and enabled and disabled sensor connections to better couple morphology with the evolved controller. Hence, the fittest controller evolved in this case often corresponded to a sensory configuration dissimilar to morphology 5 (the initial sensory configuration). Results indicated that the fittest controller evolved for morphology 5, when re-evaluated in other morphologies, yielded an average task performance that was statistically comparable to the average task performance yielded by controller evolution given morphology 5. This result was observed for all three task complexity levels (figures 4, 5, and 6, right column).

Thus, controllers evolved for fixed morphologies ($1-4$, table I), were found to be *morphologically robust*, as there was no significant difference in average maximum task performance when the fittest controller (evolved for morphology $1-4$), was re-evaluated in other morphologies. Furthermore, the fittest controller evolved for an adaptive morphology (5, table I), was similarly found to be morphologically robust, given that the average maximum task performance yielded when this fittest controller was re-evaluated in other morphologies ($1-4$), was comparable to the average maximum task performance yielded from morphology 5 controller evolution experiment. This is theorized to be a result of the complexity of co-adapting effective controller-morphology couplings [32] within limited periods of artificial evolution (100 generations in these experiments, table II), offset by the transference of evolved *connectivity patterns* [33] as functional controllers across varying robot morphologies [34]. Such connectivity patterns encode behaviors that do not rely upon specific sensory-motor mappings in controllers and thus do not necessitate specific task environment configurations, such as specific numbers of agents or objects. This in turn facilitates the transfer of controllers across varying team morphologies [35], [36], [37].

These results are corroborated by related work [34], [13], and contribute further empirical evidence that HyperNEAT yields significant benefits in evolving robot controllers that effectively operate in other morphologies. That is, this study further demonstrated HyperNEAT's capability to exploit geometric properties such as regularity, repetition and symmetry in robot morphology and environment [20], where such modularity and geometric properties are encoded in evolved connectivity patterns. This is prevalent in this study, as the configuration of sensors on each robot's periphery was symmetrical for all morphologies tested (section II-A). Also, the collective construction task required that blocks be connected together in a repeated manner in a symmetrical bounded simulation environment (20x20 units, table II). The capability of HyperNEAT evolved controllers to operate in different morphologies is further supported by other research [35] demonstrating that evolved indirect sensory-motor mappings can encapsulate effective behaviors with relatively few task environment and robot geometric relationships, such as desired positions and angles between robots and different object types.

The efficacy of HyperNEAT for evolving morphologically robust controllers for collective behavior tasks of varying complexity is also supported by related research in *multi-agent policy transfer* [35], [36], [37]. Policy transfer methods facilitate the transfer of behaviors across tasks of increasing complexity or between dissimilar tasks. Such studies have demonstrated that HyperNEAT is an effective method for evolving behaviors in one collective behavior task and then transferring the evolved behavior to a related but more complex task (for example, where robots have more complex sensory-motor configurations to process increased task complexity) with relatively little loss in average team task performance.

However, we hypothesize that the morphological robustness of HyperNEAT evolved controllers demonstrated across all morphologies tested (table I) and all levels of task complexity (table III) was facilitated by the use of morphologically and behaviorally homogenous teams. Specifically, one controller was evolved for all robots in a team and all robots used the same sensory configuration, meaning all robots had the same *collective behavior geometry* [38]. This in turn simplified the transfer of evolved controllers across varying morphologies with no significant degradation in average task performance.

Hence, overall, this study's results demonstrate that HyperNEAT is an appropriate method for evolving morphologically robust controllers. That is, controllers that are fully functional in a range of team morphologies. In order to ascertain how well HyperNEAT evolved controllers generalize, ongoing research is evaluating the evolution of morphologically robust behaviors in behaviorally and morphological heterogenous teams for complex collective behavior tasks that are irregular and without repetition or symmetry. Furthermore, current research is comparing HyperNEAT to related evolutionary approaches that have demonstrated controllers able to accomplish multiple disparate tasks in dynamic environments [39], as well as direct encoding neuro-evolution methods such as NEAT [25].

V. Conclusions

This research presented a study on the efficacy of HyperNEAT for evolving *morphologically robust* behaviors for homogenous robot teams that must solve a collective behavior task of increasing complexity. That is, the average maximum task performance of behaviors evolved for a given team morphology (robot sensory configuration) that was then transferred to a different team morphology. Controllers that did not yield degraded task performance when transferred to another morphology were considered to be morphologically robust. The objective was to test and evaluate methods that generate morphological robust behaviors, where varying morphologies emulated sensor damage or intentional changes to the sensory systems of robotic teams.

Results indicated that HyperNEAT was appropriate for generating morphologically robust controllers for a collective construction task of increasing complexity. This task required robots to cooperatively push blocks such that they connected

together to form structures. Task complexity was regulated by the number of robots required to push blocks and a construction schema mandating that specific block types be connected in specific ways. These results support the notion that developmental neuro-evolution methods, such as HyperNEAT, are appropriate for controller evolution in robotics applications where robot teams must adapt during their lifetime to damage or otherwise must dynamically adapt their sensory configuration to solve new unforseen tasks.

