Novelty Driven Neuroevolution

David Jay

Contents

1 Introduction

It is tempting to assume what we have been told throughout our lives is unquestionable. But sometimes questioning conventional wisdom can lead to an interesting conclusion, a truly novel idea.

Objective thinking is ingrained in modern day society. The idea that setting a goal is always the best way forwards is drilled into us from a very early stage in our lives. But when we actually look at great and ambitious achievements, are they really all objective driven? Did Steve Jobs really start out life with the ambition to create a revolutionary new phone?

Achievement can be thought of as a process of discovery [Stanley and Lehman, 2015]. In this way the process of achieving something ambitious can begin to look like a type of search. In search, the presence of local optima pose problems. Can a solution be found to mitigate these local optima in the search, allowing us to achieve our ambitious objectives?

As an example of a problem that suffers from getting stuck in a local optimum, imagine a misty lake with an assortment of stepping stones dotted in the water. The mist makes it impossible for you to see anything outside of a two stone radius around you. Your goal is to try and get from one bank of the lake to the other. A traditional objective view of the problem would suggest you define a measurement that would allow you to gauge how close you are to the other bank at any time. If you try and maximise this measurement, it is very likely you will reach a dead end (which is a local optimum) instead of the other bank.

This paper will introduce a solution first proposed by [Lehman and Stanley, 2011a] known as the novelty search, that does away with an objective and searches for novelty alone. This idea will then be demonstrated with the help of neuroevolution to show how novelty-based search can be advantageous to objective-based search in many tasks.

Although seldom discussed (in some part due to the advent of deep learning), evolutionary computation, of which neuroevolution is a subfield, is making a comeback [Stanley, 2017]. Its suitability for novelty search along with the fact that it is easily parallelisable coupled with the fact that deep neuroevolution can rival deep learning [Such et al., 2017] is what makes the field so interesting at the moment.

2 Background

2.1 Evolutionary Computation

Evolutionary algorithms are a class of machine learning algorithms modelled on nature's evolutionary process. The field is one of the oldest of machine learning, having first been proposed by Alan Turing as a “learning machine” which would parallel the principles of evolution [Turing, 2009]. As such they represent an intelligent use of random search to solve optimisation problems. Although they may be randomised, evolutionary algorithms exploit historical information in order to direct the search. The basic principles followed by evolutionary algorithms adhere to Charles Darwin's idea of natural selection.

As fields within evolutionary computation share many of the same properties, a quick explanation of genetic algorithms will hopefully allow for a more intuitive understanding of neuroevolution.

2.1.1 Genetic Algorithms

As with all evolutionary algorithms, genetic algorithms intend to harness the processes observed in natural evolution to solve optimization problems. After an initial population is randomly generated, the algorithm evolves through 3 operators:

- Selection
- Crossover
- Mutation

These processes mirror what is found in natural selection. A fitness is determined as a measurement to optimise and individuals with the highest fitness are allowed to reproduce. These operators give the ability to both explore and exploit which is a key characteristic of all evolutionary algorithms. This is the main reason why genetic algorithms are used in a huge variety of optimisation tasks [1]

2.1.2 Neural Networks

The use of neural networks has exploded in popularity over the past decade. They are now almost ubiquitously used in industry with buzzwords such as deep learning being used more and more often. Neural networks are a key part of neuroevolution, allowing it to exhibit the behaviours which makes it so special. But what actually is a neural network? Artificial neural networks, simply referred to as neural networks, attempt to model neural processes using a simple model of a neuron.

Neural networks are built upon simple signal processing elements that are connected together [Berger, 2016] to form a complex and non-linear system that allow neural networks to excel at non-linear classification tasks. Neural networks are typically organised in layers (often referred to as the input layer then n hidden layers, then an output layer). These layers are made up of interconnected nodes in which each apply a mathematical function called an activation function to the input. The connections are weighted so that different nodes gain different importances. These weights are what are traditionally optimised through the learning process known as backpropagation of errors [Werbos, 1990].

