

Sensory Configurations for Collective Behavior

Abstract—This paper presents a study on the impact of different robot sensory configurations (morphologies) in simulated robot teams that must accomplish a collective (cooperative) behavior task. The study’s objective was to investigate if effective collective behaviors could be efficiently evolved given minimal morphological complexity of individual robots in a homogenous team. A range of sensory configurations are tested in company with evolved controllers for a collective construction task. Results indicate that a minimal sensory configuration yields the highest task performance, and increasing the complexity of the sensory configuration does not yield an increased task performance.

I. INTRODUCTION

An open problem in *cooperative multi-robot systems* [1] is determining, *a priori* the most appropriate sensory-motor configuration (morphology) for individual robots, such that when an automated controller design (behavioral adaptation) method is applied, a robot team evolves a collective behavior to solve a given cooperative task.

This study falls within *Evolutionary Robotics* (ER) [2] research within the larger taxonomy of cooperative multi-robot systems [1], where *Neuro-Evolution* (NE) [3] is used to evolve controllers for a morphologically homogenous team. A popular approach in ER is to employ *Cooperative Co-Evolutionary Algorithms* (CCAs) [4], [5], [6] to co-adapt robot behaviors and morphologies. Such approaches have been successful for finding robot morphologies and controllers specifically suited to accomplishing various tasks and in a range of environments [7], [8], [9], [10], [11], [12], [13], [14], [15].

However, due to the computational complexity and intractable search spaces there has been relatively little research on the co-evolution of morphology and behavior, excluding self-assembly [16] in multi-robot [17] and swarm robotic [18] systems. An alternate approach to using CCAs in ER multi-robot systems is to systematically test a range of robot morphologies in company with controller evolution, in order to ascertain the best team morphology and behavior for a given task and environment. For morphologically homogeneous teams such an approach does not entail intractable search spaces or exponentially increasing computational complexity of increasing team sizes and task complexity. Rather, the experimenter must design a set of *morphological parameter tuning* experiments that test a sufficiently diverse yet functional range of robot morphologies. In this case some *a priori* knowledge of the task is assumed.

For the highlighted text below, should change it to “The objective of this research is to determine the robustness of controllers evolved using HyperNEAT with regards to their changing sensory configuration”

This study’s research objective was to ascertain the most appropriate morphology for a homogeneous robot team, that enables the team to efficiently evolve an effective team behavior that solves a collective behaviour task. The task was

collective construction, where cooperation was required for robots to gather and connect (build a structure) from all resources (blocks) in an environment. Each robot in the team has the same morphology, and an *Artificial Neural Network* (ANN) controller with a variable number of sensory inputs and hidden layer nodes and a fixed number of motor outputs.

Whilst the number of sensory input and hidden nodes and sensor ranges were determined by the experimenter, the ANN connection weights and inter-layer connectivity was adapted with HyperNEAT [19]. Teams were behaviorally homogeneous in that the current fittest ANN controller was copied P times for P robots in a team. Each robot also used the same morphology. HyperNEAT was selected as it has been successfully applied to evolve team behaviors for various tasks including *RoboCup* [20] and multi-agent *Pursuit-Evasion* [21]. Also, HyperNEAT is a generative encoding method, and such methods have been demonstrated as beneficial as they tend to produce regular and modular ANNs with increased learning capacities [22].

In this study, a *collective construction* task [23], which was a variation of *collective gathering* [24] was selected. The task was for robots to search for randomly distributed resources (blocks) in the environment and then move them such that they connected to other blocks in the environment, where the goal was for all blocks to be connected.

For the highlighted text below: “In this instance of the collective construction task, three different block types were used. The first can be moved by a single robot while the second and third type require two and three robots to move them respectively.” One block type required cooperation between robots to move, while another could be moved by individual robots.

For the highlighted text below: “In this case, the complexity of the basic task is increased/controlled/manipulated using a construction schema, which is a collection of underlying connection rules that specify which type of block can be connected to another block as well as on which side. (do not have a target structure to be built)”

In this case, blocks could be connected to any other in order that a structure be built. However, this is a simplified version of a more complex construction task that requires robots to collectively build a structure via connecting resources in specific ways such that a target structure is built [25].

Potential future applications of finding effective solutions to such a task include construction for exploration and mining in hazardous and remote environments [26], [27], where functional structures and habitats must be built from a set of prefabricated modules without human direction [28], [29].

This study is a preliminary step in producing NE methods that autonomously adapt multi-robot and swarm robotic system behaviors and morphologies such that problem solving collective behaviors are produced for high level user specified goals

[30]. This relates to recent research in *collective gathering* and *construction* [25] and *self-assembly* [31], where desired structures are specified by a user and robots adapt their individual behaviours in order to collectively build or self-assemble a desired structure.

A. Collective Construction Task:

This task requires a simulated robot team to gather blocks and cooperatively build a structure from gathered blocks in a bounded continuous environment (figure 1).

For the highlighted text below: "The complexity of this task is equated with the degree of cooperation (number of robots required) to collectively transport blocks and connect them together with other blocks in order to build a structure (resultant from connecting all blocks in the environment) as well as the difficulty of the implemented construction rules that determine how the blocks can be connected to each other." The complexity of this task is equated with the degree of cooperation (number of robots required) to collectively transport blocks and connect them together with other blocks in order to build a structure (resultant from connecting all blocks in the environment).

For the text below: "In this research there are three block types, A, B and C that require one, two and three robots to transport, respectively." In this research there are two block types, A and B that require one and two robots to transport, respectively.

The blocks must be connected into a structure according to a *construction schema*, that dictates the sequence for how block types must be connected.

However, for the testing purposes of this preliminary research the construction schema allows blocks to be connected together in any sequence.

For the highlighted text above: "The construction schema and the different block types are used to implement four levels of complexity with which the controllers are tested. These complexity levels consist of various combinations of block types and construction schema as follows:"

- Level 1: no complexity and no cooperation required. Each of the block types present in the simulation can be connected to any of the other block types (no complexity) and each block can be moved by a single robot (no cooperation).
- Level 2: no complexity and some cooperation. Each of the block types can be connected to every other block type (no complexity) but each type can only be moved by its respective number of robots (some cooperation).
- Level 3: some complexity and no cooperation. This means there are some relatively simple connection rules in the construction schema (some complexity) but each type of block can still be moved by a single robot.
- Level 4: complexity and cooperation required. For this level, the most difficult construction schema is implemented and each of the block types can only be moved by its respective required number of robots.

