How Novelty Search Escapes the Deceptive Trap of Learning to Learn

In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009)*. New York, NY:ACM
Winner of the Best Paper Award in the Artificial Life, Evolutionary Robotics, Adaptive Behavior, Evolvable Hardware Track

Sebastian Risi
School of EECS
University of Central Florida
Orlando, FL 32816, USA
sebastian.risi@gmail.com

Sandy D. Vanderbleek
School of EECS
University of Central Florida
Orlando, FL 32816, USA
sandyv@cs.ucf.edu

Charles E. Hughes
School of EECS
University of Central Florida
Orlando, FL 32816, USA
ceh@eecs.ucf.edu

Kenneth O. Stanley
School of EECS
University of Central Florida
Orlando, FL 32816, USA
kstanley@eecs.ucf.edu

ABSTRACT

A major goal for researchers in neuroevolution is to evolve artificial neural networks (ANNs) that can *learn* during their lifetime. Such networks can adapt to changes in their environment that evolution on its own cannot anticipate. However, a profound problem with evolving adaptive systems is that if the impact of learning on the fitness of the agent is only marginal, then evolution is likely to produce individuals that do not exhibit the desired adaptive behavior. Instead, because it is easier at first to improve fitness without evolving the ability to learn, they are likely to exploit domain-dependent static (i.e. non-adaptive) heuristics. This paper proposes a way to escape the deceptive trap of static policies based on the *novelty search* algorithm, which opens up a new avenue in the evolution of adaptive systems because it can exploit the behavioral difference between learning and non-learning individuals. The main idea in novelty search is to abandon objective-based fitness and instead simply search *only* for novel behavior, which avoids deception entirely and has shown prior promising results in other domains. This paper shows that novelty search significantly outperforms fitness-based search in a tunably deceptive T-Maze navigation domain because it fosters the emergence of adaptive behavior.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – connectionism and neural nets

General Terms

Algorithms

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-325-9/09/07 ...$5.00.

Keywords

Novelty Search, Neural Networks, Adaptation, Learning, Neuromodulation, Neuroevolution, NEAT

1. INTRODUCTION

Evolution and learning are two forms of biological adaptation that operate on different timescales. Whereas evolution produces phylogenetic adaptation, learning gives the individual the possibility to react much faster to environmental changes by modifying its behavior during its lifetime. There is much evidence that both processes are integral to the success of biological evolution [14, 16] and that lifetime learning can itself help to guide evolution to higher fitness [7], which is called the Baldwin Effect. Studying the interaction between evolution and learning can help not only to more fully understand biological processes but also to more efficiently create artificial adaptive systems that can learn during their lifetime. When the environment changes from what was encountered during evolution, the agent needs to adapt *online* to maintain performance. This paper introduces a method to make evolving such adaptive behavior significantly more effective.

One way that agents controlled by artificial neural networks (ANNs) can adapt is by allowing them to change their internal synaptic connection strengths during their lifetime. This approach resembles the way organisms in nature, which possess plastic nervous systems, cope with changing and unpredictable environments [5, 15, 18]. In a recent demonstration of the power of this approach, Soltoggio et al. [18] evolved adaptive Hebbian networks with *neuromodulation*, i.e. some neurons influencing how others change, that acquired the ability to memorize the position of a reward from previous trials in the T-Maze learning problem first introduced by Blynel and Floreano [2].

Although such results suggest the promise of evolving adaptive ANNs, experimental domains so far do not approach the complexity encountered by natural organisms. One reason for this gap is that learning to learn is highly deceptive with respect to objective performance on the fitness function. Reaching a mediocre fitness through non-adaptive behavior is relatively easy, but any further improvement requires sophisticated adaptive behavior with only sparse feed-

back from an objective-based performance measure. Nolfi et al. [17] also argue that there is no *a priori* reason to assume that what the individual learns during its lifetime automatically increases its chances to reproduce. Learning can even reduce fitness because of its costs (e.g. time, energy, etc.). It can take many generations for evolution to optimize the learning process sufficiently to amortize this cost.

Accordingly, this paper argues that domains that require adaptation are inherently deceptive and therefore evolution is handicapped when the goal is to evolve learning agents. In fact, deceptiveness in these domains is even more dramatic when learning is only needed in a low percentage of trials. In that case, evolution is trapped in local optima that do not require learning at all because high fitness values are achieved in the majority of trials.

Because of the problem of deception in adaptive domains, prior experiments in evolving adaptive ANNs have needed to be carefully designed to ensure that no non-adaptive heuristics exist that could potentially lead evolution prematurely astray. This awkward requirement has significantly limited the scope of domains amenable to adaptive evolution and stifled newcomers from entering the research area. To remedy this situation and open up the range of problems amenable to evolving adaptation, this paper argues that the *novelty search* algorithm [12], which abandons the traditional notion of objective-based fitness, circumvents the deception inherent in such domains.

