UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2020

Title: The Impact of a Complexity Cost on the Complexity of Robots Evolved Using an Objective Fitness Function

Author: Christina Spanellis (SPNCHR005)

Project Abbreviation: EVOBAB

Supervisor(s): Geoff Nitschke

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	5
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	

The Impact of a Complexity Cost on the Complexity of Robots Evolved Using an Objective Fitness Function

Christina Spanellis
University of Cape Town
SPNCHR005@myuct.ac.za

ABSTRACT

An open ended question in natural evolution is how and under which environmental conditions complexity evolves. Evolutionary robotics has been used to study evolving complexity in simulated robots. This paper aims to investigate the environmental conditions under which artificial robot body-brain complexity increases. This research will evaluate correlations between environmental complexity and robot complexity through the co-evolution of simulated robots for the task of locomotion in environments of increasing complexity, both with and without an imposed cost on complexity. Results produced by this study suggest that the imposition of a complexity cost evolves less complex robots across environments of both low and high complexity versus robots evolved without a cost.

1 INTRODUCTION

Evolutionary Robotics (ER) is an experimental sub field of Robotics that removes the manual process in the design of intelligent robots through the application of machine learning techniques and principles of biological evolution. These simulated robots develop control systems (controllers) and body configurations (morphologies), either independently or in tandem, to perform tasks of varying complexity without human supervision [5, 28]. A long-standing question in biological and artificial evolution is how, and under which environmental conditions complexity evolves [33]. Bedau's arrow of complexity hypothesis states that evolutionary organisms tend to increase in complexity over time [19]. Such a hypothesis is not easily proven in natural biology where evolution takes place over millions of years, but ER as a field has contributed to answering such research questions by simulating the evolution of biological organisms in artificial environments [3, 20, 21]. Understanding the conditions under which complexity evolves could aid in the development of low complexity, high performing robots. These robots could be transferred to real-world scenarios which may be impossible or dangerous for humans.

Most work in ER investigating the evolution of complexity uses Evolutionary Algorithms (EA) that evolve either robot bodies or brains in isolation. In 1994 Karl Sims published his work on successful co-evolution (simultaneous evolution of robot bodies and brains) which paved the way for modern ER techniques [25]. In his paper, artificial robots were represented by two directed graphs, the first representing the robot's controller and the second representing the morphology. The evolved robots had diverse brain-body couplings and varying behaviour patterns, importantly this study produced robots that would likely not have been thought of or developed by a human designer. Sim's evolution of diverse, sufficiently performing robots has since inspired further applications and approaches to co-evolution. However, there are few studies that have investigated the evolution of complexity in co-evolved robots. Furthermore, the sensitivity of EAs to parameters such as fitness function, robot task objective and complexity metric has lead to conflicting results in published work about the conditions under which complexity evolves.

This project aims to produce new results pertaining to the correlations between environmental conditions and the evolution of complexity. The hypothesis statement for this project is *given conducive environmental conditions, complexity increases even with imposed complexity costs.* In support of this hypothesis, the primary aims of this project are:

(1) To investigate whether system complexity (both brain and body) increases despite an imposed complexity cost (defined in section 3.3), given conducive environmental conditions.
(2) Evaluate whether the imposition of a complexity cost enables the evolution lower complexity, equally performing robots versus those evolved without a cost, where the performance of a robot is gauged by the distance moved from the starting point in an environment (task performance evaluation is discussed in section 3.5).

Previous work investigating the correlations between environmental complexity and evolved robot complexity has shown mixed results. A 2019 paper by Nagar et al. found that simple morphologies were evolved in both simple and complex environments when a cost on complexity was imposed [21]. However, the robots evolved in simple environments were in fact less complex than those evolved in complex environments. Auerbach and Bongard found more morphologically complex robots were evolved in more complex environments [3]. Other studies have provided evidence for the evolution of more complex robots in more complex environments [20, 23]. The impact that an imposed complexity cost has on the evolution of complexity is still unclear, as demonstrated by the conflicting results found by Auerbach and Bongard and Nagar et al [3, 21]. Thus, results found by this project in support of, or against, the research hypothesis will contribute to previous findings on the impact of a complexity cost on the evolution of complexity.

The hypothesis will be evaluated through the co-evolution of simulated robots in twelve different environments of increasing complexity. These robots will be evolved for the task of locomotion and evaluated using a performance based fitness function, this fitness function assigns a numeric fitness score to a robot based on the distance reached in an environment. Evolving robots for the task of locomotion is a well established bench-mark task for testing brain-body evolution in ER systems [3, 16]. Furthermore locomotion as a task is easy to understand and defining an appropriate fitness function for locomotion is straight-forward. The complexity of individuals will be evaluated using a novel combined complexity metric that accounts for both controller and morphology complexity. To

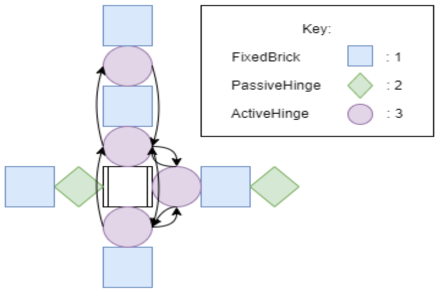


Figure 1: Example robot morphology tree where differently shaped and coloured nodes represent different robot body part types. The key shows the corresponding morphological complexity value of each body part type.

accurately evaluate the hypothesis, experiments will be run with and without an imposed complexity cost. The complexity cost penalises robots that become increasingly complex by reducing their lifetime thus giving them less chance to exhibit strong behaviour patterns and decreasing the chance for more complex robots to survive in an evolution.

Most previous work in ER investigating the evolution of complexity applies complexity metrics that account for either controller complexity or morphology complexity, but rarely both [3, 20, 21]. Complexity of the controller and morphology are ultimately bounds for the types of behaviours exhibited by simulated robots since the body and brain are so tightly coupled. Accurate and meaningful complexity metrics play a large role in the balancing of complexity and performance of simulated robots. To this end, this project also aims to produce an accurate complexity metric representative of overall system complexity. The novelty of this project lies with the combined complexity metric that accounts for both morphologies and controllers, as well as the use of a co-evolutionary Genetic Algorithm to evolve high performing, low complexity robots.

2 RELATED WORK

This section will acknowledge studies and methods within ER that have have inspired the research questions and methods used in this paper. Section 2.1 introduces techniques of co-evolution used in ER and outlines the inspiration for the body-brain representations used in this study. Section 2.2 outlines previous work by Tononoi and Sporns that provided the logic behind the neural portion of the complexity metric employed in this experiment [29]. Section 2.3 highlights previous studies in ER that have investigated the impact of an complexity cost on both neural and morphological complexity.

2.1 Co-evolution using HyperNEAT

Traditionally, most evolutionary studies in ER employ a learning algorithm that either evolves robot morphologies or robot controllers in isolation. Co-evolutionary algorithms are far more complex to implement as the brain and body representations need to fit together appropriately, and the crossover operations to produce offspring become more complicated as there are more components to consider. A further issue in the process of co-evolution is that the performance of controller-morphology couplings learnt by an algorithm can be completely offset by tiny mutations in morphologies. A 2019 paper addresses this problem through use of two evolutionary processes: one adapting the morphologies and another adapting controllers [16]. This employs the concept of lifetime learning, whereby morphologies can evolve but there is *lifetime learning* for controllers. Once a new morphology is introduced, a second evolutionary processes begins on a different timescale - evolving the controller for that specific morphology. In this method Lamarckian evolution was used, meaning individual improvements are directly encoded in genomes (candidate solutions) to be used to accelerate evolution [16]. Here the process of morphological evolution is straightforward, where the morphology is represented by a graph and each node on the graph is a body part. Parent crossover was simply implemented as a random recombination of parent morphology trees. The evolution of controllers used Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) (discussed in section 3.2.2) to set parameters in controller substrate nodes, which control movement of the robot [27]. Robogen open source framework inspired the design of their morphology and controller representations [1]. This paper takes inspiration from Jelisavcic et al.'s work, specifically focused on the body-brain representations and use of HyperNEAT for the co-evolution of robot bodies and brains [16].

2.2 Neural Complexity Metric

Tononi and Sporns presented a method of measuring neural complexity by measuring information integration [29]. Here information integration is defined as the amount of effective information that can be transferred between subsystems of a system, where effective information is the amount of information shared between two sources [29, 31]. This research suggested that networks having high information integration should be highly connected and should have highly specialised connections, whose functional roles are separate [29]. The neural portion of the complexity metric proposed in this study takes inspiration from this paper, aiming to maximise the neural complexity measure when there is a balance between local specialisation and global integration. Local specialisation is representative of learnt behaviour in specific regions of the body-brain configuration, for example a robot's knee and upper leg working in sync to achieve locomotion. Other neural metrics may measure the complexity of a neural network through the connection weight information, or the number of connections. However, this approach to measuring neural complexity is not necessarily representative of meaningful behaviour. For this reason, the measures by Tononi and Sporns are more appropriate in the context of this project, where performance and behaviour of robots is critical.