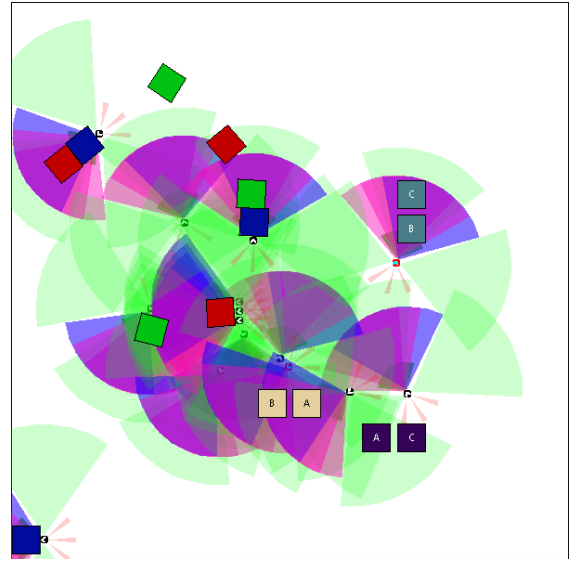


Fig. 1. Collective Construction Task Example In Simulator: Robots are the black squares with white arrows indicating the direction they are facing. The blocks are the blue, green and red squares indicating type A, B and C, respectively. The construction zones are indicated by the blocks with their respective type labels. These are the structures that have been built thus far. The translucent coloured sectors indicate the range and field of view of the sensors currently equipped to the robots. Collective Construction Task Example: Robots are the circles and blocks as the small (Type A) and large (Type B) rectangles. The green structure is that which has been built thus far via robots pushing blocks together.

It should also be noted that when two previously free blocks are connected, they form a construction zone. Other blocks in the environment can then only be connected to a construction zone.

Task performance (team fitness) is the number of blocks connected (as a built structure) during a team's lifetime (table ??).

II. METHODS

HyperNEAT [19] is an extension of the NEAT (*Neuro-Evolution of Augmented Topologies*) [32] method, where ANNs are indirectly encoded using a CPPN (*Compositional Pattern Producing Network*) [33]. In this case, HyperNEAT evolves the connection weights and the connectivity between a fixed sensory input layer, hidden layer and motor output layer. HyperNEAT was selected since geometric features in the task environment, are exploited during controller evolution. In the collective construction task, such geometric feature include the relative positions of other robots, blocks, and the direction robots and blocks are facing. HyperNEAT has also been demonstrated as being capable of exploiting regularity and modularity in multi-agent tasks in order to evolve solutions that could not otherwise be evolved [21]. In the collective construction task, HyperNEAT is potentially beneficial, given that structures to be built are modular (comprised of a set of blocks), and regular (the same sequence of blocks can be repeated).

not sure of the figure showing the ANN topology. We just created the same number of input nodes as there are sensors,

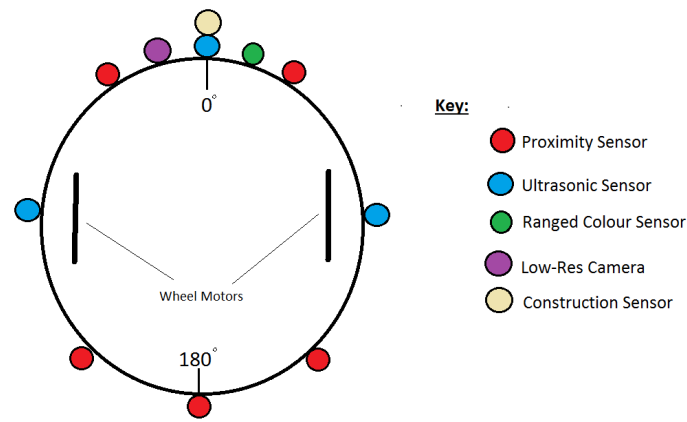


Fig. 2. Example robot sensory configuration indicating the relative positions of various sensors on the robot

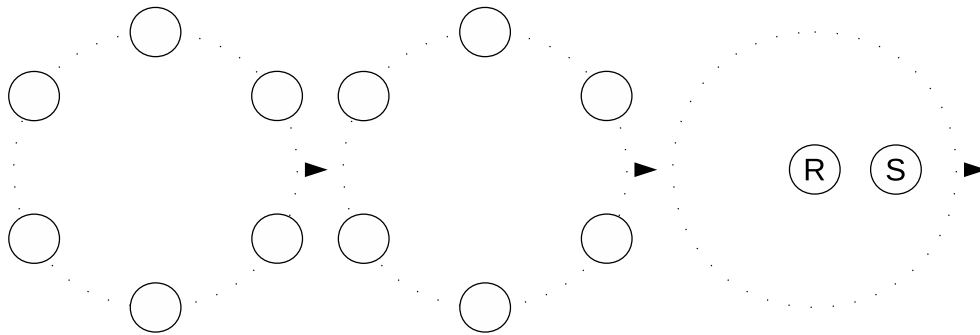


Fig. 3. ANN Topology as it relates to robot morphology: Sensory input layer (left), hidden layer (center) and motor output layer (right). Output nodes *R* and *S* determine a robot's rotation and speed, respectively. Arrows indicate the direction the agent is facing.

not sure if the geometry is matched in the way that they illustrate. Also, the outputs for our ANN's map to the left and right wheel, the output value being how fast that wheel must rotate

A. Robot Controller

All robots in a team used the same ANN controller, where each controller had N sensory input nodes (determined by a given experiment set), that adaptively mapped sensory inputs, via a hidden layer, to two motor outputs (figure 3) using HyperNEAT [19].

Figure 3 illustrates an example ANN configuration for $N = 6$. The ANN uses a 3 dimensional coordinate system for processing x, y, z positions in the CPPN in order to generate weight and bias values and connectivity. The CPPN indirect encoding of HyperNEAT allows evolved controllers to exploit the geometry of the task and the environment.