Instead of searching for a final objective behavior, novelty search rewards finding any instance whose behavior is significantly different from what has been discovered before. Surprisingly, this radical form of search has been shown to outperform traditional fitness-based search in deceptive domains [12], making it potentially appropriate to addressing the problem of deception in evolving adaptive ANNs.

To demonstrate the potential of this approach, this paper compares novelty search with fitness-based evolution in the dynamic, reward-based T-Maze scenario introduced by Blynel and Floreano [2] and further studied in the context of neuromodulated plasticity by Soltoggio et al. [18]. In this scenario, the reward location is a variable factor in the environment that the agent must learn to exploit. By varying the number of times the reward location changes, the effect of adaptation on the fitness function can be controlled to make the domain more or less deceptive for objective-based fitness. Counterintuitively, novelty search *always* outperforms regular fitness-based search and is not affected by increased levels of deception, suggesting a powerful new approach to evolving adaptive behavior.

2. BACKGROUND

This section first reviews novelty search, which is the proposed solution to deception in the evolution of learning, and then explains the neuromodulation-based model of adaptive ANNs followed in this paper.

2.1 The Search for Novelty

The problem with the objective fitness function in evolutionary computation is that it does not necessarily reward the intermediate stepping stones that lead to the objective. The more ambitious the objective, the harder it is to identify *a priori* these stepping stones.

This paper hypothesizes that evolving adaptive ANNs is especially susceptible to missing the essential intermediate stepping stones for fitness-based search and therefore highly deceptive. Reaching a mediocre fitness through non-adaptive behavior is relatively easy, but any further improvement requires sophisticated adaptive behavior with only sparse feedback from an objective-based performance measure. Such deception is inherent in most dynamic, reward-based scenarios.

A potential solution to this problem is novelty search, which is a recent method for avoiding deception based on the radical idea of ignoring the objective [12]. The idea is to identify novelty as a proxy for stepping stones. That is, instead of searching for a final objective, the learning method is rewarded for finding any behavior whose functionality is significantly different from what has been discovered before. Thus, instead of an objective function, search employs a *novelty metric*. That way, no attempt is made to measure overall progress. In effect, such a process gradually accumulates novel behaviors.

Although it is counterintuitive, novelty search was actually *more effective* at finding the objective than a traditional objective-based fitness in a deceptive maze navigation domain [12]. Thus novelty search might be a solution to the longstanding problem with training for adaptation.

The next section describes the novelty search algorithm [12] in more detail.

2.1.1 The Novelty Search Algorithm

Evolutionary algorithms are well-suited to novelty search because the population that is central to such algorithms naturally covers a wide range of expanding behaviors. In fact, tracking novelty requires little change to any evolutionary algorithm aside from replacing the fitness function with a novelty metric.

The novelty metric measures how different an individual is from other individuals, creating a constant pressure to do something new. The key idea is that instead of rewarding performance on an objective, the novelty search rewards diverging from prior behaviors. Therefore, novelty needs to be measured.

There are many potential ways to measure novelty by analyzing and quantifying behaviors to characterize their differences. Importantly, like the fitness function, this measure must be fitted to the domain.

The novelty of a newly generated individual is computed with respect to the observed *behaviors* (i.e. *not* the genotypes) of an *archive* of past individuals whose behaviors were highly novel when they originated. In addition, if the evolutionary algorithm is steady state (i.e. one individual is replaced at a time) then the current population can also supplement the archive by representing the most recently visited points. The aim is to characterize how far away the new individual is from the rest of the population and its predecessors in *novelty space*, i.e. the space of unique behaviors. A good metric should thus compute the *sparseness* at any point in the novelty space. Areas with denser clusters of visited points are less novel and therefore rewarded less.

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. Intuitively, if the average distance to a given point's nearest neighbors is large then it is in a sparse area; it is in a dense

region if the average distance is small. The sparseness ρ at point x is given by

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

where μ_i is the ith-nearest neighbor of x with respect to the distance metric *dist*, which is a domain-dependent measure of behavioral difference between two individuals in the search space. The nearest neighbors calculation must take into consideration individuals from the current population and from the permanent archive of novel individuals. Candidates from more sparse regions of this behavioral search space then receive higher novelty scores. It is important to note that this novelty space cannot be explored purposefully, that is, it is not known *a priori* how to enter areas of low density just as it is not known a priori how to construct a solution close to the objective. Thus, moving through the space of novel behaviors requires exploration. In effect, because novelty is measured relative to other individuals in evolution, it is driven by a coevolutionary dynamic.

If novelty is sufficiently high at the location of a new individual, i.e. above some minimal threshold ρ_{min}, then the individual is entered into the permanent archive that characterizes the distribution of prior solutions in novelty space, similarly to archive-based approaches in coevolution [11]. The current generation plus the archive give a comprehensive sample of where the search has been and where it currently is; that way, by attempting to maximize the novelty metric, the gradient of search is simply towards what is new, with no other explicit objective.

