Full Citation: Velez R, Clune J (2014) Novelty search creates robots with general skills for exploration. Proceedings of the Genetic and Evolutionary Computation Conference. 737-744

Novelty Search Creates Robots with General Skills for Exploration

Roby Velez
Evolving AI Lab
Computer Science Dept.
University of Wyoming
rvelez@uwyo.com

Jeff Clune
Evolving AI Lab
Computer Science Dept.
University of Wyoming
jeffclune@uwyo.com

ABSTRACT

Novelty Search, a new type of Evolutionary Algorithm, has shown much promise in the last few years. Instead of selecting for phenotypes that are closer to an objective, Novelty Search assigns rewards based on how different the phenotypes are from those already generated. A common criticism of Novelty Search is that it is effectively random or exhaustive search because it tries solutions in an unordered manner until a correct one is found. Its creators respond that over time Novelty Search accumulates information about the environment in the form of skills relevant to reaching uncharted territory, but to date no evidence for that hypothesis has been presented. In this paper we test that hypothesis by transferring robots evolved under Novelty Search to new environments (here, mazes) to see if the skills they've acquired generalize. Three lines of evidence support the claim that Novelty Search agents do indeed learn general exploration skills. First, robot controllers evolved via Novelty Search in one maze and then transferred to a new maze explore significantly more of the new environment than non-evolved (randomly generated) agents. Second, a Novelty Search process to solve the new mazes works significantly faster when seeded with the transferred controllers versus randomly-generated ones. Third, no significant difference exists when comparing two types of transferred agents: those evolved in the original maze under (1) Novelty Search vs. (2) a traditional, objective-based fitness function. The evidence gathered suggests that, like traditional Evolutionary Algorithms with objective-based fitness functions, Novelty Search is not a random or exhaustive search process, but instead is accumulating information about the environment, resulting in phenotypes possessing skills needed to explore their world.

Keywords

Novelty Search; Evolutionary Robotics

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
GECCO'14, July 12 - 16 2014, Vancouver, BC, Canada
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598225.

1. INTRODUCTION

Evolutionary Algorithms (EAs) perform automated optimization and are used in many different fields, such as designing pharmaceutical drugs [16], creating stock trading agents [15], evolving gaits for robots [2, 10, 3, 7], or optimizing traffic flow in congested cities [1]. Despite often outperforming human engineers [7, 9, 6], they have the same Achilles Heel all optimization methods possess: they get stuck in local optima. The reason is because EAs typically have a user-defined objective incorporated into a fitness function that rewards solutions the closer they are to that objective (or the better they perform on that objective). The problem is that often times the path from a current solution to the global optimum is not a series of moves that are increasingly closer to the objective; problems that are *deceptive* [8] require temporarily moving away from (i.e. scoring worse on) the objective to find better solutions. A nice visual example is the "Hard Maze" from [12], wherein robot controllers are evolved to go from the Start location to the Goal (Fig. 1, upper left). A traditional, objective-driven EA results in robots stuck in the dead end just below the Goal [12]. Such local optima are present in all challenging problems [5].

Novelty Search is an EA variant that is less susceptible to local optima because it entirely ignores the objective, except as a stopping criteria [12]. It rewards agents based on how different their behaviors are from the solutions that have already been evolved, which introduces a pressure to continuously do something new. In the Hard Maze example, controllers are rewarded for reaching not-yet-explored areas of the maze [12]. By exploring all paths, Novelty Search agents solve the maze significantly more often than objective-driven EA agents [12], which tend to get stuck in the dead end just below the Goal.

Disregarding objectives in an EA is a radical idea, leading to skepticism regarding Novelty Search's ability to scale up to more challenging problem domains [4]. Because it is too easy to be novel in high-dimensional spaces, and because those spaces are so large that randomly stumbling upon the objective is exceedingly unlikely, a common objection posed to the creators of Novelty Search is that it will "simply get lost in [a] vast search space" [11].

When considering the objection that Novelty Search will get lost in a vast search space, it is important to keep in mind the distinction between the genome search space, which is usually high-dimensional, and the *behavior space*, which is usually low-dimensional by design. The behavior space, which is where the Novelty Search algorithm searches, is

