

# Neuroevolution in Dynamically Changing Environments

Jory Schossau<sup>1, 2</sup> and Arend Hintze<sup>2,3</sup>

<sup>1</sup>Department of Integrative Biology, Michigan State University, MI, United States of America

<sup>2</sup>BEACON Center for the Study of Evolution in Action, Michigan State University, MI, United States of America

<sup>3</sup>Department of Complex Dynamical Systems and MicroData Analytics, Dalarna University, Dalarna, Sweden  
ahz@du.se

## Abstract

One goal of the Artificial Life field is to achieve a computational system with a complex richness similar to that of biological life. In lieu of the knowledge to achieve this, Open-ended evolution is often cited as a promising method. However, this is also not straightforward because it is unknown how to achieve open-ended evolution in a computational setting. One popular hypothesis is that a continuously changing fitness landscape can drive open-ended evolution toward the evolution of complex organisms. Here, we test this idea using the neuroevolution of neural network foraging agents in a smoothly and continuously changing environment for 500,000 generations compared to an unchanging static environment. Surprisingly, we find evidence that the degree to which novel solutions are found is very similar between static and dynamic environments.

## Introduction

Open-ended evolution presents a major challenge for computational evolutionary models (Bedau et al., 2000). The open-ended evolution paradigm for novel solution discovery is quite different from other computational approaches of novel discovery, such as genetic algorithms or genetic programming, because those methods perform optimization toward a well-defined – possibly complex – fitness optimum. Open-ended evolution provides only an abstract fitness optimum, such as “be more complex but also do better at meaningfully existing” and that is difficult to quantify or define. Natural systems appear to evolve indefinitely – catastrophic events aside. The specific characterization of such open-endedness has led to much debate already (Taylor et al., 2016; Juric, 1994; Maley, 1999; Channon et al., 2000; Channon, 2006; Soros and Stanley, 2014; Taylor, 2015). While an open-ended system may be trivial to implement (Channon, 2006; Hintze, 2019), and we know various algorithms to discover novel solutions (Lehman and Stanley, 2011), what these systems struggle with is to produce a continual increase in complexity that is also relevant to the research question.

A simpler way of creating novel challenges for adaptation may be changing environments: As selection pressures change, agents must keep adapting. The most naïve im-

plementation is to repeatedly switch between two environments over the course of evolution. However, we previously found that if environmental change occurs much faster than evolutionary adaptation, then the populations stop adapting and settle for a generalist neural structure (CG et al., 2018). In this kind of meta-adaptation, even a dynamic environment does not lead to open-ended neuroevolution. At the other end of the speed-change scale where changes happen rarely, the populations adapt as if in a static environment, and quickly lose neural structures common to both environments. In such a slowly changing environment, populations discover an optimum at which they remain until the environment changes again. This is also not a form of open-ended evolution, because a specific fitness maximum exists that could have also been found through other more direct optimization methods.

The question remains if we can create an environment that is evolutionarily predictable enough to discourage meta-adaptation, but unpredictable enough to discourage static adaptation *and* repeated discovery of the same solutions? Here, we create a spatial foraging environment wherein agents controlled by Markov Brains – a form of evolvable neural network not unlike Cartesian genetic programming (Miller and Harding, 2008) – must adapt to ever-changing environmental conditions (Hintze et al., 2017).

## Environment

The environment is a 2d lattice of  $1024 \times 1024$  tiles on which agents forage for food. Specifically, 10,000 pieces of four different types of food each are randomly distributed at the start of every generation. In addition, a barrier surrounds the environment and another 10,000 random locations are converted to barriers. A population of 100 agents is also distributed randomly. Agents can rotate or move in four cardinal directions (N, S, E, and W). At every step agents are given sensory information about the tile in front of them, then allowed to compute their action: turn left or right, move forward and eat, or do nothing. To resolve conflicts, agents are executed in sequential order.

When agents move to a food tile they also consume it. However, the various food types each reward agents differ-

ently depending on the current generation, and on the 1-step history of food previously consumed. The first food consumed does not provide a reward. For subsequent consumption, a matrix  $R$  defines the reward. Each element of the matrix  $R_{(i,j)}$  specifies a reward for consuming a particular food  $f$  after consuming food  $i$ . Values of  $R$  are bounded  $\forall_r \in R \in [-1, 1]$  making certain combinations of consumed resources very rewarding and others punishing.

The reward values  $R_{(i,j)}$  are newly defined every generation as:

$$R_{(i,j)} = \text{triSin}(t(\alpha_{(i,j)} + 0.5) + 2\pi\beta_{(i,j)}), \quad (1)$$

where the time  $t$  is the product of the current generation  $g$ , and a scaling factor  $s$ :  $t = gs$ . The matrices  $\alpha$  and  $\beta$  are randomly generated at the start of evolution and populated with random numbers sampled uniformly in the range  $[0, 1]$ . The `triSin` is the triangle wave function approximated using additive synthesis of `sin` function with  $N = 10$  iterations. The  $\alpha$  and  $\beta$  factors modulate the period as well as the  $y$ -intercept. The scaling factor  $s$  determines the speed with which the rewards change. While other periodic functions like `sin` or `cos` could be used, the triangle wave function has the advantage to have a uniform value function, which creates a uniform parameter space for the  $R$  matrix. As such, there should be no predictable irregularities of  $R$  for evolving strategies to exploit. Evolution proceeds through roulette-wheel selection, site-mutation rate of 0.001, with a duplication and deletion rate of 0.1 per genome, where stretches of 128 to 512 sites (sampled from a uniform random distribution) will be copied or deleted) with constant population size.

