Accelerating the Evolution of Cognitive Behaviors Through Human-Computer Collaboration

Mathias Löwe
Robotics, Evolution, and Art Lab
IT University of Copenhagen
Copenhagen, Denmark
malw@itu.dk

Sebastian Risi
Robotics, Evolution, and Art Lab
IT University of Copenhagen
Copenhagen, Denmark
sebr@itu.dk

ABSTRACT

An open problem in *neuroevolution* (i.e. evolving artificial neural networks) is to evolve complex cognitive behaviors that allow robots to adapt and learn from past experience. While previous studies on the evolution of cognitive behaviors have shown that more explorative search methods such as *novelty search*, outperform traditional objective-based approaches, evolving more sophisticated cognitive capabilities remains difficult. In this context, a major challenge is the deceptive nature of learning to learn. Because it is easier at first to improve fitness without evolving the ability to learn, evolution often converges on non-adaptive solutions. The novel hypothesis in this paper is that we can leverage human insights during the search for cognitive behaviors because of our ability to more easily distinguish between adaptive and non-adaptive solutions than novelty or fitness-based approaches. This paper shows that the recently introduced method *novelty-assisted interactive evolution* (NA-IEC), which combines human intuition with novelty search, allows the evolution of cognitive behaviors in a T-Maze domain faster than fully-automated searches by themselves.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Connectionism and neural nets*

Keywords

Interactive Evolution, Adaptive Neural Networks, Human-Computer Collaboration.

1. INTRODUCTION

Evolving artificial neural networks (ANNs) able to learn and adapt has been a major goal within the field of neuroevolution [7, 15, 19, 3]. However, evolving adaptive behaviors has proven difficult, especially in domains in which learning only improves fitness marginally. In such domains, it is easier for evolution to discover agents which utilize the static heuristics of the domain, and reach a mediocre fitness. Hereafter, further improvement is difficult without already displaying the correct adaptive behavior. Because learning to learn is highly deceptive, traditional objective-based search methods often do not reward the stepping stones that lead to the correct adaptive behavior [7, 14].

Recent evidence suggests that methods that are based on behavioral novelty, such as *novelty search* [8], may be beneficial in the evolution of learning behaviors [7, 15]. By pursuing novel behaviors, it is possible to evolve learning behaviors faster than through traditional objective-based search. Novelty search seems particularly helpful in domains in which the objective-based fitness gain of adapting to the dynamic environment is low [7, 14].

However, evolving more complex cognitive behaviors still remains difficult. Novelty search can get lost in large search spaces, particularly in unbounded environments [9]. In other domains, a certain behavior may not be remarkably novel compared to previously observed behaviors, but still a crucial stepping stone for reaching the final goal.

Other approaches that have shown promise in avoiding deception are based on *interactive evolutionary computation* (IEC) methods [21], wherein the human user guides evolution by repeatedly choosing from a set of candidates. Woolley and Stanley [23] showed that interactive evolution can help to discover artifacts which are hard to evolve with traditional evolutionary approaches.

Building on the recently introduced *novelty-assisted interactive evolution* (NA-IEC) approach [23], which combines IEC with novelty search, the new insights in this paper are that humans can more easily identify promising stepping stones that *lead towards cognitive behaviors* than either novelty or fitness-based search by themselves. The results in this paper show that humans and computers can work together synergistically to evolve adaptive agents for a T-Maze learning task. Interestingly, both novelty and fitness-based search fail to reward potentially useful stepping stones towards cognitive behaviors, while they are in fact identified by a human user. The conclusion is that in the future it might be possible to evolve more complex cognitive behaviors by humans and machines collaborating, that have heretofore proven too difficult for neuroevolution.

2. BACKGROUND

This section first describes *NeuroEvolution of Augmented Topologies* (NEAT), which evolves the ANNs in this paper. Neuromodulatory models are reviewed next, which provide the ability to evolve plastic ANNs capable of learning. This

GECCO '16, July 20-24, 2016, Denver, CO, USA
© 2016 ACM. ISBN 978-1-4503-4206-3/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2908812.2908928

is followed by a description of the novelty search algorithm and finally a review of interactive evolutionary computation (IEC) and an introduction to how novelty assisted interactive evolution (NA-IEC) enables an efficient interactive evolution to overcome the limitations of IEC.

2.1 NEAT

NeuroEvolution of Augmented Topologies (NEAT) has proven successful in a variety of domains and was originally proposed by Stanley and Miikkulainen [20] as a way to evolve ANNs capable of solving complex control tasks. NEAT is based on three main concepts: (1) NEAT gives a principled way to crossover different topologies; (2) It keeps track of and preserves innovation; (3) It is based on the principle of minimal structure which ensures that all complexity in a topology is justified. For a more complete review of NEAT see Stanley and Miikkulainen [20].