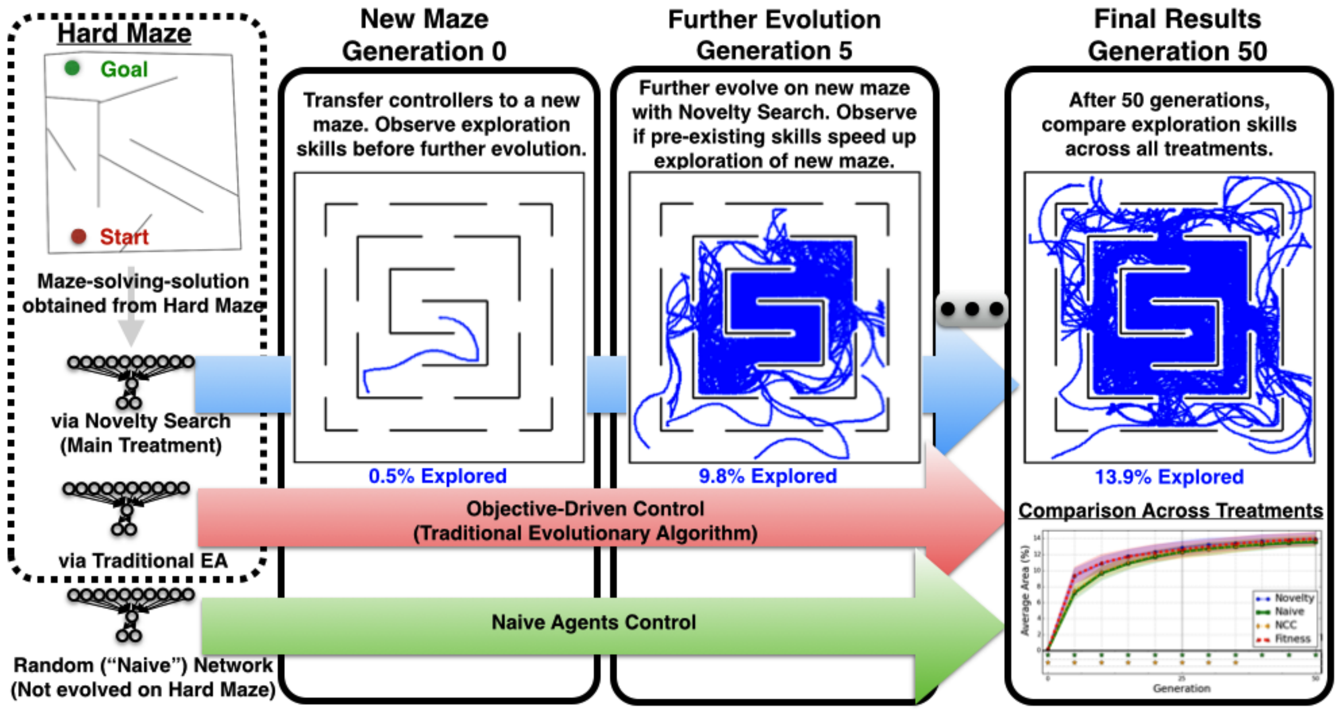


Figure 1: Experiments Designed to Test if Novelty Search Produces Exploration Skills. In the main Novelty Search treatment (blue/highest arrow), networks that solve the Hard Maze are obtained via Novelty Search. These networks are transferred into a new maze, where their initial exploration abilities are measured by quantifying the percentage of maze covered and the average distance (not pictured) from Start. The networks are then further evolved with Novelty Search to see if skills exist that speed up exploration under Novelty Search. This further evolution is stopped after 50 generations. This procedure is repeated for three control treatments: (1) a traditional, objective-driven evolutionary algorithm (middle/red arrow), (2) Naive (randomly generated) networks not evolved on the hard maze (lower/green arrow), and (3) Naive networks with the same topological complexity as Novelty Search networks (not pictured).

a user-defined description of behaviors. For example, the behavior of a robot could be described as the final location of a robot, the levers it presses, or the words it speaks. The behavior space conflates many possible genomes, phenotypes (e.g. neural networks), and actions: For example, a behavioral description might ignore an 8-legged robot's genome, the weights and topology of its neural network, and its motor commands over time, to characterize that agent just by its final location.

Novelty Search might indeed get lost in high-dimensional genome spaces, because changes in the genome due to mutations or crossover would produce new behaviors in nearly every case. However, it is not clear *a priori* that Novelty Search will get lost in low-dimensional behavior spaces, because it is non-trivial to generate new behaviors in such spaces. For example, most mutations to neural networks do not produce interesting new behaviors.

With these concepts in mind, we can return to the original objection that Novelty Search will get lost in high-dimensional search spaces. The response given by the creators of Novelty Search, is that it does not explore the search space in a random or unordered manner. Instead, to continuously produce novel behaviors, Novelty Search agents have to learn about the environment they are in and how to interact with it. As Lehman and Stanley explain:

> ...in a maze with walls, at first novel behaviors will simply crash into nearby walls, yet eventually the search must find a behavior that avoids walls to do something new. Then, once it exploits this new ability within the initial hallways, it will have to learn to go through doors to do something new. In effect, it is forced to accumulate information about the world (e.g. walls and doors) to produce novel behaviors. [11]

Over time there is constant pressure to explore even more of the environment, creating an incentive to learn general exploration skills. A key argument for Novelty Search is thus that Novelty Search causes agents to accumulate skills for exploring their environment. This paper tests that hypothesis and finds evidence that supports it.

