NOVELEAT: Combining Novelty Search and Objective-based Search

John Dinh & David Wilikofsky

May 2012

Abstract

In this paper we describe and evaluate a new system, NOVELEAT, designed to evolve neural networks to solve problems with deceptive elements. NOVELEAT combines two approaches to robotic evolution, novelty search and an objective fitness function. Our experiments show that although NOVELEAT is impractical in non-deceptive behavior landscapes, in deceptive environments it produces different behavior patterns than objective evolution alone. With better tuning, we believe NOVELEAT could prove to be a useful tool to evolve neural networks.

1 Introduction

Neural networks are commonly trained to learn a task using an objective fitness function that assesses how well a robot performs a given task. Initially, a group of neural network are randomly assigned weights and the behaviors are assessed using the fitness function. Weights in the robot's neural networks are adjusted by comparing expected behavior to actual behavior. A Darwinian-type evolution follows where the fittest robots breed the most for the next generation. The theory behind this methodology is that eventually you will produce a robot that performs your task well.

There are cases where an objective fitness function is not good enough to evolve the fittest robot possible. Some problems are deceptive because they require a robot to temporarily decrease its fitness to later attain a better one. An objective fitness function fails to evolve the best individuals in these cases because by its nature it cannot allow the robots to lose fitness. Novelty search provides a method to avoid these pitfalls. Novelty search rewards new behavior rather than good behavior so it does not get stuck on deceptive problems and often produces an optimal solution.

Because of its nature, novelty search is not directed. Finding an optimal solution using novelty search can necessitate extremely long evolutions. Novelty search is also better suited to problems with smaller behavior spaces.

The system we designed, NOVELEAT, tries to address some of these fundamental issues with novelty search. It utilizes novelty search as a tool to find neural networks that meet an objective criteria for fitness. It stores all fit individuals until a user-specified threshold of fit robots is filled. It then uses those robots to generate a new population and continues with objective evolution from that point forward. This system is aimed at harnessing the benefits of novelty search while cutting down on unnecessary searches of behavior space.

Our paper will proceed as follows. In section 2 we will discuss previous research related to our system. In section 3 we will describe the implementation of NOVELEAT. Section 4 will describe the experimental setup we used to evaluate our system. Section 5 will present our results and analyze what they tell us about NOVELEAT. Section 6 will present the conclusions we have reached and avenues for future research.

2 Related Work

2.1 Novelty Search

One of the main components of NOVELEAT is novelty search. Developed in 2008 by Joel Lehman and Kenneth Stanley, novelty search differs from traditional evolutionary algorithms with the use of a novelty metric instead of a traditional objective fitness function [3].

Objective fitness functions often have the issue of deception. Deception is when an objective is too ambitious and the fitness landscape contains local maxima. The search is deceived by a local maximum and it stops, thinking that it has reached its goal. Additionally, traditional objective fitness functions do not reward the idea of building blocks or stepping stones to an optimal solution. Novelty search mitigates the problem of deception and deceptive fitness landscapes by eliminating the objective altogether. Instead, novelty search compares an individual's behavior with an archive of past behaviors and gives the individual a novelty value that encapsulates how new or novel its behavior is when compared to previous individuals [6]. Novel behavior is rewarded and behavior continues to grow and complexify in the search for novelty until a solution is found.

The novelty metric developed by Lehman uses a behavior space archive containing previous behaviors, rewarding behaviors in sparse areas more than those in dense areas. To calculate the sparseness, ρ, novelty search uses the average distance to the k-nearest neighbors of a particular point, x. The equation is shown below where x is the individual, ρ is the sparseness value, k is an arbitrary integer found with experimentation, and μ_i is i-th nearest neighbor of x.

$$\rho\left(x\right) = \frac{1}{k}\sum_{i=0}^{k} dist\left(x, \mu_i\right)$$

Figure 1: Equation for determining sparseness [6]

In addition to mitigating deception, novelty search is believed to mimic natural evolution which runs its course without any objective. Kenneth Stanley and Brian Woolley suggest that evolving with objectives may actually be detrimental to the search for optimal solutions. In their 2011 paper, Stanley and Woolley took evolved images from Picbreeder and then attempted to reevolve the original images with an objective fitness based evolutionary algorithm. They discovered that they were not able to reevolve the original images to the same level of sophistication as the evolved Picbreeder images [10, 8]. They concluded that having an objective pulls the search away from more interesting results.