[1] More on the subject of genetic algorithms can be found at [Mallawaarachchi, 2017]

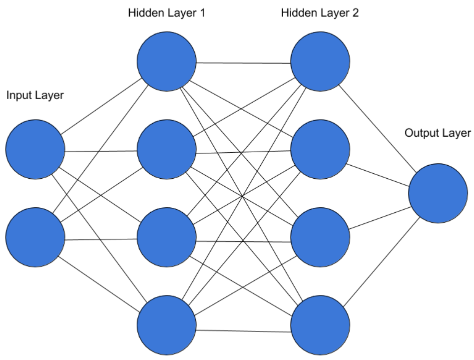

Figure 1: *An example of a simple multi-layer feedforward network.*

2.1.3 Neuroevolution

How can neural networks and evolutionary techniques be combined to try and simulate the process of evolution in our brains? The field of neuroevolution poses this question and seeks to develop the means of evolving neural networks over generations [Stanley, 2017].

Although traditional neuroevolution methods seek to offer an alternative to backpropagation and optimize the weights of a neural network using evolutionary algorithms, a persuasive argument put forward by [Gruau et al., 1996] argues that the evolving structure of the neural network in addition to the weights saves the time that is normally wasted by humans trying to decide on the topology of the network. In previous neuroevolution methods that use a fixed-topology neural network, deciding how many hidden nodes the network had was often a trial and error process. So much so that [Gomez and Miikkulainen, 1999] were able to solve the same problem 5 times faster by restarting with a random number of hidden nodes whenever the algorithm became stuck.

2.1.3.1 Neuroevolution of Augmented Topology (NEAT)

A solution to this is to evolve the topology as well as the weights and connections of the neurons through this process outlined by [Stanley and Miikkulainen, 2002]. NEAT begins its evolution process with a population of small and simple neural networks and complexifies the topology of those networks over a number of generations. This leads to increasingly sophisticated behaviour [2]. Because of its versatility, NEAT is one of the most popular neuroevolution techniques and is widely applied within the field of optimisation [Aaltonen et al., 2009; Allen and Faloutsos, 2009; SethBling, 2015].

2.2 Deception

Deception is a common problem in any search task. Take the Chinese finger trap as an example. To arrive at a solution one must push one's fingers in before they are free

[2] This section is only meant as a quick overview. A more comprehensive and technical guide can be found at [Stanley and Miikkulainen, 2002]

to move. These discrete steps can be described as *stepping stones* one of which can be described as deceptive, as it contradicts our objective. In the original conception of deception by [Goldberg and Richardson, 1987], a problem is deceptive if lower-order building blocks, when combined, do not lead to a global optimum. However this paper will define the term “deception” as intuitive measure of problem hardness. While problems such as the Chinese finger trap only have one deceptive stepping stone, most ambitious problems have many deceptive stepping stones. It turns out that when the objective is further than one stepping stone away, the objective measurement becomes a *false compass* and therefore objectives are useless even to the point of hindering progress [Lehman and Stanley, 2011a].

2.2.1 Deception in evolutionary computation

The study of deception is an important part of evolutionary computation. Researchers look for what may cause an evolutionary algorithm to fail and how to fix these failures. The more deceptive a problem, the more *rugged* the fitness functions’ landscape.

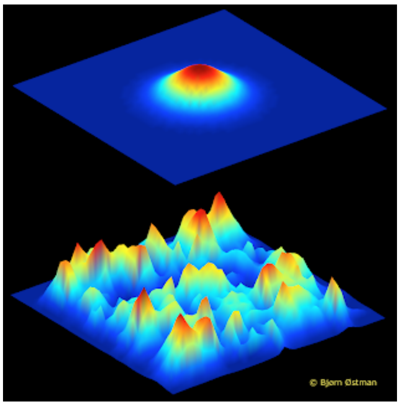

Figure 2: *Smooth vs rugged fitness landscape [Østman, 2013]*

Although it can be easy to visualise the ruggedness and therefore gauge deceptiveness, it is tricky to mitigate deception. A very common approach to prevent convergence at an unwanted local optimum in evolutionary computation is to keep diversity in the population by employing ideas from niching and speciation from natural evolution [Hutter and Legg, 2006; Shir, 2012].