2.3 Complexity Costs

The majority of previous work in ER investigating the evolution of complexity has focused on measuring either neural or morphological complexity in isoloation of one another. Work by Nagar et al. found that simpler robot morphologies were evolved when a cost on complexity was imposed on evolution [21]. This cost was imposed through a multi-objective evolutionary algorithm that selected for robots with low morphological complexity and high

	Environment	Friction	Obstacles	Tilted
Set 1	1	1.0	None	No
	2	0.6	None	No
	3	0.2	None	No
Set 2	4	1.0	Regular	No
	5	0.6	Regular	No
	6	0.2	Regular	No
Set 3	7	1.0	Irregular	No
	8	0.6	Irregular	No
	9	0.2	Irregular	No
Set 4	10	1.0	None	Yes
	11	0.6	None	Yes
	12	0.2	None	Yes

Table 1: The environment parameters per set. Each environment per set had differing friction but utilised the same obstacle set or a tilted floor. *Regular* obstacle sets contain obstacles that are regularly spaced from one another whilst *irregular* sets contain obstacles that are irregularly spaced. A *Tilted* value of *Yes* indicates that the environment floor was tilted at an incline.

fitness scores. Similar results were found by Revello and McCartney when evolving control programs [23]. The programs evolved with a dynamically scaled complexity cost proved to be more efficient and less complex than those evolved without a complexity cost. A recent paper by Hallauer and Nitschke investigated the impact of environmental complexity on robot body-brain complexity with an imposed *energy cost* on evolution [12]. This energy cost was imposed on the morphology and was equivalent to the battery cost of running the robot whilst the neural cost was imposed based on neural efficiency. This study takes inspiration from the work of Hallauer and Nitschke to impose an *energy cost* on robot evolution, where this cost is directly proportional to robot complexity (this cost is discussed further in section 3.3).

3 METHODS

This section outlines the methods used in the evolutionary learning algorithm of this study, this includes evolutionary brain-body evolution and the imposition of a complexity cost with the novel complexity metric presented in section 3.4.

3.1 Evolutionary Robot Design

The successful co-evolution of artificial robots is reliant on appropriate representations of robot morphologies and controllers that can be manipulated by the evolutionary learning algorithm. The representations used in these experiments are based on Jelisavcic et al.'s work on co-evolving simulated robots for locomotion [16]. This project will use the Robogen open source framework, discussed in section 4.2 of this paper, which provides the evolution simulation engine[1]. The source code for the evolutionary algorithm used in this paper can be found on Github[2].

[1]http://robogen.org/
[2]https://github.com/BrookeSte/EVOBAB

Experiment Parameters	Value
Population size	100
Number of generations	100
Number of children per generation	100
DTS tournament size	2
Maximum body parts	50
pNodeInsert	0.3
pSubtreeRemove	0.3
pSubtreeDuplicate	0.3
pSubtreeSwap	0.3
pNodeRemove	0.3

Table 2: Experiment parameters used by the evolutionary algorithm, where parameters prefaced with *p* represent morphological mutation probabilities and DTS is Deterministic Tournament Selection (see section 3.2).

3.1.1 Morphologies. Robot morphologies were represented as simple tree structures where the root node of the tree is the core of the robot, and each node can have up to four children representing different body parts. Figure 1 shows a simple example morphology tree of a robot. Each sub tree attached to the root node is constructed from a number of different node types and represents a limb of the robot's body. Robogen provides access to six different node (body part) types but for the purpose of this experiment the node types were restricted to *FixedBrick*, *PassiveHinge* and *ActiveHinge*[3]. The morphology trees were constrained to have a maximum of fifty nodes, this allowed for the scaling of complexity values of the robot morphologies. For the purpose of this experiment, each node type was given an inherent complexity value based on the ability of the node type to contribute to movement. The complexity values given to each node type are shown in Figure 1.

3.1.2 Controllers. Robot controllers were represented by Artificial Neural Networks (ANNs) whereby the neurons of this network are distributed throughout the robot's morphology tree. ANNs have been successfully used for developments in Neuroscience and research into information processing in natural systems and are well established for use in robotic control [11, 14, 16–18]. Each body part in the robot's morphology may have multiple neurons attached to it. Each artificial neuron can be a standard input-output neuron or can be an oscillator with an associated period, phase offset and amplitude. Each robot was initialised with an associated Compositional Pattern Producing Network (CPPN) that is evolved during its lifetime.

CPPNs are an indirect encoding technique used to encapsulate the complexities of neural networks by encoding a function which is used to approximate connection weight values in the controller network on the robot. Similar to Jelsavcic et al [16], the CPPNs are evolved using the HyperNEAT learning algorithm. CPPNs are an abstraction of natural development, aiming to capture the enormous complexities encoded in natural DNA without including unnecessary inefficiencies. They are similar to neural networks in that

[3]http://robogen.org/docs/robot-body-parts/

they are networks of functions, but can contain multiple different activation functions.

The connection weights between neurons in the morphology are encoded by the evolved CPPN. Thus, the ANN represents the phenotype of the robot whilst the CPPN represents the robots genotype. In natural biology, a genotype (or genome) stores genetic information about an individual whilst a phenotype describes the physical appearance or behaviour of an individual. To retrieve the connection weight between any two neurons in the ANN, the coordinates of the corresponding morphology tree node in which the neuron is embedded (relative to the core component of the robot) are used as query parameters for the CPPN. Figure 1 shows an example robot morphology, as well as the network of neurons embedded inside different nodes. The arrows represent possible connections between neurons, the value of which is encoded by the robot's corresponding CPPN. Since the robot's neural network is rooted in the morphology tree it is not transferable between robots, instead the CPPN can be mated and transferred between different robots as it encapsulates the learning of the controller. Each "joint" node, which could be an active or passive hinge, may have an attached oscillator. The range of motion of each joint with an embedded oscillator is specified by the amplitude of the oscillator, a value between 0 and 1.

3.2 Co-Evolution Methods

Co-evolution is the simultaneous evolution of robot bodies and brains. The co-evolutionary genetic algorithm designed for this experiment used HyperNEAT indirect encoding for evolution of controllers and implemented sub tree mutations for evolution of morphologies. Figure 6 in section 7.2 shows a diagram of the co-evolutionary learning algorithm. Genetic algorithms are evolutionary algorithms based on principles of survival of the fittest and randomised mutation and have become a default problem solving approach in evolutionary computation since they are extremely robust, yet also efficient [10]. The selection strategy employed by the learning algorithm was Deterministic Tournament Selection (DTS) as it was the only option available using the Robogen framework. DTS selects k individuals from the population for each parent that needs to be selected for reproduction, where k is the tournament size [8]. The individual with the highest fitness among the tournament is selected for reproduction. Since selection is fitness based, the fitness function exerts a selection pressure on evolution. DTS maintains a good balance of selection pressure towards high fitness individuals but should also give less fit individuals a chance to be selected if k is relatively small. In this way genetic diversity is maintained in the population. The fitness function assigns a numeric score to an individual based on how far they were able to move in the environment after one simulation, the details of this function are discussed in section 3.5.

The replacement strategy chosen for the algorithm employs the concept of elitism, where only the individuals with the highest fitness scores survive each generation. This strategy is appropriate for the project given that a secondary research aim is to investigate the correlation between robot fitness and complexity. Diversity is maintained by the selection and mutation process. The following sections explain the process of evolving the robot morphologies and controllers once suitable parents have been selected.

3.2.1 Evolution of Morphologies. Initial robot morphology trees are seeded from a pre-configured robot, they are then subject to a random mutation to ensure that the starting population has enough variance. An initial seed was used due to limitations on the number of generations experiments could be run for, without a seed low fitness and complexity values were likely. The morphology trees were evolved via Robogen's standard body evolution methods. Instead of mating two morphologies together, the robot's morphology has a probability of being randomly mutated once per generation (iteration). This random mutation could remove a part of the morphology tree, add a new sub tree, crossover existing sub trees or add a new node. Adding a new node to the tree would require a free slot on the robot's morphology tree, the new node would then be inserted. Sub trees can be exchanged or removed in the same way, simply attached to or removed from a slot on the robot's morphology tree. The mutation probabilities were set relatively high to maintain as much diversity as possible given the elitist nature of the replacement strategy. The probabilities used are shown in Table 2.

