

The Benefits of Adaptive Behavior and Morphology for Cooperation in Robot Teams

Jamie Hewland
 Department of Computer Science
 University of Cape Town
 Cape Town, South Africa
 Email: jhewland@gmail.com

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

Abstract—This is a study on the role of morphological (sensor configuration) and behavioral (control system) adaptation in simulated robot teams that must accomplish cooperative tasks. The research objective was to elucidate the necessary features and computational mechanics of a method that automates the behavior-morphology design of robot teams that must accomplish cooperative tasks (tasks that cannot be optimally solved by individual robots). Results indicate that automating behavior-morphology design is beneficial as task complexity increases, compared to evolving behaviors in fixed morphology teams. However, increased task complexity does not necessarily equate to the evolution of increased morphological complexity in teams.

I. INTRODUCTION

A challenging problem in *collective robotics* [19] and the related fields of *evolutionary robotics* [30] and *swarm robotics* [3] is behavior-morphology design in robot teams. It is usually impractical to ascertain, *a priori*, an optimal sensory-motor configuration (morphology) coupled with an optimal controller (behavior) for every robot in a team, where the team must accomplish any given collective behavior task. In this and related research, a collective behavior task requires *cooperation* [28] in a robot team [35], where cooperation emerges from the local interactions of many individuals [4].

There has been a significant amount of work done on co-evolving behavior and morphology for single (simulated) robots [36], [14], [24], [17], [34], [25], [5], [2] to find morphologies and coupled controllers specifically suited to solving various tasks. However, with notable exceptions [1], [7], [31], there has been relatively little work on co-evolving behaviors and morphologies in robot teams that must solve collective behavior (cooperative) tasks.

This study is an initial step towards automated morphology-behavior design methods that, given any collective behavior task, evolve an appropriate behavior and morphology for a robot team. The key advantage of such automated design methods is that they alleviate the time and expense associated with traditional design of multi-robot systems. Importantly, such methods solve the problem of human engineers having to anticipate the most appropriate behavior and morphology for each robot in the team and how their interactions will affect collective problem solving behavior. This becomes impractical as the number of robots and task complexity increases. Such

methods potentially allow for teams to be designed (artificially evolved), tested and verified in simulation for a given task before being built (for example with rapid prototyping and three-dimensional printing technology [23], [24]) and deployed to solve counter-part real-world tasks.

To address this end goal, this study's objective is to ascertain the most suitable team behavior and morphology for a collective gathering task of varying complexity. In this study, task complexity (ranging from low to high) is equated with the degree of cooperation required for robots to gather all blocks in a simulation environment. That is, cooperation is the number of robots needed to collectively push a block into a *gathering zone* in the simulation environment. Specifically, this study aims to produce a method that automates behavioral and morphological design in robot teams, where such teams efficiently solve a collective gathering task for varying degrees of task complexity.

The team behavior-morphology automated design method reported in this study extends the *Neuro-Evolution of Augmenting Topologies* (NEAT) [38] method to account for the evolution of robot sensory configurations (morphologies) in addition to ANN controllers. In this study, a robot's *morphology* is its sensory configuration [9]. As in related work, this is the placement, number and type of sensors on a robot's body [18], and sensor properties such as *Field of View* (FOV) and range [8], [7]. A robot's controller is an adaptive topology *Artificial Neural Network* (ANN), where behavior elicited by the ANN is evolved as a coupling to the robot's morphology (represented by the ANN's sensory input layer). Thus a robot's sensory configuration, together with ANN connectivity and connection weight values, is another parameter set in the genotype encoding of the ANN.

Given the added computational complexity and computing time needed for evolving the design of behaviorally and morphologically heterogeneous teams, this study only tests the evolution of behaviorally and morphologically *homogenous* teams. That is, one (evolved) controller was copied to each robot in the team, and each robot also used the same (evolved) morphology (sensory configuration).

II. METHODS

To test controller (behavior) and morphology (sensory configuration) evolution in robot teams, an extension to *Neuro-Evolution of Augmenting Topologies* (NEAT) [38], was de-

veloped. The extension accounts for morphological adaptation in robot teams, and is called *Neuro-Evolution of Augmenting Topologies and Morphologies* (NEAT-M). NEAT was selected as the method to extend since it and other extensions [37] have already been successfully applied to controller design in various multi-agent tasks [10], [11], [27], [29].

A. NEAT: Neuro-Evolution of Augmented Topologies

NEAT is a competitive co-evolution NE method that uses mechanisms for *historical gene marking*, *speciation*, and *complexification* [38] in its adaptation process.

Complexification is the incremental growth from minimal ANN controller topology. NEAT begins with a homogenous population of simple controllers (with no hidden nodes) and adapts connection weights and topology as a function of task complexity. Thus, NEAT biases the search towards minimal dimensional spaces and only increases search space dimensionality (adding controller structure) if the task requires it.

Speciation in NEAT calculates if two controllers will be in the same or a new species (according to a genotype compatibility threshold) after controllers have been recombined and mutated every generation. Speciating the population means controllers will only compete within their given species. This protects new innovations in controller topology adaptation.