It is important to note that novelty search resembles prior diversity maintenance techniques (i.e. speciation) popular in evolutionary computation [4, 6, 8, 9, 13]. The most well known are variants of fitness sharing [4, 6]. These also in effect open up the search by reducing selection pressure. However, in these methods, as in Hutter's fitness uniform selection [10], the search is still ultimately guided by the fitness function. Diversity maintenance simply keeps the population more diverse than it otherwise would be. (Also, most diversity maintenance techniques measure genotypic diversity as opposed to behavioral diversity [4, 13].) In contrast, novelty search takes the radical step of *only* rewarding behavioral diversity with no concept of fitness or a final objective, inoculating it to traditional deception.

It is also important to note that novelty search is not a random walk; rather, it explicitly maximizes novelty. Because novelty search includes an archive that accumulates a record of where search has been, backtracking, which can happen in a random walk, is effectively avoided in behavioral spaces of any dimensionality.

The novelty search approach in general allows any behavior characterization and any novelty metric. Although generally applicable, novelty search is best suited to domains with deceptive fitness landscapes, intuitive behavioral characterization, and domain constraints on possible expressible behaviors.

Changing the way the behavior space is characterized and the way characterizations are compared will lead to different search dynamics, similar to how researchers now change the fitness function to improve the search. The intent is not to imply that setting up novelty search is easier than objective-based search. Rather, once novelty search is set up, the hope is that it can find solutions beyond what even a sophisticated objective-based search can currently discover. Thus, the effort is justified in its returns.

The evolutionary algorithm that evolves neuromodulated plastic networks (explained in the next section) through novelty search in this paper is NeuroEvolution of Augmenting Topologies (NEAT) [22], which offers the ability to discover minimal effective adaptive topologies. The ANN topologies in NEAT start minimally and gradually add new structure, allowing it to find the right level of complexity for the task. NEAT has proven successful in diverse control and decision-making domains [1, 21, 22]. Also, importantly for this paper, novelty search is designed to work in combination with NEAT [12].

In particular, once objective-based fitness is replaced with novelty, the NEAT algorithm operates as normal, selecting the highest scoring individuals to reproduce. Over generations, the population spreads out across the space of possible behaviors, continually ascending to new levels of complexity (i.e. by expanding the neural networks in NEAT) to create novel behaviors as the simpler variants are exhausted. Thus, through NEAT, novelty search in effect searches not just for new behaviors, but for *increasingly complex* behaviors.

The next section details the model for adaptive ANNs in this paper.

2.2 Artificial Evolution of Neuromodulated Plasticity

Adaptive neural networks can *learn* by changing their internal synaptic connection strengths following a Hebbian learning rule that modifies synaptic weights based on pre- and postsynaptic neuron activity. The generalized Hebbian plasticity rule [15] takes the following form:

$$\Delta w = \eta \cdot [Axy + Bx + Cy + D], \quad (2)$$

where η is the learning rate, x and y are the activation levels of the presynaptic and postsynaptic neurons and A–D are the correlation term, presynaptic term, postsynaptic term, and constant, respectively.

Floreano and Urzelai [5] demonstrated the power of this approach by evolving only specific forms of local Hebbian learning rules for each synapse. The evolved adaptive controllers were compared to fixed-weight networks in turning on a light on one side of the environment and then navigating to a gray square area on the other side. The local learning rules in the evolved networks facilitated the policy transition from one task to the other.

Adaptive ANNs have also been successfully evolved to simulate robots in a dangerous foraging domain [21]. Although this work also showed that recurrent fixed-weight networks can be more effective and reliable than adaptive Hebbian controllers in some domains, previous studies [15, 18, 19] suggest that both network types reach their limits when more elaborate forms of learning are needed. For example, classical conditioning seems to require mechanisms that are not present in most current network models. To expand to such domains, following Soltoggio et al. [18], the study presented in this paper controls plasticity through *neuromodulation*.

In a neuromodulated network, a special neuromodulatory neuron can change the degree of potential plasticity between two standard neurons based on their activation levels (fig-

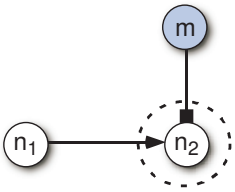

Figure 1: Neuromodulated plasticity. The weight of the connection between standard neurons n_1 and n_2 is modified by a Hebbian rule. Modulatory neuron m determines the magnitude of the weight change.

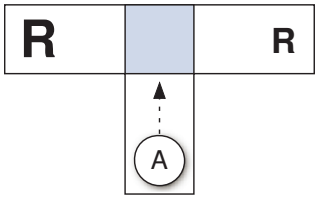

Figure 2: The T-Maze. In this depiction, high reward is located on the left and low reward is on the right side, but these positions can change over a set of trials. The goal of the agent is to navigate to the position of the high reward and back home to its starting position. The challenge is that the agent must remember the location of the high reward from one trial to the next.