2. NOVELTY SEARCH

Any evolutionary algorithm can be modified to perform Novelty Search by replacing the objective-based fitness function with a *novelty metric* and implementing an archive [12]. The archive contains a history of where the search has explored to prevent backtracking. In the initial Novelty Search software [11], which we conducted our experiments in, Novelty Search was implemented by replacing the objective-driven measure for the novelty metric in the fitness function of the neuroevolution of augmenting topologies (NEAT) algorithm [14]. NEAT starts with a simple neural network and adds complexity in the form of nodes and connections over evolutionary time. It also has a mechanism to promote diversity and an intelligent way of performing crossover. We made minor modifications to the original software to allow transferring maze-solving robots to new mazes. Our code and data are available at EvolvingAI.com. All parameters are identical to those in the original Novelty Search paper [12].

A traditional fitness function scores solutions based on how close they are to an objective. A novelty metric rates solutions based on how different or novel their behavior is from the current population and archive. Each robot's actions can be encoded as a vector of numbers representing important aspects of its behavior. This vector is a point in the space of all possible behaviors known as the *behavioral space*. For each agent the novelty metric computes the average distance between it and its k nearest neighbors in the behavioral space. An individual that has large distances to its neighbors is thus in an unexplored area of behavior space and is highly rewarded because it is performing a new behavior. Individuals in densely populated regions within the behavioral space have small average distances to neighbors and are less likely to be selected. In essence, the novelty metric encourages agents to explore and continuously perform new actions.

The novelty metric is computed as follows:

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

where $dist$ is the Euclidean distance between individual x and its ith-nearest neighbor, μ_i [12].

3. METHODS

3.1 Problem Domain

The problem domain in this study is from the papers that introduce and highlight the benefits of Novelty Search: evolving wheeled robots to solve mazes [12]. The robots are controlled by artificial neural networks (ANNs). Because Novelty Search is based on NEAT, the initial population of ANNs are generated by spawning a simple, feedforward neural network. The spawning process takes the initial neural network and makes copies, each with weights drawn from a uniform, allowable range of [-1,1]. The difference between Novelty Search and NEAT is that the initial starting network, and subsequent spawned networks, start off with one hidden neuron [12] versus none [14].

In this paper we evolve robots in the "Hard Maze" from the original Novelty Search papers [12] (Fig. 1, upper left). In the Hard Maze, robots must navigate from the Start to the Goal. Each robot's behavior is represented by a two-dimensional vector holding the final x and y position of the agent. The novelty metric therefore computes the average distance, within the behavioral space, from an agent to the k closest neighbors in the population and archive. Robots are thus rewarded to explore the maze.

We conduct 50 runs of Novelty Search in the Hard Maze (Fig. 1, upper left); each run produced one *hard-maze-solving* robot, which was the first to solve the maze by reaching its finish. To test if these solutions have general exploration skills or, alternately, if they have simply memorized a trajectory that solves the Hard Maze, we transfer them to a new maze and study their exploration abilities (Fig. 1).

Our initial test for exploration skills consists of two parts. First we transfer agents into a new maze and observe their baseline exploratory abilities. Second, we further evolve the transferred agents with Novelty Search to observe the progression of their abilities over time. Two measures, Area Covered and Distance Traveled, quantify the agents' exploratory skills. For this test there is no target in the new maze and evolution is stopped after 50 generations (a predetermined number that preliminary experiments revealed to be roughly when performance stops improving).

For an additional test of exploratory skills, we evolve agents in the new maze with different Goal locations distributed around the Start position (Fig. 2). Each target location approximates conducting an experiment in a different maze, increasing the sample size from which we can draw conclusions about the abilities of transferred agents to repurpose their exploration skills in new environments. As in the original experiment [12], each run stops if an agent reaches the Goal or if 2000 generations pass.

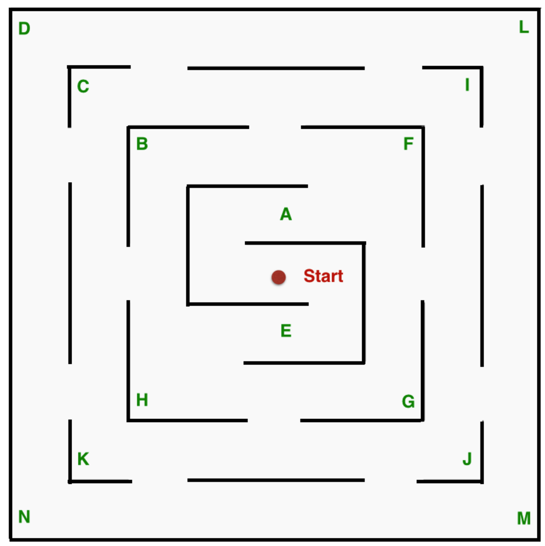


Figure 2: The Adaptation Test. The new maze, which agents are transferred into, is shown with the different Goal locations of the Adaptation Test. Novelty Search networks are transferred into this new maze and then further evolved with Novelty Search to find each Goal (in separate experiments). Each location can be considered as a separate test. If Novelty Search networks posses exploration skills, then they should find Goal locations faster than the control treatments, which are introduced in the next section.

