To appear: Proceedings of the Genetic and Evolutionary Computation Conference. 2014

Encouraging Creative Thinking in Robots Improves Their Ability to Solve Challenging Problems

Jingyu Li
Evolving AI Lab
Computer Science Dept.
University of Wyoming
Laramie High School
jingyuli@mit.edu

Jed Storie
Evolving AI Lab
Computer Science Dept.
University of Wyoming
jed.storie@gmail.com

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

ABSTRACT

Evolutionary algorithms frequently get stuck on local optima–and fail to find the global optimum–when local gradients do not point the search process toward the direction of the global optimum. A recent breakthrough called Novelty Search ameliorates this problem by enabling the search process to explore in every direction by encouraging the production of novel, or not-yet-seen, phenotypes (e.g. new robot behaviors). However, a problem with Novelty Search is that it can get lost on "novelty plateaus" wherein novel behaviors in offspring are not immediately produced by mutation and crossover (e.g. when a sequence of specific mutations is required to produce new behaviors, but the intermediate mutations are not rewarded because they do not produce novel behaviors). In such cases, Novelty Search and related approaches that reward behavioral diversity can get stuck. Here we introduce a new approach, borrowed from human psychology, that mitigates this problem: encouraging creative thinking. In addition to rewarding novel behavior, we encourage evolving neural networks to "think differently" by rewarding not-yet-seen firing patterns in hidden neurons, which we call the "Creative Thinking Approach." We hypothesize that encouraging novel thinking can reward stepping stones toward new behaviors. On a variety of challenging robotic control problems from previous publications we demonstrate that, as problem difficulty increases, adding the Creative Thinking Approach increasingly improves performance over simply encouraging novel behaviors. Our results suggest that the Creative Thinking Approach could help improve the scale and complexity of problems that can be solved by evolutionary algorithms.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Computing Methodologies:Artificial Intelligence

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598222.

Keywords

Novelty Search, Behavioral Diversity, Evolutionary Robotics, Creative Thinking Approach

1. MOTIVATION

Traditional Evolutionary Algorithms (EAs) have fitness functions that reward organisms more the closer they are to (or the higher they perform on) an objective. EAs often get stuck on local fitness optima, which are regions of high fitness surrounded by regions of low fitness. In such situations, the fitness gradient does not point in the direction that needs to be traversed to find the global optima, preventing further improvements in fitness [8]. In these deceptive problems, being unwilling to explore areas of low fitness prevents evolution from reaching its goal. Take the example of a Chinese finger trap. If we took fitness as maximizing distance between the two fingers, higher fitnesses are achieved by pulling, which is a counter-productive strategy that only more tightly traps the fingers. Instead, one has to first push together the fingers–temporarily lowering fitness–before escaping the trap [14]. The same is true for many problems: individuals must first move away from high fitness areas in order to reach the global goal [14]. It is thus helpful to reward stepping stones *on the path toward* desired solutions, which may not be *in the direction of* the final goal as judged by the fitness function; such stepping stones are often not rewarded by objective-based fitness functions [13, 14].

One approach is to reward stepping stones in deceptive problems by adding helper-objectives that are not directly related to performance, but that guide evolution to produce potentially useful behaviors. For example, it has been shown that adding objectives to promote behavioral consistency (i.e. encouraging similar behavior to be performed in the presence of noise) or reactivity (i.e. performing different behaviors in different scenarios) can increase performance [21, 15].

Most previous techniques for preventing premature convergence on local optima focus on increasing *genomic* diversity [26, 20, 24, 8]. In recent years, researchers have shifted toward rewarding diversity in the space of *behaviors*, such as by adding a behavioral diversity objective to a multi-objective algorithm [18, 6, 17]. A more extreme version of this idea is Novelty Search, which rewards only behavioral novelty and ignores the objective entirely [13, 14]. Both approaches substantially help EAs avoid getting trapped on local optima and often outperform EAs with genomic-based diversity metrics or no diversity pressure at all [18, 6, 17, 13, 14].

Algorithms that reward behavioral diversity or novelty require a *behavioral distance function* that quantifies how different the behaviors of two organisms are. Examples from previous work include the final location of a robot or its trajectory over time [13, 14, 17]. The specific behavioral distance function chosen makes a significant difference, placing a burden on the user to not only pick a behavioral distance function, but to pick a good one. Recent research has focused on creating behavioral distance functions that are generally applicable to many different problems, such as concatenating the entire sequence of inputs to and outputs from a neural network controller and then comparing this sequence to that of another robot [6, 17]. Modern computers and efficient algorithms enable comparisons even in such high-dimensional spaces, and the approach has been shown to work as well as human-designed custom behavioral distance functions [18, 6, 17].