One of the drawbacks of novelty search is its ineffectiveness in large search spaces. To address this, Lehman and Stanley examined the idea of minimal criteria in evolution [4]. Individual organisms in the real world who do not meet minimal criteria for life won't be able to reproduce nor perform basic functions. With minimal criteria novelty search, the search space is effectively pruned because individuals who do not meet the minimal criteria as defined by the researchers are not evaluated by the novelty metric at all.

Another drawback of novelty search is the question of when to stop using novelty search. To answer this question, researchers have combined novelty search with objective based search. In 2011, Lehman and Stanley proposed the idea of using novelty search along with an objective based fitness function in local competition. The idea is that novelty search would help cultivate certain ranges of behaviors and then local optimization for each species would select the best individuals from each of the different sets of behaviors. This would effectively create a diverse and strong population [5].

Also in 2011, Giuseppe Cuccu and Faustino Gomez proposed combining novelty search and an objective based fitness function into one function shown in Figure 2. ρ is a value that determines how much to weigh novelty and the objective fitness function, i is the individual, $\overline{fit}$ and $\overline{nov}$ are the respective objective and novelty functions [1].

$$\text{score}(i) = (1 - \rho) \cdot \overline{fit}\,(i) + \rho \cdot \overline{nov}\,(i)$$

Figure 2: Aggregate Fitness Function [1]

Cuccu and Gomez found that values of ρ that fall between .4 and .9 offered the best performance. Values close to 0.0 were more objective-based and values closer to 1.0 weighed novelty more heavily.

Peter Krcah and Daniel Toropila combined novelty search and an objective fitness function but applied it to the domain of robot body-brain controllers. They showed that novelty search outperforms fitness-based search in a deceptive environment. An interesting result to come out of their work is that they found no significant improvement in search performance when novelty search switched to fitness-based search [2].

The work of Jean-Baptiste Mouret in 2009 explored the possibilities of novelty search in a multi-objective evolutionary algorithm and discovered that multiobjectivism outperforms single objective novelty search at fine-tuning behaviors [7].

The contributions by these researchers show that combining novelty search with other fitness based methods is an area of evolutionary algorithms that is quite active and prompts interesting solutions to the question of overcoming deception.

2.2 NEAT

Another component of NOVELEAT is its artificial neural network. NOVELEAT's neural network is implemented with NEAT (NeuroEvolution of Augmenting Topologies), a complexifying neural network developed in 2004 by Stanley and Miikkulainen [9]. NEAT was created with the goal of solving problems that exist in traditional neural networks. According to the creators, the first problem is figuring out how to allow meaningful crossover between varying topologies. The creators of NEAT use historical markings to align genes during crossover to solve this issue. Another problem with traditional neural networks is that interesting individuals in the population may be eliminated by an unforgiving topology. For instance, a great individual might take a few generations to evolve, but a topology may not allow for the individual to take that time and eliminate it. Stanley and Miikkulainen propose the idea of speciation to protect these special groups in a population. Lastly, the creators of NEAT saw that complex topologies defined a priori in traditional neural networks may be structurally inefficient and confusing. They propose the idea of a gradually complexifying network that will only add structures when necessary. We use NEAT because the gradual complexification of the network mesh well with gradually complexifying behaviors in novelty search. The search for novelty starts out with simple behaviors and complexifies when all of the basic behaviors have been explored by the search in the same way that NEAT's structure is simple and builds as it becomes more necessary.

3 Methods

In this section we will discuss the modifications made to NEAT to create NOVELEAT.

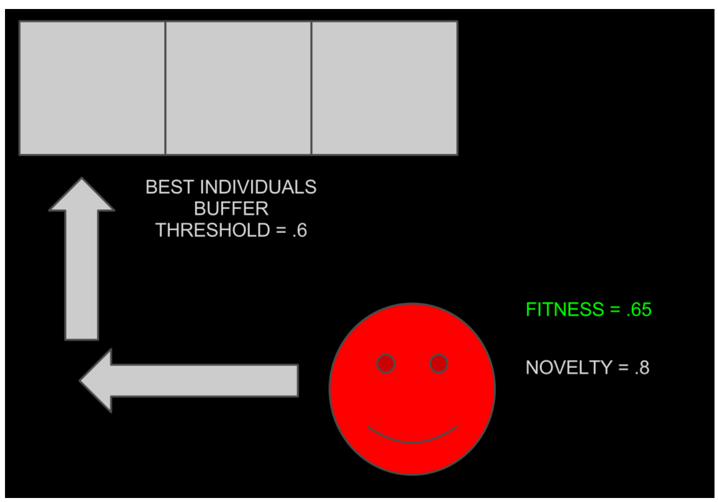

Figure 3: In this image, a fitness and novelty value for the robot have been calculated. The fitness threshold for the best individuals buffer is .6, so the robot meets the criteria with a novelty value of .65. It is added to the best individuals buffer.