3.2.2 Evolution of Controllers. The connection weights between a robot's neurons are encoded by the associated CPPN. Since the CPPNs approximate the controller of the robot, it is the CPPNs of robots that are mated. This learning algorithm uses an adapted version of HyperNEAT to evolve CPPNs [27]. Due to unexpected complications with the Robogen framework and time limitations, only a simple version of HyperNEAT could be implemented. HyperNEAT is an extension of NeuroEvolution of Augmenting Topologies (NEAT) that allows neuroevolution to scale topology connections to that of a comparable scale of the natural brain [27]. NEAT optimises and complexifies candidate topology-weight couplings simultaneously and favours incremental growth of simple solutions rather than the traditional approaches using numerous random seeds [26]. These solutions grow in complexity as evolution continues, allowing minimisation of the dimensionality of the search space. Typically, HyperNEAT evolves a population of CPPNs by incremental complexification of networks through the additions of new neurons and links. A diverse CPPN population is maintained through speciation, which groups together similar topologies based on neuron and connection parameters.

The version of HyperNEAT implemented for this project forgoes the technique of speciation, it is not necessary when maintaining a significantly smaller population of CPPNs. Each robot has one associated CPPN, thus a population of 100 CPPNs is maintained. The process of mating CPPNs is less dramatic than that of morphologies, usually with one or two nodes in the network being exchanged per reproduction. Thus the probability of a dysfunctional network is highly unlikely but in the event that such a reproduction occurred, CPPNs would be re-mated. The mutation probabilities used for mating CPPNs were kept the same as Robogen's default HyperNEAT parameters, these are available on Robogen's website [1].

Algorithm 1 Co-evolutionary genetic algorithm

```
Initialise population of 100 robots from seed
for Each generation g = 0...100 do
    Select one hundred pairs of robots to produce offspring for next generation
    for Each pair of parents do
        Generate offspring by mutating one of the parent's bodies
        Mate CPPNs of the two parents and assign to offspring
    end for
    Group and rank offspring and parents based on fitness
    Select the top 100 individuals for the next generation
end for
```

Algorithm 2 Fitness Function

```
Begin simulation of robot in environment
startPos = positionOfCoreComponent
for Each each evaluation do
    minDist = MaxVal
    for Each body part do
        xDist = partPosition − startPos
        yDist = partPosition − startPos
        totalDist = sqrt(partPosition^2 − startPos^2)
        if totalDist < minDist then
            minDist = totalDist
        end if
    end for
end for
Return minDist as fitness
```

3.3 Imposing a Complexity Cost

The approach taken to impose a complexity cost is inspired by previous work by Hallauer and Nitschke who imposed an *energy cost* on body-brain complexity [12]. In this paper, an energy cost was imposed by decreasing the simulation lifetime of a robot based on body-brain complexity. In this paper, the terms *energy cost* and *complexity cost* will be used interchangeably. A complexity cost was imposed on individuals in the second experiment by limiting the number of simulation ticks a robot experienced. It is thought that in nature, the complexity of an individual is a constraint on the rate at which they adapt and evolve, compared to simpler organisms [22]. The fitness of a robot is directly linked to how it performs over a simulation run, thus high complexity robots will need to perform at least as well as low complexity robots in a smaller time frame to achieve decent fitness scores. This complexity cost served to filter out machines that were complex because of environmental biases instead of those that were complex due to the resulting behavioural advantage [3]. Per generation, each robot's fitness was evaluated during a simulation in an environment. Each simulation runs for a number of simulation ticks, emulating the passing of time in an artificial environment. Instead of starting at tick $t = 0$, robots in the experiment with an imposed complexity cost started the simulation with a tick value of

$$t = (s - (1 - c) * s)$$

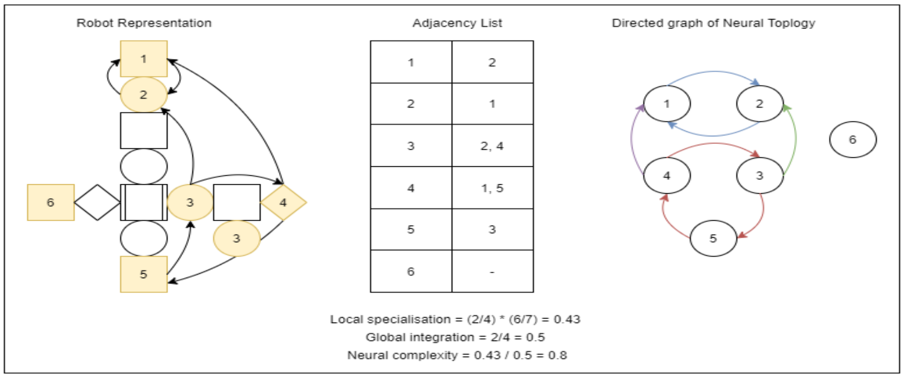
where s is the total simulation run time and c is the complexity of the robot. The imposed cost is directly proportional to the robot's complexity. This directly affects the amount of simulation time each robot receives, thus the lower a robot's complexity, the longer they have in the simulation. This gives lower complexity robots a fair chance to achieve high fitness scores and requires that high complexity robots exhibit strong task performance immediately if they are to obtain high fitness scores. Various other works investigating the evolution of complexity have imposed costs on complexity in different ways, most using multiobjective evolutionary algorithms [20, 21]

3.4 Complexity Metric

As mentioned earlier in this paper, ER has and will continue to contribute to understanding the environmental conditions under which complexity evolves in natural and artificial evolution. However, research into the effects that environments have on neural or morphological complexity is lacking due to difficulties developing meaningful complexity metrics (a method for measuring the complexity of a system) for environmental, neural and morphological complexity [3]. When devising a complexity metric, both objective (i.e. system size) and subjective complexity (depends on the representation) need to be considered [7]. Besides early papers from Larry Yaeger studying evolving complexity in AL systems through use of co-evolution methods [15, 33], there have been few true co-evolution studies into the evolution of complexity using a complexity metric that accounts for both body and brain complexity.

The neural complexity component on the metric presented in this paper is based off of work by Tononi et al. and Tononi and Sporns [30, 32]. These papers provide information theoretic methods for measuring neural complexity. Specifically, they present a metric that is maximised when there is a balance between local specialisation and functional segregation of regions of the neural topology and global integration between regions of the neural topology. The premise of the metric is to predict meaningful neurological behaviour instead of measuring the objective size of the network. The neural metric used in this project adopts the logic used in the work of Tononi et al. but adapts the methods of calculation to suit the representation of robot controllers used [32]. Such a complexity metric should aid in the evolution of modular (locally specialised, sparsely globally integrated) ANNs, which are believed to allow higher levels of evolvability (ability to adapt to novel environments) in organisms [4].

This paper presents a complexity metric that accounts for both morphological and neural complexity. The morphological and neural complexity values are calculated separately and combined to give the overall system complexity of the robot. The morphological component of the metric assumes an inherent complexity value is assigned to each body part type. As seen in figure 1, the three different node types are assigned different complexity values - *ActiveHinge* having the highest value since it has the greatest potential impact on locomotion. The higher morphological complexity value assigned to *ActiveHinge*s also account for a portion of the neural complexity since oscillators are embedded in hinge nodes only. To



1	2
2	1
3	2, 4
4	1, 5
5	3
6	-

Figure 2: Example neural complexity calculation. Left: example robot representation where yellow nodes have embedded neurons, arrows between nodes represent connections between these neurons. Middle: corresponding adjacency list (list of adjacent neurons per neuron). Right: directed graph of the neural topology of the example robot. Middle-bottom: example neural complexity calculation.

calculate the overall morphological complexity, the robot's morphology tree is traversed and the complexity value for each node encountered is summed.

$$c(n) = complexity\ of\ node\ n$$

$$mComplexity = \sum_{n=1}^{t} c(n)$$

where t is the number of body parts in the morphology tree. This morphological complexity value is then scaled to be between zero and one.

To calculate total neural complexity, an adjacency list storing the adjacent neurons of each neuron is determined. This adjacency list is used to create a directed graph representing the neural topology of the controller. Neurons are then grouped into strongly connected components (sub-graphs of a directed graph where there exists a path between each pair of nodes in the sub-graph). Strongly connected components are representative of areas of local or functional specialisation in the network. Figure 2 (left) shows an example robot body-brain configuration, (middle) adjacency list and (right) the associated directed graph. Nodes in the morphology tree with embedded neurons are shown in yellow where arrows between nodes represent connections between the embedded neurons. In figure 2 (right), example strong components can be seen in the directed graph of the neural topology. In this example there are four strong components, the members of which are indicated by the differently coloured arrows between nodes. Cycles within these strongly connected components are identified and recorded as they are representative of areas of local specialisation. In the example figure there are two different cycles within two different strong components (blue and red). The total specialisation, s, of the brain was then calculated according to the following equation

$$s = (c/s) * (n/t)$$

where c is the number of cycles identified, s is the number of strong components, n is the number of neurons contained in strong components and t is the total number of neurons in the topology. Figure 2 (middle-bottom) shows an example calculation of local specialisation based on the strong components identified in the directed graph of the neural topology.