All approaches that encourage behavioral diversity will have a problem if mutations to the current population do not produce novel behaviors. We call this phenomenon a *novelty plateau*. After a prolonged search for novel behaviors, it would not be surprising if the population reached a point wherein the creation of new behaviors required more than one mutation. One can imagine, for example, a situation in which five specific mutations would be required to produce a new behavior, but in which none of the intermediate mutations cause the behavior. In this situation, there is no selection gradient towards having all five mutations, since having four of the five is no better than having none of the five. Such novelty plateaus may significantly hurt the performance of EAs that encourage behavioral diversity, although the extent to which they do so is currently unknown. Of course, neutral genetic drift could lead the search out of a novelty plateau, just as drift could lead a population out of a fitness plateau [11, 23], but drift requires luck in exponentially increasing amounts as the number of specific mutations required to exit the plateau grows. For the same reason you may want to encourage behavioral diversity in objective-based search when there is no helpful gradient toward the objective, it could also be helpful to encourage thought diversity in behavior-based search when there is no behavioral diversity gradient. As seen in psychology, being stuck on a problem is a result of being locked into a certain way of thinking and thus new and creative thinking plays a large factor in getting unstuck on problems [22].

2. METHODS

2.1 The Creative Thinking Approach

To test the idea of encouraging thought diversity, we introduce the Creative Thinking Approach (CTA), which encourages robots to "think differently" than robots that have come before in the same run. Of course, suggesting that robots controlled by neural networks "think" at all is controversial: we only use the term "think" metaphorically. To the extent that neural networks do think in the most loose sense of the word, in that they perform computations to process inputs and generate outputs, our algorithm encourages them to think differently. In less metaphorical and more technical language, rewarding "creative thinking" simply means rewarding novel patterns of neural firing. In a nutshell, then, our algorithm rewards novel thoughts in addition to novel behaviors.

Technically, our work is an extension of the general behavioral diversity mechanism from Doncieux and Mouret [6, 17]. For the general behavioral diversity mechanism, over the course of a robot evaluation, all of the robot's inputs and outputs from each time step (or a sub-sample of time steps) are stored in a binary vector to represent the behavior of the individual (values > 0 are stored as 1, otherwise 0). This method is general because a user does not have to design a problem-specific behavioral characterization (such as the number of doors opened): the method will provide a behavioral metric and distance function for any problem in which a robot has inputs and outputs. The behavioral distance between two organisms is the Hamming distance between their binary vectors, which is computationally efficient to calculate with modern algorithms. The Hamming distance is the number of bits that differ between two binary sequences. This behavioral distance factors into a behavioral diversity objective in the NSGA-II multi-objective algorithm [4]: the other objective is performance on the task. Briefly, multi-objective algorithms seek to improve performance on all objectives in a way that maintains a diverse set of solutions that perform well to varying degrees on all objectives. For a review of multi-objective algorithms, the reader is directed to [5, 3].

In this paper, the general behavioral diversity mechanism just described [6, 17] is compared to the Creative Thinking Approach. Because that general behavioral diversity mechanism looks only at behavior (characterized by input and output values) [6, 17], we refer to this control as the *behavioral diversity only* (BDO) treatment.

To implement the Creative Thinking Approach, we extend the BDO method by adding the firing patterns of the hidden neurons to the binary vector describing the robot's behaviors (Figure 1). Note that the CTA is a superset of the BDO, in that it includes the behaviors (characterized by inputs and outputs) and adds to that a consideration of the firing patterns of hidden neurons. Thus, two robots that have identical inputs and outputs, but have different neural firing patterns, will be recognized as being different and thus rewarded. While technically a small change, there are consequences to consider. By adding hidden neurons, the computational cost of calculating the difference between two organisms increases. In our problems, this additional cost is minor compared to the cost of running the simulation to evaluate organisms, although it could become an issue for faster problem domains or larger neural networks. For the problems in this paper, we did not notice that the CTA ran slower than BDO. An additional complication is how to compare the hidden neurons in neural networks with different topologies, as can be the case if the topology evolves. To avoid that complexity in this first paper on the Creative Thinking Approach, the topology of the neural networks is fixed. In the discussion we propose ways of dealing with variable topologies.

Following [17], the neural network controllers for our experiments are Elman-Jordan networks [7], which are simple recurrent networks. The networks are represented with a direct encoding [27, 2].