ure 1). In addition to its standard activation value a_i, each neuron i also computes its modulatory activation m_i:

$$a_i = \sum_{j \in Std} w_{ij} \cdot o_j, \tag{3}$$

$$m_i = \sum_{j \in Mod} w_{ij} \cdot o_j, \tag{4}$$

where w_{ij} is the connection strength between presynaptic neuron j and postsynaptic neuron i and o_j is calculated as $o_j(a_j) = tanh(a_j/2)$. The weight between neurons i and j then changes following the m_i-modulated plasticity rule

$$\Delta w_{ji} = tanh(m_i/2) \cdot \eta \cdot [Axy + Bx + Cy + D]. \tag{5}$$

The benefit of adding modulation is that it allows the ANN to change the level of plasticity on specific neurons at specific times. This property seems to play a critical role in regulating learning behavior in animals [3] and neuromodulated networks have a clear advantage in more complex dynamic, reward-based scenarios: Soltoggio et al. [18] showed that networks with neuromodulated plasticity significantly outperform fixed-weight and traditional adaptive ANNs without neuromodulation in the double T-Maze domain, and display nearly optimal learning performance.

Few modifications to the standard NEAT algorithm are required to also encode neuromodulated plasticity. NEAT's genetic encoding is augmented with a new modulatory neuron type and each time a node is added through structural mutation, it is randomly assigned a standard or modulatory role. The neuromodulatory dynamics follow equations 2–5.

Thus the main idea is to evolve neuromodulatory ANNs with NEAT through novelty search, which we hypothesize should help to escape the deception inherent in many adaptive domains. The next section describes such a domain, which is the basis for testing this hypothesis.

3. THE DECEPTIVE T-MAZE DOMAIN

An appropriate domain for testing novelty search should have a deceptive fitness landscape [12]. In such a domain, an algorithm that follows the fitness gradient is susceptible to local optima.

How does deception arise in a dynamic, reward-based scenario in which the goal is to evolve adaptive agents? The problem occurs when the impact of learning on the fitness of an individual is only marginal. For example, an individual that performs well in the 99 out of 100 trials wherein learning is not required and only fails in the one trial that requires learning will most likely score a high fitness value. Thus such a search space is highly deceptive to evolving learning and the stepping stones that ultimately lead to an adaptive agent will not be rewarded. The problem is that learning domains often have the property that significant improvement in fitness is possible by discovering hidden heuristics that avoid lifetime adaptation entirely, creating a pathological deception against learning to learn.

The domain in this paper is based on experiments performed by Soltoggio et al. [18] on the evolution of neuromodulated networks for the T-Maze learning problem. The single T-Maze (figure 2) consists of two arms that either contain a high or low reward. The agent begins at the bottom of the maze and its goal is to navigate to the reward position and return home. This procedure is repeated many times during the agent's lifetime. One such attempted trip to a reward location and back is called a *trial*. A *deployment* consists of a set of trials. The goal of the agent is to maximize the amount of reward collected over deployments, which requires it to memorize the position of the high reward in each deployment. When the position of the reward sometimes changes, the agent should alter its strategy accordingly to explore the other arm of the maze in the next trial. In Soltoggio's original experiments [18], the reward location changes at least once during each deployment of an agent, which fosters the emergence of learning behavior.

However, the *deceptiveness* of this domain with respect to the evolution of learning can be increased if the reward location is not changed in all deployments in which the agent is evaluated. If adaptation is thus only required in a small subset of deployments, the advantage of an adaptive individual over a non-adaptive individual (i.e. always navigating to the same side) in fitness is only marginal. The hypothesis is that novelty search should outperform fitness-based search with increased domain deception.

4. EXPERIMENT

To compare the performance of NEAT with fitness-based search and NEAT with novelty search, each agent is evaluated on ten deployments, each consisting of 20 trials. The number of deployments in which the high reward is moved after ten trials varies among one (called the *1/10 scenario*), five (called the *5/10 scenario*), and ten (called the *10/10 scenario*), effectively controlling the level of deception. The high reward always begins on the left side at the start of each deployment.

Note that all deployments are deterministic, that is, a

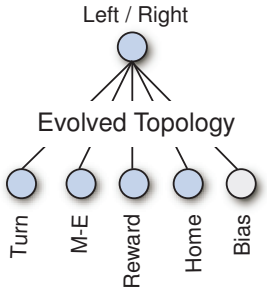

Figure 3: ANN topology. The network has four inputs, one bias, and one output neuron [18]. *Turn* is set to 1.0 at a turning point location. *M-E* is set to 1.0 at the end of the maze, and *Home* is set to 1.0 when the agent returns to the home location. The *Reward* input returns the level of reward collected at the end of the maze. The bias neuron emits a constant 1.0 activation that can connect to other neurons in the ANN. Network topology is evolved by NEAT.

deployment in which the reward does not switch sides will always lead to the same outcome with the same ANN. Thus the number of deployments in which the reward switches is effectively a means to control the proportional influence of adaptive versus non-adaptive deployments on fitness and novelty. The question is whether the consequent deception impacts novelty as it does fitness.

