Competition Between Objective and Novelty Search on a Deceptive Task

Billy Evers and Michael Rubayo

Abstract

It has been proposed, and is now widely accepted within use of genetic algorithms that a directly focused, objective based search is not always the most efficient way to best solve a problem or maximize a fitness score. Past work has shown that on deceptive tasks, in which local maxima are common in the search space, direct search is outperformed by novelty search, which is in essence a direct search for behavioral uniqueness instead of increased fitness. Our past efforts to recreate this result have been underwhelming, and so we have crafted what we believe is a far more deceptive task and corresponding fitness function on which to test the efficiency of objective search versus novelty search. We find that while neither method reaches a fitness peak that is higher than the other by a statistically significant margin, novelty search reaches top fitness far quicker on average. These results indicate that for tasks with built in deception, novelty search can be a more effective strategy for problem solving.

1 Introduction

In our previous lab work, we attempted to test the validity of rewarding behavioral novelty as a search algorithm. Our first experience with genetic algorithms dealt with the more intuitive direct search, as a general explanation for genetic algorithms as a whole, and stemmed from Mitchell's 1999 work, *An Introduction to Genetic Algorithms.* The underlying principles for direct search are easy to grasp: the quickest way from any point in the search space to the ideal point is a straight line. Individuals who succeed in outperforming their population peers should be the ones who natural selection rewards and are chosen to procreate, and the only measure of success should be proximity to the perfect solution as measured by a fitness function. This fitness maximization strategy felt unbreakably logical, and so we were skeptical upon reading about the apparent benefits of rewarding novelty instead of fitness in Lehman and Stanley's *Abandoning Objectives: Evolution Through the Search for Novelty Alone.* How could a search that seems to be, at its very essence, exhaustive possibly be more efficient than direct objective based search? The simple answer is deception, and we will discuss this concept in greater detail later. We were not convinced that novelty could outperform objective, but we were curious, so we attempted to create a deceptive vacuum coverage task in which agents had the task of covering certain parts of a simulated area without covering others. Our results were inconclusive and left us eager to further pursue the performance differences between direct and novelty search. In order to provide a common arena for these two strategies, we used and will again use NeuroEvolution of Augmenting Topologies (NEAT) as it is detailed by Stanley and Miikkulainen in *Competitive Coevolution through Evolutionary Complexification.* The algorithms creation of species and continued promotion of a certain quality in a population, whether it be fitness or unique behavior, will allow the two strategies to compete on an even playing field

with the same simulated task and evolution algorithm. After testing this competition, we believe that because of the deception level of our task will allow novelty search to outperform objective search.

2 Genetic Algorithms and Objective Search

In many ways, objective search mirrors the processes of Natural Selection in evolution, and as a result the loosely defined field of genetic algorithms borrows many concepts from the field of Natural Biology. Indeed, Mitchell highlights the benefits of using natural selection as a search tool, saying Evolution is a massively parallel search method: rather than work on one species at a time, evolution tests and changes millions of species in parallel [2]. In a genetic algorithm, each individual in a population is a different chromosome, each comprised of different genes which can be as simple as just binary bits, or built up to be weights and connections in a complex neural network. Individual chromosomes are tested as possible solutions to a problem and their suitability is rated by a fitness function. Organisms, which we call chromosomes, that score higher as dictated by the fitness function are more likely to reproduce and continue into the next generation. When two of these chromosomes reproduce, they are subject to more processes that mirror what occurs in evolution, the most important of which is mutation. When any two solutions reproduce their topologies and genes are brought together to form a representation of their combined strategy, but they are also subject to the possibility of a mutation occurring in which a new node or connection is added at random, according to some predetermined probability. Through this process, complex solutions can arise to tackle the problem in question. When the fitness function on which the individuals are measured is directly related to solving the task, the search is objective. In other words, the individuals who are chosen to reproduce are those who achieve the highest, although of course not necessarily a perfect, fitness score. Additionally, when a chromosome is chosen to advance to the next generation, it takes information from its previous run and attempts to better itself, tweaking the weights of connections between nodes in order to raise its fitness for the next generation. As a result, through both reproduction and self-improvement, organisms in a genetic algorithm population either strive or die off, and theoretically converge to the optimal solution over time. As Robert Axelrod demonstrated in 1987, both to evolve solutions to an interesting problem and to model evolution and coevolution in an idealized way. [2] Undoubtedly, objective search seems like the most intuitive strategy for guiding genetic algorithms towards the solution to a complicated problem.

