Neuroevolution and Evolutionary Robotics

A Literature Review

Tristan Joseph
Department of Computer Science

University of Capet Town
Cape Town, South Africa
JSPTRI002@myuct.ac.za

ABSTRACT

This paper will present discuss the ideas and concepts behind Neuroevolution and Evolutionary robotics, which together provide for the ability to construct collective robotics experimental environments for scientific observation. We begin with a deconstruction of all the integral parts of Neuroevolution, firstly we will present the structural characteristics of an Artificial Neural Network and how it functions to learn information. After that we delve into its pairing with Evolutionary Algorithms to construct adaptive and incrementally evolving networks that closer resemble the human brain. The underpinnings of this research is in the theories laid out by The Social Brain Hypothesis, where it is theorised that certain brains have grown more complex due to the requirements for them to accomplish increasingly difficult tasks, so we discuss how Neuroevolution can be used to construct the variables and structures required to simulate and test these hypotheses. Certainly we show that allowing for the evolution of network complexity via the TWEANN and NEAT methods brings us closer to understanding how the human brain adapts and grows to deal with different tasks. Previous studies have shown however that the definition of complexity, specifically morphological complexity does affect the conditions for which complexity may grow or reduce when a cost on fitness has been imposed.

1 INTRODUCTION

The following will give a review of Neuroevolution by laying out it's individual working parts before concluding with a an overview of the results when combining it with Cooperative Coevolution and Multi-objective Optimisation.

2 MULTI-OBJECTIVE NEUROEVOLUTION

2.1 Neuroevolution

Neuroevolution pairs the relative strengths of Artificial Neural Networks and Evolutionary Algorithms together to develop and evolve networks in order to learn a specific task. Parameters of a network are optimised to maximise the fitness of task performance [1].

The advantage of this combination is that learning and evolution can be achieved at the same time, allowing for a more flexible definition of task performance as well as for defining network characteristics to be evolved at the same time [2].

2*.1.1 Artificial Neural Networks.*
Artificial Neural Networks (ANNs) is an adaptive system that alters and adapts its structure according to the influence that information flowing through the network has during the training and learning phase [3].

It take the brain as inspiration, where neurons or nodes are tiny cells that send information back and forth between each other in attempt to model something similar to that of the human brain. It is a computational model that is composed of interconnect processing units called nodes or neurons.

The neurons are connected via synaptic weights that represent and influence the strength of the connection between two neurons. Nodes are grouped into layers that are categorised as either input, output or hidden. Input nodes represent points of information that are being fed into the network and output nodes house the result of the networks computation given those inputs, they are connected to the external environment. Hidden layer nodes connect to other nodes but they do not directly connect to the environment. A neuron or node houses the result of a weighted sum of all incoming signals from connections to other nodes, these sums have been transform by some activation function. The activation function maps a value to some other value that gives meaning to its influence in the network. Take for example the Sigmoid activation function, which takes any value within its domain and maps it to another value in the range of 0 to 1, this can interpreted as a percentile which describes the proportion affect that something has on the network and can be used to propagate changes back for better accuracy.

There are various kinds of topologies that an ANN can have, different types have different primary advantages that are problem dependent. Two common types are feedforward, where data flows from input to output alone, or recurrent where precious activations are stored and used in sequential processes.

There are two primary methods training a network with data, those supervised and reinforcement learning. Supervised learning employs an algorithm called back propagation that take the outputs produced by feeding data through the network and calculates a margin of error with a predetermined expected result. This error is used to distribute changes backwards through the network, updating parameters and weights in order improve accuracy incrementally.

Reinforcement learning is unsupervised learning that does not have predetermined and expected results used to compare against the network. This style of learning attempts to approximate explicitly or implicitly the value of the states experienced by the agents. The aim is to favour the choice of those actions to maximise positive reinforcement received by that agent [4].

2.*1.2 Evolving a Neural Network.*

The primary objective of Neuroevolution (NE) is the breeding of ANNs in order to discover which are the most efficient at solving a particular problem. Evolutionary Algorithms (EAs) are used to iteratively build and evolve ANNs in a process that is inspired by Charles Darwin's theory of natural selection. Genomes are a data structure modelled after chromosomes that store the encoding of a solution to a particular problem. The characteristics of each genome are the specific parameters that are used for a problem.

An EA begins with an initial random population of these genomes, the population size is problem dependent and might change during optimisation. In the simplest approach, each individual in this initial population is assessed for their fitness, or in other words their level of competency at solving the given problem. At this stage, fit individuals are bred in order to produce a new generation of offspring with the expectation that their performance should be in some ways better than the previous one.

When two individual genomes are mates together, one of two operations can occur: A crossover exchanges the genes of two chromosomes and a random mutation can change part of a gene which will potentially lead to entirely new phenotype. A phenotype is the behaviour of the genome, its potential solution to the problem. Random mutation is used to limit premature convergence, which occurs when leaps in evolution are either too big or too small and lead to the discovery of a local optima that is not the global optima.