Historical gene markings allow NEAT to add new structure and recombine controllers with differing topologies, since gene markings are evidence of controller homology (if genes comprising different controllers share the same origin). This protects topological innovations made as part of each controller's structural adaptation, and ensures that evolving controllers are given sufficient time to reach their potential before being removed from the adaptive process [38].

B. NEAT: Genetic Operators

Mutation: Adapts both connection weights and controller topology. To adapt controller weights, this application of NEAT uses mutation with a Gaussian distribution to change each gene in each genotype with a given probability (table I).

Controller topology mutation is the basis of complexification and works via adding genes to genotypes. New genes are new connections or controller nodes represented by a mutated genotype. These mutations are simultaneously applied to each genotype with a given probability (table I). Added connection weights connect two previously unconnected nodes in a mutated controller. When a new node c is added, the existing connection between two existing nodes a and b is disabled. A new connection between nodes a and c (weight value = 1) is initialized. A second connection is initialized between node c and b , with the same weight value that previously connected nodes a and b . When a new gene is added, the new gene is assigned an incremented *global innovation number*.

Recombination: NEAT tracks the historical origin of all genes using innovation numbers, so only homologous genes in two given controllers will be recombined. That is, NEAT only recombines genotypes (controllers) with ancestral genes in common. Matching genes are randomly selected for child

genotypes. Disjoint and excess genes are inherited from the fitter parent, or at random in the case of equal fitness.

C. NEAT-M

NEAT-M equates the evolving topology of an ANN's input layer with a robot's adaptive sensory configuration. Hence, NEAT-M includes additional genes corresponding to sensors that are subject to the neuro-evolution process. These additional genes are: *sensor type*, *bearing*, *orientation*, *FOV* and *range*. NEAT-M also includes three additional genetic operators, *sensor position*, *sensor number*, and *sensor FOV and range*. The first operator changes the *bearing* and *orientation* of a given sensor on the robot body, the second operator adds and removes sensors from the robot body, and the third operator changes the FOV and range of a given sensor type (figure 1). Adapting sensor position and orientation was achieved via placing a sensor anywhere on the circular periphery of the robot, but not within a given minimum distance of sensors already on the robot (table I).

D. NEAT-M: Genetic Operators

NEAT-M included three genetic operators to adapt the morphology (sensory configuration) of a robot: *sensor position mutation*, *sensor FOV and range mutation*, and *sensor number mutation*. Table I specifies the mutation rates for these operators.

Sensor Position Mutation: Changes the bearing (position) or orientation (sensed direction) of a given sensor on the robot body. The operator is applied to one randomly selected sensor, with a given degree of probability and perturbs the sensor's bearing or orientation (randomly selected) via adding a randomly selected value from a Cauchy distribution (table I). This incrementally perturbs the position of a given sensor on the robot body and also changes the direction the sensor is facing with respect to the heading of the robot (figure 1). If a sensor's bearing value exceeds its valid range (table I), the mutated value that exceeds the upper limit is added to the lower limit. If a sensor's orientation value exceeds its valid range, then the maximum or minimum orientation is used (table I).

Sensor FOV and Range Mutation: Either the Infrared or Ultrasonic sensor type is randomly selected. This sensor type's FOV or range (one is randomly selected) is then mutated. The selected sensor type's current FOV or range is then perturbed via adding a randomly selected value from a Cauchy distribution (table I), where the range of values for FOV ($(0, \pi]$ Radians) and range $(0, 1.0]$ remains fixed.

Sensor Number Mutation: Given that adding a new sensor equates to adding a node to the ANN controller's input layer, NEAT's existing mutation operators were used to add and remove sensors from the robot body. When adding a new sensor to the controller, the goal was to affect a change in controller behavior, but not to inhibit controller functionality. Hence, new connections between new sensors and other nodes in the controller were randomly initialised but with a low connection density (table I) so as to limit the number of overall connections in the ANN, as a rapid increase in the number of connections for each additional sensor node would increase the likelihood of a dysfunctional controller resulting.

Similarly, the weight values of new connections between new and existing nodes were specially initialised (table I) so as not to cause significant changes in ANN functionality, because of no or very high sensor node activation values resulting from zero or high connection weight values. As presented in table I, initialization was different for the different sensor types used in this study’s experiments (section V).

E. NEAT-M: Speciation

The speciation mechanism was adapted to account for varying sensor positions, sensor bearings, orientation, ranges and FOV and thus speciation equation 1 was used.

$$\delta = \delta_n + c_4 \cdot \bar{B} + c_5 \cdot \bar{O} + c_6 \cdot \bar{R} + c_7 \cdot \bar{F} \quad (1)$$

Where, δ_n is the result of the NEAT speciation equation [38], c_4 and c_5 are constant factors, \bar{B} is the vector of differences in sensor bearings, \bar{O} is the vector of differences in sensor orientations, \bar{R} is the vector of differences in sensor ranges, and \bar{F} is the vector of differences in sensor FOV. The sensor bearing, \bar{B} , and orientation, \bar{O} , range, \bar{R} , and FOV, difference vectors use values normalized to: $[-1.0, 1.0]$.