The dimensions of the new maze, which agents are transferred into, are 450×450 (Fig. 1, 2), while those for the Hard Maze are 200×200. Since it is much larger than the Hard Maze, we refer to this new maze as the *Large Maze*. The added size provides more corridors and passageways to explore. Moreover, the Large Maze is designed so that a memorized trajectory that is successful on the Hard Maze will not lead to much exploration, since going right and up from the Start will immediately cause collisions with walls. When testing whether robots can find different Goals, the target locations are placed symmetrically around the Start position (Fig. 2) to mitigate the benefits of any memorized trajectory a transferred network may posses.

3.2 Controls

For the first control treatment, 50 randomly generated networks were obtained via the same process that generated the initial population of ANNs for the Novelty Search treatment. These *Naive* networks reveal the performance of agents that do not posses any domain knowledge or skills.

If the hard-maze-solving-networks produced by Novelty Search outperform the Naive networks, it supports the claim that Novelty Search facilitates the accumulation of exploratory skills. However, an alternate explanation exists that we would like to rule out: that the increased skill level simply results from the fact that networks evolved via Novelty Search have more complex topologies (i.e. more neurons and connections) than Naive networks. Recall that NEAT adds nodes and connections across generations and that the Novelty Search networks undergo many generations of evolution on the Hard Maze before being transferred to the new maze, whereas the Naive networks start with a single hidden neuron. To control for this explanation, we created a *network complexity control* (NCC). As with the Naive control, 50 random networks were generated. The difference is that the ith NCC network has the same number of neurons as the ith hard-maze-solving network in the Novelty treatment, where i represents the run number (for all treatments $i = 1...50$). For simplicity, the pattern of connections in the Novelty Search networks, including the presence of recurrent connections, is not mirrored in the NCC control. The initial NCC networks are all feed forward, following the convention of how initial networks are created in NEAT [14] and Novelty Search [12].

A third random control is the Random Search treatment from the original Novelty Search paper [12]: Agents were evolved with the objective-based EA, but given random fitness values. For each of 50 runs, the population evolved for E evaluations, where E is the number of evaluations it took the i^{th} Novelty Search run to solve the Hard Maze. For each run, the agent from all of these E evaluations that got closest to the Goal became the i^{th} member of the stock of initial networks for the Random Search treatment.

Along with Naive and NCC, the Random Search treatment is another interpretation of a random process. Because we experimentally found that this treatment performs qualitatively the same as Naive and NCC (data not shown), we do not separately report on it in the Results section.

In addition to random controls, it is informative to include in the comparison agents evolved in the original Hard Maze via a traditional, objective-driven evolutionary algorithm (EA). This *Fitness Control* sheds light on what skills exist in a process that is known to follow an information gradient (i.e. a search process that is clearly not random or exhaustive). Because objective-driven search is less likely to succeed on the Hard Maze, even if run for large numbers of generations [12], we had to conduct 306 runs of the objective-driven EA to obtain 50 hard-maze-solving-networks. Specifically, we run the objective-driven EA until an agent either solves the maze or 2000 generations pass. If the latter occurs, we start a new run. Each generation consists of N evaluations, where N is the population size. As in the original experiment, the population size is 250.

Each treatment has an initial stock of 50 networks. Each of these networks are used to seed the initial population for a Novelty Search process on the new maze.

3.3 Metrics for Measuring Skills

Each of the four treatments starts with an initial, distinct set of 50 networks. For each treatment the ith network is cloned N times to generate an initial population for each run R_i. The population for R_i is then transferred to the new maze to evaluate its inherent exploration ability prior to any further evolution (Fig. 1). Thus, any increased ability to explore in the Novelty Search networks vs. controls suggests the presence of general exploration skills that transferred from one maze to a different one. The R_i population is then further evolved on the new maze with Novelty Search for a predetermined number of generations. Since there is no Goal to cause an early stop to the evolutionary process in the new maze, each run terminates at the final generation, which is 50. Fig. 1 illustrates the entire process.

Every 5 generations, the exploration skills for each treatment are measured via two metrics: Area Covered and Distance Traveled. An example of the Area Covered metric is shown in Fig. 1. In this figure the ith Novelty Search network is first placed in the new maze to examine its baseline exploratory skills. As explained in the preceding paragraph, it is cloned N times to create the initial population for run R_i. The population for run R_i is then further evolved with Novelty Search for 50 generations. Every 5 generations the Area Covered by the N individuals within the current population is calculated.