2.2 Neuromodulated Plasticity

In plastic ANNs, the internal connection strengths can change following a Hebbian learning rule that modifies synaptic weights based on pre- and postsynaptic neuron activity. Following Soltoggio et al. [19], the plastic Hebbian networks evolved in this paper are augmented with neuromodulation by using two kinds of neurons, *modulatory* and *standard*. The standard neurons act as normal information processing units that activate other connected neurons. The output from the modulatory neurons modulate the synaptic growth of the target neuron's incoming synapses from standard neurons. The synaptic growth or decay follows a *Hebbian learning rule*. The Hebbian learning rule modifies the synaptic strength based on the pre- and postsynaptic neuron stimulation. The generalized Hebbian learning rule [11] is a follows:

$$\delta_{ji} = \eta \cdot [A o_j o_i + B o_j + C o_i + D], \tag{1}$$

where η is the learning rate, o_j is the presynaptic neuron output, and o_i is the postsynaptic neuron output. A, B, C, and D are the correlation term, presynaptic term, postsynaptic term, and a constant. Classic Hebbian learning only uses one type of neurons and therefore makes use of homosynaptic plasticity [11]. Every time both neurons are stimulated simultaneously the synaptic strength changes [4]. As in recent studies [13, 15, 16], this paper extends the classic Hebbian learning rule to make it function in a heterosynaptic mechanism or neuromodulation. In contrast to the classic Hebbian learning, it is the modulatory neurons that are responsible for the plasticity in the ANN.

By using heterosynaptic plasticity with two types of neurons, it is possible to modulate the synaptic strength between neurons regardless of whether the pre- and/or postsynaptic neuron has been activated [11]. The heterosynaptic plasticity enables the ANN to change the level of plasticity at specific neurons on demand [13]. In a neuromodulated ANN, the weight change is calculated by:

$$\Delta w_{ji} = \delta_{ji} \cdot tanh(m_i/2), \tag{2}$$

where δ_{ji} is the output from equation 1, and m_i called the modulatory value which is calculated for each neuron as

$$m_i = \sum_{j \in Mod} w_{ji} \cdot o_j + m_d. \tag{3}$$

Here Mod is the collection of all the modulatory neurons, w_{ji} is the synaptic strength between neuron j and i, and m_d is a default modulation value. O_j is the presynaptic neuron's output, a_j, computed with the standard activation function, which in this paper is the tangent function

$$o_j(a_j) = tanh(a_j/2). \tag{4}$$

Lastly, the standard activation value, a_i, is

$$a_i = \sum_{j \in Std} w_{ji} \cdot o_j, \tag{5}$$

where Std is the collection of all the standard neurons, w_{ji} is the synaptic strength between neuron j and i, and o_j is the presynaptic neuron's output.

2.3 The Search For Novelty

Objective-based fitness measures have shown to struggle in deceptive domains (i.e. domains with many suboptimal solutions) because they do not reward the stepping stones that lead towards the final solution [8, 15]. Lehman and Stanley [8] proposed a new algorithm called *novelty search* that abandons the traditional objective driven search paradigm, and only looks for novel behaviors. This method has shown promise in a variety of deceptive domains such as biped walking [10], maze navigation [8], or the evolution of cognitive behaviors in the T-Maze domain studied in this paper [7, 14]. Novelty search is able to avoid local optima in these deceiving domains by actively pursuing exploration in the search space [8, 15].

2.3.1 The Novelty Search Algorithm

Tracking novelty requires little change to any evolutionary algorithm aside from replacing the fitness function with a novelty metric. The novelty metric compares the individual's behavior to an archive of past individuals that were highly novel together with the current population. The novelty score is calculated as the average distance to the k-nearest neighbors:

$$p(x) = \frac{1}{k} \sum_{i=1}^{k} dist(x, \mu_i), \tag{6}$$

where x is the individual at hand, μ_i is the ith nearest neighbor from either the archive or the population, and $dist$ is the distance between two behaviors. Equation 6 ensures, that a behavior in a sparse part of the search space will have a high distance to its kth nearest neighbors and thus a high novelty score. Notice that the novelty score of a behavior is relative and dynamic; it changes during evolution as the search progresses. Individuals are added to the archive if the behavior's novelty score is above a certain threshold p_{min}.

2.4 Interactive Evolutionary Computation

In *interactive evolutionary computation* (IEC) the traditional objective fitness function is replaced by a human selecting the candidates for the next generation. IEC has traditionally been used in optimization tasks of subjective criteria or in open-ended domains where the objective is undefined [12, 21]. A problem in IEC is *user fatigue*, and many attempts have been made to minimize it [2, 6, 12]. User fatigue is a problem because evolution takes time and humans tend to suffer from fatigue after evaluating relatively few generations.

One approach to limit user fatigue is *collaborative interactive evolution*, in which users can continue evolution from promising starting point generated by other users. Picbreeder[1] is an example of such a system, which allows users to collaboratively evolve images online, building on the intermediate results published by others [17].

Another way to alleviate the problem of user fatigue is to combine the intuition of humans with the speed of machines to limit the number of candidates the user has to evaluate [1, 22]. Bernatskiy et al. [1] introduced an approach called a *Fitness-Based Search with Preference based Policy Learning* (FS-PPL) that trains a user model based on collected user preferences. After the model is trained, it guides the search along with the fitness function. However, here we build on a complementary approach that is more novelty than objective-based driven, which is explained next. Novelty search seems to be particularly well suited to overcome deception in adaptive domains [7, 14, 18].

2.5 Novelty-assisted Interactive Evolutionary Computation

An approach recently introduced by Woolley and Stanley [23], called *novelty-assisted interactive evolutionary computation* (NA-IEC), combines human intuition with an automated search to make the most of the limited number of possible evaluations from a human user. As in traditional IEC, the user is presented with a series of behaviors and has to pick one or more parents for the next generation. However, unlike IEC, NA-IEC enables users to choose between three different strategies: a traditional IEC step, fitness-based optimization, or a short-term novelty search. The novelty search step runs until it has found a certain amount of novel behaviors (i.e. behaviors that have a novelty score above the novelty threshold p_{min}).