3 Novelty Search

As early as 1990, W. Daniel Hillis had begun to decry objective search in search spaces with multiple local maxima. Hillis was unsatisfied with the solutions that his genetic algorithm evolved to optimize a search network, and wondered how it could struggle so mightily to maximize efficiency. As Mitchell explains, Hillis concluded that, the GA was getting stuck at local optimalocal "hilltops" in the fitness landscaperather than going to the globally highest hilltop. The GA found a number of moderately good (65comparison) solutions, but it could not proceed further [2]. The implementation for a novelty based search which exclusively rewards unique behavior that we will draw from was proposed in tested in 2011 by Joel Lehman and Kenneth Stanley. Because objective search, particularly when adjusting error with gradient descent, behaves in such a way that actually

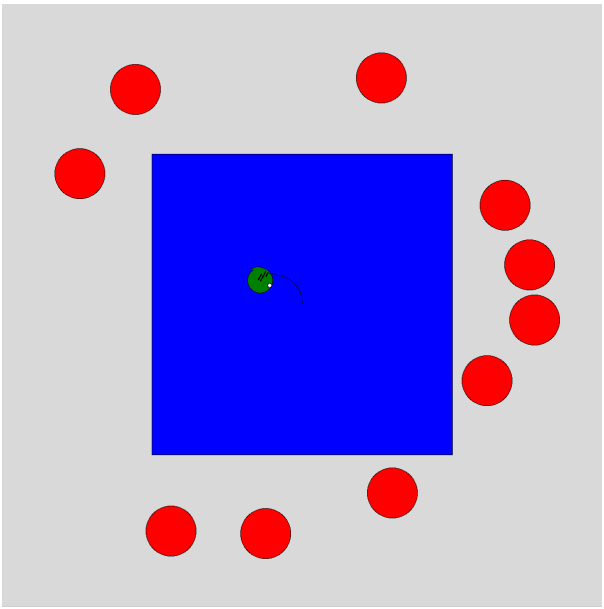


Figure 1: Each gameboard contains ten goal spaces radnomly placed throughout the world.

encourages getting stuck in these small hilltops, it can be beneficial to attack certain search spaces without an explicit objective. Problems that are particularly tricky for a direct search algorithm to solve are said to be deceptive, and the Chinese finger trap is presented as an example [1]. If the overall goal is to separate the two fingers, a direct fitness function that grades solutions based on end distance between the two fingers will reward solutions that pull the fingers apart, even though the fitness maximizing solution would be to push the fingers together in order to loosen the trap. The key insight that gives validation to novelty based search as a legitimate strategy is that, while seeking behavioral novelty will reach all possible solutions over time, novelty search is not necessarily an exhaustive search. If novelty were exhaustive, it would be no different than a linear scan of the search space, but novelty search proponents argue that task domains on their own provide sufficient constraints on the kinds of behaviors that can exist or are meaningful, without the need for further constraint from an objective function [1]. In other words, most problems, by their very nature, limit the amount of feasible solutions a search algorithm could return. With this insight in mind we know that novelty search is at least a computationally viable option, but how do we implement it? Lehman and Stanley argue that a sparseness metric should be computed for every demonstrated behavior, and solutions with sparseness greater than the solutions already existing in an archive of sparse behaviors are added to the archive. The sparseness metric can be based on a number of behavioral aspects, but a simple measure of sparseness at a point is the average distance to the k-nearest neighbors of that point, where k is a fixed parameter that is determined experimentally. If the average distance to a given points nearest neighbors is large then it is in a sparse area; it is in a dense region if the average distance is small [1] Using this measurement to quantify sparseness and therefore novelty, we can institute an evolutionary search algorithm in which those that survive are the most unique, instead of those who record the highest fitness score.

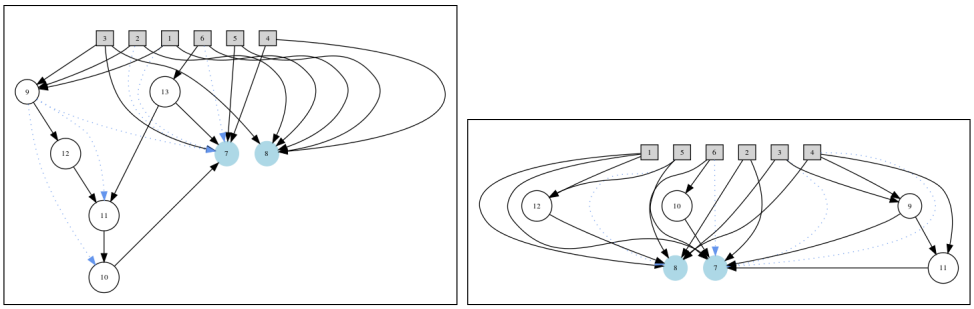


Figure 2: The neural network on the left is an example of an evolved novelty network, while the one on the right is an evolved objective network.