Figure 3 shows the inputs and outputs of the ANN. The *Turn* input is set to 1.0 when a turning point is encountered. *M-E* is set to 1.0 at the end of the maze and *Home* becomes 1.0 when the agent successfully navigates back to its starting position. The *Reward* input is set to the amount of reward collected at the maze end. An agent crashes if it does not (1) maintain a forward direction (i.e. activation of output neuron between -0.3 and 0.3) in corridors, (2) turn either right ($o > 0.3$) or left ($o < -0.3$) when it encounters the junction, or (3) make it back home after collecting the reward. If the agent crashes then the current trial is terminated.

The fitness function for fitness-based NEAT (which is identical to Soltoggio et al. [18]) is calculated as follows: Collecting the high reward has a value of 1.0 and the low reward is worth 0.2. If the agent fails to return home by taking a wrong turn after collecting a reward then a penalty of 0.3 is subtracted from fitness. On the other hand, 0.4 is subtracted if the agent does not maintain forward motion in corridors or does not turn left or right at a junction. The total fitness of an individual is determined by summing the fitness values for each of the 20 trials over all ten deployments.

Novelty search on the other hand requires a *novelty metric* to distinguish between different behaviors. The novelty metric for this domain distinguishes between learning and non-learning individuals and is explained in more detail in the next section.

4.1 Measuring Novelty in the T-Maze

The aim of the novelty metric is to measure differences in behavior. In effect, it determines the behavior-space through which the search explores. Because the goal of this paper is to evolve adaptive individuals, the novelty metric must distinguish a learning agent from a non-learning agent. Thus it is necessary to characterize behavior so that different such behaviors can be compared.

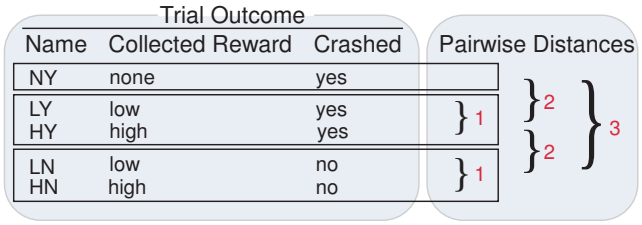

Figure 4: The T-Maze novelty metric. Each trial is characterized by (1) the amount of collected reward (2) whether the agent crashed. The pairwise distances (shown at right) among the five possible trial outcomes, NY, LY, HY, LN, and HN, depend on their behavioral similarities.

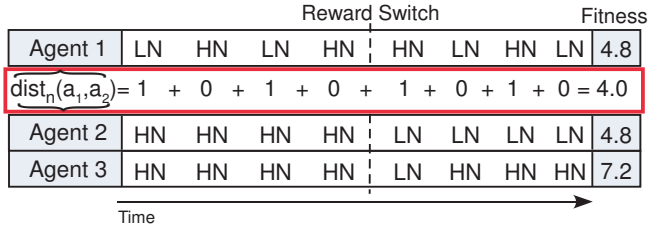

Figure 5: Three sample behaviors. These learning and non-learning individuals all exhibit distinguishable behaviors when compared over multiple trials. Agent three achieves the desired adaptive behavior. The vertical line indicates the point in time that the position of the high reward changed. While agents one and two look the same to fitness, novelty search notices their difference, as the distance calculation (inset line between agents 1 and 2) shows.

The behavior of an agent in the T-Maze domain is characterized by a series of trial outcomes (i.e. 200 trial outcomes for ten deployments with 20 trials each). It is necessary to include multiple trials because an agent that learns can only be distinguished from one that does not by observing its behavior before and after the reward switch.

Each trial outcome is characterized by two values: (1) the amount of reward collected (*high*, *low*, *none*) and (2) whether or not the agent crashed. These outcomes are assigned different *distances* to each other depending on how similar they are (figure 4). In particular, an agent that collects the high reward and returns home successfully without crashing (HN) should be more similar to an agent that collects the low reward and also returns home (LN) than to one that crashes without reaching any reward location (NY). The novelty distance metric $dist_{novelty}$ is ultimately computed by summing the distances between each trial outcome of two individuals over all deployments.

Figure 5 depicts outcomes over several trials of three example agents. The first agent always alternates between the left and the right T-Maze arm, which leads to oscillating low and high rewards. The second agent always navigates to the left T-Maze arm. This strategy results in collecting the high reward in the first four trials and then collecting the low reward after the reward switch. The third agent exhibits the desired learning behavior and is able to collect the high reward in seven out of eight trials. (One trial of explorative behavior is needed after the reward switch.)

Interestingly, because both agents one and two collect

the same amount of high and low reward, they achieve the same fitness, making them indistinguishable to fitness-based search. However, novelty search discriminates between them because $dist_{novelty}(agent_1, agent_2) = 4.0$.

Importantly, fitness and novelty both use the same information (i.e. the amount of reward collected and whether or not the agent crashed) to explore the search space, though in a completely different way. Thus the comparison is fair.