F. NEAT and NEAT-M Generational Process

For this study’s experiments (section V), the generational process of NEAT and NEAT-M functioned as follows.

- 1) N genotypes (controllers) are initialised with two motor outputs and a bias node connected to these outputs. NEAT-M controllers are initialized with two sensory input nodes. NEAT controllers are initialized with six sensory input nodes. In both cases, all sensory input nodes are fully connected to motor output nodes and connection weights randomly initialized to within a given range (table I).
- 2) Genotypes are ranked by fitness. A genotype is randomly selected from the genotype population’s *elite portion* (table I) and copied M times as the M controllers for a robot team. For NEAT-M teams the genotype is also decoded into a morphology that is copied to each of the M robots in simulation.
- 3) At each generation, each genotype is evaluated Q times (represented as a *lifetime* of Y simulation iterations for the robot team). An average fitness is then assigned to the genotype (team) being evaluated.
- 4) One generation is the evaluation of all N genotypes. At the end of each generation speciation and genetic operators are applied within the population’s elite portion and a new population generated.
- 5) The new population replaces the previous population and steps 2-4 are iterated for X generations.

III. COLLECTIVE GATHERING TASK

In this study, a variation of the well established *collective gathering* task [4] was selected given its pertinence to real-world multi-robot applications [20] including toxic waste clean-up [33], sweeping mine fields [15], and exploration of remote environments [6]. Collective gathering requires a simulated robot team to search a bounded environment for resources

(blocks) randomly distributed throughout the environment. The team must collectively gather blocks via cooperatively pushing them back to a *gathering zone*. Blocks are of types *small*, *medium*, or *large*, and varying degrees of complexity are required to accomplish the task. *Small* blocks could be pushed by individual robots, *medium* blocks could only be moved by two or more robots pushing against one side, and *large* blocks could only be moved by three or more robots pushing against one side (table I).

Task complexity is adapted via changing the distribution of block types in the environment. For example, a high number of *small* blocks versus *medium* and *large* blocks in the environment indicates a low degree of cooperation and thus low task complexity, as robots could work concurrently and individually in order to gather most blocks. Whereas a distribution of only *large* and *medium* blocks in the environment indicates a high degree of cooperation and thus high task complexity, given that robots would always have to work cooperatively (in pairs or groups of three) in order to push blocks back to the gathering zone. Collective gathering task performance (team fitness) is the total number of blocks pushed into the gathering zone during a team’s lifetime.

IV. ROBOT CONTROLLERS AND SENSORS

Experiments (section V) tested only homogenous teams so all robots used the same ANN controller for a given generation of the NEAT and NEAT-M adaptive process.

Teams used either a fixed morphology (controllers evolved with NEAT) based on the sensory-motor configuration of Khepera III robots [21] or an adaptive morphology (controllers and sensory configuration evolved with NEAT-M). In the latter case, the robot body is modeled on the Khepera III, but NEAT-M adapts the number and type of sensors and their placement on the robot chassis. In this adaptive morphology case, all robots started with one randomly placed Infrared (IR) proximity sensor, and one proximity sensor for detecting the gathering zone. This proximity sensor is positioned in a fixed position on the under-side of the robot and is not subject to the NEAT-M adaptive process.

A. Sensor numbers and types:

The two types of sensors used were *Infrared* (IR) proximity and *Ultrasonic* sensors. Sensor FOV was modeled as conical fields emanating from the outer edge of a robot’s body for a given range (figure 1).

Each robot had three *infrared* proximity ([4, 6] in figure 1) and three *ultrasonic* sensors ([1, 3] in figure 1) giving it partial sensory coverage within 360 degrees. An additional ground facing sensor (not shown in figure 1) was included by default in both fixed and adaptive morphology robots. This sensor detected when the robot was positioned in the gathering zone.

The IR proximity sensors were placed at bearings of $\pm 40^\circ$, $\pm 75^\circ$ and 180° with respect to the center of the robot’s body. The ultrasonic sensor positions were placed at bearings of 0° , $\pm 90^\circ$. This configuration is dissimilar to standard sensor complement of the Khepera III, which includes 11 IR proximity sensors and five ultrasonic sensors [21]. However, this reduced sensor complement was used to speed up simulations

TABLE I. SIMULATION AND NEURO-EVOLUTION PARAMETERS: COLLECTIVE GATHERING TASK.