NOVELEAT begins evolution using a novelty metric. The user initializes NOVELEAT with an objective fitness function, a method to assess novelty, a fitness threshold and a number of fit individuals. The system initializes a population of neural networks with random weights and a 'best individuals' buffer. It then evaluates each robot by testing it on a given task and calculating both its fitness and novelty value. If its objective fitness value is greater than or equal to the fitness threshold set by the user, the robot is placed into the 'best individuals' buffer, otherwise no action is taken (see Figure 3). Regardless of its objective fitness, every robot's novelty value is then passed into NEAT for evolution. If appropriate, the robot is also placed into the novelty archive of past behaviors as a part of novelty search.

Once the buffer of fit individuals is filled, a new population is generated from those robots. The system destroys all previous NEAT species data before creating a new population (essentially starting NEAT evolution over from scratch). First, all robots in the 'best individuals' buffer are placed into the new population. We then crossbreed and mutate the fit robots to fill out the rest of the population. The theory behind this is that even if all crossbreeding and mutation produces unfit robots, the fit robots from the 'best individuals' buffer should perform well.

After the new population is bred, evolution proceeds just as any objective evolution would using NEAT. We stop calculating novelty values and only use objective fitness values for evolution.

4 Experiments

We ran multiple experiments to assess NOVELEAT. In the following section, we describe our experimental setups. We will first explain our task and then explain the different variations of this task we designed.

All experiments were run using the Pyrobot simulator on five CS lab machines: elderberry, cardamom, cilantro, vanilla, and avocado.

4.1 Task

Because novelty search is best suited to deceptive problems, we hypothesized that NOVELEAT would be best suited to tasks that have deceptive elements to them. We designed an objective light eating task that could take place in deceptive and non-deceptive environments. A robot is placed into an environment with lights and walls. The robot is initialized with some amount of energy, and each step it takes depletes its energy by a fixed amount. Eating lights replenishes the robot's energy levels. Fitness is determined by how many steps the robot takes compared to how many steps it could have optimally taken, meaning seeking light and avoiding walls are behaviors essential to succeeding at this task. Each robot is equipped with two front sonar sensors, two light sensors, and one low energy 'warning' sensor.

For this task, NOVELEAT was set to measure novelty based on a tuple of the number of steps the robot took and the Euclidean distance it traveled. These values can be distinct; because the robot can rotate in place the number of steps it takes can be distinct from the distance it traveled.

4.2 Complexifying Environments

We altered the light eating environment in an attempt to make the task more deceptive. The original light eating task consists of a room with no walls filled with lights. We added walls into

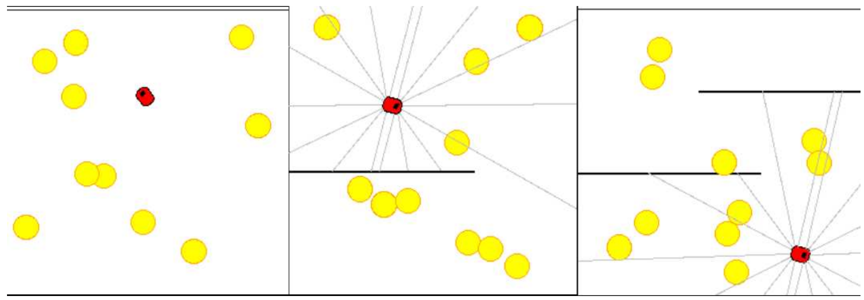

Figure 4: Left: Least Complex World, Center: One Wall World, Right: Two Wall World

the environment in an effort to make it more deceptive. By splitting the environment into several chambers, we ensure that the robot must evolve wandering behavior to fully succeed. This is necessitated because being in one chamber prevents the robot from detecting the lights in other chambers. For comparison, this task was run using NEAT with only an objective fitness function and NOVELEAT.

4.3 Best Population Generation

We played with tuning parameters used to generate the best population within NOVELEAT. These are parameters added to NEAT to create NOVELEAT. We increased the number of fit individuals needed to switch from novelty to objective evolution.We also varied the rates of mutation used on the best individuals to generate our 'best' population.