Woolley and Stanley [23] showed that NA-IEC finds solutions much faster than using either objective fitness or novelty fitness alone in deceptive navigation tasks. The speed increased both in terms of evaluations and wall-clock time. Furthermore, the resulting solution ANNs had significantly fewer hidden nodes than the ones evolved with a traditional objective fitness.

In this paper, NA-IEC is applied for the first time in a dynamic domain, in which the objective is to evolve online learning behavior. The hope is that a combination of our human ability to identify promising stepping stones together with novelty search can facilitate the evolution of these cognitive behaviors.

3. THE DECEPTIVE T-MAZE DOMAIN

The domain in this paper is based on experiments performed by Soltoggio et al. [19] on the evolution of neuromodulated networks for the T-Maze learning problem. Different variations of the domain have been used to study the evolution of learning behavior [13, 14, 15, 19]. We use the harder continuous version of the T-Maze, in which the agent has to develop both collision avoidance and the ability to learn during its lifetime in a continuous world [13].

Figure 1 shows the maze's two arms with two rewards, a high and a low reward. The task is for the agent to avoid collision with the walls, and collect the high reward as many times as possible during its lifetime. The agent has three rangefinders that detect its distance to the walls. These values are scaled into the range [0,1], where higher values indicate a closer proximity to a wall.

[1] http://picbreeder.org/

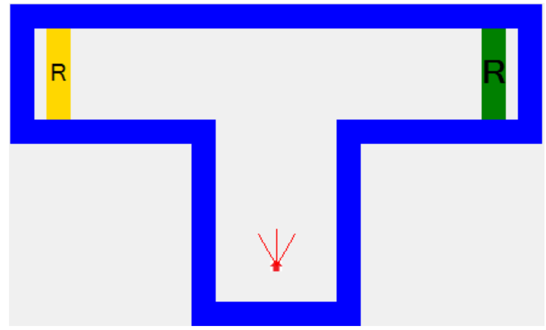


Figure 1: The Continuous T-maze Domain. In this depiction, high reward is located on the right and low reward is on the left side, but these positions can change over a set of trials. The goal of the agent is to navigate to the position of the high reward. The challenge is that the agent must remember the location of the high reward from one trial to the next.

The agent begins at the bottom of the maze and its goal is to navigate to the high reward position. This procedure is repeated many times during the agent's lifetime. One such attempted trip to a reward location is referred to as 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. The rewards switch sides during the agent's lifetime. Thus, the agent is required to recognize such a change when collecting the low reward, and alter its strategy accordingly to explore the other arm of the maze in the next trial to collect the high reward again. In the experiments reported in this paper, each agent is evaluated on two deployments that consist of ten trials each. The rewards always switch sides after five trials to get a noise free fitness evaluation and make behaviors between generations easier to compare for the user. In the first deployment, the high reward starts on the right side of the maze and on the left side in the second deployment.

Figure 2 shows the setup of the ANN that controls the agent. The ANN has three inputs for the agent's rangefinders and one reward input, which is set to the amount of reward collected at the maze end (1.0 for the high and 0.2 for the low reward). The agent is evaluated for up to 500 steps, where the agent is constantly moved forward each timestep. The robot turns $17(R - L)$ degrees, where L is the left effector output, and R is the right effector output.

3.1 Measuring Novelty in the T-Maze

To run the NA-IEC algorithm, a novelty metric for the T-Maze is needed that must distinguish a learning agent from a non-learning agent. In this context, 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.

The novelty metric in this paper is a simplified version of the T-Maze novelty metric introduced by Risi et al. [15], which was able to evolve learning behaviors faster than traditional objective-based search. Table 1 shows how the behavioral distance between different trial outcomes is calculated. As shown in Table 2, this metric can more easily distinguish

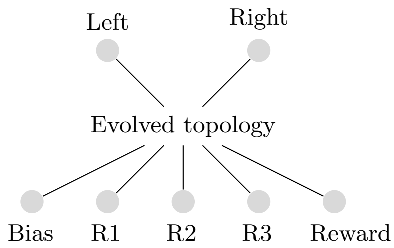


Figure 2: The basic ANN topology. The network has five inputs, one for bias, one for each of the three range sensors (R1, R2, and R3), and one for sensing the collected reward. The bias gives an output of 1.0 during the whole evaluation. When collecting the low reward, the reward sensor receives a 0.2 stimulation, and when collecting the high reward, it receives a 1.0 stimulation. The network has two outputs, simulating a robot with two motors controlling each wheel and thereby how much the robot turns to either side.

Table 1: Novelty Metric. The pairwise distances between the different trial outcomes depend on their behavioral similarities. The distance between two behaviors that collect rewards is low compared to one that just crashes.

Name	Collected Reward	Pairwise Distances
H	High Reward	} 1
L	Low Reward	} 3
C	None (crashed)	

between different types of adaptive behaviors than fitness. Agent 1 always navigates to the right side of the maze, independently of the reward switch, resulting in a mediocre fitness (12.0). Agent 2 on the other hand, keeps alternating between which arm it navigates to for each trial. In the end, both behaviors receive the same fitness score, however, novelty search succeeds in distinguishing between them, as their novelty distance is 4.0.