Also, HyperNEAT exploits the configuration of nodes in the ANN controller, thus the sensory input and motor output nodes must be in an appropriate configuration reflecting their position as part of a robot's morphology. Nodes for processing sensory inputs correspond to the direction each sensor faces. Thus, the input layer of the ANN controller is a circle of N evenly distributed nodes.

For the text below: The FOV of each equipped sensor is shown in the image, they do not make a complete 360 degree FOV, not sure if this is relevant Each node is a sensor,

where the sensory *Field of View* (FOV) of all sensors forms a complete 360 degree FOV (figure 2).

For the text below: The output nodes correspond to the left and right wheel actuators, producing an output $[-1.0, 1.0]$ that indicates how fast each respective wheel must turn. Equal output means straight forward motion and unequal output results in the robot rotating about its own axis.

The rotation output node is in the center to preserve the angle between sensory input nodes. The speed output node is offset in the direction the robot is facing to signify forward movement at a given speed.

while they keep mentioning that the output is mapped to a rotation and speed node, later on in the section titled Movement Actuators, they talk about how the output is mapped to the left and right wheels. Not sure if this is an inconsistency or I am misunderstanding something

The intermediate hidden layer reflects the configuration of the input layer, in order to preserve the geometry of the sensory input layer, that is the direction of each sensor's FOV (figure 3). The ANN is initialized with full connectivity between adjacent layers, however, partial connectivity can be evolved via the CPPN generating a zero weight. During the neuro-evolution process, the CPPN is developed via having nodes and connections added and removed, as well as connection weight values mutated [19]. Neuro-evolution parameters used in this study are given in table II.

TABLE I. A TABLE SHOWING THE DIFFERENT TYPES OF SENSORS USED IN THE EXPERIMENTS IN THE LEFT COLUMN AND THE RIGHT COLUMN SHOWS THE SENSORY CONFIGURATION (MORPHOLOGY NUMBER) THAT THEY ARE PART OF. THE SECOND SET OF NUMBERS IN PARENTHESES INDICATE THE NUMBER OF THAT SENSOR PRESENT IN EACH RESPECTIVE MORPHOLOGY LISTED BEFORE.

Sensor Type	Morphology Number
Proximity	1,2,4,5,6,7 (5,4,2,2,5,5)
Ultrasonic	1,2,5,6,7 (3,1,2,3,3)
Colour Ranged	1,2,3,4,5,7 (1,1,1,1,1,1)
Low-Res Camera	1,2,3,4,5 (1,1,1,1,1)
Construction	1,2,3,4,5,6,7 (1,1,1,1,1,1,1)

The subsection below called Block Detection Sensors is the original content from the paper. The subsection below that called Detection Sensors is the content that I added/changed

1) *Block Detection Sensors*: Each robot has N block detection sensors each with a range of r (portion of the environment's length). Given that N and r are the subject of the experimental comparisons of this research, they are determined by the experimenter III. Robots are unable to detect each other, thus all cooperative interactions are *stigmergic* [34], taking place via the blocks and the built structure in the environment. A robot's 360 degree sensory FOV is split into N sensor quadrants 2. Block detection sensors are constantly active for the duration of the robot's lifetime. Sensor q returns either 0 (no blocks detected) or 1 (one or more blocks detected) in sensor quadrant q . All detection sensor values are normalized within the range [0.0, 1.0].

2) *Detection Sensors*: As shown in 2, each robot is equipped with a variety of different types of abstracted sensors so as to simulate the different types of sensors that might be used in the real world. Each sensor has a specified field of view (FOV), range and bearing (its relative position on the robot) that is predetermined by the experimenter. While the range and FOV of the sensor remains constant, its bearing changes depending on the sensory configuration that the robot is equipped with. The range and FOV of each type of sensor is shown in table II.

Each robot has N sensors each with a specified range, field of view and bearing (its position on the robot). The subject of these experimental comparisons is the various combinations of these different types of sensors (referred to as sensor morphologies) that are implemented in the different experiments. These sensor morphologies are predetermined by the experimenter and are outlined later on in this table. Robots are unable to detect each other, thus all cooperative interactions are *stigmergic* [34]. All detection sensor values are normalized within the range [0.0, 1.0].

3) *Movement Actuators*: not sure about this section because the output nodes map to left and right wheels and not to rotation and speed nodes

Two motor outputs control a robot's heading and speed (R and S in figure 3) of movement. Motor output values are normalized within the range [-1.0, 1.0], where R , S = 0.0 corresponds to a default heading and being stationary, for the rotation and speed motors, respectively. Negative values of R and S correspond to anti-clockwise rotation and decreasing speed, where as, positive values correspond to clockwise rotation and increasing speed. R values of -1.0 and 1.0 equate

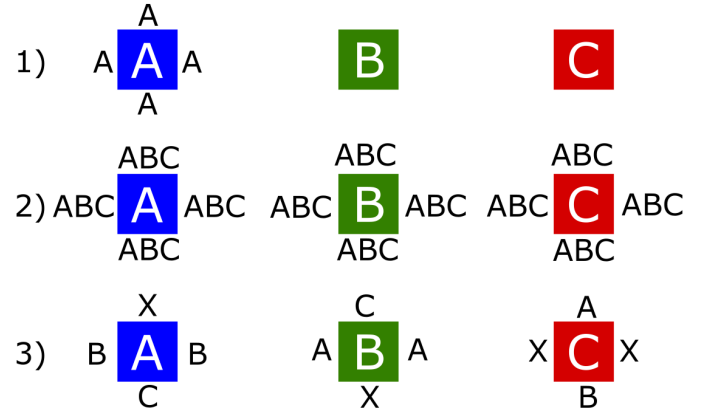


Fig. 4. An example of the various connection rules implemented in this experiment's construction schema. This is an old image, still need to create one for this years experiments

to keeping the same heading, and an S value of 1.0 equates to the maximum distance a robot can traverse in one simulation iteration (*MaxDistance* in table ??).