4.2 Generalization Performance

The goal of the comparison between fitness and novelty is to determine which learns to adapt most efficiently in different deployment scenarios, e.g. 1/10, 5/10, and 10/10. Thus it is important to note that, because performance on different scenarios will vary based on the number of trials in which the reward location switches, for the purpose of analyzing the results there is a need for an independent measure that reveals the overall adaptive capabilities of each individual.

Therefore, to test the ability of the individuals to generalize independently of the number of deployments in which the position of the high reward changes, they are tested for 20 trials on each of two different initial settings: (1) high reward starting left and (2) high reward starting right. In both cases, the position of the high reward changes after 10 trials. An individual passes the *generalization test* if it can collect the high reward and return back home in at least 18 out of 20 trials from both initial positions. Two low reward trials in each setting are necessary to explore the T-Maze at the beginning of each deployment and when the position of the high reward switches.

The generalization measure does not necessarily correlate to fitness. An individual that receives a high fitness in the 1/10 scenario can potentially perform poorly on the generalization test because it does not exhibit adaptive behavior. Nevertheless, generalization performance does follow a general upward trend over evaluations and reveals the ultimate quality of solutions.

4.3 Experimental Parameters

NEAT with fitness-based search and novelty search run with the same parameters in the experiments in this paper. The steady-state real-time NEAT (rtNEAT) package [20] is extended to encode neuromodulatory neurons. The population size is 500, with a 0.001 probability of adding a node (uniformly randomly chosen to be standard or modulatory) and 0.01 probability of adding a link. The weight mutation power is 1.8. Runs last up to 125,000 evaluations. They are stopped when the generalization test is solved. The number of nearest neighbors for the novelty search algorithm is 15 (following Lehman and Stanley [12]). The novelty threshold is 2.0. This threshold for adding behaviors to the archive dynamically changes every 1,500 evaluations. If no new individuals are added during that time the threshold is lowered by 5%. It is raised by 20% if the number of individuals added is equal to or higher than four. The novelty scores of the current population are reevaluated every 100 evaluations to keep them up to date (the archive does not need to be reevaluated).

The coefficients of the generalized Hebbian learning rule used by all evolved neuromodulated networks are $A = 0.0$, $B = 0.0$, $C = -0.38$, $D = 0.0$ and $\eta = -94.6$, resulting in the following m_i-modulated plasticity rule:

$$\Delta w_{ji} = tanh(m_i/2) \cdot 35.95y. \quad (6)$$

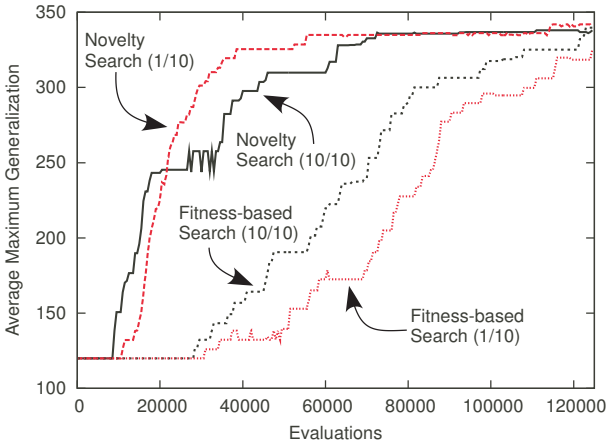

Figure 6: Comparing generalization of novelty search and fitness-based search. The change in performance over evaluations on the generalization test is shown for NEAT with novelty search and fitness-based search in the 1/10 and 10/10 scenarios. All results are averaged over 20 runs. The main result is that novelty search learns a general solution significantly faster.

These values worked well for a neuromodulated ANN in the T-Maze learning problem described by Soltoggio et al. [18]. Therefore, to isolate the effect of evolving based on novelty versus fitness, they are fixed at these values in the experiment in this paper. However, modulatory neurons still affect the learning rate at Hebbian synapses as usual. For a more detailed description of the implications of different coefficient values for the generalized Hebbian plasticity rule see Niv et al. [15].

5. RESULTS

Because the aim of this experiment is to determine how quickly a general solution is found by both methods, an agent that can solve the generalization test described in Section 4.2 counts as a solution.

Figure 6 shows the average performance of the current best-performing individuals on the generalization test across evaluations for novelty search and fitness-based search, depending on the number of deployments in which the reward location changes. Novelty search performs consistently better in all scenarios. Even in the 10/10 domain that resembles the original experiment [18], it takes fitness significantly longer to reach a solution than novelty search. The fitness-based approach initially stalls, followed by gradual improvement, whereas on average novelty search rises sharply from early in the run.

Figure 7 shows the average number of evaluations over 20 runs it took fitness-based and novelty-based NEAT to solve the generalization test in the 1/10, 5/10, and 10/10 scenarios. If no solution was found within the initial 125,000 evaluations, the current simulation was restarted. This procedure was repeated until a solution was found, counting all evaluations over all restarts.