4.4 Comparing Methods

In our most complex environment (2 walls) we ran three methods of evolution: NEAT with an objective fitness function, NEAT with a fitness function aggregating novelty and objective fitness values [1], and our best known version of NOVELEAT to compare performance.

5 Results and Analysis

In this section we will describe the results of our experiments and discuss what these results indicate. Graphs generated from our experiments are not included because we believe that they are not indicative of the quality of the robots. An example of a graph generated by NOVELEAT is given in Figure 5. Instead we describe the robots' behaviors as a better approach to evaluating the robots.

5.1 Complexifying Environments

In the simplest environment, the one with no extra walls, we see no benefit to using NOVELEAT. An objective fitness function at least matches NOVELEAT's performance, which is to be expected. Without the deceptive element to the task, it makes little sense to search the behavior space with novelty. An objective fitness function can get to the desired behavior quicker.

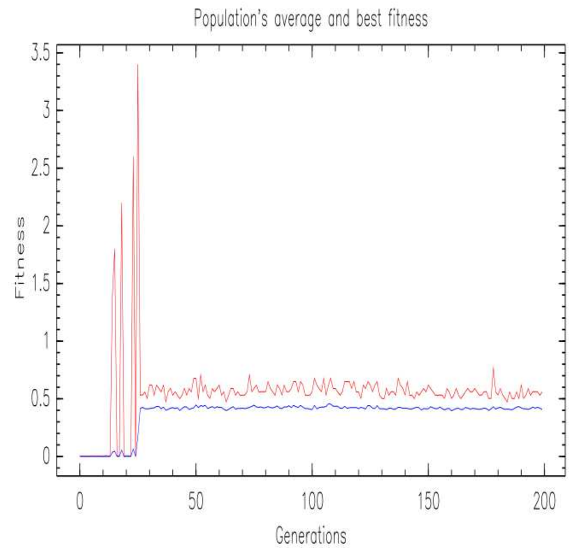

Figure 5: In this graph (from Best Population Generation, .3 experiment), there is a buffer of 5 individuals. Each time NOVELEAT finds an individual that is above the threshold, a flag is sent to the system to signify that NOVELEAT has found a decent indvidual. Each spike in this graph represents an individual added to the buffer. The lighter line (red) represents the best fitnesses and the darker line (blue) represents the average fitness for a particular generation.

As the environment becomes more complex, there appears to be no quantitative advantage to using NOVELEAT over NEAT. The evolutions of both methods yield robots with similar fitnesses. In the single wall environment both NEAT and NOVELEAT produce robots with best fitnesses of about .725 and average fitnesses of .4. The double wall environment also yields almost identical quantative results, with NOVELEAT and NEAT both producing best robots with fitness of about .68 and average fitnesses of .4.

Although the results don't appear different from a quantitative perspective, when watching the robots perform it becomes clear that NOVELEAT produces different behavior than NEAT. In the most complex environment, the NOVELEAT robots explores, often moving between the three chambers. Sometimes this wandering comes at the cost of light seeking; the robot can seem more focused on exploring the area than eating lights. The NEAT robots typically stay in one chamber, occasionally moving into a second one. Although it doesn't wander, it often eats most of the lights in a single chamber. Although the fitnesses are relatively similar, the behaviors evolved by the two methods differ dramatically

5.2 Best Population Generation

Making the rates of mutation very high (a .8 probability of adding a connection or node) and very low (a .15 probability) had little effect on producing successful individuals. Although high mutation appeared to produce individuals with slightly higher fitness (best individuals having fitness of .8 compared to .7), this quantitative difference seems to be a result of chance. Watching the best robots perform they exhibit similar behavior. Robots produced by both of these trials tend to enter one chamber and wander through it. There is little variation in their trajectory through the chamber. They often stop in front of lights and turn in a different direction, indicating they do not seek light.

A moderate rate of mutation (.3 probability, see Figure 5) showed impressive wall avoidance behavior. However, the agent showed no evidence of light seeking nor any exploratory behaviors. Individuals evolved with a moderate rate of mutation had fitness values similar to those evolved with a high rate of mutation and a low rate of mutation.

5.3 Comparing Methods

NEAT using only an objective fitness function failed to produce a robot that performs the two wall light eating task well. The best individuals had a fitness of about .65 while the average fitnesses hovered around .4. The actual behavior exhibited by the robots was erratic. Somewhat confusingly, it seemed that the robot didn't really develop any light seeking or wall avoiding behavior. Instead, the robots tended to just move backwards and forwards.