4. EVOLVING COGNITIVE BEHAVIORS WITH NA-IEC

The NA-IEC user interface for the T-Maze domain is shown in Figure 3. A population of 15 individuals is shown from which the user can select candidate behaviors to apply one of the three different NA-IEC strategies: *step*, *optimization*, and *explore*.

As in traditional IEC, *Step* creates a new generation by combining and mutating the user-selected individuals. *Optimize* runs a traditional objective-based search for five generations. *Explore* performs a short-term novelty search until at least 15 individuals are novel enough to enter the archive. The novelty threshold to add individuals to the archive can be modified by the user to allow for more fine-grained control. The *back* button enables the user to discard the current

Table 2: The fitness and novelty calculation for two distinct behaviors. While the two different behaviors are indistinguishable for fitness as they collect the same amount of reward, novelty search is able to distinguish them.

	Reward Switch								Fitness
Agent 1	H	H	H	H	L	L	L	L	12.0
Agent 2	H	L	H	L	L	H	L	H	12.0
Novelty Dist.	0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 = 4.0								

population and go back to the population from the previous generation. The back button does not modify the number of evaluations performed or the novelty search archive.

For each of the 15 individuals, a visual representation of the agent's behavior across the 20 trials is shown to the user. Each row is a separate deployment, and the color of each cell represents a trial outcome. A green cell implies that the agent collected the high reward, and a yellow cell implies it collected the low reward. The red vertical line represents when the rewards switched locations during the deployment. The agent's trajectory throughout all trials and its novelty and fitness score are also shown.

In the NA-IEC approach in this paper, both the automated and the IEC version use the same population size of 500 individuals. Through a drop-down menu, the user can decide to either see the 15 most novel behaviors or the 15 behaviors with the highest fitness.

5. EXPERIMENT

To evaluate the performance of NA-IEC, a user test was conducted on site at the IT University of Copenhagen, with ten inexperienced users. Before the test, a user guide was distributed to the participants that included a description of the domain, the target learning behavior, and the user interface. In an additional experiment an expert user (one of the authors) tried to evolve 15 solutions to the T-Maze. The purpose of this comparison was to measure how well NA-IEC performs depending on the level of experience with the particular framework and domain, and insights gained on which behaviors are promising stepping stones. These experiments are compared to automated versions with both fitness-based search and novelty search. The experiments were conducted with SharpNEAT [5] as the underlying framework, which was extended to support neuromodulation.

5.1 Experimental Parameters

The population size was 500 with 30 species for all experiments. The mutation rates used for NA-IEC was a 3% chance of adding a neuron, 10% chance of adding a connection, and 1% chance of deleting a connection. For the fitness-based search and novelty search, there was 0.1% chance of adding a neuron, 1% chance of adding a connection, and 0.1% chance of deleting a connection. In the event of adding a neuron, there was a 50% chance of it being a modulatory neuron. In order to achieve dynamic behavior faster, all genomes in the initial population were mutated by adding a neuron and a connection. The initial mutation of the population was performed for all approaches. Evolution ran for up to 125,000 evaluations. If no solution was found by then, it counted as a failure. A NA-IEC session was counted as a failure if the user stopped the current session before finding a solution or when 125,000 evaluations were reached.

The number of nearest neighbors for novelty search was set to 15. In the automated version, the threshold p_{min} for adding individuals to the archive was initially set to 2.0. Following Risi et al. [15], p_{min} was automatically adjusted every 1500 evaluations: it increased by 20% if four or more individuals were added to the archive during the 1500 evaluations. If no individuals were added to the archive, p_{min} was decreased by 5%. In the NA-IEC experiments, the p_{min} variable was static during novelty search, but the user was able to adjust it during the IEC sessions. The track bar used to set the p_{min} value (see Figure 3) can be set to values in