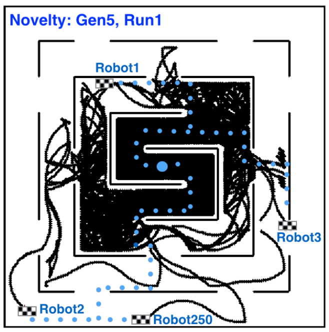
The curved paths in Fig. 1 show the trajectories of the individuals of R_i at generation g, where g goes from 0 to 50 at intervals of 5. Each trajectory is encoded in a 2×400 vector that represents the x and y positions of a robot over 400 time steps. A 450×450 matrix, initially composed of 0's, is created to represent the Large Maze. The x and y coordinates, originally floating point values, are rounded up to whole integers. For each (x, y) point within the vector trajectories a 1 is recorded on the Large Maze matrix. The percent of 1's to 0's within the Large Maze gives a measure of the area (A_i) covered by the N individuals in the current population for run R_i at generation g. Once the A_i values for each run are calculated the median is obtained as the measure for generation g.

The Distance Traveled metric, like the Area Covered metric, is sampled every 5 generations. Like the Area Covered metric at generation g, the population for run R_i is examined. The Start position and final position for each individual are obtained and the length of the shortest path between these points is calculated. The distance traversed for all N individuals within the current population is averaged to produce D_i, which is the average distance traveled for R_i at generation g. The median D_i, over all 50 runs, is then obtained and used as the measure for generation g.

Figure 3 illustrates the procedure for obtaining the length of the shortest path for four example individuals from R_1 at generation 5. The curved lines represent the robots' trajectories and the dotted lines are the shortest path from the Start position to the final location for these agents. Each agent receives a distance score that is then averaged across all the individuals in the population to produce a single value (A_i) for the run. The A* algorithm is used to measure the shortest path. The length of the shortest path is used instead of the length of the robot trajectory, because the robot could simply go in circles within a small section of the maze. The purpose of the Distance Traveled metric is to see how far away from the Start location the agents get. Since getting away from the Start location involves avoiding collisions and navigating corridors, it is a measure of a robot's exploratory skills.

Length of paths from start location (circle) to end location (checkered flags) for each individual. Distances averaged to produce a single value for each run.

	Length
Robot1	14.4
Robot2	28.3
Robot3	17.8
. . .	
Robot250	24.6
Average	



Paths are obtained via an A* algorithm. Maze and all locations were scaled from 450x450 to 90x90 to ease computation.

Figure 3: The Distance Covered Metric. Many robot trajectories (curved lines) are shown. The dotted lines represent the shortest path from the Start location to the final location (via the A* algorithm [13]) for four of the individuals within the population. A* calculates the distance traveled from Start to end so as not to count inefficient loops, while still respecting the walls of the maze (which a straight-line distance would not do). The distance values for all the individuals within the population are averaged to produce a single distance measure for this run.

Lastly to make A* computationally tractable, the maze and locations are scaled from a 450×450 maze to a 90×90 maze. This means that a matrix representing the Large Maze (0's for open space and 1's for walls) is reduced by a factor of 5 ($450 \div 90 = 5$), in the same way that images are shrunk with typical image processing. The Start and final locations for each individual are also scaled by $\frac{1}{5}$.

3.4 Adaptation Test

Aside from measuring general exploratory skills, it is interesting to measure how fast Novelty Search networks can find Goals in new mazes. If the Novelty Search agents posses exploratory skills, they should solve the new tasks faster than the Naive agents, which are starting from scratch.

As in the original experiments [12], agents are placed within a maze and evolved until they find the Goal location or a set number of evaluations elapses. Instead of evolving the treatments on a few different mazes, each with its own Goal, we conducted 14 separate experiments that involved evolving agents on the Large Maze with 14 different Goal locations (Fig. 2). The networks for each treatment begin evolution in the Large Maze for a particular Goal in the same state as they were when transferred from the Hard Maze. In other words, the results for one location do not affect the results for another. The 14 different locations allows us to generate a large sample size of data.

4. RESULTS AND DISCUSSION

4.1 Novelty Search vs. Naive

The results provide three lines of evidence showing that networks evolved with Novelty Search have greater exploration skills than Naive networks.

Second, when further evolved with Novelty Search on the new maze, Novelty Search networks explore significantly more area than Naive networks at every sampled generation (Fig. 6, Left) and travel significantly further than Naive networks in early generations (Fig. 6, Right). These results suggest that Novelty Search on the Hard Maze produces a skill set that gives networks in the Novelty Search treatment a head start for exploring the new maze. While that head start makes a significant difference for many generations, eventually the gap shrinks, suggesting that over time the Naive networks accumulate their own exploratory skills from Novelty Search and begin to catch up in terms of exploration abilities.