Evolution was conducted in the Sferes2 evolutionary computation framework [16]. As previously mentioned, all runs feature the NSGA-II [4] multi-objective evolutionary algo-

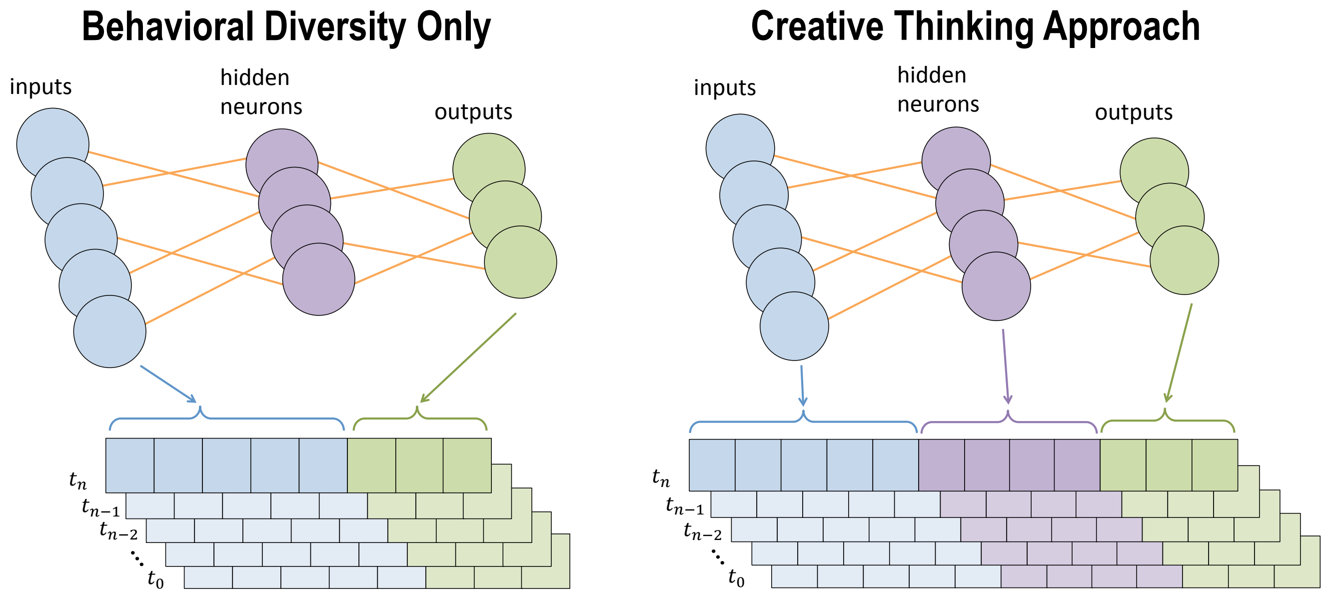

Figure 1: Comparing the behavioral diversity only (BDO) method from [6, 17] to the Creative Thinking Approach (CTA) proposed in this paper. At each time step t, a vector of the robot's input and output values for that time step are concatenated to a large vector that characterizes the robot's behavior.

rithm with two objectives: performance and diversity (either BDO or CTA). All statistics are performed with the non-parametric Mann-Whitney Wilcoxon test. Plots show medians (lines) and 95% bootstrapped confidence intervals of the median (shaded areas), which are calculated by resampling the data 5000 times. Confidence intervals are smoothed by a median filter with a window size of 101 to reduce noise. Asterisks below each plot indicate if the Creative Thinking Approach performs significantly better than the behavioral diversity only approach at that generation. The code and data for all experiments are available upon request.

3. EXPERIMENTS AND RESULTS

3.1 Easiest Problem: The Deceptive Maze

In this experiment, the objective is to evolve a neural network that is able to navigate its way through the maze from the start to the goal (Fig. 2). The problem is from Mouret and Doncieux 2012 [17] and is inspired by the deceptive "Hard Maze" from Novelty Search papers [14, 13]. It is deceptive because the performance objective rewards organisms for getting closer to the goal, but doing so results in getting trapped in the dead end above the start. While hard for traditional, objective-only EAs, solving this maze is relatively simple for EAs that do encourage behavioral diversity [17, 14, 13].

The neural network robot controllers for this problem have 7 input (sensor) neurons and two output neurons (Fig. 3). Sensors include three range sensors, which provide the normalized distance to the nearest obstacle in the direction that sensor points, and four pie-slice goal sensors that fire if the goal is within their purview, irrespective of intervening obstacles, and thus serve as a compass toward the goal. The two output neurons control the speed of each wheel. There are 7 hidden neurons. The experiment is run 50 times with a population size of 200 for 1500 generations.

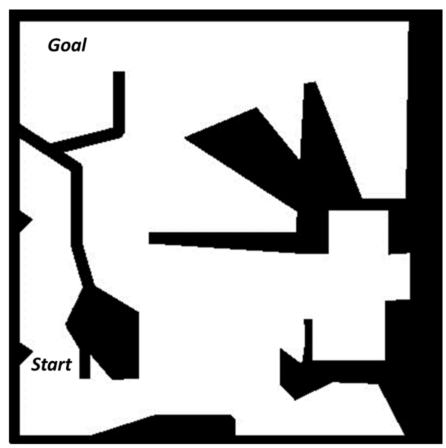

Figure 2: The Deceptive Maze Environment

The performance measure rewards getting closer to the goal [17]. Specifically, it is

$$performance = 1 - \frac{distance\ to\ goal}{size\ of\ maze} \tag{1}$$

On this problem there is no significant difference between the treatments ($p > 0.05$) and both treatments solve the problem in less than 1500 generations (Fig. 4). We hypothesize that the problem is not difficult enough for the Creative Thinking Approach to make a difference, and it may in fact slow the search process down by diluting the selection pressure on generating easily-produced behavioral diversity. We next investigate whether the Creative Thinking Approach is more advantageous on a harder problem.