III. EXPERIMENTS

Experiments test n robots in a bounded two dimensional continuous environment (20 x 20 units) containing a random distribution of type A , B and C blocks (table ??). Robots are randomly initialized in the environment.

Figure 1 illustrates an example environment, containing 15 robots and 15 blocks of type A , B and C .

Figure 1 illustrates an example environment, containing 30 robots and 10 type A and B blocks.

How different block types are connected is dictated by a *construction schema* defining the sequence of block types that must be connected together in order for a structure to be built.

should possibly remove the highlighted text below. Or say: "Since the preliminary studies have been done, this investigation will implement the various connection rules that are illustrated in figure 4

However, since this is a preliminary study to demonstrate a simple collective construction task, blocks can be connected in any sequence.

The difficulty of the construction task is regulated via requiring varying degrees of cooperation to make specific block connections. In this task, cooperation refers to at least two robots simultaneously gripping and pushing a block to touch another block, to which it automatically connects.

Once two previously free blocks are connected for the first time, a construction zone is formed. A construction zone refers to any structure that has been built in the environment. Once a construction zone is created, all resources that get attached to it are fixed in position and can not be disconnected again. There are only allowed to be three construction zones at a time and free blocks can only be connected to existing construction zones.

One robot only is required to push type *A* blocks, two robots are needed to push type *B* blocks and three robots are needed to push type *C* blocks.

A. Experiment Design

For the highlighted text below: "These experiments determine the robustness of HyperNEAT evolved multi-robot controllers with regards to completing the collective construction task given a varying sensor morphology"

OR

Experiments measure the ability of HyperNEAT evolved controllers to complete the collective construction task of varying complexity given changes in its sensory configuration

Experiments measure the impact of varying robot morphology upon the evolved group behavior of a homogeneous robot team given a collective construction task and a simulation environment.

For the highlighted text below: "The collective construction task requires robots to search the environment for blocks and then push the blocks connecting them to other construction zones in the environment, in order to build a structure."

The collective construction task requires robots to search the environment for blocks and then push the blocks connecting them to others in the environment in order to build a structure.

Team task performance equals the number of blocks connected together during a team's lifetime (equation 1). An environment is defined as a distribution of block types, robots, and a construction schema.

For the highlighted text below: "To investigate the behavioural robustness of multi-robot team controllers evolved using the HyperNEAT algorithm with respect to sensory configurations when implemented in solving increasingly complex tasks requiring collective behaviour."

The research objective was to efficiently ascertain an appropriate sensory morphology (number of sensors and range), such that a homogeneous team maximizes its task performance.

For the highlighted text below: This investigation consists of testing four different sensory configurations across three levels of complexity. There is also a fifth experiment case in which the sensory configuration is randomly changed at the start of each generation.

Exploratory experiments of 12 different sensor morphologies were run to guide experiment design. These preliminary tests indicated that experiments testing a set of 6 sensory morphologies would be sufficient to satisfy the research objective. That is, a set of three sensor counts (3, 6, 10) and two different sensor ranges (20, 50) for each robot.

For the paragraph below: Each experiment tested sensor morphology for a fixed team size of 15 robots. Each experiment applied HyperNEAT to evolve team behaviour for 100 generations. A generation comprised 5 team lifetimes (simulation task scenarios). One team lifetime was 1000 simulation timesteps, representing a task scenario that tested different

TABLE II. SENSOR, EXPERIMENT AND NEURO-EVOLUTION PARAMETERS

Generations	100	
Sensors per robot	11, 8, 4, 6, random	
Evaluations per genotype	5	
Experiment runs	20	
Environment length, width	20	
MaxDistance	1	
Team size	15	
Team Lifetime (Task scenario length)	1000	
Lifetimes per generation	5	
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	
Mutation rate	Add neuron	0.25
	Add connection	0.8
	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	1.0f	0.2f
Ultrasonic Sensor	4.0f	1.22f
Ranged Colour Sensor	3.0f	1.5f
Low-Res Camera	3.0f	1.5f
Colour Proximity Sensor	3.0f	3.0f

robot starting positions, orientations, and block locations in an environment. For a given morphology, the task performance (fitness) of teams is an average calculated over 20 simulation runs, where the network with the maximum team task performance is selected from each run.

Each of six experiments tested one combination of the number and range of sensors for a fixed team size of 30 robots. Each experiment applied HyperNEAT to evolve team behaviour for 250 generations. Initially, 500 generations were tested, but 250 was found to be sufficient to observe the convergence of team behaviour for all morphologies tested. A generation comprised 3 *team lifetimes* (simulation task scenarios). One team lifetime was 120 simulation iterations, representing a task scenario that tested different robot starting positions, orientations, and block locations in an environment. For a given morphology, the task performance (fitness) of teams is an average calculated over 30 simulation runs, where the maximum team task performance is selected from each run.

For each given morphology, 20 experiment runs were conducted in which HyperNEAT was used to evolve controllers for 100 generations. The highest scoring network of all 20 runs was then re-implemented in the simulator to perform another 20 task scenarios so that an average fitness can be calculated. This is the fitness that is used in the results section below.

Experiment and neuro-evolution parameters are given in tables ?? and II, respectively. In table II, the CPPN inputs which affected the weight or bias of a given node were the x , y , z position of the connecting nodes, the difference between their

TABLE III. TWO-WAY ANOVA INVESTIGATING THE IMPACT OF SENSOR CONFIGURATION AND TASK COMPLEXITY ON TEAM TASK PERFORMANCE. **THIS IS FOR LEVEL 1**

	Df	SS	MS	F	p
Complexity	2	27.88	13.94	1187.36	1.6×10^{-181}
Morphology	7	3.68	0.53	44.8	3.9×10^{-48}
Complexity:Morphology	14	2.28	0.16	13.86	1.29×10^{-27}
Residuals	456	39.19	0.12		

positions *delta*, and the angle between them. These parameter values were determined experimentally. Minor value changes produced similar results for all morphologies. Except those parameters given in table II, other neuro-evolution parameters were set to values previously used for HyperNEAT [21].