Third, when Novelty Search is given a specific target Goal, as in the original experiments [12], Novelty Search networks reach that Goal significantly faster ($p < 0.05$) than Naive networks for a majority of the locations: Figure 7 shows the median number of evaluations until a solution was found, for all the treatments, across 14 different Goal locations within the new maze. For this task a smaller number of evaluations is better (i.e. lower numbers on the plot are better). The Large Maze, with all the Goal locations, is shown in Figure 2 and as the inset of Figure 7.

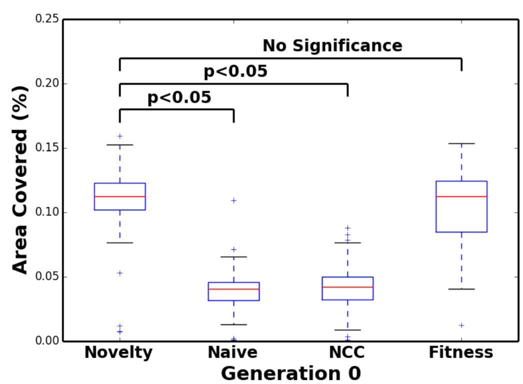


Figure 4: Area Covered on the New Maze Before Further Evolution. The data plotted show the percent area of the maze covered by the networks for the four treatments when they are evaluated in the Large Maze, prior to any further evolution on the Large Maze. Novelty Search networks explore significantly more of the maze than Naive or NCC networks. There is no significant difference in the area covered between Novelty Search networks and those evolved via a traditional, objective-driven evolutionary algorithm.

First, without further evolution, Novelty Search networks cover a significantly larger region than the Naive control (Figs. 4 and 5). Because no further evolution has occurred, and because a memorized trajectory cannot produce the diversity of paths and trajectories observed (Fig. 5), the evidence supports the claim that Novelty Search produces *general* exploration skills during evolution on the Hard Maze.

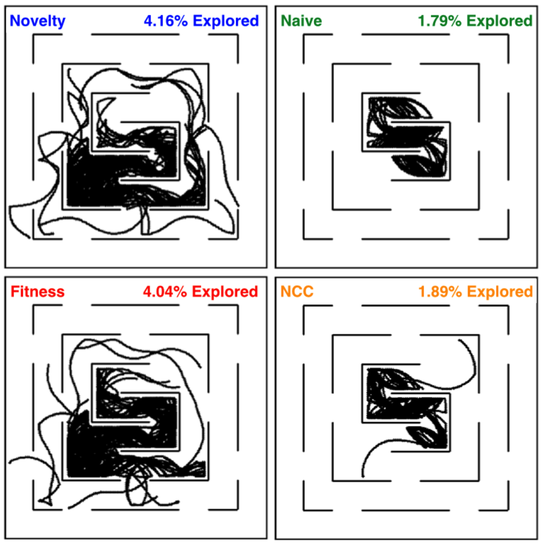


Figure 5: Paths Traveled on the New Maze Prior to Further Evolution. The networks for each treatment are run on the Large Maze without further evolution to evaluate their ability to explore a new environment. Novelty Search and Fitness networks demonstrate the capability to explore more areas of the maze, while Naive and NCC networks tend to stick to the inner sections.

The different target locations are arranged on the bottom axis from left to right, by innermost to outermost. The innermost locations (A and E) are on the far left, and the outermost locations (D, N, M, and L) are on the far right. The results reveal that the further the Goal location is from the Start position, the more evaluations are required to reach the Goal. Consistent with results described above, the longer the evolutionary process takes, the more likely the Naive treatment is to catch up to the Novelty Search treatment: Three of the four Goal locations that do not show a significant difference between Novelty Search networks and Naive networks are at the furthest reaches of the maze. The other Goal without a significant performance difference is near the Start, where substantial exploration skills are not needed to reach it.

4.2 NCC

The Novelty Search treatment significantly outperforms the NCC treatment in all but the last three sampled generations for the Area Covered metric (Fig. 6, Left), all but the last five sampled generations for the Distance Traveled metric (Fig. 6, Right), and all but three locations in the Goal tests in the new maze (Fig. 7). Novelty Search also explored a significantly larger area in the pre-evolution case (Fig. 4). That NCC networks significantly underperform Novelty Search networks is evidence that simply having more neurons and neural connections is not what gives Novelty Search networks the advantage over Naive networks.

Looking closely at the Area metric, Distance metric, and Adaptation tests it can be seen that NCC slightly outperforms Naive. This difference is only significant in generation 0 and generation 5 for the Distance metric, and location H and B in the Adaptation test (significance not shown on plots). While adding more neurons is not enough to bump up NCC's performance so that it is on par with Novelty it does seem to give a slight boost in comparison to Naive.