2.2.2 Natural Evolution

Let us have a look at natural evolution. The idea that evolutionary lineages tend towards increasing complexity is one that is widely accepted in the evolutionary community [McShea, 1996; Heylighen, 1999] but what causes this and how can it be simulated to produce the kinds of results that evolution does? A strong argument can be made that evolution isn’t an inherently objective driven process [Stanley and Lehman, 2015] and therefore one idea is to abandon our objective view and think of evolution as a sort of stepping stone collector. Individual species are adapted in their own right but aren’t part of a path to an ultimate objective.

2.2.3 Novelty Search

A kind of forced stepping stone collector can be created by using novelty as a metric - rewarding any time that the agent find something that is different and new from what has been done previously. This approach doesn't reward stepping stones that lead to the objective, meaning that there is no fitness landscape or deceptiveness. This discovery of novel solutions which has been modelled off natural evolution, results in, like in evolution, behaviours that are ordered from less complex to more complex [Lehman and Stanley, 2011a].

The pairing of novelty search with NEAT is ideal due to the fact that during successive generations of NEAT, the behaviours that the neural network is able to model become more complex. This mirrors the ordering of the behaviours that result from a novelty metric (simple to complex).

3 The Experiment

This papers experiment utilises NEAT with both an objective-based fitness measurement and a novelty-based measurement. This novelty metric is formally defined along with the maze navigation experiment.

3.1 The Novelty Search Algorithm

The novelty metric that is used to determine the novelty of a new behaviour is unique to the domain. The idea of a novelty metric is to characterise how far away the new individual's behaviour is from its predecessors. A simple way to do this is to compute the sparseness of a point in the behaviour space. This can be done by measuring the average distance from the individual's point to the k-nearest neighbours of that point. The sparseness s is given at a point x by

$$s(x) = \frac{1}{k} \sum_{i=0}^{i} \mathrm{dist}(x, \mu) \tag{1}$$

where μ is the ith nearest neighbour of x. The distant metric is domain specific [Lehman and Stanley, 2008]. If the novelty of the new individual is sufficiently high, then the individual is entered into an archive similar to archive based approaches in coevolution [De Jong, 2004]. The archive allows for the novelty metric to have a comprehensive sample of where the search has been before. A further development of this idea is to dynamically adjust this threshold to allow for a constant flow of individuals into the archive [Gomes et al., 2015].

Evolutionary algorithms are very well suited to novelty search. In fact, implementing a novelty search just requires the replacing of the fitness function with our novelty metric. Once objective-based fitness is replaced with novelty, the NEAT algorithm operates as normal. This results in NEAT generating novel behaviours that become more complex as the simple ones get discovered first. Due to the principle of Occam's razor [Blumer et al., 1987], it is likely that a solution to our problem will occur as we ascend complexity.

3.2 The Maze Experiment

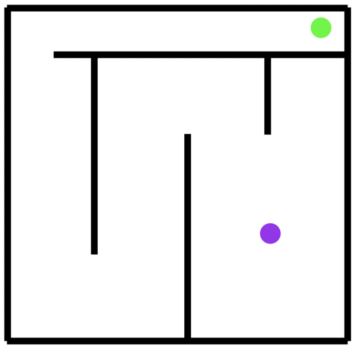

Figure 3: *A diagram of the maze used. The purple circle represents the starting point of the agent and the green represents the target point.*

This maze situation is a perfect representation of a rugged fitness landscape where the agent has to navigate through multiple local optima to reach the final objective. The agent has successfully found a solution if it gets within 5 units of the target. This property is why it acts as good demonstration of novelty-based search.

This agent, in the form of a bouncy ball, is given 4 inputs: x and y euclidean distance to the target and x and y velocity. The outputs of the network are fed into the x and y velocity of the agent.

The fitness function of the objective-based NEAT is defined by $b - d_g$ where b is a bias and d_g is the euclidean distance of the agent from the goal. The constant makes sure that all individuals will have a positive fitness. The fitness of function of the novelty-based NEAT is just defined at the sparseness of an individual at its ending point meaning that the novelty metric rewards the agent for their ending position rather than their path [Lehman and Stanley, 2011b].