(Should be mentioned somewhere?) The randomised morphology was implemented such that each generation starts off with the same morphology as the first experiment but then various random sensors in the configuration are deactivated, resulting in an all new morphology.

For the text below: The fitness function (ref the equation) used in team (controller) evaluation was a weighted sum of the number of each type of block that was successfully connected to a built structure

cant highlight the text below because of the many latex commands (below is the original)

The fitness function (equation 1) used in team (controller) evaluation was a weighted sum that included, the number of times a robot successfully found blocks (*a* in equation 1), the number of times *type A* blocks were pushed by *one robot* and connected with a built structure, and the number of times *type B* blocks were pushed by *two robots* and connected with a built structure (*b* in equation 1).

this is the new content 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, respectively, resulted in functional controller evolution

this is the original Parameter tuning experiments found that setting the weights (reward values r_a and r_b in equation 1) both to 1.0 resulted in functional controller evolution.

this is the new content Fitness was normalized to the range [0.0, 1.0] using the total number of each type of block present in the task scenario.

this is the original Fitness was normalized to the range [0.0, 1.0] using the number of blocks and agents required to move a given block i (s_i).

$$f = \frac{r_a a + r_b b + r_c c}{r_a A + r_b B + r_c C} \quad (1)$$

IV. RESULTS & DISCUSSION

For a given experiment, the highest team task performance (fitness) achieved during an evolutionary run was recorded. The network that produced the highest fitness of all the evolutionary runs for each morphology is then re-implemented in

TABLE IV. TWO-WAY ANOVA INVESTIGATING THE IMPACT OF SENSOR CONFIGURATION AND TASK COMPLEXITY ON TEAM TASK PERFORMANCE. **THIS IS FOR LEVEL 2**

	Df	SS	MS	F	p
Complexity	2	17.81	8.92	649.21	3.8×10^{-134}
Morphology	7	0.19	0.03	1.98	0.06
Complexity:Morphology	14	0.15	0.01	0.79	0.68
Residuals	456	6.26	0.01		

TABLE V. TWO-WAY ANOVA INVESTIGATING THE IMPACT OF SENSOR CONFIGURATION AND TASK COMPLEXITY ON TEAM TASK PERFORMANCE. **THIS IS FOR LEVEL 3**

	Df	SS	MS	F	p
Complexity	2	24.67	12.34	920.91	7.3×10^{-161}
Morphology	7	0.25	0.04	2.69	0.01
Complexity:Morphology	14	1.49	0.11	7.94	3.41×10^{-15}
Residuals	456	6.11	0.01		

the simulator for 20 task trials and an average fitness is then calculate over those 20 runs.

For a given experiment, the highest team task performance (fitness) achieved during an evolutionary run was recorded. An average of these maximum fitness values was then calculated over the 30 runs performed for each experiment.

For the text below: Figure ?? presents the average fitness achieved, by teams using each of the four team morphologies (table ??) in terms of the box plots.

This is the original text: Figure ?? presents the average fitness achieved, by teams using each of the six team morphologies (section III-A) in terms of box plots.

Figures ?? also illustrates fitness means, variances and outliers.

below is the original text Figure ?? indicates that teams using only three sensors yield a significantly higher task performance, compared to the other morphologies tested. This difference was found to be statistically significant using a two-way analysis of variance (ANOVA) [35] (factors were *sensor range* and *sensor count*). ANOVA confirmed that the number of sensors has a significant impact on maximum fitness ($F = 31.737$, $p = 1.78 \times 10^{-12}$). However, the impact of sensor range was not found to be significant ($F = 0.261$, $p = 0.610$), and no interaction effect was found ($F = 1.070$, $p = 0.345$). The results of these statistical tests are summarized in table ??.

This result implies that for homogenous teams that must accomplish a collective behavior task, then simpler morphologies (in this case, sensory configurations), leads to significantly higher team task performance. This is theorized to be the result of the demonstrated benefits of HyperNEAT applied to controller evolution, such as its capability to exploit modularity and geometric regularities in the task environment [21], [19], coupled with the lower dimensionality of the search space for ANNs with only three sensory nodes. Thus, in this case, the modular and regular nature of the task, requiring blocks to be connected together in a repeated fashion and the lower dimension search space facilitated a rapid convergence upon an effective collective behavior solution by HyperNEAT.

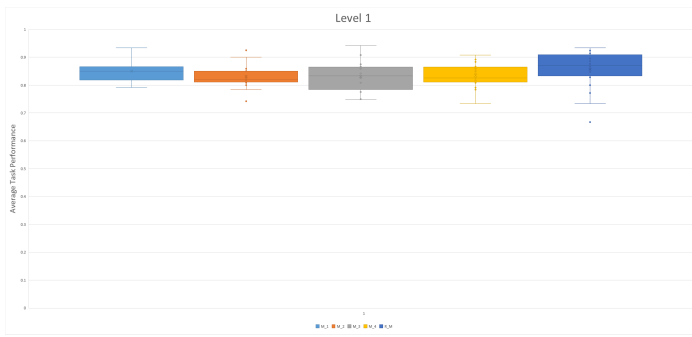


Fig. 5. Average task performance of each controller re-evaluated using the morphologies outlined in table I

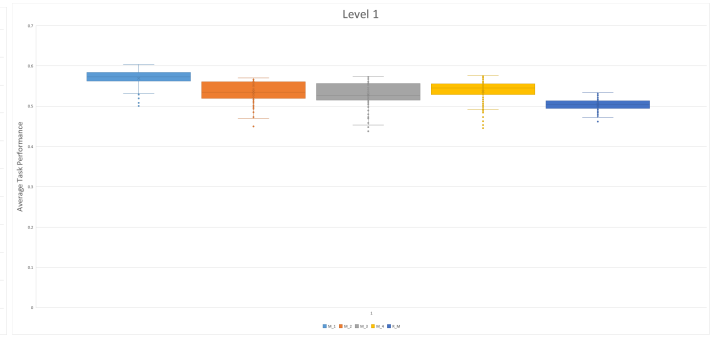


Fig. 6. Progression of best team task performance over the course of controller evolution, averaged over 20 runs