4.3 Novelty Search vs. Objective-Driven EA

There is no significant difference between the Novelty Search and Fitness treatments in regard to the Area Covered metric (Fig. 6, Left), initial pre-evolution test (Fig. 4), or Adaptation test (Fig. 7). For the Distance Traveled metric, Novelty Search networks significantly outperform Fitness networks at generation 10 and generation 45 (Fig. 6, Right). Since this only happens in two, non-consecutive sampled generations, it is difficult to assign any real weight to the difference. In aggregate, the data suggest that the Fitness and Novelty Search treatments are qualitatively the same (Figs. 4, 6, 7).

The Fitness treatment was created with a traditional, objective-based EA, which is regarded as an algorithm that is non-random and follows an information gradient. That the Novelty Search and Fitness treatments perform equally suggests that Novelty Search is not a random or exhaustive search, but is also encoding information about the world within its evolved solutions.

5. FUTURE WORK

The evidence presented in this paper supports the claim that Novelty Search is not random, but instead instills information (skills) in its solutions. To further test that claim, in future work we will conduct additional tests with new types of mazes. The different types of mazes (e.g. those with curved walls, narrow corridors, and other variations) will be tried both as new mazes to transfer into, new mazes to transfer from, or both. Additionally, we will conduct tests in other problem domains, including those where Novelty Search has already been shown to perform well, such as biped locomotion [12].

Another subject we will explore further is the relationship between the difficulty of domains and the exploration skills that Novelty Search develops. We hypothesize that on more difficult domains that require more skills to explore, Novelty Search will instill much more knowledge that will produce increased exploration abilities in new environments where those skills are also helpful. For example, if a robot was evolved in a house and learned to open doors, go up and down stairs, turn on faucets, etc., then those skills would also likely transfer to office buildings and possibly even to different environments like forests (where stair climbing could provide basic skills that could help on rugged, hilly terrain).

A final interesting subject we plan to study is the difference between skills acquired via Novelty Search and Fitness. It is surprising how similar the exploration skills were for networks evolved under these two very different algorithms. We plan to test whether this similarity holds, or whether there are types of problems on which the skills produced by the two algorithms differ.

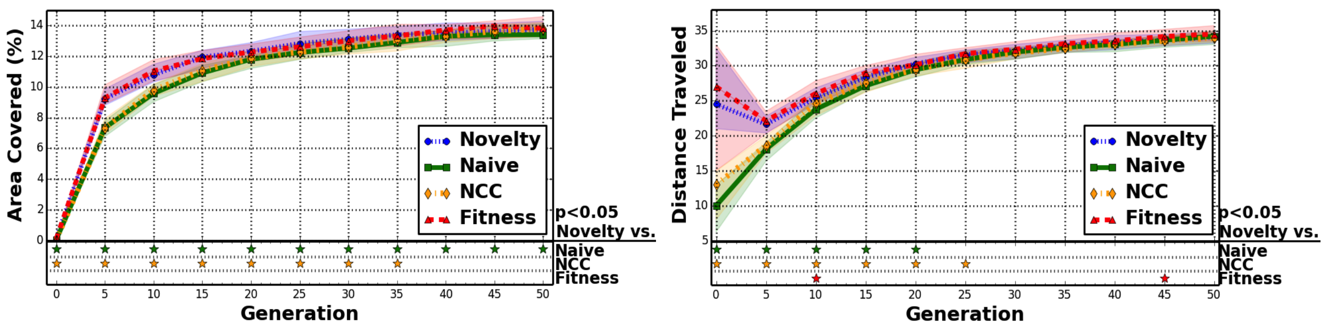


Figure 6: Novelty Search Networks Cover More Area and Travel Further Than Naive Networks. Left: Median area covered by treatments. Both the Novelty Search and Fitness treatments start off exploring more of the maze than Naive or NCC treatments. The Novelty Search treatment is significantly different than Naive for all sampled generations, and significantly different from NCC for all but the last three sampled generations. Right: Median average distance from the Start location. Both Novelty and Fitness start off traveling further than Naive and NCC. Novelty is significantly different from NCC and Naive for about half of the first sampled generations. Note the drop in distance traveled at generation 5 for the networks originally evolved with Novelty Search and the objective-driven EA (Fitness). This drop occurs because areas near the Start have not been explored yet early on, causing a temporary selective pressure to shed skills for traveling far: Recall that all treatments are being further evolved under Novelty Search, which rewards getting to all not-yet-explored places, including those near the Start. Confidence intervals represent $75th$ and $25th$ percentiles. Data are plotted every 5 generations. Asterisks indicate a significant difference ($p < 0.05$) between Novelty Search and the corresponding treatment.