Global integration, g, in the brain can be represented by connections between strong components, as these connections imply communication between areas of local specialisation in the brain. The global integration of the brain is the ratio of connections between strong components to the number of strong components, an example calculation of this is shown in figure 2 (middle-bottom). Total neural complexity, according to the logic of Tononi et al. should then be representative of a balance between the local specialisation and global integration calculated. It unimportant whether global integration is marginally larger than the specialisation score, but one being significantly larger than the other should reflect in the final neural complexity calculation. Thus, neural complexity is calculated according to Equation (1) or Equation (2) to avoid excessively large numbers should there be a significant difference between s and g.

$$nComplexity = g/s \tag{1}$$

$$nComplexity = s/g \tag{2}$$

The result is that a disproportionate difference between these values will result in a low neural complexity score. Following the neural complexity calculation, the morphological complexity and neural complexity are combined as follows to give the total complexity score.

$$Complexity = (0.7) * mComplexity + (0.3) * nComplexity$$

Morphological complexity is given a greater weighting since the *ActiveHinges* in the robot's morphology may contain embedded neural oscillators - these oscillators influence meaningful locomotion patterns. Thus, morphological complexity is representative of neural complexity to an extent. Furthermore, from the perspective of Automated Engineering and real world applications for artificial robot design, the morphology of robots would be a larger design constraint than neural complexity due to the involved manufacturing cost of a complex morphology [6].

3.5 Objective Fitness Evaluation

Fitness functions are an imperative part of the evolutionary process as they determine the selection pressure in the algorithm. Selection pressure is largely known as the driving force of evolution as it is responsible for diversifying the genetics of the population. The fitness function decides which individuals are selected for reproduction, without an appropriate fitness function individuals would effectively be randomly selected and reproduction would be equivalent to genetic drift (random changes in genome) [9].

Objective fitness functions evaluate how *fit* a robot is based on some tangible objective, the most commonly used objective fitness function is the performance based fitness function. Performance based fitness functions, or behavioural fitness functions, provide a measure for how well the robot is performing physically, usually in direct relation to the completion of some task. Behavioural fitness functions are useful for well-defined tasks in which a successful outcome is immediately obvious, for example locomotion, and have see much success in various ER experiments investigating the evolution of complexity [3, 20, 21].

The fitness function used by the learning algorithm in this study is a performance based fitness function, quantifying the success of a robot based on the minimum distance reached in an environment. At each evaluation the distance from each body part to the starting position of the robot is calculated, the minimum of these distances is then recorded. The minimum distance is used instead of the maximum because the minimum distance a robot is able to travel is it's biggest performance constraint. At the end of the simulation, the minimum distance travelled is returned as the fitness. Pseudo code of the fitness calculation is given in Algorithm 2.

3.6 Testing and Validation

There is no single way to validate a Genetic Algorithm as results produced from each run will vary due to the random nature of EAs and evolution. Since the Robogen framework provides functionality for most of the evolutionary process, validation had to ensure that the modified learning algorithm performed at least as well as the existing co-evolutionary algorithm available in Robogen. Therefore, runs of the experiment using the same random number and robot seeds were conducted with Robogen's learning algorithm versus the adapted algorithm used for this project. After comparing the average fitness produced at each stage of evolution it could be seen that the adapted algorithm performed just as well as Robogen's own. Furthermore, the evolved robots produced from either run were visualised using Robogen's visualisation software, this allows you to view how a robot progresses in an environment. The robots evolved by the adapted algorithm were able to physically perform just as well as those evolved by Robogen's learning algorithm.

4 EXPERIMENT DESIGN AND EXECUTION

This section outlines the design of evolutionary environments (section 4.1 and the simulation framework (section 4.2) used to execute experiments.

4.1 Experiment Overview

Two experiments were run using the simulation framework provided by Robogen [1]. The first experiment evolved an initial population, seeded from a pre-defined robot, of one hundred simulated robots in 12 different environments of increasing complexity for the task of locomotion. Each evolution, where an evolution is one hundred iterations of evolution run in one environment, then had to be repeated 10 times so that results could be averaged to account for the random nature of EAs. The second experiment was identical to the first with the addition of an imposed complexity cost on evolution. In both experiments, the fitness of robots was evaluated using the objective fitness function discussed in section 3.5. The results gathered from the experiments were the fitness, the complexity and the body-brain configuration of each robot in the population per environment per run (10 runs per environment). Comparison of the complexities of populations evolved in environments of differing complexity, with and without a complexity cost, will provide evidence to refute or support research objective 1. Correlations between complexity and fitness of evolved populations will be compared in experiment 1 and 2 to provide evidence for or against research objective 2.

Part of this experiment is investigating the correlations between the complexity of an environment and the complexity of evolved robots. To this end, twelve environments of increasing complexity were designed. Four sets, of three environments each, were designed where set 1 is the least complex and set 4 the most complex. Each environment per set had differing friction values but used the same obstacle set or tilted (inclined) floor. This approach to the design of evolutionary environments is based off of Auerbach and Bongard's previous work on investigating complexity [2, 3], where low friction areas are meant to simulate patches of ice that are more difficult to traverse than high friction areas. Table 1 shows the friction, obstacle set and tilt parameters used per set. Gradual increases in complexity were introduced within each set by decreasing friction values whilst complexity was increased between sets through the addition of obstacles or inclined floors. Images of environments 1 (least complex, flat, full friction), 5 (average complexity, flat, regular obstacles, medium friction) and 12 (most complex, tilted, low friction) can be seen in figure 5 left, middle and right respectively.

4.2 Simulation Framework

Robogen open source framework was used to run all experiments. The simulation framework provided by Robogen handles the physics simulation of robots in each of the configured environments and calculates the fitness of a robot based on this simulation. Furthermore, Robogen handles the evolution of robots in the population based on the aforementioned fitness scores. The Robogen framework had to be extended to allow for the co-evolution of robot bodies and brains using an adapted version of HyperNEAT [27], outlined in section 3.2.2. Other modifications to the Robogen framework included the imposition of a complexity cost on evolution and the addition of a complexity metric that accounts for both robot bodies and brains. This is discussed in sections 3.3 and 3.4.

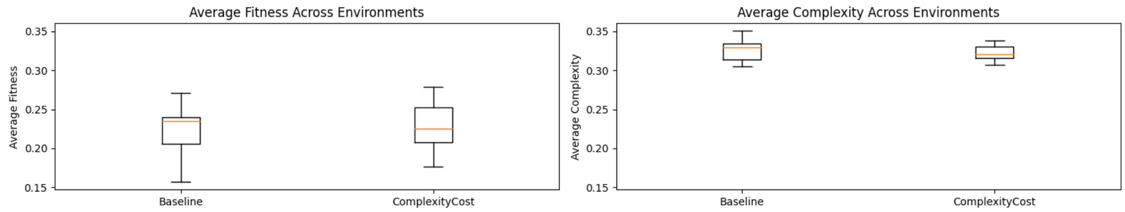


Figure 3: Left: Average fitness of the populations evolved across all environments (see Table 1) in the *Baseline* experiment (without an imposed complexity cost) and in the *ComplexityCost* experiment (with an imposed complexity cost). **Right** Average complexity of population in Baseline versus ComplexityCost.

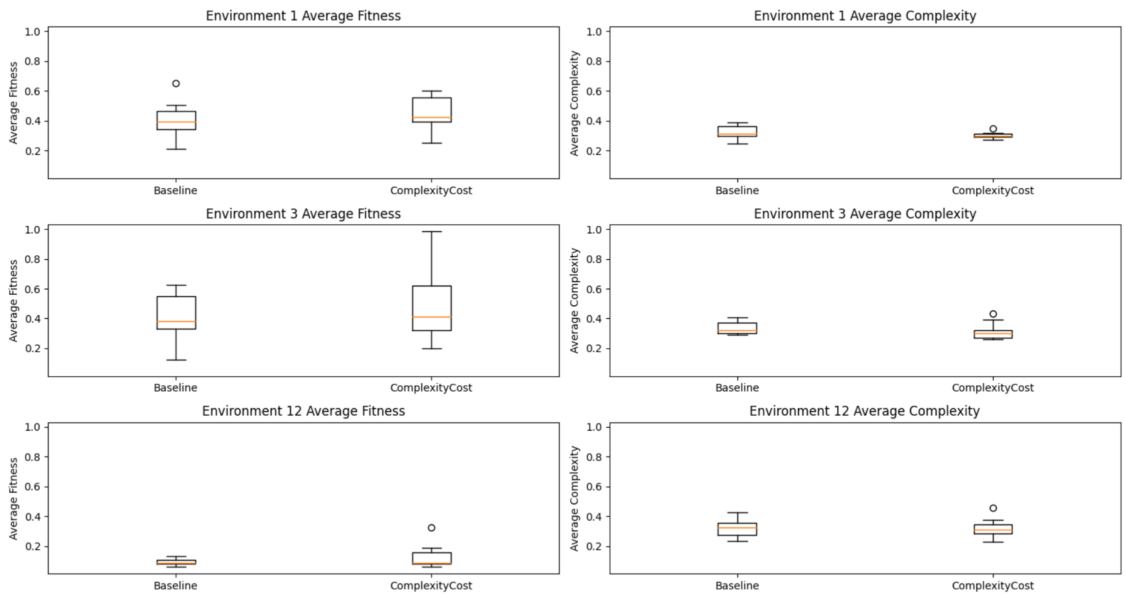


Figure 4: Row 1, left: Average fitness of the population evolved in Environment 1 in Baseline versus ComplexityCost. **Row1, right:** Average complexity of individuals in Baseline versus ComplexityCost in Envrionment 1. **Rows 2 and 3, left and right:** As above, for Environments 3 and 12 respectively. See table 1 for environmental complexity parameters.