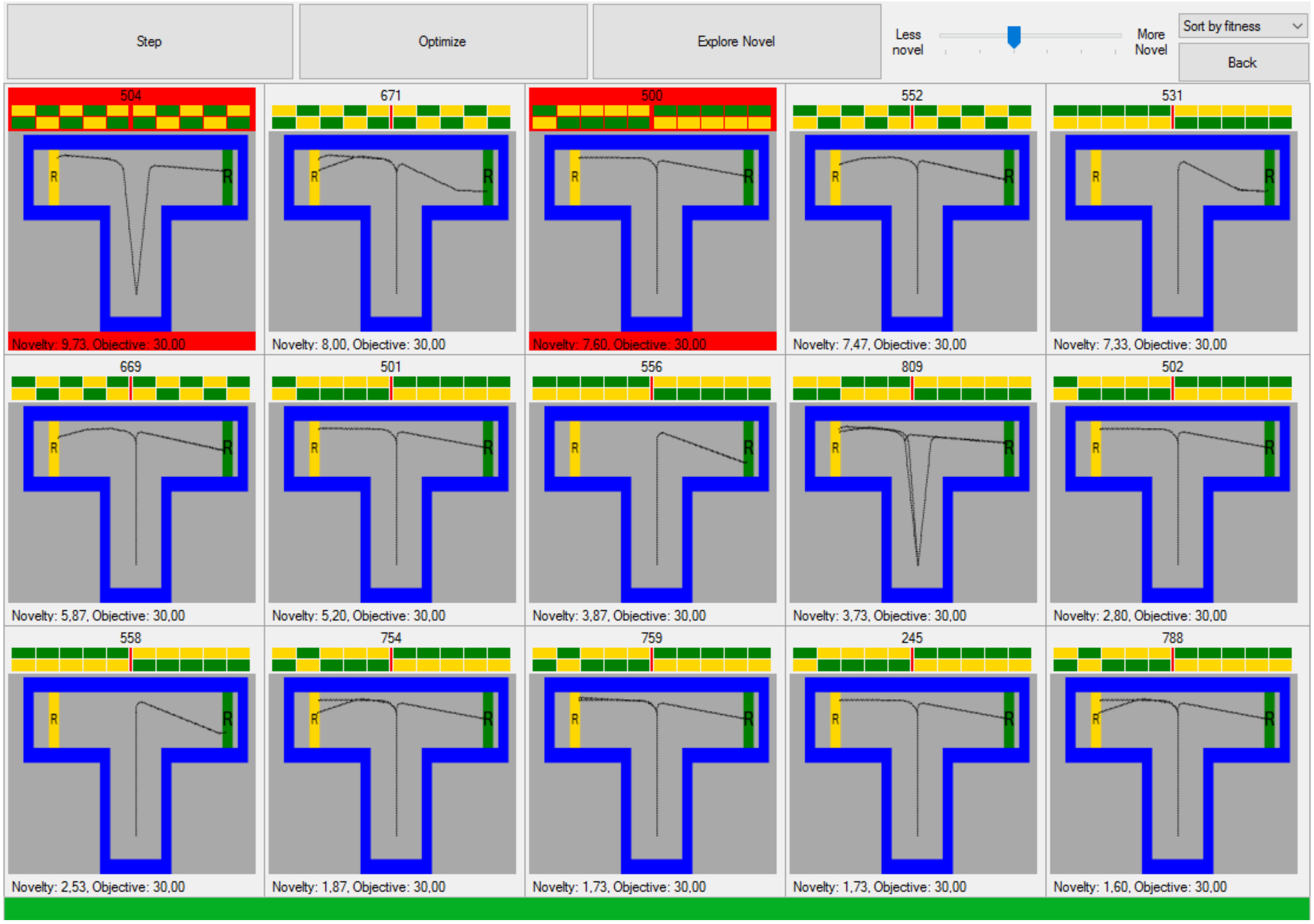


Figure 3: The NA-IEC Framework for the T-Maze Learning Task. The user interface enables the collaboration between a human user and a computer. The behaviors of the agents' across trials are shown as trails together with the type of reward they collected. Selected candidates are marked in red. Unlike in traditional IEC approaches, the user can also perform a short-term novelty search.

the range [0.5, 3.0].

The coefficients for the Hebbian learning rule (Equation 1) were set to $\eta = -94.6$, $A = 0.0$, $B = 0.0$, $C = -0.38$, and $D = 0.0$, following previous work [14, 15, 16].

6. RESULTS

Figure 4 depicts the average number of evaluations it took NA-IEC, novelty search, and objective-based fitness to find a solution. All results are based on 20 independent runs for both novelty and objective-based approaches. A total of 36 sessions were recorded for NA-IEC with ten inexperienced users (Table 3). Fitness-based search found solutions in 60% of the runs after an average of 77,581 evaluations ($\sigma = 23,344$) when successful. Novelty search found a solution in 80% of the runs and with 41,994 evaluations on average ($\sigma = 30,295$). The difference in performance between the two approaches is significant, confirming previous results (*Mann-Whitney U test:* $U = 33, n_1 = 12, n_2 = 16, P < 0.005$ *two-tailed*) [15]. NA-IEC with inexperienced users found a solution in 44% of all sessions after 8,850 evaluations on average ($\sigma = 6,738$). NA-IEC with an expert user found a solution in 93.33% of all sessions after 4,581 evaluations on average ($\sigma = 3,416$). When comparing the performance of

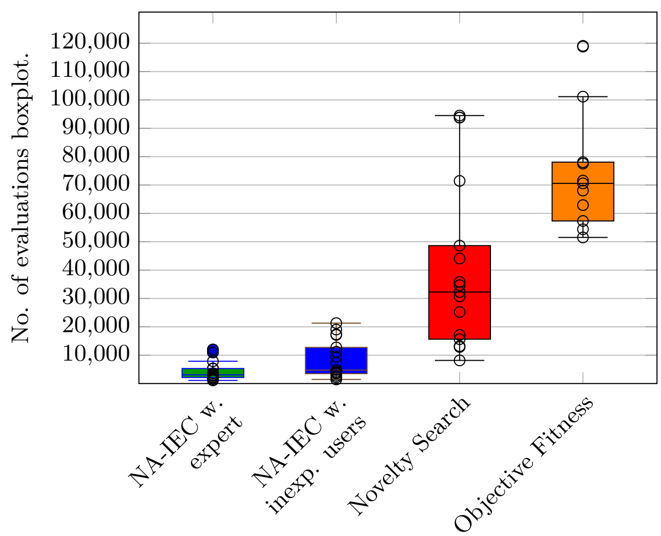


Figure 4: Evaluations to Solution. A boxplot of the number of evaluations it took to find a solution for each approach. When successful, NA-IEC finds solutions significantly faster than any other approach.

Table 3: User Data. Each user's success rate and the average evaluations required to find a solution for the inexperienced users, an expert, and both.