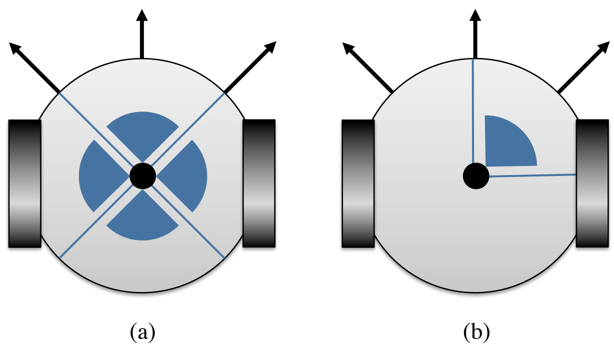

Figure 3: The robots and their sensors. (a) The maze navigating robot has three distance-to-wall sensors (arrows) and four pie-slice "compass" sensors (blue wedges) that indicate the direction of the goal. The angular range of the pie-slice sensors are shown; these sensors can see through walls, acting as a compass toward the goal. (b) The ball collecting robot has three distance-to-wall sensors (arrows), two pie-slice sensors that detect the presence of balls, and two similar sensors with the same range that detect the basket. The range of the right pie-slice sensors is shown; the left pie-slice sensors are symmetric. See text for a full description of all sensors. Figure adapted from [17].

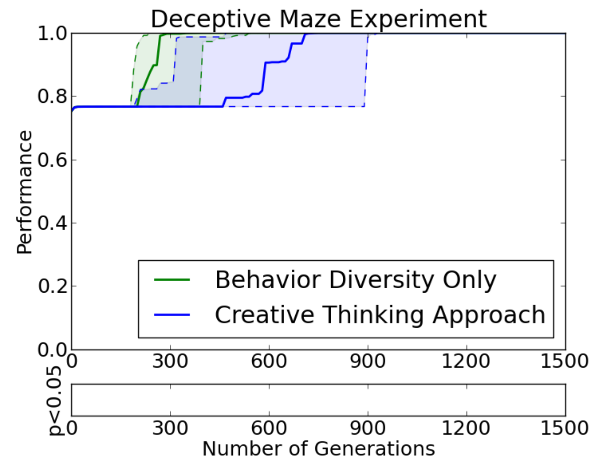

Figure 4: Results for the Deceptive Maze problem. In this and all subsequent plots, solid lines are medians over 50 runs and colored regions indicate 95% bootstrapped confidence intervals of the median. Asterisks appear in the lower panel if the treatments are statistically significantly different from each other with p $<$ 0.05.

3.2 Harder Problem: Ball Collection

In this problem—from [6, 17]—a robot must navigate its way around an unknown environment, pick up a ball, place it in an illuminated area, and then repeat that process for eleven other balls (Fig. 5). A trial consists of three sub-trials. In each sub-trial, the robot starts at a different location and must collect four balls, the initial positions of which are the same for each sub-trial[1]. The three different starting locations for the robot are shown in Fig. 5. Those three different starting locations are the same for all subsequent versions of the ball collection problem, but are not pictured in subsequent figures to reduce clutter. For this version of the problem, trials stop at 9000 evaluations (3000 evaluations for each sub-trial). The setup, including the robot's neural network, is reset before each sub-trial.

Performance on this problem is measured as the fraction of balls placed in the lighted area. Note there is no pressure on speed or efficiency aside from the fact that being efficient may enable collecting more balls. In this version of the problem, the sequence in which balls are collected does not matter. If a ball is dropped outside the lighted region, the robot will be unable to pick it up again and fails to collect that ball.