Simulation and Neuro-Evolution Parameters	
Simulation runs / Generations per run	20 / 250
Epochs (team lifetimes) per generation / Iterations per epoch	5 / 10000
Genotype population size / elite portion / Robot team size	100 / 30 / 20
ANN connection weight range	[-1.0, 1.0]
Robot size (diameter) / Gripping distance	0.004 / 0.002 (Portion of environment size)
Wait for help (cooperation) time	Remaining <i>lifetime</i>
Maximum robot movement per simulation iteration	0.013 (Portion of environment size moved per iteration)
Initial robot / block positions	Random (Outside gathering zone)
Environment width x height / Type / Gathering zone size	1.0 x 1.0 / Continuous / 0.5 x 0.1
Small / Medium / Large block size (Width / Height)	0.01 x 0.01 / 0.015 x 0.015 / 0.02 x 0.02
Infrared sensor range	(0.0, 1.0] (NEAT-M) / 0.05 (NEAT)
Ultrasonic sensor range	(0.0, 1.0] (NEAT-M) / 0.1 (NEAT)
Infrared sensor Field Of View (FOV)	(0, π] Radians (NEAT-M) / 0.2 Radians (NEAT)
Ultrasonic sensor FOV	(0, π] Radians (NEAT-M) / 1.22 Radians (NEAT)
Infrared / Ultrasonic sensor bearing range	$[-\pi, \pi]$ Radians
Infrared / Ultrasonic sensor orientation range	$[-\pi/2, \pi/2]$ Radians
Sensor position / number mutation (x_0, γ)	Cauchy mutation (0, 5)
Sensor FOV / range mutation (x_0, γ)	Cauchy mutation (0, 5)
Add or remove Infrared / Ultrasonic sensor probability	0.6 / 0.4
Sensor connection weight mutation (μ, σ range)	Gaussian mutation (0, [0.04, 0.2])
Sensor position / number mutation probability	0.1
Connection weight mutation probability / mutation range	0.1 / [-1.0, 1.0]
Initial / new node connection density	0.5 / 0.1 (Portion of maximum connection density)
Speciation constants	$c_1 = 1.0, c_2 = 1.0, c_3 = 0.4, c_4 = 0.4, c_5 = 0.1, \delta_t = 3.0$
Crossover / (Add or Remove) node connection probability	0.4 / 0.1
Initial sensory input nodes	3 (NEAT-M) / 6 (NEAT)
Maximum sensory input nodes / Motor output nodes	10 (NEAT-M) / 6 (NEAT) / 2

TABLE II. EXPERIMENTS RANGE FROM A BLOCK TYPE DISTRIBUTION REQUIRING A HIGH DEGREE OF COOPERATION (ENVIRONMENT 1) TO ACCOMPLISH (GATHER ALL BLOCKS) TO MEDIUM COOPERATION (ENVIRONMENT 2), TO LOW COOPERATION REQUIRED (ENVIRONMENT 3). VALUES IN PARENTHESES ARE THE NUMBER OF ROBOTS NEEDED TO MOVE A BLOCK OF THE GIVEN TYPE.

	Small (1) blocks	Medium (2) blocks	Large (3) blocks
Environment 1	0	5	10
Environment 2	5	5	5
Environment 3	10	5	0

and approximate incomplete sensory information indicative of physical robotic experiments. The IR and ultrasonic sensor types were selected as they are currently what is available on the standard Khepera III robot [21]. Also, the properties of these sensors can also be readily adapted for experiments with physical Khepera robots.

B. Motor outputs:

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 ([7, 8] 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 I). 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 is used.

C. NEAT and NEAT-M: Initial ANN Controllers

NEAT evolved robots: Begin with a simple ANN controller that fully connects six sensor input nodes ([1, 6] in figure 1, right) to two motor output nodes ([7, 8] in figure 1, right). These initial connections were randomly initialized with connection weights within a pre-specified range (table I), and the ANN controller was then subject to *complexification* during NEAT adaptation. The ANN controller of NEAT evolved teams used six sensory input nodes connected to H hidden layer and to two output nodes (section IV-B). The number of hidden layer nodes, connections and weight values between inputs nodes, hidden nodes, and output nodes was evolved by NEAT.

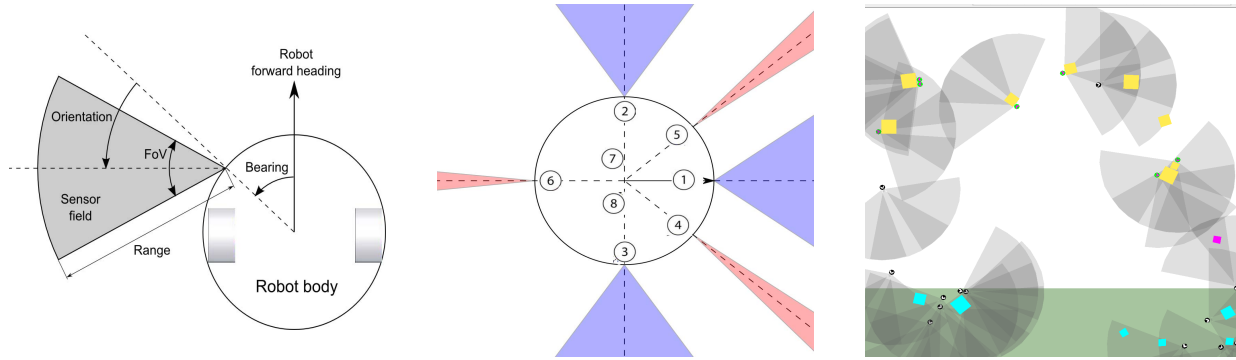


Fig. 1. *Left*: Example robot with one sensor. Bearing determines sensor position on robot body periphery with respect to the robot’s heading. Orientation then determines the direction the sensor faces with respect to this bearing. A robot’s heading is a preset forward facing direction (parallel to its wheels, shown as gray rectangles on each side of the robot). *Center*: Sensory-motor configuration of robots in fixed morphology teams. Nodes [1, 2, 3]: Ultrasonic sensors. Nodes [4, 5, 6]: IR proximity sensors. Nodes [7, 8]: Wheel motors. *Right*: Example simulation environment containing 20 robots, a distribution of small, medium, and large blocks. The *gathering zone* is highlighted at the bottom. Sensory views (FOV and range) of individual robots also highlighted.