5 RESULTS AND DISCUSSION

To the end of refuting or supporting the research hypothesis, the experiments run aimed to evaluate the impact of an *energy cost* on the evolution of complexity in simulated robots. The 12 environments in which experiments were run were of increasing complexity where environment 1 was the simplest and environment 12 the most complex (see table 1). To evaluate the correlation between environmental complexity and overall robot complexity, the *average complexity* of evolved populations was computed over the ten runs per environment as well as the *average complexity* of evolved populations across all runs on all environments (12*10*2 = 240 evolutionary runs). Figure 3 (left) presents the average fitness and (right) average complexity of the evolved populations, averaged across all environments in the *Baseline (B)* experiment (no complexity cost) versus *ComplexityCost (CC)* experiment (with an imposed complexity cost). Table 3 shows the average complexity and average fitness values of CC and B over all runs in each environment. In this table, environments where average complexity was lower in CC than B are shown in red text, environments where average fitness was higher in CC than B are shown in blue text and environments exhibiting both lower average complexity and higher average fitness in CC than B are shown in green.

Figure 4 presents the average fitness and average complexity of the populations evolved in environments 1 (row 1), 3 (row 2) and 12 (row 3) respectively, both with and without an imposed complexity cost. Figures for the 9 remaining environments can be found in 7.4. Additional figures plotting the *average maximum fitness and complexity* per environment in CC and B can be found in section 7.5 (the maximum average results displayed similar trends to figures included in the main text). A complexity value of 0.0 indicates the least possible complex robot body-brain configuration, whilst a complexity value of 1.0 indicates the most complex configuration. The complexity values demonstrated in the figures presented in this section are within a small range, this is due to the seeding

Environment	Average Complexity (B)	Average Complexity (CC)	Average Fitness (B)	Average Fitness (CC)
1	0.324	0.302	3.166	3.457
2	0.327	0.323	3.291	3.007
3	0.335	0.314	3.139	3.621
Avg Set 1	**0.329**	**0.331**	**3.199**	**3.362**
4	0.316	0.323	1.579	1.807
5	0.350	0.361	2.084	1.860
6	0.329	0.298	1.148	0.953
Avg Set 2	**0.332**	**0.327**	**1.604**	**1.540**
7	0.323	0.323	1.526	1.662
8	0.327	0.317	1.549	1.505
9	0.310	0.309	1.088	0.903
Avg Set 3	**0.320**	**0.316**	**1.388**	**1.357**
10	0.335	0.384	2.334	1.983
11	0.318	0.293	2.314	2.661
12	0.319	0.320	1.216	1.436
Avg Set 4	**0.324**	**0.332**	**1.955**	**1.693**

Table 3: The average complexity and average fitness of the *Baseline (B)* and *ComplexityCost (CC)* experiments, per envrionment. Rows with green text indicate environments where average complexity was lower in CC than B and average fitness was higher in CC than B. Rows with red text indicate environments where average complexity in CC was lower than B. Rows with blue text indicate environments where average fitness in CC was higher than in B. The averages per environment set (environment sets shown in table 1) are in bold text.

of the robot population from an initialisation robot. More varied complexity values may have been discovered if robots were evolved for more generations, due to time constraints this was not possible. It is worth noting that the x and y axes of all graphs have been normalised to be between 0.0 and 1.0 for the sake of comparison of general trends in average fitness or complexity between the Baseline and ComplexityCost experiments.

Figure 4 (rows 1-3, left) shows that the average fitness evolved in environments 1, 3 and 12 with a complexity cost was higher than that of evolutions without a complexity cost. As shown in Table 3, the same result was found in 50% of the evolutionary environments (rows in green and blue) - although these differences were not found to be statistically significant. However, statistical differences between environmental fitness values were found which suggests that there is significance in six of the environments having higher average fitness in CC than B. The lack of statistical significance between fitness values within each environment may be due to a number of factors. The seeding of robots from an initial design meant that the behaviours exhibited by the evolved population were similar and consequently, discovered fitness values of the population were relatively similar. Thus, this study would have benefited greatly from both more runs per environment and more generations (evolutions) per environment. The average fitness values of the remaining six environments of the ComplexityCost experiment were 11% lower than the average fitness values of the Baseline experiment. Although an 11% difference is relatively high, the discrepancies in results could be due to the severity of the complexity cost imposed on individuals.

The complexity cost was directly proportional to the complexity of the robot and given that the seed robot's initial complexity was relatively high, robots may not have had enough time to evolve appropriate low-complexity, high fitness configurations in CC versus B. Running experiments with a greater population size and less severe complexity cost would most likely yield more concrete evidence for the correlation between an imposed complexity cost and fitness. A further explanation for the contrasting results is the random nature of evolutionary algorithms, one random mutation might result in large fitness gains whilst another might result in significant decreases. To account for the randomness of the algorithm it would have been favourable to have more runs per experiment, but due to time constraints this was not possible.

Statistical significance between data sets was calculated using the Shapiro-Wilk test for data normality and the Mann-Whitney U test [13, 24]. Tests for statistical significance were computed between both average complexity and average fitness in CC and B per environment. Furthermore, both average complexity and average fitness (in B and CC) between environments was tested for statistical significance. The lack of statistical difference in average fitness of B and CC within environments has been explain previously. It was found that there was no statistical difference between average complexity of B and CC within environments. This is most likely due to the lack of sensitivity of the neural portion of the complexity metric to small neural differences and a lack of neural diversity in evolved populations, resulting in similar complexity values throughout the experiments.

As demonstrated by figure 4 (rows 1-3, right), the average complexity of populations evolved in environments 1, 3 and 12 was lower than that of populations evolved without a complexity cost. Table 3 shows that eight of the environments (rows in green and red) demonstrated average complexity lower in CC than or equal to that of B. These results show that a greater selection pressure for low-complexity, high fitness individuals was imposed in experiments run with a complexity cost. Furthermore, figure 4 (rows 1 and 3, right) suggests that environmental complexity did not have a direct impact on robot complexity. Environment 12 was defined to be the most complex environment however, as seen in Table 3, average population complexity in environments 4 and 5 was greater than that of environment 12, with a complexity cost. Similar results can be seen by comparing the average complexity of environment 2 to that of environment 12, with a complexity cost, where environment 2 displayed a higher average complexity. These differences in average complexity between environments were found to be statistically significant. Figure 3 summarises these results, demonstrating that both higher average fitness and lower average complexity values, when averaged across all environments, were evolved with an imposed complexity cost.

Table 3 shows that average complexity per environment set (rows in **bold** text) was lower in CC than in B for sets 2 and 4. These results suggest that lower complexity individuals were evolved in most environments with an imposed complexity cost versus the same environments without a complexity cost. However, there is no immediate correlation found, in this data, between environmental complexity and robot body-brain complexity. Upon comparing the average complexity values in B and CC in Table 3 per environment set, there is no immediate link to increasing environmental complexity and increasing robot complexity. The average complexity in CC for environment set 1 was greater than that of environment sets 2 and 3, which were designed to be more complex. Furthermore, the average complexity of environment set 2 in CC was greater than that of environment set 3 despite the latter having higher environmental complexity. These differences were found to be statistically significant.

It is worth noting the possibility that the definitions of environmental complexity used in these experiments may have affected results if an environment designed to be complex in theory did not require a complex robot configuration in practice to achieve decent fitness values. Figure 7 in section 7.3 presents the average complexity trend over 100 generations of evolution in each environment in both CC and B. Although there does not seem to be a correlation between corresponding environments in CC and B, the general trend shows that complexity values were lower towards the last fifty generations of evolution in CC versus B. This implies that the fittest individuals found in the later half of evolution in CC also had low complexity, this is further evidence of an imposed selection pressure for high fitness, low complexity robots. The disproportionately high complexity values for environments 10 and 5 in CC are most likely results of random mutations that led to a high complexity population early on in evolution. It can also be seen that the inherent complexity of the environments did not affect the evolution of complexity, an example of this can be seen in the plots of environments 1 and 11, which were two of the lowest average complexity environments in CC even though environment 11 was defined to be the second most complex environment.