There are 10 inputs to the neural network (Fig. 3) [17]: three laser range sensors that return normalized distances to the nearest obstacles, two bumpers that return a 1 when touching an obstacle and 0 otherwise, two basket-presence sensors that detect the basket when in the angular range of its sensors (Fig. 3) and not blocked by obstacles (contrary to the maze navigating robot, this robot cannot see through walls), two ball presence sensors with the same range and properties as the basket sensors, and one ball-carrying sensor that returns a 1 if the robot is carrying a ball and 0 otherwise. The robot has 3 outputs (Fig. 3): two that control the speed of each wheel and a third that controls an effector that can pick up a ball. To pick up a ball this output must be above 0.5 when the robot is over the center of the ball. Once a ball is collected, it is dropped any time this effector output drops below 0.05. There are 10 hidden neurons and 10 context neurons. Each input neuron is connected to all hidden neurons. Each hidden neuron is connected to exactly one context neuron and each context neuron is connected to itself and all hidden neurons. The recursive nature of these connections enables memory because the networks can maintain data from previous states. Each hidden neuron is connected to all output neurons [7, 17]. The population size is 200 individuals. In each experiment, both treatments are run 50 times. All other parameters are identical to those in [17].

This problem is hard because no reward is given unless a robot navigates to a ball, closes its gripper to pick the ball up, keeps that gripper in the closed position while navigating to the lighted area, and releases the ball in the lighted area. Because none of these intermediate stepping stone behaviors are rewarded via the performance objective, evolutionary algorithms that are guided by only performance never solve the task [17].

The BDO and CTA treatments in this paper, which both reward for behavioral diversity, achieve perfect performance on this problem on average (Fig. 5). The increased difficulty of the problem is reflected in the larger number of generations required to solve it compared to the Deceptive Maze Problem (compare the range of the y axes of Fig. 4 and Fig. 5) and the fact that some runs in both treatments do not solve the problem after 15000 generations. On this more difficult problem, the addition of the Creative Thinking Ap-

[1] A video showing some of the evolved robot behaviors can be viewed at `EvolvingAI.com`

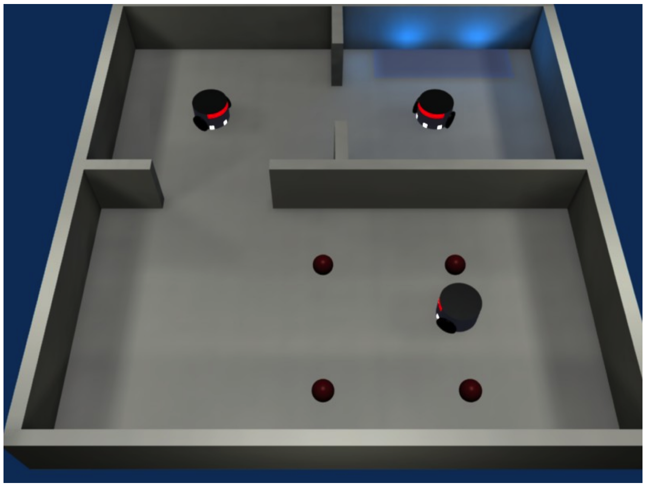

Figure 5: The Ball Collection Problem. The robot must collect balls one at a time and place them in the lighted area. There are twelve balls in total (four pictured). The robot's three starting locations (different location for each sub-trial) are shown. There are four balls per sub-trial whose initial locations are the same for all sub-trials.

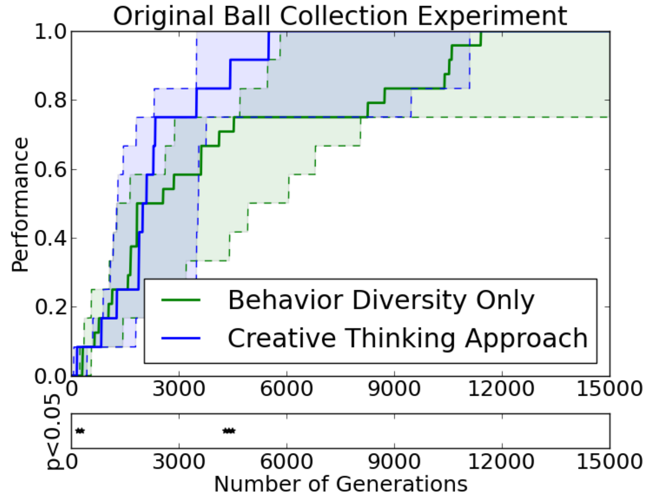

Figure 6: Results for the ball collection problem. Asterisks below the plot indicate generations in which the Creative Thinking Approach (CTA) performs significantly better than the behavioral diversity only (BDO) approach.

proach improves performance, although there is rarely any statistically significant advantage (Fig. 6).

3.3 An Even Harder Problem: Sequenced Ball Collection

To increase the difficulty further, in this problem a reward is only given if balls are placed in the lighted area in a predefined order (shown in Fig. 7). This order is the same for the three sets of four balls. On this more difficult problem, the addition of the Creative Thinking Approach improves performance, and does so significantly from around 38000 to 53000 generations (Fig. 8).