	Valid solutions	**Failures**	**Success rate (%)**
Expert user	14	1	93.33
User 1	1	5	16.67
User 2	4	5	44.44
User 3	1	2	33.33
User 4	2	1	66.67
User 5	1	0	100.00
User 6	3	3	50.00
User 7	1	0	100.00
User 8	2	2	50.00
User 9	1	1	50.00
User 10	0	1	0.00
Total for inexp. users	16	20	44.44
Total including expert	30	21	58.82
Average evaluations for inexp. users	8850.94		
Average evaluations for expert	4581.36		

NA-IEC with inexperienced users with novelty search for the runs they found a solution, the difference is statistically significant ($U = 23, n_1 = n_2 = 16, P < 0.005$) (NA-IEC with expert user: $U = 2, n_1 = 14, n_2 = 16, P < 0.005$). Comparing NA-IEC with fitness-based search, the difference is statistically significant (inexperienced users: $U = 0, n_1 = 16, n_2 = 12, P < 0.005$; expert user: $U = 0, n_1 = 14, n_2 = 12, P < 0.005$). This makes NA-IEC with inexperienced users 8.77 times faster than fitness-based search and 3.89 times faster than novelty search. NA-IEC with an expert user was 16.93 times faster than fitness-based search and 7.52 times faster than novelty search.

To validate that the results were not merely a result of the higher mutation for NA-IEC, the fitness-based search and novelty search ran for 20 times each with the same mutation rates as used in NA-IEC. All approaches performed significantly worse, with fitness-based search failing in 17 out of 20 runs, and novelty search failing in 13. Furthermore, only three of the evolved solutions were able to solve the generalization test as described below.

Using NA-IEC, the wall-clock time it takes to find a solution is also relevant. NA-IEC consumes less wall-clock time than fitness-based search; on average it took NA-IEC 10 minutes and 39 seconds ($\sigma = 0.0063$) for inexperienced users to find a solution, and 2 minutes and 53 seconds ($\sigma = 0.0023$) for an expert user. This is significantly ($U = 34, n_1 = 16, n_2 = 12, P < 0.005$ for inexperienced users and $U = 0, n_1 = 14, n_2 = 12, P < 0.005$ for expert user) faster than fitness-based search with an average wall-clock time of 19 minutes and 46 seconds ($\sigma = 0.0044$). Comparing NA-IEC with novelty, NA-IEC with inexperienced users is twice as fast (2.13) as novelty search. The average wall clock time for novelty was 22 minutes and 11 seconds ($\sigma = 0.03$). The reason why novelty search is slower in terms of wall clock time but not in terms of evaluations, is that it is a rather computationally heavy algorithm.

Woolley and Stanley reported that NA-IEC found significantly simpler ANNs able to solve a navigation task in a deceiving maze. Using NA-IEC in the T-maze domain, inexperienced users found solutions with an average of 11.69 connections ($\sigma = 3.88$). The expert user found solutions

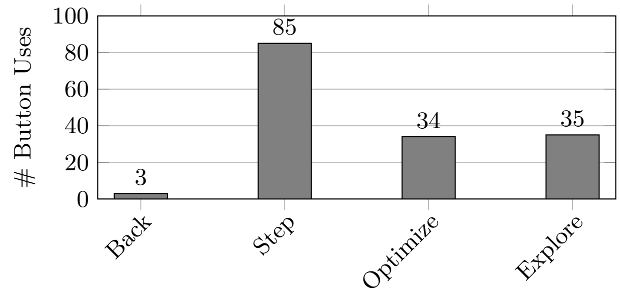


Figure 5: Button Presses. The number of times each button was used during the user session with NA-IEC.

with an average of 10.36 connections ($\sigma = 1.98$). In comparison, novelty search found solutions with an average of 11.47 connections ($\sigma = 6.44$) and fitness-based search found solutions with an average of 20.62 connections ($\sigma = 6.60$). While there is a significant difference in the complexity of the evolved ANNs by NA-IEC and fitness-based search ($U = 15, n_1 = 16, n_2 = 12, P < 0.005$ for inexperienced users and $U = 3, n_1 = 14, n_2 = 12, P < 0.005$ for an expert user), the difference between novelty search and NA-IEC is not significant in any case.

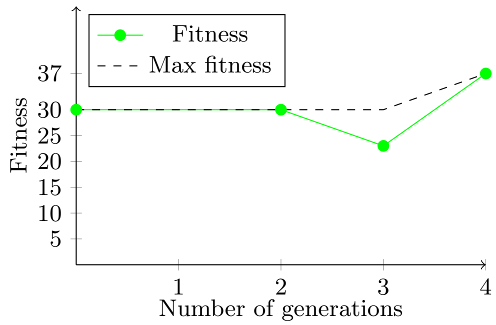
Figure 5 illustrates how often each of the buttons was clicked in total during the experiments with inexperienced users. Unlike previous results with NA-IEC [23], the step button was still the most used button, but it is noteworthy that many users frequently used both of the two sorting options to get a broader view of the population. The back button was only used three times, making it the by far least used button. This could indicate that the user found it better to start over, than going back one generation when getting undesired results. On average, each user had to make 9.63 ($\sigma = 12.2$) evaluations of a population to find a solution.