To ensure a fair test, both the novelty and objective-based NEAT algorithm are given the same starting parameters. The only thing that has been changed is the fitness function that is used by the algorithm.

3.3 Results

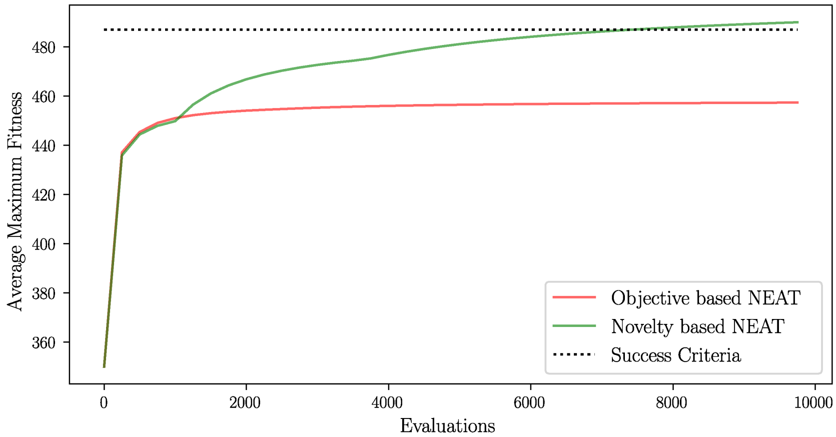

Figure 4: *Comparing novelty-based search to the objective-based search. The results are the mean from 5 runs of each algorithm.*

The novelty-based search method significantly outperformed the objective-based search as shown by figure 4. The objective-based search didn't manage to solve the problem in any of its runs while the novelty-based search method found a solution in 4 out of the 5 runs.

3.3.1 Behaviour

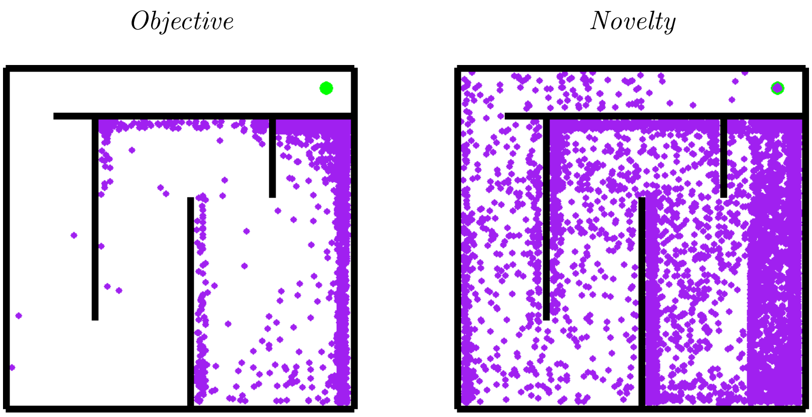

Figure 5: *Comparing the spread of behavioural points (purple) of the novelty search with those of the objective search.*

Figure 5 depicts the behaviour (represented as the last point visited by the individual) discovered during both the objective-based NEAT algorithm and the novelty-based algorithm over 10,000 evaluations. The novelty search exhibits a much wider spread of behaviours around the maze than the objective search. This is due to the fact that the objective search has fallen into a local minimum.

4 Discussion

Although the results seem counter intuitive, they only highlight the limitations of the objective driven search method rather than discrediting it a whole. Based on objective performance of the maze problem, it would normally be concluded that NEAT was not able to solve it. Yet when a novelty metric is applied to NEAT, changing the fitness metric rather than the algorithm, it is able to solve the maze problem. This is not because NEAT doesn't employ a diversity metric as it does [Stanley and Miikkulainen, 2002], even with a state of the art diversity metric [Goldberg and Richardson, 1987] NEAT is still fundamentally deceived as it seeks a higher objective-based fitness.