REFERENCES

[1] R. Kube and H. Zhang, "Collective robotics: from social insects to robots," *Adaptive Behaviour*, vol. 2, no. 2, pp. 189–218, 1994.

[2] G. Beni, "From swarm intelligence to swarm robotics," in *Proceedings of the First International Workshop on Swarm Robotics*. Santa Monica, USA: Springer, 2004, pp. 1–9.

[3] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008.

[4] 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.

[5] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic life forms," *Nature*, vol. 406, no. 1, pp. 974–978, 2000.

[6] H. Lund, "Co-evolving control and morphology with lego robots," in *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, H. Fumio and R. Pfeifer, Eds. Berlin, Germany: Springer, 2003, pp. 59–79.

[7] 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.

[8] J. Auerbach and J. Bongard, "Environmental influence on the evolution of morphological complexity in machines," *PLoS Computational Biology*, vol. 10(1), p. e1003399. doi:10.1371/journal.pcbi.1003399, 2014.

[9] C. Mautner and R. Belew, "Evolving robot morphology and control," *Artificial Life and Robotics*, vol. 4(3), pp. 130–136, 2000.

[10] 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.

[11] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson, "Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Amsterdam, Netherlands: ACM Press, 2013, pp. 167–174.

[12] Y. Asai and T. Arita, "Coevolution of morphology and behavior of robots in a multiagent environment," in *Proceedings of the SICE 30th Intelligent System Symposium*. Tokyo, Japan: The Society of Instrument and Control Engineers, 2003, pp. 61–66.

[13] 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.

[14] 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.

[15] R. O'Grady, A. Christensen, and M. Dorigo, "Swarmorph: Morphogenesis with self-assembling robots," in *Morphogenetic Engineering, Understanding Complex Systems*, R. Doursat, Ed. Berlin, Germany: Springer-Verlag, 2012, pp. 27–60.

[16] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345(6198), pp. 795–799, 2014.

[17] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314(5802), pp. 1118–1121, 2006.

[18] A. Cully, J. Clune, D. Tarapore, and J. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521(1), pp. 503–507, 2015.

[19] R. Brooks and A. Flynn, "Fast, cheap and out of control: A robot invasion of the solar system." *Journal of the British Interplanetary Society*, vol. 1, no. 1, p. 478485, 1989.

[20] K. Stanley, D'Ambrosio, and J. Gauci, "Hypercube-based indirect encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 1, pp. 185–212, 2009.

[21] P. Verbancsics and K. Stanley, "Constraining connectivity to encourage modularity in hyperneat," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2011, pp. 1483–1490.

[22] D. D'Ambrosio and K. Stanley, "Scalable multiagent learning through indirect encoding of policy geometry," *Evolutionary Intelligence Journal*, vol. 6, no. 1, pp. 1–26, 2013.

[23] 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.

[24] J. Werfel and R. Nagpal, "Three-dimensional construction with mobile robots and modular blocks," *The International Journal of Robotics Research*, vol. 27(3-4), pp. 463–479, 2008.

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

[26] 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.

[27] 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.

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

[29] R. Beckers, O. Holland, and J. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Proceedings of the International Workshop on the Synthesis and Simulation of Living Systems*. Cambridge, USA: MIT Press, 1994, pp. 181–189.

[30] 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.

[31] B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge, UK: Cambridge University Press, 1986.

[32] R. Pfeifer and J. Bongard, *How the body shapes the way we think*. Cambridge, USA: MIT Press, 2006.

[33] J. Gauci and K. Stanley, "Autonomous evolution of topographic regularities in artificial neural networks," *Neural Computation journal*, vol. 22, no. 7, pp. 1860–1898, 2010.

[34] S. Risi and K. Stanley, "Confronting the challenge of learning a flexible neural controller for a diversity of morphologies," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Amsterdam, The Netherlands: ACM, 2013, pp. 255–261.

[35] P. Verbancsics and K. Stanley, "Evolving static representations for task transfer," *Journal of Machine Learning Research*, vol. 11(1), pp. 1737–1769, 2010.

[36] S. Didi and G. Nitschke, "Hybridizing novelty search for transfer learning," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Athens, Greece: IEEE Press, 2016, pp. 10–18.

[37] ——, "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.

[38] D. D'Ambrosio, J. Lehman, S. Risi, and K. Stanley, "Evolving policy geometry for scalable multi-agent learning," in *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*. Richland, USA: ACM Press, 2010, pp. 731–738.

[39] E. Izquierdo-Torres and T. Buhrmann, "Analysis of a dynamical recurrent neural network evolved for two qualitatively different tasks: Walking and chemotaxis," in *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (Artificial Life IX)*. Winchester, UK: MIT Press, 2008, pp. 257–264.