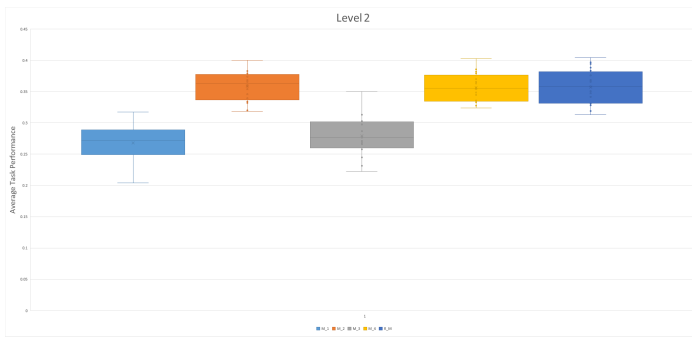


Fig. 7. Average task performance of each controller re-evaluated using the morphologies outlined in table I

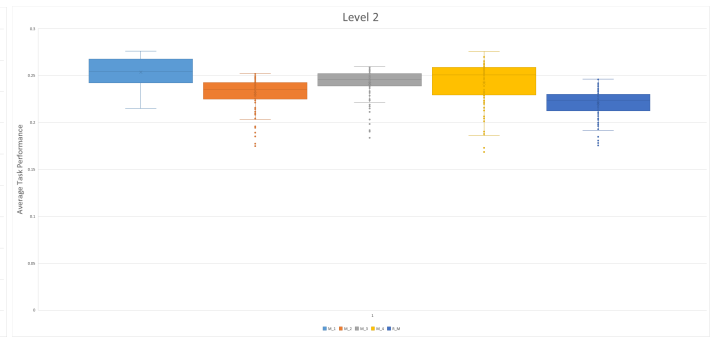


Fig. 8. Progression of best team task performance over the course of controller evolution, averaged over 20 runs

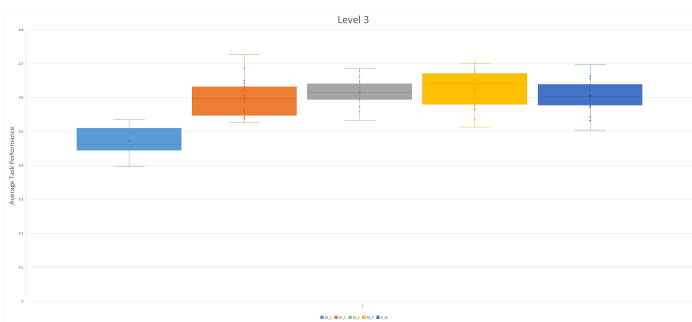


Fig. 9. Average task performance of each controller re-evaluated using the morphologies outlined in table I

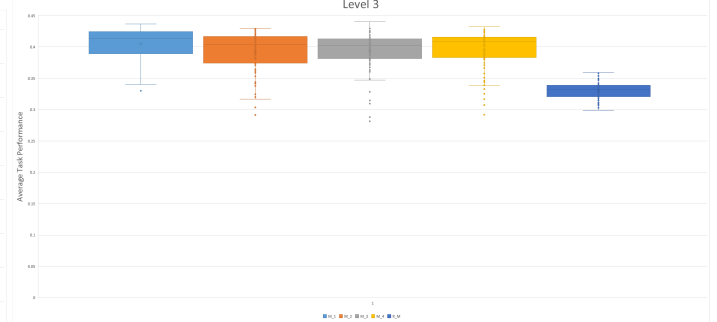


Fig. 10. Progression of best team task performance over the course of controller evolution, averaged over 20 runs

However, attaining these results depended upon morphological design choices by the experimenter. A robot's sensory configuration, that is, the input neurons of the ANN, were set relative to a robot's rotation, rather than being absolute in the environment. This design decision had two main motivations. First, it allowed evolved behaviours to be more robust in that it assumed the robot has no information about its orientation with respect to global markers in the environment. Second, a robot's heading is not aligned with a specific sensor, rather it was exactly between two sensory FOVs (figure ??). This allowed a robot to more accurately move towards blocks with a feedback-control loop of minor adjustments to rotation. This effectively resulted in quick direct movement towards detected blocks, thus reducing the time needed to move detected blocks, and increasing average team task performance.

Also, the geometric relationship between the sensory inputs

and the rotation and speed motors (figure 3), benefited from relatively few sensors, given that less sensory information had to be processed by the ANN. This relationship could be most effectively exploited by HyperNEAT for low dimensional search spaces, that is, a low number of sensors.

Figures ??, ?? and ?? present the progression of maximum team fitness over the course of the controller evolution for each complexity level respectively. The values were averaged over 20 runs

Figure 11 presents the progression of maximum team fitness over the course of controller evolution, averaged over 30 runs. Whilst figure ?? indicated that teams using three sensors achieved the highest overall fitness, figure 11 indicates that team fitness, for this lower number of sensors, after a steep increase during the first 50 generations, remains relatively

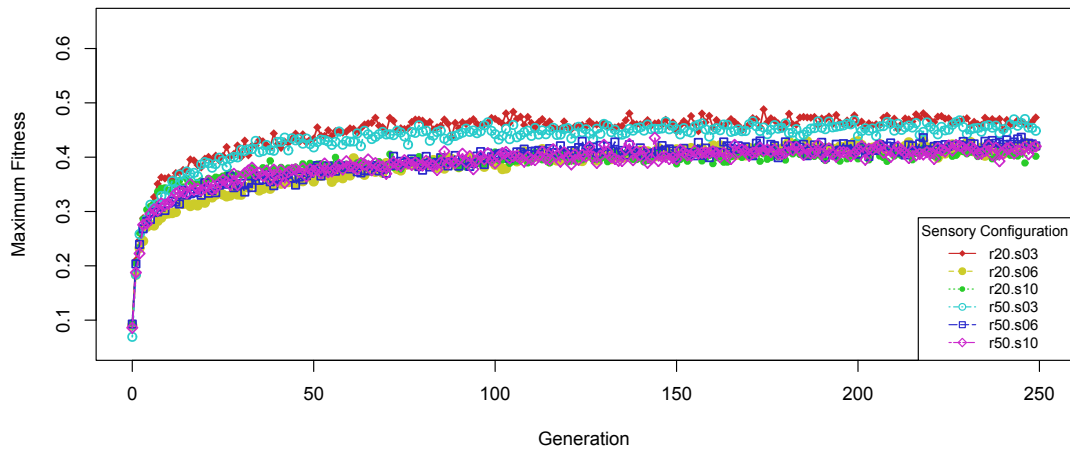


Fig. 11. Progression of best team task performance over the course of controller evolution, averaged over 30 runs. Red and blue curves at the top correspond to teams using three sensors, and yield a higher average task performance (with statistical significance), compared to the other sensory configurations tested.

constant throughput the rest of the evolutionary run. This implies that if controller evolution can only be run for relatively few generations, then selecting an appropriate sensory configuration yields significantly beneficial task performance advantages.