4 Experiments

NEAT, which as stated before stands for NeuroEvolution of Augmenting Topologies, and it is a system used to evolve a brain to optimize a certain task. It does this through the usage of an artificial neural network, or a system of nodes that fall under three types: input, hidden and output. The network takes the given input values and those values are altered and changed depending on their values through different hidden nodes, until the output node is reached and the output is decided upon. The network evolves by introducing new hidden nodes, altering the values of the hidden nodes, and removing unnecessary hidden nodes every generation to create a more optimized outcome based on the task at hand [3].

This is how the network evolves over the course of a simulation. For our experiment we had 6 input values, all normalized to be between 0 and 1, and two output values. The six input values were: inSafe (a check to see if the agent was in the safe zone), energy level, distance to wall (simulator boundary), stall (not able to move in the direction it is facing), distance to nearest goal space, and heading to nearest goal space. The outputs were the two needed to move the agent, translation and rotation. We decided to give the agent these six inputs as we felt it provided it with enough information to figure out the deceptiveness to the task, without overloading the network with too much information to process. Now for the actual implementation of NEAT in both objective and novelty searches, we decided to have a population of 100, run for 30 generations, a total of 30 unique times, for both search algorithms. This allows for a total of 90,000 unique runs being made for each search algorithm. They were both judged on the same fitness function, F= $\frac{f1+f2+f3}{3}$. Now the novelty of each run is the Euclidean distance between two points,which for our case is defined as (f1, f2, f3), as using the coordinate points does not necessarily define a unique solution in the terms of our task.

In order to create a more deceptive task to test novelty and objective search methods, we devised a task utilizing our simulator made from in-class labs to create an area 1200 by 1200 in size. Within this region, there was a centrally located square of size 600 by 600 with the four corners located at points (300, 300), (300, 900), (900, 300), and (900, 900). This center square, was colored blue. Then, around the central square was ten distinct red circles with radius 50. These ten circles represented goal spaces and their location was randomized for every run, with the only limitations being that the center point of the circle must be within a range of [100, 1100] for both the x and y values, none of the circles may overlap or touch at all with the center square, and none

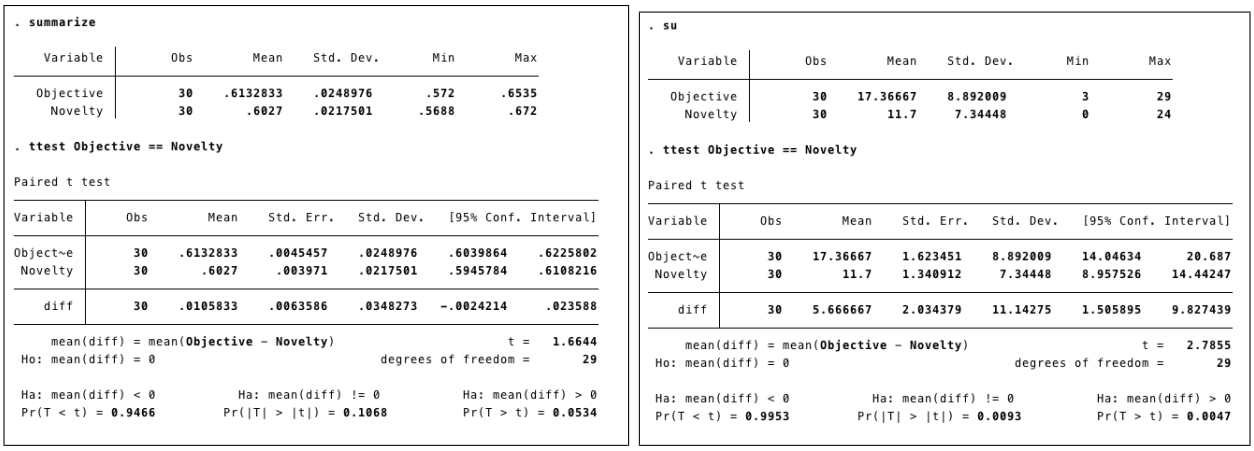
Parameter	Setting
Generations	30
Population size	100
Input nodes	6
Hidden nodes	0
Output nodes	2
Prob. to add link	0.1
Prob. to add node	0.05

Table 1: NEAT parameter settings used in the experiments.