Whenever genes are crossed over and new generations is iterated upon, the algorithm converges towards a solution, sometimes a solution is reached too quickly before the entire search space can be explored and evaluated. Throwing a random mutation at the algorithm will act as a deterrence from reaching these optima too soon. Mutation is performed at a rate that is proportional in some way to population fitness.

2.*1.3 Encoding a Neural Network.*

The genome of an ANN contains two important features, /node genes/, which specify a single neuron, and /connection genes/ which specify the connection between neurons, the weight of the connection and wether that connection is enabled or not.

An initial population of networks are generated and evolved in the same way as mentioned previously. In reference to the random initial population, the fact that their performance in solving a problem is likely to be very poor is not important, what is important is that some will be better than others, and this I the driving force behind fitness evaluation in order to produce progressively better networks via breeding.

There are two approaches to encoding an ANN in EAs, they are direct and indirect. Direct encoding describes every part of the network by some gene listed in the artificial DNA.

Impressive results have been achieved using this method on smaller networks but as the size of the network grows so does the complexity and infeasibility of being able to encode the genetic strings and perform the mutations necessary in a reasonless amount of time [5].

Indirect encoding is a compressed representation that allows for the number of genes to be much fewer than the number of neurons or connections. This results in the ability to evolve more impressive and complex ANNs. The downside is that search can biased in unpredictable ways due to the fact that individuals are not mapped directly to their phenotypes [6].

2.*1.4 Complexity.*

There is a problem with the typical structure of an ANN, and that is that their topologies are fixed. The weight parameters are encoded from birth and locked beyond that point. This means that everything is learned through breeding and not through experience.

To make progress in NE, the complexity of an ANN needs to be evolved in order to overcome its limitations. As mentioned above, having a fixed topology does not allow for this and so the magnitude of nodes in a structure cannot grow. In reality, brains have indeed grown in complexity over time in order to accomplish increasingly difficult tasks, as is detailed by the theories laid in the Social Brain Hypothesis. Evidently, a solution is required to move this process further.

2.*1.5 Topology and Weight Evolving Neural Networks.*

Topology and Weight Evolving Neural Networks (TWEANNs) are a style of network structure and algorithm that allows for the adding of new connections or neurons which in turn results in the potential for networks to evolve and become larger and more complex. Although they are a step in the right direction, TWEANNs still not place a strong enough emphasis on evolving complexity.

There are some common problems when attempting to evolve ANNs. When two networks produce the functionality but with different characteristics and structure, it is difficult to determine how to crossover their genes for their offspring, this is referred to as competing conventions.

When elitism is employed, which is choosing the fittest of every generation, premature convergence occurs as mentioned previously. In this scenario, networks or approaches to a solution are eliminated before they have the chance to mature and potentially succeed [7].

Algorithms have been developed to try and combat this, these include SANE and ESP, which are an example of symbiotic evolution. The idea is that neurons within a network can be evolved in their own subpopulations in order to cooperate with other neurons. These neurons are grouped incrementally with those from other subpopulations to progressively form a full working network. These approaches favour both cooperation and specialisation [8].

2.*1.6 Neuroevolution of Augmenting Topologies.*

Neuroevolution of Augmenting Topologies (NEAT) is a style of reinforcement learning algorithm that allows for the ability to solve increasingly complex ANNs. In contrast to TWEANNs, NEAT explicitly aims to evolve complexity by employing a method for evolving the architecture and weights of neural networks. A key feature is that topologies are minimised during evolution instead of at the end, this has resulted in significant gains in learning speed [9].

Three important functions of NEAT that help to solve some of the previously mentioned problems with evolving neural networks are detailed by the following:

Historical marking is a part of the connection genome that stores information regarding the ancestral history of the gene. This allows for a coherent crossover between parents and solved the competing convention. Something called an innovation number is assigned to a new genome when it is created via structural mutation. The innovation numbers of two individuals in a population are matched during crossover, they are aligned so that the elements between them that differ are exchanged in order to produce offspring.

When a new structure in introduced into the network via topological innovation, it might perform poorly which would result in it being rejected before ever receiving the time to grow and optimise. Speciation allows for innovative new structures to have the time to mature before premature elimination from the population. NEAT specialties a population by grouping them into niches. The dissimilarity of individuals is computed using historical markers and alignment during crossover to separate them into these groups. The individuals in these species now get the chance to optimise in these subpopulations before competing with the larger population.

NEAT also ensure that complexity is increased incrementally over time which solves the previously mentioned problem.