NEAT-M evolved robots: Begin with one randomly placed IR and one ultrasonic sensor, and no hidden nodes, fully connected to the two output nodes. Connections were initialized with a pre-specified density and weight values (table I). Figure 1 (left) presents an example of an initial NEAT-M sensory configuration. The ANN in this case would be one sensor node directly connected to two motor output nodes.

To ensure that NEAT and NEAT-M controllers are initially able to execute actions and accomplish the collective gathering task with some degree of success, motor outputs were fixed throughout the adaptation process. For both NEAT and NEAT-M controllers, used hidden and output sigmoidal [16] nodes. Also, all controllers were initialized with a bias node (constant weight value of -1.0) connected to motor outputs. The bias node was not subject to adaptation.

Heuristics: Given that the research focus was on evolving cooperative behavior in teams with respect to fixed or adaptive morphologies, behavioral heuristics were included to speed up the evolution of collective gathering behaviors. If a robot was within *gripping distance* (table I) 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 I), 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.

V. EXPERIMENTS

Collective gathering experiments measured the impact of controller evolution in fixed (NEAT) versus adaptive (NEAT-M) morphology teams on the number of *blocks gathered* (team fitness) for a given simulation *environment*.

Experiments executed simulations of 20 robots in a bounded two dimensional continuous environment containing a distribution of *small*, *medium*, and *large* blocks (table II). Blocks are randomly distributed throughout the environment, excluding the *gathering zone*, whereas robots are initially randomly placed in the gathering zone. Figure 1 illustrates

an example simulation environment containing 20 robots and a distribution of *small*, *medium*, and *large* blocks.

The three block type distributions given in table II correspond to three environments that test the impact of collective gathering tasks requiring high, medium and low cooperation (environments 1, 2, and 3, 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 in teams during controller evolution [28]. For example, the block distribution in experiment 1 (table II) requires cooperation between at least two robots in order for any blocks to be gathered. Whereas, experiment 3 did not encourage cooperation as most blocks could be gathered by individual robots.

Each experiment applied NEAT (controller evolution in fixed morphology teams) or NEAT-M (controller evolution in adaptive morphology teams) to evolve team behavior for 250 generations. A generation comprised five *epochs*, where one epoch was 10000 simulation iterations, representing the execution of one team *lifetime* (table I). One team lifetime was a simulated task scenario that tested different robot starting positions, orientations, and block locations in a given environment (table II). At each generation, average team task performance (fitness) was taken over the five epochs and used for NEAT and NEAT-M genotype selection (section II). Average team fitness was then taken at the end of each run and calculated over 20 runs.

Only homogenous teams were tested, meaning that at each NEAT and NEAT-M generation, the selected controller was copied 20 times to represent the team. NEAT-M evolved teams were also morphologically homogenous per generation, meaning that morphology was adapted at each generation but each robot’s morphology was the same for a given generation.

Table I presents the simulation and neuro-evolution (NEAT and NEAT-M) parameter settings. These parameter values were determined experimentally. Minor changes to these values produced similar results for both NEAT and NEAT-M evolved teams. All other NEAT parameters used the same settings as in previous work [29], [38].

A. Team Fitness Evaluation

A team's fitness (task performance) was calculated based on the time taken to move blocks (of a given type) from an initial position in the environment into the gathering zone. *Small*, *medium*, and *large* blocks yield progressively higher reward values (table I) for being moved into the gathering zone. These varying fitness rewards reflect the degree of difficulty (cooperation) required to move the different block types. A fitness bonus was rewarded to teams that gather all blocks before the end of the team's lifetime.

Team fitness f (equation 2) was an average taken over five epochs (team *lifetimes*) per generation (table I).

$$f = 100 \times \frac{v_c}{v_t} + 20 \times (1.0 - \frac{s_e}{s_t}) \quad (2)$$

where v_c was the total value of gathered blocks, v_t the total value of all the resources, s_e the number of elapsed simulation iterations, and s_t the team lifetime. Since only homogenous teams were tested, each robot attempted to maximize its own and thus team fitness f . At the end of a team's lifetime, f was normalized to the range: [0.0, 1.0].

VI. RESULTS AND DISCUSSION

Results indicated that for any given collective gathering task complexity (section III), teams with adaptive morphology and controller evolution out-performed teams using a fixed morphology and evolving controllers. This result is supported by a statistically significant difference ($p < 0.05$, two-tailed t-test [13]) between the adaptive and fixed morphology teams in terms of *best* and *average* team task performance (fitness) for all environments.