Both novelty and fitness-based NEAT were restarted three times out of 20 runs in the 10/10 scenario. Fitness-based search took on average 90,575 evaluations ($\sigma = 52,760$) while novelty search was almost twice as fast at 48,235 eval-

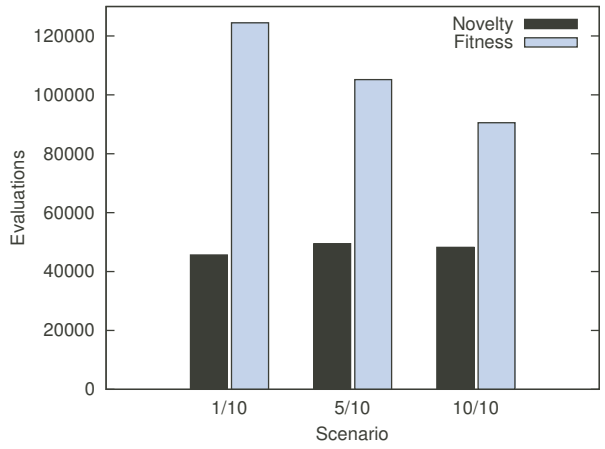

Figure 7: Average evaluations to solution for novelty search and fitness-based search. The average number of evaluations over 20 runs that it took novelty search and fitness-based search to solve the generalization test is shown. Novelty search performs significantly better in all scenarios and fitness-based search performs even worse when deception is high. Interestingly, novelty search performance does not degrade at all with increasing deception.

uations on average ($\sigma = 55,638$). This difference is significant ($p < 0.05$). In the more deceptive 1/10 scenario, fitness-based search had to be restarted six times and it took 124,495 evaluations on average ($\sigma = 81,789$) to find a solution. Novelty search only had to be restarted three times and was 2.7 times faster ($p < 0.001$) at 45,631 evaluations on average ($\sigma = 46,687$).

Fitness-based NEAT performs worse with increased domain deception and is 1.4 times slower in the 1/10 scenario than in the 10/10 scenario. It took fitness on average 105,218 evaluations ($\sigma = 65,711$) in the intermediate 5/10 scenario, which is in-between its performance on the 1/10 and 10/10 scenarios, confirming that deception increases as the number of trials requiring adaptation decreases. In contrast, novelty search is not significantly affected by increased domain deception: The performance differences among the 1/10, 5/10, and 10/10 scenarios is insignificant for novelty search.

6. DISCUSSION AND FUTURE WORK

Interestingly, novelty search not only outperforms fitness-based search in the highly deceptive 1/10 scenario but also in the intermediate 5/10 scenario and even in the 10/10 scenario in which the location of the reward changes every deployment. Fitness-based search gradually deteriorates with increased domain deception (figure 7) while novelty search is not significantly affected because it explores the space of possible behaviors in a completely different way.

There is no obvious deception in the 10/10 scenario that resembles Soltoggio's original experiment [18]; however the long plateaus in fitness common to all scenarios (figure 6) suggest a general problem for evolving learning behavior in dynamic, reward-based scenarios.

Agents initially learn to always navigate to one arm of the maze and back, resulting in collecting 20 high rewards (i.e. ten high rewards for each of the two starting positions) on the generalization test. Yet because the reward location changes after ten trials for both initial settings, to be more successful requires the agents to exhibit learning behavior. The problem is that evolving the right neuromodulated dynamics to be able to achieve learning behavior is not an easy task. There is little information available to incentivize fitness-based search to pass this point, making it act more like random search. In other words, the stepping stones that lead to learning behavior are hidden from the objective approach behind long plateaus in the search space.

While in some domains the fitness gradient can be improved, i.e. by giving the objective-based search clues in which direction to search, such an approach might not be possible in dynamic, reward-based scenarios. The problem in such domains is that reaching a certain fitness level is relatively easy, but any further improvement requires sophisticated adaptive behavior to evolve from only sparse feedback from an objective-based performance measure. That is, novelty search returns more *information* about how behavior changes throughout the search space.

In this way, novelty search removes the need to carefully design a domain that fosters the emergence of learning because novelty search on its own is capable of doing exactly that. The only prerequisite is that the novelty metric is constructed such that learning and non-learning agents are separable, which is not necessarily easy, but is worth the effort if objective-based search would otherwise fail.

In fact, because NEAT itself employs the *fitness sharing* diversity maintenance technique [6, 22], the significant difference in performance between NEAT with novelty search and NEAT with fitness-based search also suggests that traditional diversity maintenance techniques do not evade deception as effectively as novelty search.

Novelty search's ability to build gradients that lead to stepping stones is evident in figure 6. The increase in generalization performance is steeper than for fitness-based NEAT, indicating a more efficient climb to higher complexity behaviors. In effect, by abandoning the objective, the stepping stones come into greater focus [12]. Although it means that the search is wider, as Lehman and Stanley [12] write, "*It is better to search far and wide and eventually reach a summit than to search narrowly and single-mindedly yet never come close.*"

Of course, there are likely domains for which the representation is not suited to discovering the needed adaptive behavior or in which the space of behaviors is too vast for novelty search to reliably discover the right one. In some cases, novelty search might search to within the vicinity of the answer but it may not sufficiently fine-tune the results. On the other hand, because higher fitness may often be associated with novel behavior, it is likely that novelty search will sometimes implicitly fine-tune fitness.