of the circles may overlap or touch with each other. Finally, located at point (600, 600) at the start of every simulation was the agent, or autonomous aspect of the simulation. The agent, colored green, had a radius of 15, and its movement was controlled by a brain evolved through NEAT. The deceptive task that we gave the simulator was based on three parts. First, the simulation started with an energy level of 2000 that would deplenish linearly on every time step. In order to avoid the termination of the simulation, if the simulator reached the center of any of the ten red circles, which we called goal spaces, it would receive back 50 percent of the energy it had lost from the original 2000, basically replenishing the simulators life to continue on. Every time the simulator achieved this we took a goal value of 100/e, or 1 if e¡100 with e being the energy level prior to the replenishing, while removing the goal space. Once the simulation had either run out of energy or attained every goal space, the simulation ended, and we took the average of all the accrued goal values, which was the first value used within our fitness function. The second aspect was simply the percentage of time steps that the center point of the agent was located within the center square, or safe zone. The third and final aspect to our fitness function was the percentage of the ten goal spaces that were attained during the simulation. These three aspects, which we labelled f1 , f2, and f3 respectively, were then averaged out in the equation F = $\frac{f1+f2+f3}{3}$, so each aspect was a third of the overall fitness function, which was normalized to fall within [0,1]. This task was created to be deceptive, as it is physically impossible to achieve the maximum fitness value of 1, for the simulation cannot stay in the safe zone while reaching the goal spaces, creating a task where the agent has to knowingly suffer a lower fitness score in one aspect of the task to increase the other two. Also since the agent benefits from gaining the goal spaces with lower energy levels, it complexifies the deceptiveness as the agent must realize that it needs to waste time between goal spaces, while still ensuring it can reach the goal space before it runs out of energy. With these multiple layers of deception we hypothesised that the runs utilizing novelty search would achieve a higher average fitness level over the course of 30 runs, and achieve the peak fitness in an earlier generation than its counterpart, objective search, for the randomized task.

5 Results

After running 30 runs for both objective and novelty search, we see in Figure 3 that objective search actually outperforms in terms of average peak fitness across the runs. On average, the highest fitness recorded over thirty runs for objective search was .613, while for novelty it was .6027. At first glance it seems that our hypothesis was incorrect, objective search is actually more



```
. summarize

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
   Objective |         30    .6132833    .0248976        .572       .6535
     Novelty |         30      .6027    .0217501       .5688        .672

. ttest Objective == Novelty

Paired t test

------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
Object~e |      30    .6132833    .0045457    .0248976    .6039864    .6225802
 Novelty |      30      .6027     .003971    .0217501    .5945784    .6108216
---------+--------------------------------------------------------------------
    diff |      30    .0105833    .0063586    .0348273   -.0024214     .023588
------------------------------------------------------------------------------
     mean(diff) = mean(Objective - Novelty)                        t =   1.6644
 Ho: mean(diff) = 0                         degrees of freedom =       29

 Ha: mean(diff) < 0           Ha: mean(diff) != 0           Ha: mean(diff) > 0
 Pr(T < t) = 0.9466         Pr(|T| > |t|) = 0.1068          Pr(T > t) = 0.0534
```

```
. su

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
   Objective |         30    17.36667    8.892009          3         29
     Novelty |         30        11.7     7.34448          0         24

. ttest Objective == Novelty

Paired t test

------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
Object~e |      30    17.36667    1.623451    8.892009    14.04634     20.687
 Novelty |      30        11.7    1.340912     7.34448    8.957526    14.44247
---------+--------------------------------------------------------------------
    diff |      30    5.666667    2.034379    11.14275    1.505895    9.827439
------------------------------------------------------------------------------
     mean(diff) = mean(Objective - Novelty)                        t =   2.7855
 Ho: mean(diff) = 0                         degrees of freedom =       29

 Ha: mean(diff) < 0           Ha: mean(diff) != 0           Ha: mean(diff) > 0
 Pr(T < t) = 0.9953         Pr(|T| > |t|) = 0.0093          Pr(T > t) = 0.0047
```

Figure 3: On the left: The average peak fitness for both novelty and objective. On the right: The average peak fitness generation.

efficient even on a task with various local maxima. However, must check to see if the difference between the two means is statistically significant. We deliberately chose a sample size of 30 so that it would be large enough to assume normality in our data, so that we can run a t-test on the two means to indicate if our results are statistically significant. Using a two tailed t-test and assuming a null hypothesis that there is a difference between objective search and novelty search, we get a p-value of just over .1. This tells us that we cannot say with enough certainty that there is any difference between the true means of objective and novelty search average peak performances, and therefore our results are not statistically significant.