3.4 Another Hard Problem: The Balls in Corner Problem

To create another difficult version of the ball collection problem, we placed the balls in a compact area in a corner (Fig. 9). Doing so means that the robot is far less likely to encounter a ball by chance, for three reasons. First, the balls are in a smaller area, making it less likely for a robot to randomly encounter one of them. Secondly, in the default version of the problem, the balls surround the robot for one of its three starting positions (Fig. 5). While the robot's three starting positions remain the same for this version of the problem, because the balls are in the corner they no longer surround the robot, meaning that most directions from that starting location will not immediately lead to collecting a ball. Third, it is much less likely for a robot conducting a random or semi-random walk to make it deep into a corner due to collisions with walls. All three reasons make it less likely for the robot to initially, randomly encounter a ball, which greatly increases the difficulty of the problem. Recall that no fitness is gained until a ball is collected and placed in the lighted area, so there is no chance of a robot receiving any performance feedback if it does not at least first encounter a ball (let alone the other things it needs to do after picking a ball up).

On this difficult problem, the advantage of adding the Creative Thinking Approach is statistically significant at nearly every generation (Fig. 10). The median CTA run solves the problem in around 15000 generations, whereas the median run in the BDO treatment never solves the problem.

4. DISCUSSION

An objection to CTA may be its extra computational cost, especially as the size of the neural network and length of the evaluations increase. This same objection was mentioned as a potential problem with the behavioral diversity only (BDO) approach [6, 17], which includes all input and output values in the behavioral characterization [6, 17]. However, there are reasons why this objection is not a major concern. There are efficient algorithms for calculating Hamming distances between large vectors, and usually the computational cost of fitness evaluations dominates this cost [6, 17]. Moreover, while adding hidden neurons to the behavioral characterization increases the overall vector size, the number of hidden neurons is usually only a small factor larger than the number of input and output neurons, at least for most neural networks currently experimented with in the field of Evolutionary Robotics. As such, the algorithm is not slowed down exponentially. However, this concern could become a factor for neural networks with many times more hidden neurons than input and output neurons.

Techniques can also be applied to reduce computational costs while maintaining benefits of CTA. We can sample only a subset of neurons and/or time points. Additionally, techniques can be used to reduce the dimensionality of the data in the hidden-neuron vector, such as Principal Components Analysis [12], Independent Components Analysis [10], or more modern machine learning techniques [19, 1].

Even if the approach is computationally slower, it is better to have an approach that will eventually succeed if given

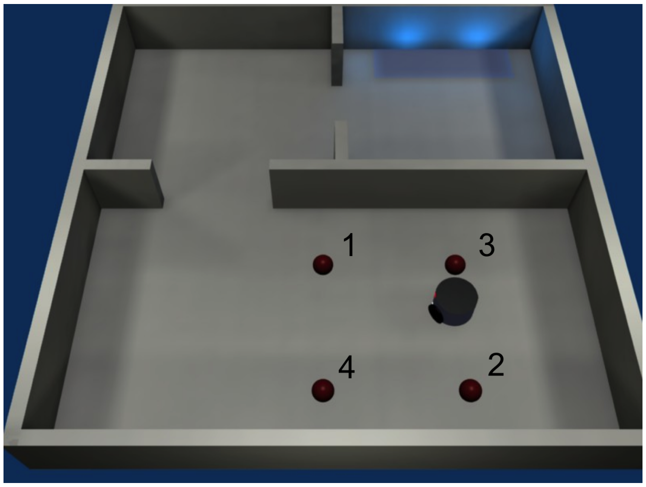

Figure 7: Sequenced Ball Collection Problem. Robots must collect the balls and place them in the lighted area in a predefined sequence. There are 12 balls in total (four pictured). The sequence for each set of four balls per sub-trial is the same.

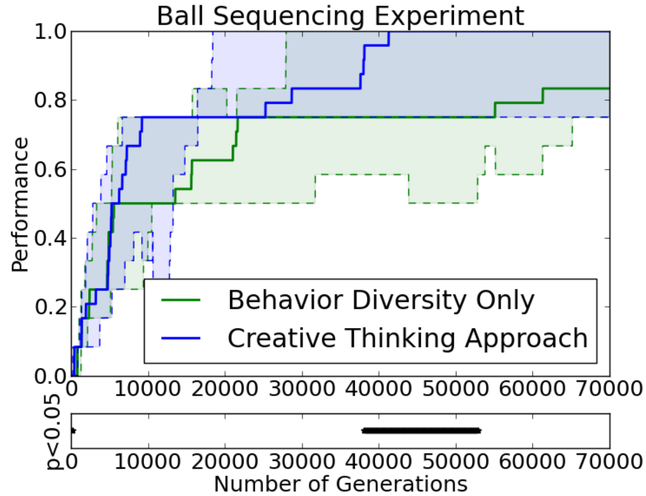

Figure 8: Results from the Sequenced Ball Collection problem.

enough computation, rather than an algorithm that will spin its wheels indefinitely. Similar arguments have been made for Novelty Search [14]. As such, if the CTA approach does increase the ability to solve some challenging tasks, as it appears to, it seems worthy of future investigation. Moreover, that the CTA can solve some problems in fewer generations than BDO suggests that it may actually save computation (Figs. 8 and 10). While these comparisons were made on generations, not actual running time, we did not notice that CTA ran slower: the fitness evaluations and the idiosyncrasies of evolutionary runs were the dominant factors that determined computational costs.