Characterizing when and for what reason novelty search fails is an important future research direction. Yet its performance in this paper and in past research [12] has proven surprisingly robust. While it is not always going to work well, this paper suggests that it is a viable new tool in the toolbox of evolutionary computation to counteract the deception inherent in evolving adaptive behavior.

Thus the results in this paper are important because research on evolving adaptive agents has been hampered largely as a result of the deceptiveness of adaptive tasks. Yet the promise of evolving adaptive ANNs is among the most intriguing in artificial intelligence. After all, our own brains

are the result of such an evolutionary process. Therefore, a method to make such domains more amenable to evolution has the potential to further unleash a promising research direction that is only just beginning.

To explore this opportunity, a promising future direction is to apply novelty search to other adaptive problems without the need to worry about mitigating their potential for deception.

7. CONCLUSIONS

This paper showed how novelty search, which abandons the objective to search only for novel behaviors, can facilitate the evolution of adaptive behavior. Results on a T-Maze domain demonstrated that novelty search can significantly outperform objective-based search and fosters the emergence of adaptive individuals. It also performed consistently under varying levels of domain deception. The main conclusion is that it may now be more realistic to learn interesting adaptive behaviors that have been heretofore seemingly too difficult. Furthermore, the results presented in this paper add to the growing body of evidence [12] that novelty search can overcome the deception inherent in a diversity of tasks.

Acknowledgments

This research was partially supported by the National Science Foundation under grants DRL0638977 and IIP0750551. Special thanks to Andrea Soltoggio for sharing prior experience in the T-Maze domain.

8. REFERENCES

[1] T. Aaltonen et al. Measurement of the top quark mass with dilepton events selected using neuroevolution at CDF. *Physical Review Letters*, 2009. To appear.

[2] J. Blynel and D. Floreano. Exploring the T-Maze: Evolving Learning-Like Robot Behaviors using CTRNNs. In *2nd European Workshop on Evolutionary Robotics (EvoRob'2003)*, Lecture Notes in Computer Science, 2003.

[3] T. Carew, E. Walters, and E. Kandel. Classical conditioning in a simple withdrawal reflex in Aplysia californica. *The Journal of Neuroscience*, 1(12):1426–1437, 1981.

[4] P. Darwen and Y. Yao. Every niching method has its niche: Fitness sharing and implicit sharing compared. *Parallel Problem Solving from Nature (PPSN IV)*, pages 398–407, 1996.

[5] D. Floreano and J. Urzelai. Evolutionary robots with online self-organization and behavioral fitness. *Neural Networks*, 13:431–443, 2000.

[6] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[7] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, 1, 1987.

[8] G. S. Hornby. Alps: the age-layered population structure for reducing the problem of premature convergence. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 815–822, New York, NY, USA, 2006. ACM.

[9] J. Hu, E. Goodman, K. Seo, Z. Fan, and R. Rosenberg. The hierarchical fair competition (hfc) framework for sustainable evolutionary algorithms. *Evolutionary Computation*, 13(2):241–277, 2005. PMID: 15969902.

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

[11] E. D. Jong. The incremental pareto-coevolution archive. In *Proceedings of the Genetic and Evolutionary Computation Conference, (GECCO-2004)*, Berlin, 2004. Springer.

[12] J. Lehman and K. O. Stanley. Exploiting open endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life*, Cambridge, MA, 2008. MIT Press.

[13] S. W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, Champaign, IL, USA, 1995.

[14] G. Mayley. Guiding or hiding: Explorations into the effects of learning on the rate of evolution. In *Fourth European Conference on Artificial Life*, pages 135–144. MIT Press, 1997.

[15] Y. Niv, D. Joel, I. Meilijson, and E. Ruppin. Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior*, 10(1):5–24, 2002.

[16] S. Nolfi and D. Floreano. Learning and evolution. *Autonomous Robots*, 7(1):89–113, July 1999.

[17] S. Nolfi, D. Parisi, and J. L. Elman. Learning and evolution in neural networks. *Adaptive Behavior*, 3:5–28, 1994.

[18] A. Soltoggio, J. A. Bullinaria, C. Mattiussi, P. Dürr, and D. Floreano. Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenarios. In *Artificial Life XI*, pages 569–576, Cambridge, MA, 2008. MIT Press.

[19] A. Soltoggio, P. Dürr, C. Mattiussi, and D. Floreano. Evolving neuromodulatory topologies for reinforcement learning-like problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2007.

[20] K. O. Stanley. rtNEAT C++ software homepage: **www.cs.utexas.edu/users/nn/keyword?rtneat**. 2006-2008.

[21] K. O. Stanley, B. D. Bryant, and R. Miikkulainen. Evolving adaptive neural networks with and without adaptive synapses. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC-2003)*. Canberra, Australia: IEEE Press, 2003.

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