Figure 2 presents, as box plots, these average and best team fitness results (calculated over 20 runs) for experiments labeled 1, 2, and 3. *Experiment 1* applied NEAT-M to adapt the behavior and morphology of teams, and *experiment 2* applied NEAT to adapt the behavior in fixed morphology teams. Both experiments tested the adaptation of teams for three task environments. *Environment 1* contained mostly large blocks and required a high degree of cooperation to optimally solve (that is, for all blocks to be gathered). *Environment 2* contained mostly medium sized blocks and required a medium degree of cooperation to optimally solve. *Environment 3* contained mostly small blocks and required low cooperation to optimally solve (table II). In figure 2 environments 1, 2, and 3, are labeled *large*, *medium* and *small*, respectively.

The fittest team evolved in environment 1 (mostly *large* blocks) of experiment 1 (adaptive morphology using NEAT-M) evolved after 135 generations and attained a maximum task performance efficiency of 65%, whereas, the fittest team evolved in environment 1 of experiment 2 (fixed morphology teams using NEAT), evolved after 116 generations but only achieved a maximum task performance efficiency of 45%.

The fittest team evolved in environment 2 (mostly *medium* blocks) of experiment 1 (NEAT-M) evolved after 227 generations and attained a maximum task performance efficiency of 87%, whereas, the fittest team evolved in environment 2 of experiment 2 (NEAT), evolved after 121 generations but only achieved a maximum task performance efficiency of 60%.

The fittest team evolved in environment 3 (mostly *small* blocks) of experiment 1 (NEAT-M) evolved after 235 generations and attained a maximum task performance efficiency of 93%, whereas, the fittest team evolved in environment 3 of experiment 2 (NEAT), evolved after 217 generations but only achieved a maximum task performance efficiency of 88%. These results support previous work supposing that whilst fixed morphology and controller evolution approaches are sufficient for relatively simple tasks, adapting behavior and morphology is advantageous as task complexity increases [18], [26], [25], [12], [1], [7], [32], [22]. This previous research used single agent tasks, however, this study's results indicate that the notion is extendable to collective behavior tasks.

Results in figure 2 also provide a valuable complement to current co-evolution methods for adapting simulated robot behaviors and morphologies. That is, these results demonstrate that a simple extension (encoding controller behavior and robot morphology on a single genotype) to an existing ANN controller design method (NEAT) is sufficient for automating behavior-morphology design for robot teams that must accomplish a collective behavior task. This contrasts with other behavior-morphology automated design methods that use *Cooperative Co-evolution Algorithms* (CCAs) to co-evolve populations of behaviors and morphologies [36], [24], [17], [34], [25], [2] [7]. Due to the computational complexity of such CCAs, with few exceptions [1], [7], [31], they are typically applied to adapt the behavior and morphology of single robots that must accomplish relatively simple tasks, not requiring cooperation. This contributes to the research objective of ascertaining the most suitable defining features for automated robot team behavior-morphology design methods that function for collective behavior tasks with varying complexity.

Results also indicate that a more complex collective gathering task encourages the selection of a simpler team controller and morphology. This is evidenced by the fittest controller¹ evolved for *environment 1* (table II) in *experiment 1*. In this case NEAT-M was applied to adapt team behavior and morphology in the collective gathering task requiring a high degree of cooperation. The fittest ANN controller evolved three IR proximity sensors and three ultrasonic sensors with varying properties. For example, proximity sensor 1 (figure 1, <http://people.cs.uct.ac.za/%7Egnitschke/SSCI2015/>) has a bearing of 0.88 radians (with respect to a preset forward heading of the robot). At this position on the robot's body, the sensor has an orientation of 1.57 radians, and a FOV of 1.36 radians. The sensor's range equates to a distance of 0.38 (as a portion of the environment's size). Figure 1 (<http://people.cs.uct.ac.za/%7Egnitschke/SSCI2015/>) also depicts the bias node and the sensor node corresponding to the proximity sensor on the underside of each robot. However, these nodes were not subject to morphological adaptation.

This fittest controller evolved in *environment 1* (mostly large blocks and high degree of cooperation required) of experiment 1 (NEAT-M) is a relatively simple reactive controller with seven sensory input nodes directly connected to two motor output nodes, yet evolved after only 135 generations and achieved a task performance of 65% of optimal.

¹Fittest controller topologies for experiment 1 (adaptive team morphologies) and experiment 2 (fixed team morphologies) can be viewed at: <http://people.cs.uct.ac.za/%7Egnitschke/SSCI2015/>.