Another challenge for the Creative Thinking Approach is dealing with algorithms that evolve the number of hidden nodes, such as the NeuroEvolution of Augmenting Topologies algorithm [26]. Fortunately, that very algorithm offers a technical solution that makes it easy to conduct a principled comparison of vectors of hidden node activity even when those vectors are of different lengths. The answer is NEAT's historical marking scheme, which can keep track of which hidden neurons should be considered the same, from a historical (phylogenetic) perspective [26]. Explaining the details of that algorithm are beyond the scope of this paper, but NEAT's historical marking scheme, which can be applied to any topology-evolving neural network algorithm, should enable the CTA to align hidden node activation vectors in a principled fashion before calculating the Hamming distance between them. Neurons that are phylogenetically new in one network–and thus do not exist in the comparator network–can be compared to binary slots always set to zero in the comparator network's vector.

Another objection could be that the CTA is simply carrying out an exhaustive search. Importantly, however, while the CTA is exhaustively searching in neural firing space, that is far different from exhaustively searching in the much larger genomic search space. In fact, papers on the behavioral diversity only approach found that BDO worked as well as hand-designed behavioral characterizations that were much lower-dimensional [6]. That is because algorithms that reward behavioral diversity, like Novelty Search, search in the space of behaviors, which is not exhaustive search in the genome space [14, 13]. Like those algorithms, the CTA approach encourages evolution to learn how to produce new patterns of hidden neuron activity (in addition to behavior), which is a challenging space in which to search. It is not possible to simply search directly in the space of neural firing, because it is not known ahead of time which neural network topologies will lead to which patterns of behavior and hidden neuron firing. That problem is especially true when evolving neural networks with generative encodings, which feature an indirect mapping from the genome to a neural network, such as the popular HyperNEAT algorithm [25, 2, 9]. Thus CTA is not an exhaustive or random search in the extremely large genotype space (or even neural network space, for generative encodings). Rather, it searches for genomes that tend to produce novel thinking and novel behavior when mutated.

5. FUTURE WORK

As seen on the deceptive maze problem (Fig. 4), some problems do not require the Creative Thinking Approach and may be slowed down by it. It is thus useful to adjust the weight of rewards for diversity in hidden neurons accordingly. On harder problems, we see that the CTA and BDO are similar in performance (Figures 6, 8, 10) in early generations. We assume this is because generating novel behaviors is easier in early generations. We hypothesize that increasing the weight for diversity in hidden neurons as generation number increases will improve performance. Another, perhaps more informed and thus more effective approach, is to increase the reward for hidden neuron diversity as an inverse function of the behavioral diversity in the current generation. In other words, if the search is currently able to generate novel behaviors, do not add a reward for novel thinking; if the search for novel behaviors stagnates, increase the weight for creative thinking. Preliminary experiments suggest these approaches improve performance and in future work we will investigate them more thoroughly.

Another direction of future work is to investigate the Cre-

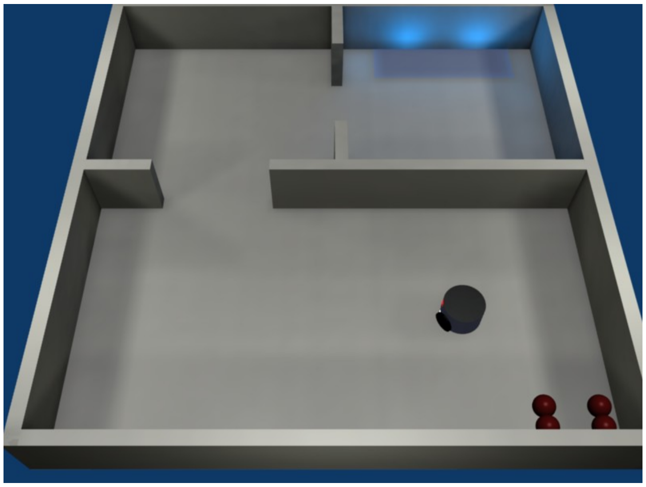

Figure 9: A harder version of the collection problem in which balls are hidden in the corner. Placing the balls in the corner makes the problem hard for three different reasons (see text), significantly increasing the problem difficulty.