Figure 6 shows an example of the type of stepping stones towards an adaptive behavior that are discovered by a human but would not be properly rewarded by either fitness or novelty; fitness would discard the chosen champion in the third generation, and novelty would fail to reward any of the chosen champions as other individuals in the population are more novel. Interestingly, the stepping stone in the third generation had the exact opposite behavior of the target behavior (i.e. an agent that learned to always collect the low reward). While it is intuitive for a human user to recognize this behavior as a potential stepping stone, it receives the worst possible fitness value without crashing into a wall.

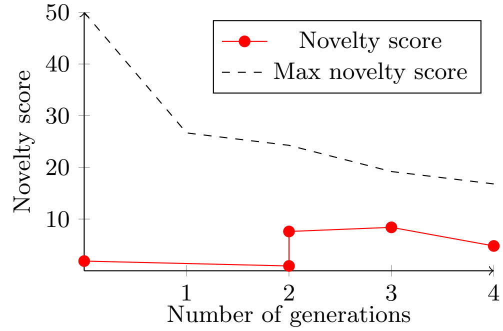
6.1 Generalization Test

Some of the evolved solutions were classified as false positives because they exploited certain aspects of the task instead of displaying the correct adaptive capabilities. One such example is an agent that just counts when to switch sides independently of the collected reward. To discover such false positive solutions, each solution candidate was further evaluated on a generalization test (following Risi et al. [14]). The generalization test consists of three deployments with 20 trails each, in which the rewards switch position after 12 trials in two out of the three deployments. In the last deployment, the rewards never switch position.

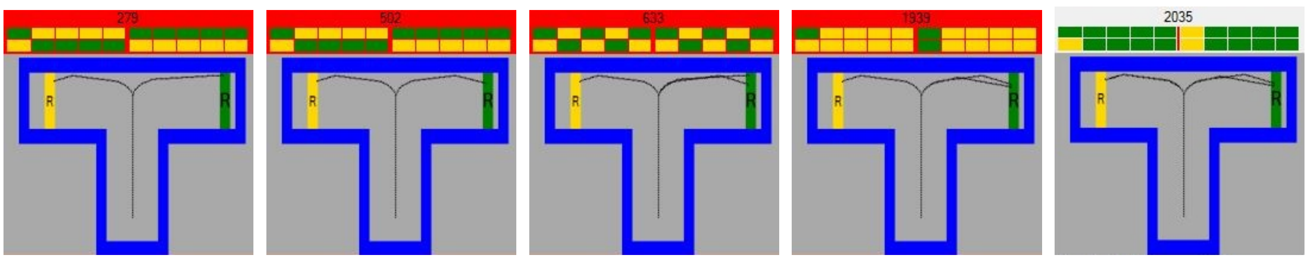
Figure 7 shows the performance of the different approaches on the generalization test. Novelty search succeeds 10% more often than fitness-based search in finding solutions that



The fitness of the behaviors picked by the user, and the highest fitness in the population for that generation.



The novelty score of the behaviors picked by the user, and the highest novelty score in the population for that generation.



Generation 0
Novelty: 1.87, Fitness: 30.00

Generation 2
Novelty: 0.93, Fitness: 30.00

Generation 2
Novelty: 7.60, Fitness: 30.00

Generation 3
Novelty: 8.40, Fitness: 23.00

Solution found
Novelty: 4.80, Fitness: 37.00

Figure 6: The sequence of selected individuals during NA-IEC with an expert user. The last individual in the lower right corner is the solution shown to the user. The main result is that a human user is able to reward stepping stones that lead to the final adaptive behavior that would have been neither chosen by a purely novelty nor fitness-based approach (i.e. humans do neither always choose the behavior with the highest fitness nor the behavior with the highest novelty score).

pass the generalization test. For NA-IEC with inexperienced users, the users succeeded in finding a solution 33.34% of the times, 55.56% of their attempts failed, and 11.12% of their attempts found false positive solutions. This makes NA-IEC with inexperienced users fail the most often of all methods. However, as shown in Table 3 the failure rate among the users is very diverse, some users had a good intuition about promising behaviors, whereas it was harder for others. Furthermore, NA-IEC finds fewer false positive solutions than the automated versions.

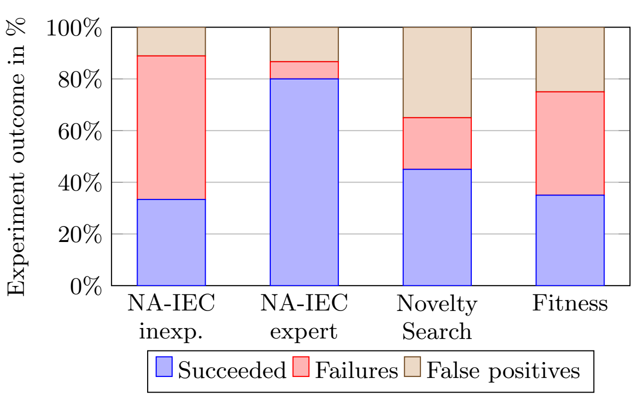


Figure 7: The relative amount of successful runs, failures and false positives.

7. DISCUSSION AND FUTURE WORK

Involving a human in the evolutionary process raises a couple of issues concerning user fatigue, being able to recognize promising behaviors, and understanding the properties of the domain at hand. In the T-maze domain, it is a challenge to illustrate the changing properties of the domain in a way that is easy to understand and fast to evaluate. Admittedly, the results show that NA-IEC with inexperienced users fails more often than both novelty search and fitness-based search. Nonetheless, when an inexperienced user finds a solution, it is done significantly faster in terms of evaluations than novelty search and with fewer false positive solutions. Importantly, the NA-IEC approach is also faster in terms of wall clock time, even though the automated experiments ran on a high-performance cluster, whereas the NA-IEC experiments were executed on a normal consumer desktop PC.