On the topic of elitism and premature convergence that was previously discussed, choosing the most fit parents based on their objective performance can be detrimental in the long run. A useful selection was developed that chooses parents based on their novelty rather than their fitness, it’s called novelty search. This means that if a given individual in a network displays behaviour that is functionally different to some others in the network but that approach currently performs poorly, it will be chosen over fitter individuals simply because of its unique approach to solving the problem, those approaches or genes could later lead to far superior solutions. Novelty search is not so much about solving a particular problem, but rather about discovering all the interesting ways that individuals in the search space might be able to solve the problem.

2.*1.7 HyperNEAT.*

HyperNEAT is an extension of NEAT that used indirect encoding. This approach has been shown to produce results that are biased towards regular solutions over irregular ones. [On the Performance of Indirect Encoding Across the Continuum of Regularity]

Composition Pattern Producing Networks (CPPNs) are the style of ANNs used in HyperNEAT. CPPNs generate the connectivity patterns in ANNs and are a way of expressing the regularities and symmetries of a pattern with a small set of genes. This approach takes another step closer to simulating the human brain, where there are numerous irregular patterns such as in the visual cortex.

2.2 Multiobjective Neuroevolution

2.*2.1 Cooperative Coevolution.*

Cooperative Coevolution is an evolutionary computation method that solves problems by breaking them down into smaller subproblems. This has been shown to give more diverse solution when compared with that of algorithms that have single populations. It specifically takes advantage of the pairing of NE and back propagation. NE has limitations in terms of training time but back propagation makes use of gradient descent to help the network converge to a solution much faster, but it does however suffer from premature convergence. Combining NE and back propagation helps to maximise the strengths and limit the weaknesses of both approaches.

2.*2.2 Multiobjective Optimisation.*

Multi-objective optimisation handles the issue of conflicting objectives, it aims to produce multiple optimal solution instead of one single global one. In this approach, no one solution can be better than any other when compared with all the objective functions.

2.*2.3 Multi-objective Cooperative Coevolution.*

Multi-objective Cooperative Coevolution (MOCC) has been used to create a series algorithms such as Non-Dominating Sorting Cooperative Coevolutionary Genetic Algorithms (NSCCGA) which compares well with NSGA-II on some benchmark functions. MOCC has been used for optimisation of larger populations as well with special niching mechanisms that make use of an extending operator in order to maintain diversity.

3 Evolutionary Robotics

Evolutionary Robotics (ER) is the creation of artificial robots to be used in scientific research and experiments in order to study and better understand the behaviours of the of complex neurological creatures such as humans. Often these robots are modelled after some neurological structure that resembles that of the human brain. This allows scientists the opportunity to study phenomena and behaviour that is often not possible or feasible to observe in the real world simply due to the time steps and evolutionary requirements.

An area of ER that is relevant to this topic is the study of task performance and behaviour in a collective group of co-adapting robots. These robots are given some definition of morphological complexity and required to perform certain tiers of progressively difficult group tasks such as moving a set of various size blocks to a marked zone. The main goal is to observe whether or not imposing a cost on complexity or fitness would result in the need for a more or less complex neurological structure in the robots. This is achieved by applying multi-objective optimisations where task performance is maximised and morphological complexity is minimised. This is compared against another population that has not had a cost imposed.

In general, results have leaned towards the hypothesis that a more complex environment will result in a great morphological complexity for the robots. However, a recent paper has shown that this is not the case, but the important take aways are that the validity of the hypothesis depends particularly on the definition of morphological complexity and that it further illuminates just how complex the conditions for certain evolution in nature can be.

4 Conclusions

Evolutionary robotics is an area of research that is rich with possibility, there has been significant work done already to help quantify and observe neurological behaviour and evolution. There are various algorithms and approaches that can be applied and all produce their own unique and interesting results. Although there are examples of good research having been done, the area can always use further illumination. We intend to closely study the results of giving increasingly complex tasks to a collective robotics group and report the findings, some parameters and environments will be altered and extended form previous experiments.

REFERENCES

[1] Miikkulainen R. (2011) Neuroevolution. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[2] Dario F, Peter D, Claudio M. (2008) Neuroevolution: from architectures to learning. In: Evol. Intel. (2008) 1:47–62

[3] Sammut C. (2011) Miikkulainen R. (2011) Neuroevolution. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA

[4] Sutton RS, Barto AG (1998) Reinforcement learning. an introduction. MIT Press, Cambridge.

[5] Floreano, Dario & Dürr, Peter & Mattiussi, Claudio. (2008). Neuroevolution: From architectures to learning. Evol Intell. 1. 10.1007/s12065-007-0002-4

[6] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. Evolutionary comp., 10(2):100, 2002.

[7] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. Evolutionary comp., 10(2):100, 2002.

[8] Moriarty DE, Miikkulainen R (1996) Efficient reinforcement learning through symbiotic evolution. Machine Learn 22:11–32

[9] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. Evolutionary comp., 10(2):99–127, 2002.