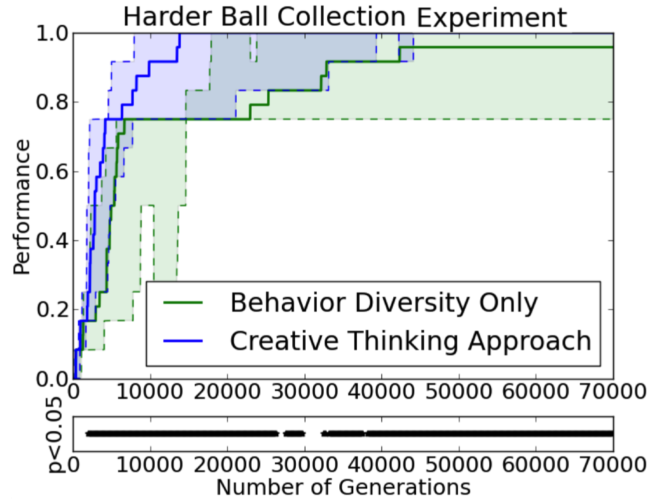

Figure 10: Results for the harder version of the ball collection experiment in which balls are hidden in the corner (Fig. 9). The addition of the Creative Thinking Approach significantly improves performance at nearly every generation.

ative Thinking Approach with recurrent neural networks on problems where remembering information would be helpful, but does not provide an immediate benefit. On such problems, the CTA could be a successful way of encouraging the search process to try creating robots that remember different things.

6. CONCLUSION

This paper introduced the Creative Thinking Approach, which encourages different patterns of firing in hidden neurons in addition to rewarding for novel behavior. In other words, thinking of neural networks as models of animal brains, it encourages robots to exhibit novel or creative thinking to help solve problems. We showed that, on challenging problems, the Creative Thinking Approach significantly outperforms a similar algorithm that rewards behavioral diversity only. We also introduced the concept of a novelty plateau, wherein the search process has no gradient to follow to produce new behaviors. Such novelty plateaus may help explain why rewarding creative thinking can help a search process: encouraging creative thinking rewards stepping stones on the path to generating new behaviors.

7. ACKNOWLEDGMENTS

We thank Jean-Baptiste Mouret, Stéphane Doncieux, and the members of the Evolving AI lab for their helpful roles in this research, especially Joost Huizinga.

8. REFERENCES

[1] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] J. Clune, K.O. Stanley, R.T. 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.

[3] C.A. Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, Feb 2006.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, April 2002.

[5] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[6] S. Doncieux and J.-B. Mouret. Behavioral diversity measures for evolutionary robotics. In *WCCI 2010 IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation (CEC)*, pages 1303–1310, 2010.

[7] J.L. Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

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

[9] J. Gauci and K.O. Stanley. Autonomous evolution of topographic regularities in artificial neural networks. *Neural Computation*, 22(7):1860–1898, 2010.

[10] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Netw.*, 13(4-5):411–430, May 2000.

[11] T. Jansen and I. Wegener. Evolutionary algorithms - how to cope with plateaus of constant fitness and when to reject strings of the same fitness. *Trans. Evol. Comp*, 5(6):589–599, December 2001.

[12] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag.

[13] J. Lehman and K.O. Stanley. Exploiting open-endedness to solve problems through the search

for novelty. In *Proceedings of Artificial Life XI*, volume 11, pages 329–336, 2008.

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

[15] J. Lehman, S. Risi, D.B. D'Ambrosio, and K.O. Stanley. Rewarding reactivity to evolve robust controllers without multiple trials or noise. In *Proceedings of the Artificial Life Conference*, volume 13, pages 379–386, 2012.

[16] J.-B. Mouret and S. Doncieux. SFERESv2: Evolvin' in the multi-core world. In *WCCI 2010 IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation (CEC)*, pages 4079–4086, 2010.

[17] J.-B. Mouret and S. Doncieux. Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evolutionary Computation*, 1(20), 2012.

[18] J.-B. Mouret and S. Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 1161–1168. IEEE, 2009.

[19] K.P. Murphy. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, August 2012.

[20] E.S. Nicoară. Mechanisms to avoid the premature convergence of genetic algorithms. *Petroleum - Gas University of Ploiesti Bulletin, Mathematics - Informatics - Physics Series*, 61:87–96, 2009.

[21] C. Ollion, T. Pinville, and S. Doncieux. With a little help from selection pressures : evolution of memory in robot controllers. *Artificial Life*, 13:407–414, 2012.

[22] T. Rickards. *Creativity and Problem Solving at Work*. Gower Publishing, Ltd., 1997.

[23] A. Rogers and A. Prügel-Bennett. Genetic drift in genetic algorithm selection schemes. *IEEE Transactions on Evolutionary Computation*, 3(4):298–303, October 1999.

[24] C. Ryan. Reducing premature convergence in evolutionary algorithms, 1996.

[25] K.O. Stanley, D.B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

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

[27] K.O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.