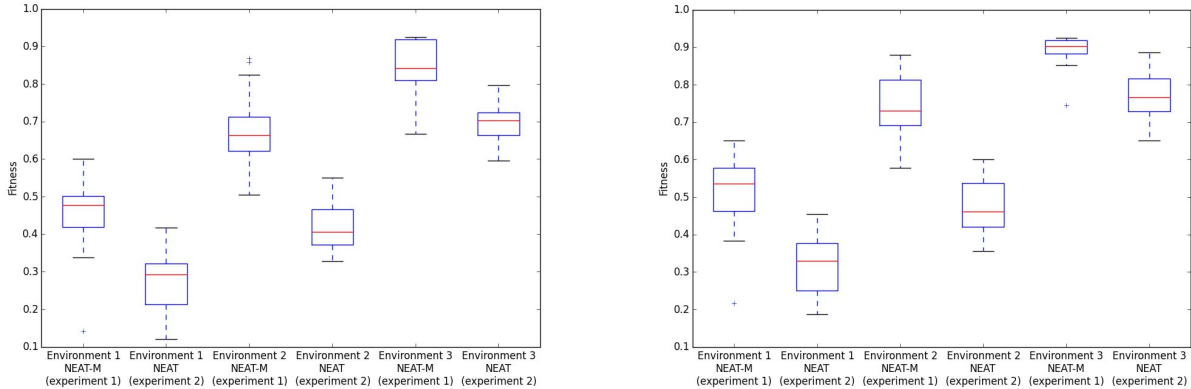


Fig. 2. Box plots of *average* (left) and *maximum* (right) team fitness for experiments 1 and 2. *Experiment 1*: Team behavior and morphology adapted with NEAT-M. *Experiment 2*: Team behavior (fixed morphology) adapted with NEAT. Both experiments were tested in three environments. Environment 1: Required *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.

In contrast, the controller and morphology evolved for the fittest team in *environment 2* (mostly medium blocks and medium degree of cooperation required) of experiment 1 is considerably more complex. This controller took 227 generations to evolve and comprises four proximity sensors and four ultrasonic sensors connected to 16 hidden nodes and the two motor output nodes. However, this additional complexity resulted in a higher team task performance (that is, 87% of optimal). Also, the fittest controller and morphology evolved for the fittest team in *environment 3* (mostly small blocks and low cooperation required) of experiment 1 was of comparable complexity to that evolved in environment 1, but achieved a maximum task performance of 93%.

This result demonstrates that a relatively simple controller and morphology is sufficient for teams to attain a high fitness in the most complex and the simplest versions of the collective gathering task tested. This is theorized to be a product of the speciation mechanism in NEAT-M which removes poorly performing species (genotype sub-populations) and protects morphological innovations of new species. This, for example, allows for the selection of teams with relatively simple morphologies that eventually adapt their behaviors to outperform more complex teams. The selection of simpler controllers is also enabled and directed by the task environment. That is, the distribution of block types in the environment has been demonstrated to have a significant impact on behavior adaptation and underlying controller complexity in related collective gathering tasks [28].

Also, the lack of any cost associated with increasing morphological complexity [2] (as observed in environment 2 of experiment 1) is hypothesized to facilitate the selection of more complex morphologies given specific task environment features (in this case, the distribution of small, medium and large blocks in environment 2, II). However, why such simple controllers and morphologies were not evolved for environment 2 remains the subject of ongoing research.

Future work will focus on extending NEAT-M to account for the evolution of behaviorally and morphologically heterogeneous teams that must accomplish a range of collective

behavior tasks. Also, the scalability of NEAT-M is still to be tested. For example, evolving robot swarms that must accomplish various collective behavior tasks with a range of complexity. Additionally, in this study there was no cost imposed on increasing morphological complexity, however, future work will do so. For example, a greater number of sensors and increasing the range and effectiveness of sensors did not carry any cost, such as reducing robot lifetime (to emulate increased battery drain). This would allow one to investigate the relationship between morphological and task complexity and test current hypotheses [2]. The notion being that simpler morphologies would be selected for together with behaviors that optimize task performance within a robot's lifetime, however morphological complexity would not necessarily increase with task complexity (as observed in this study).

VII. CONCLUSION

This research investigated the impact of behavior (elicited by ANN controllers) and morphology (sensory configuration) evolution versus behavior only evolution in robot teams that must accomplish collective behavior tasks of varying complexity. In this study the task was *collective gathering* where varying degrees of cooperation were required in order to optimally solve the task (gather all blocks into a *gathering zone*). This study was an initial contribution to the derivation of automated morphology-behavior design methods for robot teams. The goal is for such methods, given any collective behavior task, to evolve a behavior and morphology for robots such that the task is optimally solved. Results indicated that evolving behavior and morphology in teams out-performs controller evolution for fixed morphology teams in collective gathering for varying task complexity. For the most complex collective gathering task (requiring a high degree of cooperation) morphologically simple teams were evolved. Though for a simpler version of the task (requiring medium level cooperation), a more complex controller and morphology was evolved for the fittest team. This selection for morphologically complex teams was hypothesized to be enabled by the lack of any cost associated with increasing morphological complexity [2] and as such is the subject of current research.