In summary, evidence refuting research objective 1 was found through the comparison of average population complexity between environments of varying complexity. Results suggest that robot complexity decreases in both complex and simple environments when a cost is imposed on robot brain-body complexity. Conversely, support for research objective 2 was found by analysing correlations between population fitness and complexity in CC versus B. Trends were discovered suggesting that the imposition of a complexity cost may evolve simpler robots that perform just as well as those evolved without a cost. These findings provide evidence against the proposed research hypothesis, finding that complexity decreases, despite environmental conditions, when a cost on complexity is imposed.

6 CONCLUSIONS AND FUTURE WORKS

The experiments conducted provided evidence against the primary research hypothesis 1, *given conducive environmental conditions, complexity increases even with imposed complexity costs*, by investigating the relationship between environmental complexity and average complexity of simulated robots. It was found that there was no correlation between environmental complexity and the complexity of evolved robot body-brain configurations. However, it was found that, on average, the imposition of a complexity cost on evolution evolved robots of lower complexity despite environmental complexity. The secondary findings of these experiments were the correlations between fitness and complexity when a cost on complexity was imposed. The results presented show that, on average, lower complexity, equally performing robots were evolved across environments when a complexity cost was imposed versus without a complexity cost. These results support the findings of previous work by Nagar et al., who found that simpler morphologies were evolved in both simple and complex environments when a cost on complexity was imposed [21]. Previous work by Auerbach and Bongard found that more complex robot morphologies were evolved in more complex environments, adversely this study found that there was no clear correlation between increasing environmental and robotic complexity (both with and without a complexity cost) [3].

A further conclusion to be drawn from results is the finding that, on average, the imposition of a complexity cost results in a greater selection pressure for low complexity, high performing individuals thus lending support to the secondary research objective 2. This supports previous findings of Nagar et al. who found that a multi-objective evolutionary algorithm selecting for low neural complexity and high fitness was effective in maximising the performance of robots and minimising the complexity of the robot brains, producing robots just as behaviourally successful as those evolved with a single objective evolutionary algorithm selecting for fitness [20]. A different study by Nagar et al. found similar results when imposing a complexity cost on morphological complexity [21].

This research would benefit from fine tuning of the evolutionary parameters to the end of producing more consistent results. Future work might implement the full version of HyperNEAT, which may encourage a more diverse and competitive CPPN population, to the end of producing larger differences between complexity values for different environments. Additionally, this research would benefit from a study implementing less evolutionary environments with larger, more obvious differences in complexity. This might illuminate the differences of evolved complexity in simple versus complex environments more clearly than this paper.

REFERENCES

[1] Joshua Auerbach, Deniz Aydin, Andrea Maesani, Przemyslaw Kornatowski, Titus Cieslewski, Grégoire Heitz, Pradeep Fernando, Ilya Loshchilov, Ludovic Daler, and Dario Floreano. 2014. RoboGen: Robot Generation through Artificial Evolution. 136–137. https://doi.org/10.7551/978-0-262-32621-6-ch022

[2] Joshua Auerbach and Josh Bongard. 2012. On the Relationship Between Environmental and Mechanical Complexity in Evolved Robots. 309–316. https://doi.org/10.7551/978-0-262-31050-5-ch041

[3] Joshua E. Auerbach and Josh C. Bongard. 2014. Environmental influence on the evolution of morphological complexity in machines. *PLoS computational biology* 10, 1 (2014), e1003399–e1003399. https://doi.org/10.1371/journal.pcbi.1003399

[4] Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. 2013. The evolutionary origins of modularity. *Proceedings. Biological sciences / The Royal Society* 280 (07 2013), 20122863. https://doi.org/10.1098/rspb.2012.2863

[5] S. Doncieux, N. Bredeche, J.-B. Mouret, and A.E. Eiben. 2015. Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI* 2, 4 (2015). https://doi.org/10.3389/frobt.2015.00004

[6] Stéphane Doncieux, Jean-Baptiste Mouret, Nicolas Bredeche, and Vincent Padois. 2011. Evolutionary Robotics: Exploring New Horizons. In *New Horizons in Evolutionary Robotics*, Stéphane Doncieux, Nicolas Bredèche, and Jean-Baptiste Mouret (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 3–25.

[7] Mahmoud Efatmaneshnik and Michael J. Ryan. 2016. A general framework for measuring system complexity. *Complexity* 21, S1 (2016), 533–546. https://doi.org/10.1002/cplx.21767 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cplx.21767

[8] Iñaki Fernández Pérez, Amine Boumaza, and François Charpillet. 2014. Comparison of Selection Methods in On-Line Distributed Evolutionary Robotics. *Artificial Life Conference Proceedings* 26 (2014), 282–289. https://doi.org/10.1162/978-0-262-32621-6-ch046

[9] D Floreano and Joseba Urzelai. 2000. Evolutionary Robots with on-line self-organization and behavioral fitness. *Neural networks : the official journal of the International Neural Network Society* 13 (06 2000), 431–43. https://doi.org/10.1016/S0893-6080(00)00032-0

[10] David E. Goldberg. 1988. Genetic Algorithms in Search Optimization and Machine Learning.

[11] M. T. Hagan and H. B. Demuth. 1999. Neural networks for control. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, Vol. 3. 1642–1656 vol.3.

[12] S. Hallauer and G. Nitschke. 2020. Energy and Complexity in Evolving Collective Robot Bodies and Brains. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8.

[13] T. Hettmansperger and J. McKean. 2011. *Robust Nonparametric Statistical Methods*. CRC Press. 467 pages.

[14] John Hopfield. 1982. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79 (05 1982), 2554–8. https://doi.org/10.1073/pnas.79.8.2554

[15] Greg Hornby, Hod Lipson, and Jordan Pollack. 2003. Generative Representations for the Automated Design of Modular Physical Robots. *Robotics and Automation, IEEE Transactions on* 19 (09 2003), 703 – 719. https://doi.org/10.1109/TRA.2003.814502

[16] Milan Jelisavcic, Kyrre Glette, Evert Haasdijk, and A. E. Eiben. 2019. Lamarckian Evolution of Simulated Modular Robots. *Frontiers in Robotics and AI* 6 (2019), 9.

[17] Henry Markram. 2006. The Blue Brain project. *Nature reviews. Neuroscience* 7 (03 2006), 153–60. https://doi.org/10.1038/nrn1848

[18] Henry Markram, Karlheinz Meier, Thomas Lippert, Sten Grillner, Richard Frackowiak, Stanislas Dehaene, Alois Knoll, Haim Sompolinsky, Kris Verstreken, Javier Defelipe, Seth Grant, Jean-Pierre Changeux, and Alois Saria. 2011. Introducing the Human Brain Project. *Procedia Computer Science* 7 (12 2011), 39–42. https://doi.org/10.1016/j.procs.2011.12.015

[19] Thomas Miconi. 2008. Evolution and Complexity: The Double-Edged Sword. *Artificial life* 14 (02 2008), 325–44. https://doi.org/10.1162/artl.2008.14.3.14307

[20] D. Nagar, A. Furman, and G. Nitschke. 2019. The Cost of Big Brains in Groups. *Artificial Life Conference Proceedings* 31 (2019), 404–411. https://doi.org/10.1162/isal_a_00193

[21] D. Nagar, A. Furman, and G. Nitschke. 2019. The Cost of Complexity in Robot Bodies. In *2019 IEEE Congress on Evolutionary Computation (CEC)*. 2713–2720.

[22] H. Allen Orr. 2000. Adaptation and the Cost of Complexity. *Evolution* 54, 1 (2000), 13–20.

[23] Timothy E Revello and Robert McCartney. 2000. A cost term in an evolutionary robotics fitness function. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, Vol. 1. IEEE, 125–132 vol. 1.

[24] S. S. Shapiro and M. B. Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* 52, 3/4 (1965), 591–611. http://www.jstor.org/stable/2333709

[25] K. Sims. 1994. Evolving 3D Morphology and Behavior by Competition. *Artificial Life* 1, 4 (1994), 353–372.

[26] Kenneth Stanley. 2007. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines* 8 (06 2007), 131–162. https://doi.org/10.1007/s10710-007-9028-8

[27] Kenneth Stanley, David D'Ambrosio, and Jason Gauci. 2009. A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. *Artificial life* 15 (02 2009), 185–212. https://doi.org/10.1162/artl.2009.15.2.15202

[28] Dario Floreano Stefano Nolfi. 2001. *Evolutionary Robotics The Biology, Intelligence and Technology of Self-Organising Machines*. Massachuesetts Institute of Technology.

[29] Giulio Tononi and Olaf Sporns. 2003. Measuring information integration. *BMC Neuroscience* 4, 1 (2003), 31. https://doi.org/10.1186/1471-2202-4-31

[30] Giulio Tononi and Olaf Sporns. 2003. Measuring information integration. *BMC neuroscience* 4, 1 (2003), 31.