Of course, novelty search also faces limitations. For example, as it ignores the objective, there is no incentive for it to optimise its solution once one is found. A solution could be to take the rough solution from the novelty search and then optimise it using standard objective-based optimisation techniques, which is good at tuning approximate solutions [Michalewicz, 1996]. Due to the fact that novelty search is a much more expansive search, it is susceptible to getting lost. An example of this is if the outer walls of Figure 3 were to be removed, the search space would allow for small changes to result in a large increase in novelty [Lehman and Stanley, 2011a].

The suggestion that the concept of novelty-based goal is more effective than objective-based search in certain contexts is unexpected but the evidence suggests that it is indeed the case. Although there are limitations, applying this to search problem may reveal solutions to problems that were seen as incredibly difficult under an objective-based view. It may be time to apply these ideas to our search for achievement suggesting a sort of paradox: the best way to achieve ambitious objectives is to abandon our objectives completely [Stanley and Lehman, 2015].

5 Conclusion

This paper serves as an introduction to the idea of novelty search and shows that in certain domains, such as spatial maze solving, novelty search significantly outperforms objective search. Novelty has been suggested as an alternative to traditional objective based thinking in ambitious tasks in the everyday world [3].

Although the results presented challenge common intuition, it is important not to view them as just presenting novelty search as a "better" search methodology. Instead these results should be used to question objective-based search and explore its limits. Allowing us to move forward, past this local optimum, in our quest for search.

In challenging the assumed suitability of objective-based search an intriguing alternative has been found allowing us to rethink how we search at the most basic level. It may be time to follow suit by questioning our own assumptions and asking ourselves: why?

[3] More on this abstraction can be found in [Stanley and Lehman, 2015]

6 Bibliography

References

Aaltonen, T., Adelman, J., Akimoto, T., Albrow, M. G., González, B. A., Amerio, S., Amidei, D., Anastassov, A., Annovi, A., and Antos, J. (2009). Measurement of the top-quark mass with dilepton events selected using neuroevolution at CDF. *Physical Review Letters*, 102(15):152001.

Allen, B. and Faloutsos, P. (2009). Complex networks of simple neurons for bipedal locomotion. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference On*, pages 4457–4462. IEEE.

Berger, C. (2016). Perceptrons - the most basic form of a neural network. https://appliedgo.net/perceptron/.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Information processing letters*, 24(6):377–380.

De Jong, E. D. (2004). The incremental pareto-coevolution archive. In *Genetic and Evolutionary Computation Conference*, pages 525–536. Springer.

Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Hillsdale, NJ: Lawrence Erlbaum.

Gomes, J., Mariano, P., and Christensen, A. (2015). Devising Effective Novelty Search Algorithms: A Comprehensive Empirical Study. pages 943–950.

Gomez, F. J. and Miikkulainen, R. (1999). Solving non-Markovian control tasks with neuroevolution. In *IJCAI*, volume 99, pages 1356–1361.

Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In *Proceedings of the 1st Annual Conference on Genetic Programming*, pages 81–89. MIT Press.

Heylighen, F. (1999). The growth of structural and functional complexity during evolution. *The evolution of complexity*, pages 17–44.

Hutter, M. and Legg, S. (2006). Fitness uniform optimization. *IEEE Transactions on Evolutionary Computation*, 10(5):568–589.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX*, pages 37–56. Springer.

Mallawaarachchi, V. (2017). Introduction to Genetic Algorithms — Including Example Code. https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.

McShea, D. W. (1996). PERSPECTIVE METAZOAN COMPLEXITY AND EVOLUTION: IS THERE A TREND? *Evolution*, 50(2):477–492.

Michalewicz, Z. (1996). Evolution strategies and other methods. In *Genetic Algorithms+ Data Structures= Evolution Programs*, pages 159–177. Springer.

Østman, B. (2013). Smooth and rugged fitness landscapes.

SethBling (2015). MarI/O - Machine Learning for Video Games.

Shir, O. M. (2012). Niching in evolutionary algorithms. In *Handbook of Natural Computing*, pages 1035–1069. Springer.

Stanley, K. O. (2017). Neuroevolution: A different kind of deep learning. https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning.

Stanley, K. O. and Lehman, J. (2015). *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer.

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

Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., and Clune, J. (2017). Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. *arXiv:1712.06567 [cs]*.

Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer.

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.