ACKNOWLEDGMENTS

The financial assistance of the *National Research Foundation* (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF. Computations were performed using facilities provided by ICTS High Performance Computing team: <http://hpc.uct.ac.za>.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford, England: Oxford University Press, 1998.
- [5] J. Bongard, "The utility of evolving simulated robot morphology increases with task complexity for object manipulation," *Artificial Life*, vol. 16, no. 3, pp. 201–223, 2010.
- [6] 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.
- [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] G. Buason and T. Ziemke, "Exploration of task-dependent visual morphologies in competitive co-evolutionary experiments," in *Proceedings of the seventh European Conference on Artificial Life*. Seattle, USA: Springer-Verlag, 2003, pp. 763–770.
- [9] D. Cliff, P. Husbands, and I. Harvey, "Evolving visually guided robots," in *Proceedings of the Second International Conference on Simulation of Adaptive Behaviour*. Cambridge, USA: MIT Press, 1992, pp. 374–383.
- [10] D. D'Ambrosio, J. Lehman, S. Risi, and K. Stanley, "Evolving policy geometry for scalable multiagent learning," in *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*. Richland, USA: ACM Press, 2010, pp. 731–738.
- [11] D. D'Ambrosio, J. Lehman, S. Risi, and K. Stanley, "Task switching in multiagent learning through indirect encoding," in *Proceedings of the International Conference on Intelligent Robots and Systems*. Piscataway, USA: IEEE Press, 2013.
- [12] K. Endo, F. Yamasaki, T. Maeno, and H. Kitano, "Co-evolution of morphology and controller for biped humanoid robot," in *RoboCup 2002: Robot Soccer World Cup VI*, G. Kaminka, P. Lima, and R. Rojas, Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 327–341.
- [13] B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge: Cambridge University Press, 1986.
- [14] P. Funes and J. Pollack, "Evolutionary body building: adaptive physical designs for robots," *Artificial Life*, vol. 4(4), pp. 337–357, 1998.
- [15] N. Hazon and G. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *Proceedings of the IEEE International Conference on Robotics and Automation*. Barcelona, Spain.: IEEE Press, 2005, pp. 735–741.
- [16] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*. Redwood City: Addison-Wesley, 1991.
- [17] 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.
- [18] W. L. J., Hallam, and H. Lund, "A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks," in *Proceedings of IEEE International Conference on Evolutionary Computation*. Nagoya, Japan: IEEE Press, 1996, pp. 384–389.
- [19] S. Kernbach, "Introduction to collective robotics: Reliability, flexibility, and scalability," in *Handbook of Collective Robotics - Fundamentals and Challenges*, S. Kernbach, Ed. Singapore: Pan Stanford Publishing, 2013, pp. 1–49.
- [20] R. Kube and H. Zhang, "Collective robotics: from social insects to robots," *Adaptive Behaviour*, vol. 2, no. 2, pp. 189–218, 1994.
- [21] F. Lamercy and J. Tharin, *Khepera III User Manual: Version 3.5*. Lausanne, Switzerland: K-Team Corporation, 2013.
- [22] K. Larpin, S. Pouya, J. v. Kieboom, and A. Ijspeert, "Co-evolution of morphology and control of virtual legged robots for a steering task," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. Phuket, Thailand: IEEE Press, 2011, pp. 2799–2804.
- [23] H. Lipson, "Homemade: The future of functional rapid prototyping," *IEEE Spectrum*, vol. May, pp. 24–31, 2005.
- [24] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic life forms," *Nature*, vol. 406, no. 1, pp. 974–978, 2000.
- [25] H. Lund, "Co-evolving control and morphology with lego robots," in *Morpho-functional Machines: The New Species*, F. Hara and R. Pfeifer, Eds. Tokyo, Japan: Springer-Verlag, 2003, pp. 59–79.
- [26] C. Mautner and R. Belew, "Coupling morphology and control in a simulated robot," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Orlando, USA: Kaufmann, 1999, pp. 1350–1357.
- [27] G. Nitschke, "Behavioral heterogeneity and collective construction," in *Proceedings of the IEEE Congress on Evolutionary Computation*. Brisbane, Australia: IEEE Press, 2012, pp. 387–394.
- [28] 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.
- [29] G. Nitschke and M. Tolkamp, "Approaches to dynamic team sizes," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Singapore: IEEE Press, 2013, pp. 66–73.
- [30] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, USA: MIT Press, 2000.
- [31] 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.
- [32] G. Parker and P. Nathan, "Concurrently evolving sensor morphology and control for a hexapod robot," in *Proceedings of the Congress on Evolutionary Computation*. Barcelona, Spain: IEEE Press, 2010, pp. 1–6.
- [33] L. Parker, "An architecture for fault tolerant multi-robot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.
- [34] J. Pollack, H. Lipson, P. Funes, and G. Hornby, "Three generations of coevolutionary robotics," in *Computational Intelligence: The Experts Speak*, D. Fogel and C. Robinson, Eds. Piscataway, USA: IEEE Press, 2003, pp. 1–10.
- [35] C. Schultz and L. Parker, in *Multi-robot Systems: From Swarms to Intelligent Automata*. Washington DC, USA: Kluwer Academic Publishers, 2002.
- [36] K. Sims, "Evolving 3d morphology and behavior by competition," in *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. Cambridge, USA: MIT Press, 2004, pp. 28–39.
- [37] K. Stanley, D'Ambrosio, and J. Gauci, "Hypercube-based indirect encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 2, pp. 185–212, 2009.
- [38] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.