NOVELEAT produced much more interesting behavior than using solely NEAT with an objective fitness function. In both the 1-wall and 2-wall tasks, robots evolved with the NOVELEAT system exhibited wandering behavior and light seeking behavior. Though at times, the robot's behaviors could also be described as wall-following instead of wandering and light-seeking. The best fitness values achieved with NOVELEAT was between .7 and .8 in the 1-wall task and around .65 in the two world task. Average fitnesses for both tasks ranged from .39 to .36.

The aggregate technique allowed us to create a fitness metric that considered both a novelty and objective fitness function. Novelty and objective fitness were weighted and added together. We tried weighting novelty higher than objective fitness, objective fitness higher than novelty and both novelty and objective fitness having equal weights. Of these three combinations, weighting objective fitness higher produced the best results. Robots exhibited some light seeking and wandering behavior, but sometimes the robots would begin to spin around in circles even when there were lights nearby. Weighting novelty and objective fitness equally resulted in very poor behavior, where most robots just moved in a straight line into a wall. Weighting novelty higher resulted in something that looked like wall following behavior.

6 Summary and Conclusions

The results we have seen thus far from NOVELEAT seem disappointing. From the experiments we have run, it seems like there isn't a concrete benefit to using NOVELEAT over other evolutionary methods. In fact, there could be major disadvantages to using NOVELEAT. There is no guarantee when or if NOVELEAT switch from novelty to objective evolution, which could leave some evolutions completely useless. Additionally, since NOVELEAT was built as a combination of NEAT and novelty search, there are many parameters which have to be tuned for tasks. Yet despite these disadvantages, in some cases NOVELEAT produced behavior that was different than purely objective evolution. As a result, we think there are future avenues of research that could produce interesting results.

It is clear that NOVELEAT is only useful for certain types of tasks. As our experiments showed, we saw no benefit to using NOVELEAT over a purely objective evolution in non-deceptive tasks. The light task with no walls showed that NOVELEAT was both slower than objective evolution and not necessary to evolve fit robots. In some cases on the more deceptive light eating tasks NOVELEAT produced distinct behavior from NEAT. Extensive testing with all types of tasks (deceptive, non-deceptive and everything in-between) is necessary to fully understand the uses of NOVELEAT and how the behaviors it produces differ from NEAT alone.

A problem with systems like NEAT and NOVELEAT can often be tuning parameters. There are so many parameters built into the NEAT system that finding the optimal configuration can take a lot of time. It doesn't help that optimal configurations can differ based on the task. NOVELEAT only adds more parameters on top of NEAT's, leaving more room for both tuning and error. Some of our results seem promising because the behaviors that NOVELEAT produced seem different from only using NEAT; however, it is possible that we could see even better performance with a different configuration. Testing must be done on NOVELEAT to determine its optimal configuration before its potential can be fully evaluated.

NOVELEAT needs extensive testing to fully assess its benefits. The results thus far indicate that it could provide a useful method for evolving robots that solve tasks with deceptive elements, but more experimentation is necessary to draw any concrete conclusions.

Acknowledgments

We would like to thank Professor Meeden for being awesome.

References

[1] Giuseppe Cuccu and Faustino Gomez. When novelty is not enough. Technical report, IDSIA, 2011.

[2] Peter Krcah and Daniel Toropila. Combination of novelty search and fitness-based search applied to robot body-brain co-evolution. Technical report, Charles University, 2010.

[3] Joel Lehman and Kenneth Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life*, 2008.

[4] Joel Lehman and Kenneth Stanley. Revising the evolutionary computation abstraction: Minimal criteria novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2010.

[5] Joel Lehman and Kenneth Stanley. Evolving a diversity of creatures through novelty search and local competition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2011.

[6] Joel Lehman and Kenneth Stanley. *Novelty Search and the Problem with Objectives*, chapter 1. Springer, 2011.

[7] Jean-Baptiste Mouret. Novelty-based multiobjectivization. Technical report, Universite Pierre et Marie Curie, Institut des Systemes Intelligents et de Robotique, 2009.

[8] Jimmy Secretan, Nicholas Beato, David B. D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O. Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1759–1768, New York, NY, USA, 2008. ACM.

[9] Kenneth Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21, 2004.

[10] Brian Woolley and Kenneth Stanley. On the deleterious effects of a priori objectives on evolution and representation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2011.