[31] G Tononi, O Sporns, and G M Edelman. 1994. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences* 91, 11 (1994), 5033–5037. https://doi.org/10.1073/pnas.91.11.5033

[32] G Tononi, O Sporns, and G M Edelman. 1994. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences* 91, 11 (1994), 5033–5037. https://doi.org/10.1073/pnas.91.11.5033

[33] Larry Yaeger. 2009. How evolution guides complexity. *HFSP journal* 3 (10 2009), 328–39. https://doi.org/10.2976/1.3233712

7 SUPPLEMENTARY MATERIALS

7.1 Example Evolutionary Environments

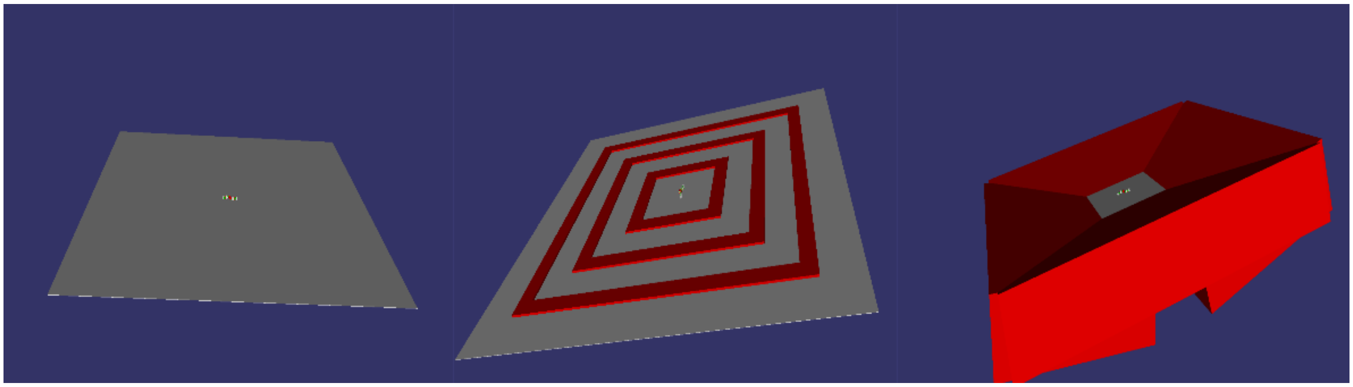


Figure 5: Three example evolutionary environments. Left: simple, flat environment, Middle: average complexity environment with obstacles, Right: high complexity, tilted environment. Environment parameters are shown in Table 1.

7.2 Co-evolutionary Learning Algorithm

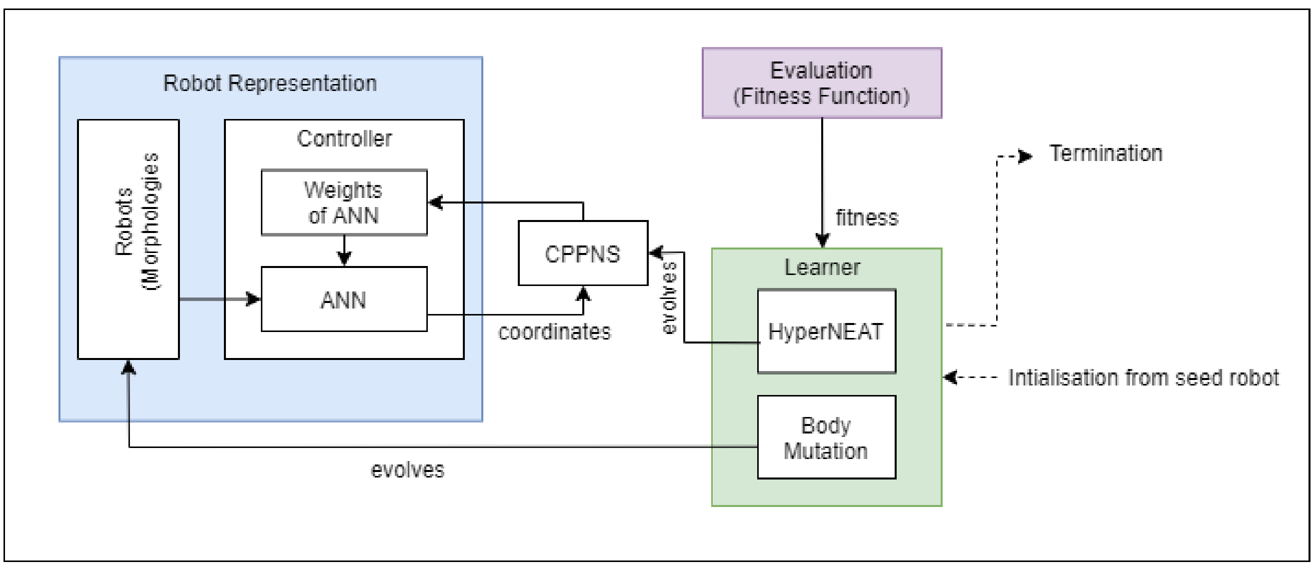


Figure 6: The co-evolutionary learning algorithm. HyperNEAT and body mutation are used to evolve robot body-brain configurations, where CPPNs evolved by HyperNEAT update weight information in the Artificial Neural Network of the controller. Diagram inspired by work by Jelsavcic et al. [16]

7.3 Evolution of Average Fitness and Complexity per Generation

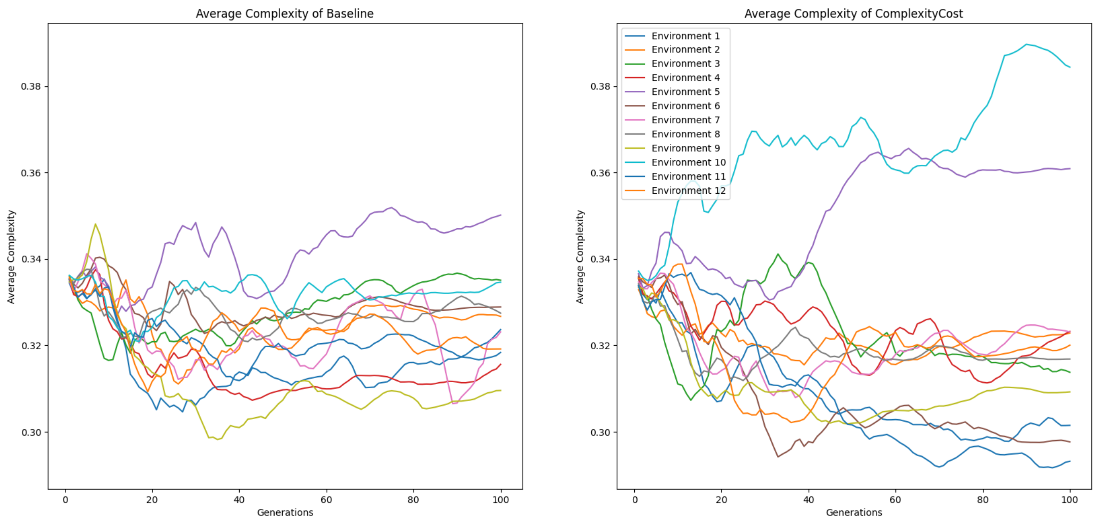


Figure 7: Left: Average complexity of the populations evolved across 100 generations for each environment in the *Baseline* experiment (without an imposed complexity cost). **Right:** Average complexity of the populations evolved across 100 generations for each environment in the *ComplexityCost* experiment (with an imposed complexity cost)

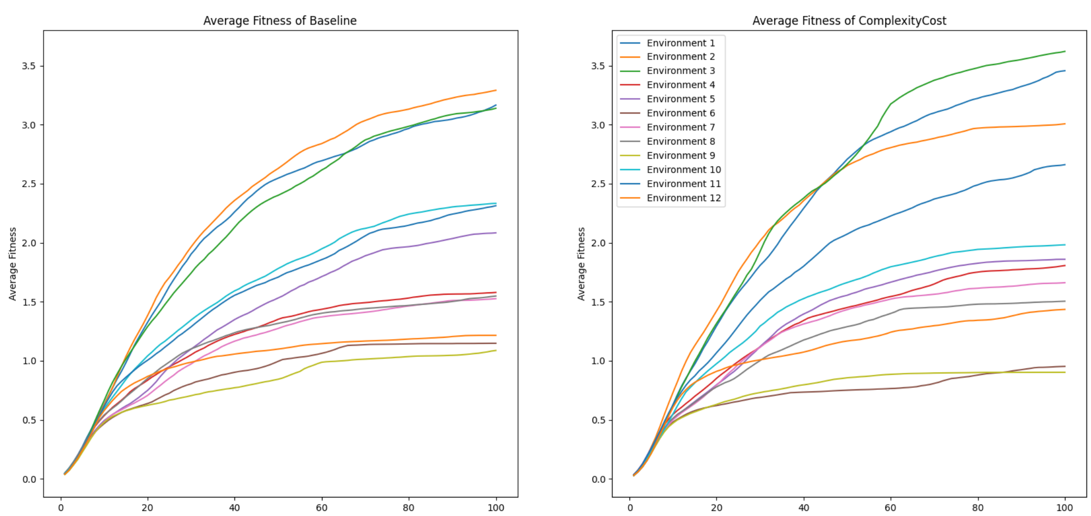


Figure 8: Left: Average fitness of the populations evolved across 100 generations for each environment in the *Baseline* experiment. **Right:** Average fitness of the populations evolved across 100 generations for each environment in the *ComplexityCost* experiment.