These results are supported by related research [36] that similarly demonstrates a minimal number of sensors and an appropriate sensory configuration leads to desired task performance, and increasing the complexity of the sensory configuration does not necessarily result in increased task performance.

This study's research objective was to find a minimal morphology (sensory configuration) such that a morphologically and behaviorally homogenous team evolve effective collective construction behaviors. The results are relevant to the larger fields of ER [2] and swarm robotics [18] that aim to minimize cost, weight and power requirements of robots. Another goal of these research fields is to synthesize problem solving collective behaviors that can be efficiently evolved either *a priori* in simulation and then transferred to counterpart physical robots [37], [38], [39] or evolved in real time as the robots interact with their environment [40], [41]. This research aims to address future applications of the former approach.

This study deviated from related research that co-evolves both morphology and behavior. Such approaches let a given artificial evolution process search a space of defined behaviors and morphologies in order to derive a sensory-motor configuration and coupled controller that is suited to solving a given task in a given environment [7], [42], [43], [8], [9], [44], [10], [45], [13], [46], [47], [48], [14], [15].

In this study, the goal was to determine the ability of HyperNEAT evolved controllers to complete the collective construction task across a variety of different sensory configurations

In this study the goal was for the experimenter to design and test a range of morphologies and then have a neuro-evolution process evolve an appropriate ANN controller. The

motivation for this approach was the desire to find a minimal sensory configuration such that solutions to a collective behavior task are efficiently evolved. This falls in line with ER and swarm robotic research objectives of designing swarm robotic systems able to accomplish collective behavior tasks with minimal cost, weight (number sensors) and power requirements, where the problem solving collective behaviors of such robotic systems can be efficiently evolved in minimal time.

In related research that co-evolves behavior and morphology, especially that dealing with robot teams, there is a significant computational and time expense involved. That is, invariably large behavioral and morphological spaces must be searched and a huge number of possible behavior-morphology interactions that must be tested and evaluated. This research demonstrated that in collective behavior tasks where experimenters have some *a priori* knowledge from previous experiments [49] to guide morphological design, then a limited set of exploratory experiments that test a range of morphologies is sufficient (section III-A). Furthermore, methods that co-evolve behavior and morphology must often place significant constraints on the types of behaviors and morphologies that can be evolved in order to reduce the size of the search spaces and the number of behavior-morphology combinations that must be tested and evaluated. In such a case, morphological design by the experimenter, and then evolving controllers for these morphologies, is often just as effective as the behavior-morphology co-evolution approaches.

In this research the collective construction task was more akin to widely studied collective gathering tasks [24] that require agents (robots) to search an environment for resources, and then transport these gathered resources to a home area. In this case there was no specific home area, rather the first gathered block could be connected to any other, and then this initial structure became the focal point where all gathered blocks were transported to. Also, the task was made more complex via requiring cooperation (two robots) to move half of the blocks. Thus cooperative behavior and division of labor was mandated for the team to achieve optimal task

performance. That is, if all blocks were to be gathered, teams had to effectively split into those that gathered *type A* blocks (one robot only) and those that gathered *type B* blocks (two robots needed). Experimental results achieved on average a maximum of approximately 60 percent of optimal team task performance (figure ??). This was a result of limiting the team lifetime duration (table ??) to lower computational expense, and the lack of sensors to detect fellow robots. Both were in the interest of satisfying the research objective, and having purely stigmergic interactions between robots [34].

The inclusion of sensors for discriminating between team mates and resources in the environment has been demonstrated as beneficial in collective gathering and construction tasks [49] and will thus potentially form a part of future work that aims to test the impact of various morphologies (sensory configurations) as collective behavior task complexity increases. Specifically, future work will focus on testing a range of morphologically and behaviorally homogenous and heterogenous teams for varying degrees of complexity of a collective gathering and construction task. Increasing complexity in such a task entails having more block types, where each type requires a different degree of cooperation between robots to move, as well as testing a range of construction schemas. Construction schemas dictate the sequence and number of each block type that must be connected together in order for a structure to be built. Task difficulty can be regulated via the construction schema requiring that relatively scarce block types be the first building blocks used in construction [49].

V. CONCLUSIONS

This research presented a study on the impact of different robot sensory configurations (morphologies) on evolving team behaviors in a robot team that had to accomplish a collective construction task. The collective construction task required cooperation in order for the team to optimally accomplish it. The team was behaviorally and morphological homogenous where the morphology was determined *a priori* by the experimenter and an accompanying ANN controller was evolved. The research objective was to investigate what degree of morphological complexity would be amenable to the efficient evolution of effective collective behaviors. This objective addresses future ER goals of designing robot swarms that have minimal cost, power and weight requirements, where problem solving collective behaviors must be efficiently evolved. Results indicate that minimal sensory configurations yield the highest task performance, and that effective collective behaviors can be more efficiently evolved for such minimal morphologies. In this collective construction task, increasing morphological complexity did not result in increased task performance over minimal morphology-behavior couplings.