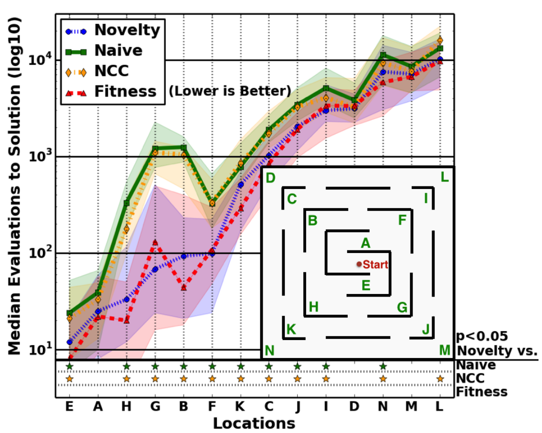


Figure 7: Adaptation Test. Median of number of evaluations for each treatment at the various locations within the Large Maze. Low values are good because it took less time for the treatment to solve the new tasks. For all locations Novelty Search solved the new locations faster than Naive. The different is also significant for all locations except A, D, M, and L. Confidence intervals represent $75th$ and $25th$ percentile.Asterisks indicate a significant difference ($p < 0.05$) between Novelty Search and the corresponding treatment.

6. CONCLUSIONS

The impetus for this study stems from concerns in regards to Novelty Search, which is an algorithm whose usefulness and popularity is slowly growing within the Evolutionary Algorithm community. Many researchers object to Novelty Search's abandonment of the objective. They argue that without a target to direct the evolutionary process, the algorithm is simply either a random search or an exhaustive search. The response by Novelty Search's creators (Lehman and Stanley), which we examine in this study, is that Novelty Search does not search randomly (or exhaustively), but instead that information (skills) relating to exploring the environment accumulate within a population evolved under Novelty Search [11]. If true, the resulting Novelty Search solutions should exhibit exploration skills that generalize to new environments where such skills are useful.

To test this claim we took successful robot controllers evolved with Novelty Search on one maze and transferred them to a different maze. We observed their general exploratory skills in the new maze. When we compared their performance to randomly generated robot controllers, we found a significant increased ability to explore in networks evolved under Novelty Search, suggesting that such agents had indeed acquired some general exploration skills.

Moreover, we compared agents evolved under Novelty Search to those evolved under an algorithm known not to be exhaustive or random, a traditional evolutionary algorithm. We found no significant difference between the two types of controllers, further suggesting that Novelty Search is following an information gradient that encodes information about the environment into its phenotypes.

Overall, our results rebut the argument that Novelty Search is similar to an exhaustive or random search process. Instead, as its creators argue, it is an interesting variant of an evolutionary algorithm well worth continued investigation.

7. ACKNOWLEDGMENTS

We thank Jean-Baptiste Mouret and the members of the Evolving AI Lab, specifically Lucas Helms, Henok Sahilu Mengistu, and Spencer Sharpe.

8. REFERENCES

[1] H. Ceylan and M. G. Bell. Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B: Methodological*, 38(4):329–342, May 2004.

[2] J. Clune, B. Beckmann, C. Ofria, and R. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2764–2771, 2009.

[3] J. Clune, K. Stanley, R. Pennock, and C. Ofria. On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(4):346–367, 2011.

[4] G. Cuccu and F. Gomez. When novelty is not enough. In *Applications of Evolutionary Computation*, pages 234–243. Springer, 2011.

[5] D. Floreano and C. Mattiussi. *Bio-inspired artificial intelligence: theories, methods, and technologies*. The MIT Press, 2008.

[6] G. Hornby, J. Lohn, and D. Linden. Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission. *Evolutionary Computation*, 19(1):1–23, 2011.

[7] G. Hornby, S. Takamura, T. Yamamoto, and M. Fujita. Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402–410, 2005.

[8] S. Kistemaker and S. Whiteson. Critical factors in the performance of novelty search. In *GECCO*, pages 965–972, 2011.

[9] J. Koza. *Genetic programming IV: Routine human-competitive machine intelligence*. Kluwer, 2003.

[10] S. Lee, J. Yosinski, K. Glette, H. Lipson, and J. Clune. Evolving gaits for physical robots with the hyperneat generative encoding: the benefits of simulation. In *Applications of Evolutionary Computing*. Springer, 2013.

[11] J. Lehman and K. Stanley. Frequently asked questions. http://eplex.cs.ucf.edu/noveltysearch/userspage. Accessed: 2014-01-02.

[12] J. Lehman and K. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[13] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, 1995.

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

[15] H. Subramanian, S. Ramamoorthy, P. Stone, and B. J. Kuipers. Designing safe, profitable automated stock trading agents using evolutionary algorithms. *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06*, page 1777, 2006.

[16] V. Venkatsubramanian. Computer-aided molecular design using neural networks and genetic algorithms. *Genetic Algorithms in Molecular Modeling*, pages 1–18, 1996.