The lack of significant evidence in our performance test is disappointing, but we do see a significant difference between the two search strategies in terms of average peak fitness generation. By average peak fitness generation we mean that we use the average generation at which each strategy recorded a fitness maximum for the 30 runs as an indication of how quickly the different strategies find optimal solutions. If this metric is lower, that means that the algorithm is reaching its peak fitness at an earlier generation, and therefore more efficiently. The average peak fitness generation for novelty search is much lower than objective: 11.7 as compared to 17.367. Again running a t-test to see if this difference is significant, we can confirm with over 99 percent certainty the null hypothesis that the average peak fitness generation for novelty search is lower than for objective search.

After reaching these results, we became curious about which elements of the fitness function were being maximized over time in each of the different alogirthms. The average value of each fitness term over time is displayed in Figure 4, and the results are intuitive with our hypothesis. Objective search acts as expected, and clearly alternates between focusing primarily on f2 and f3 in order to find the optimal combination. Novelty, however, shows no inclination to maximize the individual terms, it simply pursues behavioral uniqueness. The most telling observation in this graph is negative correlation between f2 and f3 in the objective graph, as it struggles to work out a balance between the two.

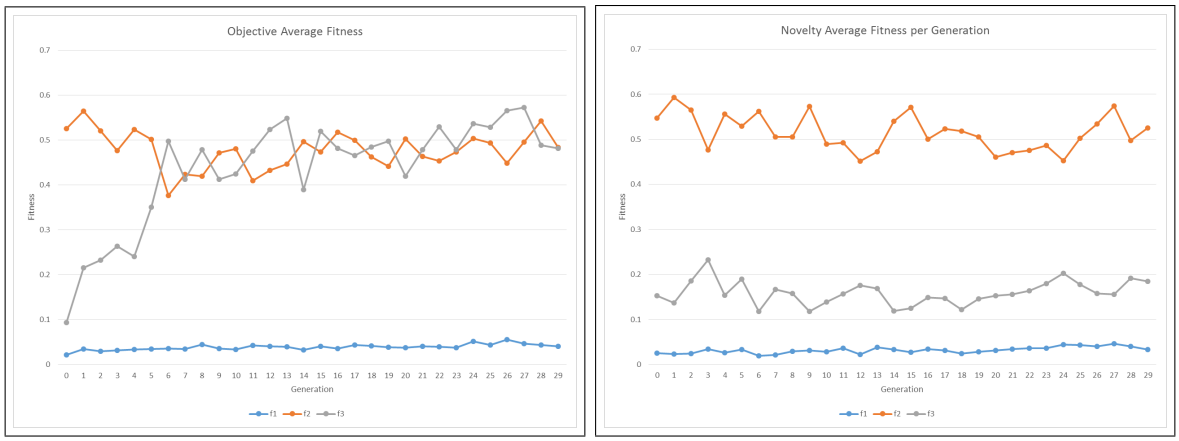


Figure 4: The average fitness term value acorss generation for novelty compared to objective.

6 Discussion

Upon first glance it seems that we erred in our hypothesis, and that novelty search offers no performance benefits over objective search. Ultimately we seek the search method that will solve the problem successfully and efficiently, and after 30 generations we see no significant difference between the best solutions provided by each strategy. We observe no significant variation in the max fitness reached between novelty search and objective search. However, efficiency of search is just as important as accuracy, and in this metric we see significant evidence that novelty search is more efficient in reaching the best solution than objective search. As we expected, because novelty search is not getting repeatedly stuck in local maximum solutions, it finds the best solution more quickly and therefore efficiently. This fits within the framework of our hypothesis, and indicates perhaps a potential flaw in our experiment. It makes sense that, given enough time, both search algorithms will reach an optimal solution. In retrospect, it is possible that our number of generations was too large in order for us to see any meaningful results in terms of average peak performance. Limiting the number of generations per run would put the two search algorithms in more of a time crunch, and the more efficient strategy would theoretically have shown significantly better results in fitness as well as generation. This is possibly an idea for future work, in which we vary the number of generations throughout the expriment. Results from this experiment could demonstrate that for a task of a given difficulty and deception level, we can estimate the amount of generations each search strategy will need. However, even with no significant difference in peak fitness, we believe that the significant difference in average peak generation is sufficient to demonstrate that novelty search is indeed more efficient than objective search when the search space is full of local maxima and the task in question is of a high level of deception.

References

[1] Joel Lehman and Kenneth Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2), 2011.

[2] Melanie Mitchell. *Genetic Algorithms: An Overview*. 1999.

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