7.4 Average Complexity and Fitness of Environments 2 and 4-12

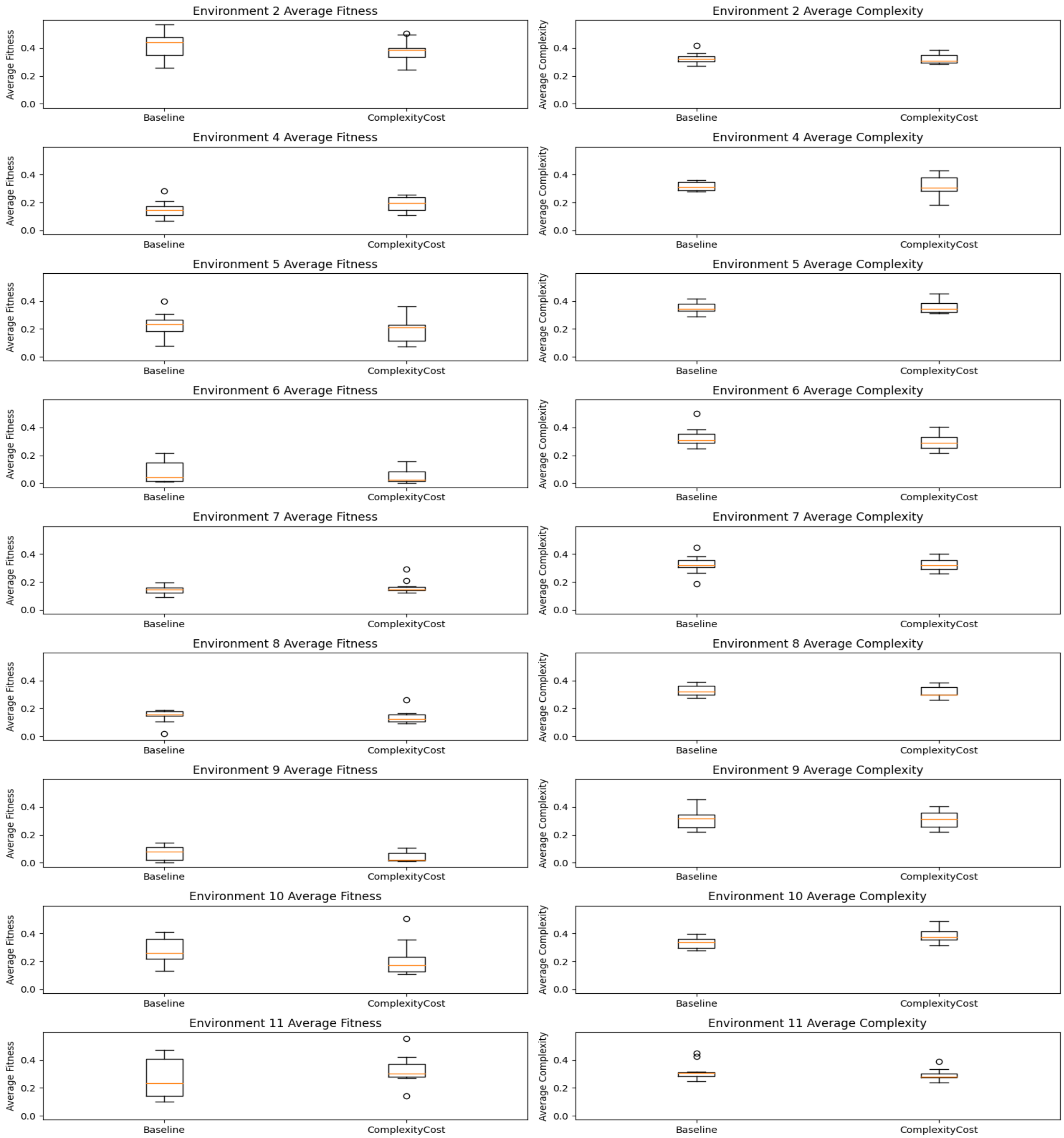


Figure 9: Row 1, left: Average fitness of the population evolved in Environment 2 in Baseline versus ComplexityCost. **Row1, right:** Average complexity of individuals in Baseline versus ComplexityCost in Envrionment 1. **Rows 2-9, left and right:** As above, for Environments 4-12 respectively. See table 1 for environmental complexity parameters.

7.5 Average Maximum Complexity and Fitness

Figure 10: Row 1, left: Average maximum fitness of the population evolved in Environment 1 in Baseline versus ComplexityCost. **Row1, right:** Average maximum complexity of individuals in Baseline versus ComplexityCost in Envrionment 1. **Rows 2-9, left and right:** As above, for Environments 2-9 respectively. See table 1 for environmental complexity parameters.

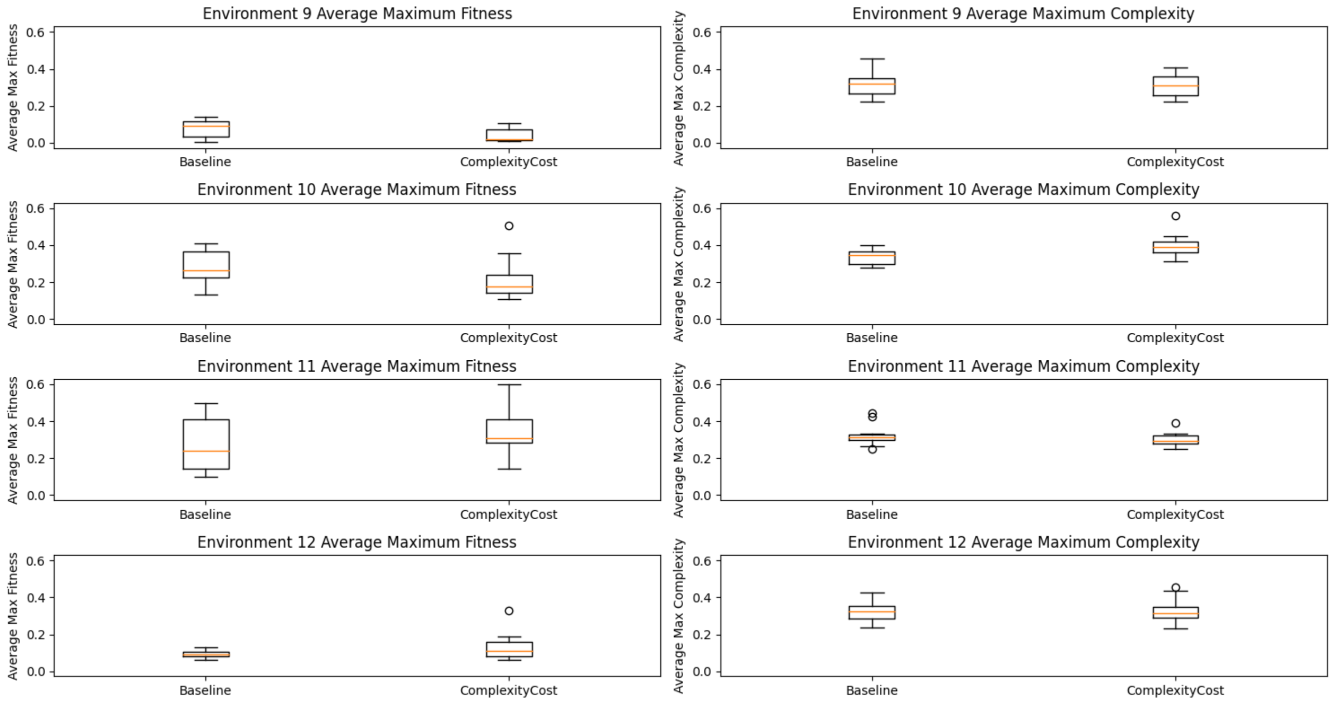


Figure 11: Row 1, left: Average maximum fitness of the population evolved in Environment 9 in Baseline versus ComplexityCost. **Row1, right:** Average maximum complexity of individuals in Baseline versus ComplexityCost in Envrionment 9. **Rows 2-4, left and right:** As above, for Environments 10-12 respectively. See table 1 for environmental complexity parameters.