REFERENCES

- [1] A. Farinelli, R. Farinelli, L. Iocchi, and D. Nardi, "Multi-robot systems: A classification focused on coordination," *IEEE Transactions on Systems Man and Cybernetics B*, vol. 34, pp. 2015–2028, 2004.
- [2] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, USA: MIT Press, 2000.
- [3] D. Floreano, P. Dürri, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008.
- [4] F. Gomez and R. Miikkulainen, "Solving non-markovian control tasks with neuroevolution," in *Proceedings of the International Joint Conference on Artificial Intelligence*. Stockholm, Sweden: Morgan Kaufmann, 1999, pp. 1356–1361.
- [5] M. Potter and K. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.
- [6] R. Wiegand, W. Liles, and K. De Jong, "An empirical analysis of collaboration methods in cooperative coevolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, USA: AAAI Press, 2001, pp. 1235–1242.
- [7] 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.
- [8] C. Mautner and R. Belew, "Evolving robot morphology and control," *Artificial Life and Robotics*, vol. 4(3), pp. 130–136, 2000.
- [9] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic life forms," *Nature*, vol. 406, no. 1, pp. 974–978, 2000.
- [10] G. Hornby and J. Pollack, "Evolving l-systems to generate virtual creatures," *Computers and Graphics*, vol. 25(6), pp. 1041–1048, 2001.
- [11] —, "Creating high-level components with a generative representation for body-brain evolution," *Artificial Life*, vol. 8(3), pp. 1–10, 2002.
- [12] 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.
- [13] J. Bongard and R. Pfeifer, "Evolving complete agents using artificial ontogeny," in *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, H. Fumio and R. Pfeifer, Eds. Berlin, Germany: Springer, 2003, pp. 237–258.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] P. Verbancsics and K. O. Stanley, "Evolving static representations for task transfer," vol. 11, pp. 1737–1769.
- [21] 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.
- [22] P. Tonelli and J. Mouret, "On the relationships between generative encodings, regularity, and learning abilities when evolving plastic artificial neural networks," *PLoS ONEs*, vol. 8(11), p. e79138. doi:10.1371/journal.pone.0079138, 2013.
- [23] J. Werfel, "Building blocks for multi-robot construction," in *Distributed Autonomous Robotic Systems 6*. Tokyo, Japan: Springer, 2007, pp. 285–294.
- [24] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford, England: Oxford University Press, 1998.
- [25] 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.
- [26] R. Brooks, P. Maes, M. Mataric, and G. More, “Lunar base construction robots,” in *Proceedings of Intelligent Robots and Systems*. Tokyo, Japan: IEEE Press, 1990, pp. 389–392.
- [27] W. Shen, P. Will, and B. Khoshnevis, “Self-assembly in space via self-reconfigurable robots,” in *IEEE International Conference on Robotics and Automation*. Taipei, Taiwan: IEEE Press, 2003, pp. 2516–2522.
- [28] J. Werfel, Y. Bar-Yam, and R. Nagpal, “Building patterned structures with robot swarms,” in *International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland, United Kingdom: AAAI Press, 2005, pp. 1495–1502.
- [29] 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.
- [30] C. Jones and M. Mataric, “From local to global behavior in intelligent self-assembly,” in *Proceedings of IEEE International Conference on Robotics and Automation*. Taipei, Taiwan: IEEE Press, 2003, pp. 721–727.
- [31] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable self-assembly in a thousand-robot swarm,” *Science*, vol. 345(6198), pp. 795–799, 2014.
- [32] K. Stanley, *Efficient Evolution of Neural Networks Through Complexification*. Ph. D. Dissertation. Austin, USA: Department of Computer Sciences, The University of Texas, 2004.
- [33] —, “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.
- [34] 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.
- [35] B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge, UK: Cambridge University Press, 1986.
- [36] A. Schultz and M. Bugajska, “Co-evolution of form and function in the design of autonomous agents: Micro air vehicles project,” in *Proceedings of the Workshop on Evolution of Sensors in Nature, Hardware, and Simulation*. Chicago, USA: AAAI Press, 2000, pp. 154–166.
- [37] S. Koos, J. Mouret, and S. Doncieux, “Crossing the reality gap in evolutionary robotics by promoting transferable controllers,” in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. Portland, USA: ACM, 2010, pp. 119–126.
- [38] J. Zagal and J. Ruiz-Del-Solar, “Combining simulation and reality in evolutionary robotics,” *Journal of Intelligent and Robotic Systems*, vol. 50, no. 1, pp. 19–39, 2007.
- [39] C. Hartland and N. Bredeche, “Evolutionary robotics, anticipation and the reality gap,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. IEEE Press, 2006, pp. 1640–1645.
- [40] G. Baele, N. Bredeche, E. Haasdijk, S. Maere, N. Michiels, Y. Van de Peer, T. Schmickl, C. Schwarzer, and R. Thenius, “Open-ended on-board evolutionary robotics for robot swarms,” in *Proceedings of the IEEE Congress on Evolutionary Computation*. Trondheim, Norway: IEEE Press, 2009, pp. 1123–1130.
- [41] A. Eiben, G. Karafotias, and E. Haasdijk, “Self-adaptive mutation in on-line, on-board evolutionary robotics,” in *Self-Adaptive and Self-Organizing Systems Workshop*. Budapest, Hungary: IEEE Press, 2010, pp. 147–152.
- [42] H. Lund and J. Lee, “Evolving robot morphology,” in *Proceedings of the IEEE International Conference on Evolutionary Computation*. Indianapolis, USA: IEEE Press, 1994, pp. 197–202.
- [43] A. Adamatzky, M. Komosinski, and S. Ulatowski, “Software review: Framsticks. cybernetes,” *The International Journal of Systems and Cybernetics*, vol. 29(9), pp. 1344–1351, 2000.
- [44] G. Hornby and J. Pollack, “Body-brain co-evolution using l-systems as a generative encoding,” in *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, USA: IEEE Press, 2001, pp. 868–875.
- [45] P. Eggenberger, “Evolving morphologies of simulated 3d organisms based on differential gene expression,” in *Proceedings of the Fourth European Conference on Artificial Life*. MIT Press: Brighton, United Kingdom, 1997, pp. 205–213.
- [46] J. Auerbach and J. Bongard, “Dynamic resolution in the co-evolution of morphology and control,” in *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems*. Odense, Denmark: MIT Press, 2010, pp. 451–459.
- [47] 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.
- [48] 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.
- [49] 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.