NA-IEC with an expert user reduces the number of failures drastically, making it succeed in almost all cases. While these results are anecdotal and the expert user has not only domain experience but also intimate knowledge of the underlying algorithm, they do hint at the full potential of the platform. The results also stress the importance of experience with the NA-IEC framework, and learning to understand how the dynamic properties of the domain are reflected in the images shown to the user.

While it was not taken fully advantage of by the users in this study, NA-IEC's ability to experiment with one or more behaviors and selection strategies, while being able to revert to a previous generation, gives freedom to explore each individual's potential. Additionally, the ability to sort the population based on fitness or novelty enabled the users to gain an efficient overview of a large population when looking for promising stepping stones. In future iterations of the interface, these aspects could be highlighted more prominently.

This paper supports the claim that NA-IEC is not just applicable to the maze navigation domains investigated by Woolley and Stanley [23]. In the T-Maze task, even inexperienced users were able to recognize potential stepping stones towards adaptive behaviors such as if (1) the agent followed many different paths for each trial, (2) the behavior indicated some response to the change in the reward's location, or (3) the agent was able to show the correct learning behavior in one of the two deployments. These intuitive insights allow the human to guide evolution through the search space in a more informed and efficient way than novelty or fitness-based search could by themselves.

8. CONCLUSION

This paper applied the NA-IEC approach to the evolution of plastic neural networks in a T-Maze learning task. The results show that NA-IEC outperforms both novelty search and fitness-based search both in terms of evaluations and wall-clock time, confirming the hypothesis that humans have critical insights into which stepping stones lead to adaptive behaviors. Also importantly, we show that with training, users can learn to evolve solutions faster and more consistently. In the future, the combination of novelty search together with human intuition might enable the evolution of more complex cognitive behaviors.

References

[1] A. Bernatskiy, G. Hornby, and J. Bongard. Improving Robot Behavior Optimization by Combining User Preferences. *ALIFE 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, 2014.

[2] J. C. Bongard and G. S. Hornby. Combining Fitness-based Search and User Modeling in Evolutionary Robotics. *Proceeding of the 15th Annual conference on Genetic and Evolutionary Computation Conference - GECCO '13*, pages 159–166, 2013.

[3] K. O. Ellefsen, J.-B. Mouret, and J. Clune. Neural modularity helps organisms evolve to learn new skills without forgetting old skills. *PLoS Comput Biol*, 11(4):e1004128, 2015.

[4] W. Gerstner. Hebbian Learning and Plasticity. *From neuron to cognition via computational neuroscience*, pages 0–25, 2011.

[5] C. Green. Sharpneat homepage, 2003-2006. URL http://sharpneat.sourceforge.net/.

[6] R. Kamalian, Y. Zhang, H. Takagi, and A. Agogino. Reduced human fatigue interactive evolutionary computation for micromachine design. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 9, pages 5666–5671 Vol. 9, 08 2005.

[7] J. Lehman and R. Miikkulainen. Overcoming deception in evolution of cognitive behaviors. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 185–192. ACM, 2014.

[8] J. Lehman and K. O. Stanley. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. *Artificial Life XI*, (Alife Xi):329–336, 2008.

[9] J. Lehman and K. O. Stanley. Revising the Evolutionary Computation Abstraction: Minimal Criteria Novelty Search. *Proc. of the International Conference on Genetic and Evolutionary Computation (GECCO'10)*, pages 103–110, 2010. ISSN 09252312.

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

[11] 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.

[12] D. Pallez, P. Collard, T. Baccino, and L. Dumercy. Eye-Tracking Evolutionary Algorithm to minimize user fatigue in IEC applied to Interactive One-Max problem. *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, pages 2883–2886, 2007.

[13] S. Risi and K. O. Stanley. A unified approach to evolving plasticity and neural geometry. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, (Ijcnn):1–8, 2012.

[14] S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 153–160, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9.

[15] S. Risi, C. E. Hughes, and K. O. Stanley. Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18 (6):470–491, 2010. ISSN 1059-7123.

[16] M. Schlesinger, L. Berthouze, and C. Balkenius. Neuromodulation Increases Decision Speed in Dynamic Environments. *Lucs.Lu.Se*, pages 119–126, 2008.

[17] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: Evolving Pictures Collaboratively Online. *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, (Chi):1759–1768, 2008. ISSN 1879-2782.

[18] A. Soltoggio and B. Jones. Novelty of behaviour as a basis for the neuro-evolution of operant reward learning. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 169–176. ACM, 2009.

[19] A. Soltoggio, J. A. Bullinaria, C. Mattiussi, P. Durr, and D. Floreano. Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. 2008.

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

[21] H. Takagi. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275 – 1296, 2001. ISSN 00189219.

[22] M. D. Wagy and J. C. Bongard. Combining computational and social effort for collaborative problem solving. *PloS one*, 10(11):e0142524, 2015.

[23] B. G. Woolley and K. O. Stanley. A novel human-computer collaboration: Combining novelty search with interactive evolution. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 233–240, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2662-9.