### Discovered Novelty

Agents are controlled by Markov Brains using 4 input sensors, 4 output actuators and 24 hidden states. They evolved as populations of 100 individuals for 500,000 generations. Two environmental conditions were tested using 50 independent replicate experiments. One condition used a changing environment with  $s = 0.001$ , and the other was static with  $s = 0$ . At the end of the experiment we reconstructed the line of descent (Lenski et al., 2003) and the topology of the evolved Markov Brains. Each of these brains has a connectivity matrix of identical dimensionality, and we compared their values using the Frobenius distance (Weisstein, 2003).

To determine the novelty arising from evolution, we use the concept defined in Novelty Search (Lehman and Stanley, 2008), specifically the idea of keeping an archive of unique solutions by some metric. Here, a solution produced by evolution is the brain connectivity matrix. The first 100,000 agents discovered by evolution seed the initial archive. A new solution is considered novel and added to the archive if it has a shortest distance to all members of the archive that is greater than the shortest distance between all current members of the archive. Contemporaneously, the solution with

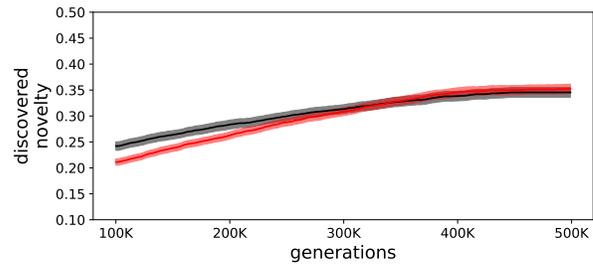


Figure 1: Average discovered novelty for 50 replicate evolutionary experiments in a dynamic environment (red) compared to a static environment (black). The areas indicate the standard error. Observe that the plot begins with generation 100,000, before that all solutions are allowed in the archive and thus no meaningful measure of discovered novelty can be derived.

the shortest intra-archive distance is removed. In this way, the archive defines the space of all discovered novelty. We quantify this space by computing the mean distance between all members of the archive.

### Results

We find that the size of solution space discovered in a dynamic environment compared to the size discovered in a static environment is approximately the same (see Figure 1). At the same time, we find that under both experimental conditions the space of discovered solutions becomes saturated around 400,000 generations, and no more novel solutions are found. This saturation point – as well as the average distance between solutions – is also far from the theoretical maximum of a fully explored solution space (the average distance between randomly generated connection matrices is large, data not shown).

### Discussion

We expected that a continuously changing environment would produce significantly more novel solutions than a constant one. Surprisingly, both conditions explore the solution space approximately the same after 400,000 generations. Clearly, more tests with different rates of change are necessary to arrive at a definite conclusion. It is possible that the speed of change chosen is still too fast to allow proper adaptation to each new environment. However, the current results indicate that ever-changing environments do not allow for the continuous discovery of novel neural network structures.

### Acknowledgements

This work was in part sponsored by the BEACON Center for the Study of Evolution in Action NSF Cooperative Agreement No. DBI-0939454. The computations were performed on resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project SNIC 2020/15-18.

## References

- Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., and Ray, T. S. (2000). Open problems in artificial life. *Artificial life*, 6(4):363–376.
- CG, N., Lundrigan, B., Smale, L., and Hintze, A. (2018). The effect of periodic changes in the fitness landscape on brain structure and function. In *Artificial Life Conference Proceedings*, pages 469–476. MIT Press.
- Channon, A. (2006). Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment. *Genetic Programming and Evolvable Machines*, 7(3):253–281.
- Channon, A., Maley, C., and Boudreau, E. (2000). Three evolvability requirements for open-ended evolution. In *Artificial Life VII Workshop Proceedings*, pages 39–40. Portland, OR.
- Hintze, A. (2019). Open-endedness for the sake of open-endedness. *Artificial life*, 25(2):198–206.
- Hintze, A., Edlund, J. A., Olson, R. S., Knoester, D. B., Schossau, J., Albantakis, L., Tehrani-Saleh, A., Kvam, P., Sheneman, L., Goldsby, H., et al. (2017). Markov brains: A technical introduction. *arXiv preprint arXiv:1709.05601*.
- Juric, M. (1994). An anti-adaptationist approach to genetic algorithms. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 619–623. IEEE.
- Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.
- Maley, C. (1999). Four steps toward open-ended evolution. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, pages 1336–1343. Morgan Kaufmann Publishers Inc.
- Miller, J. F. and Harding, S. L. (2008). Cartesian genetic programming. In *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation*, pages 2701–2726.
- Soros, L. B. and Stanley, K. O. (2014). Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. In *H. Sayama, J. Rieffel, S. Risi, R. Doursat, & H. Lipson (Eds.), Artificial life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 793–800. Citeseer.
- Taylor, T. (2015). Requirements for open-ended evolution in natural and artificial systems. *arXiv preprint arXiv:1507.07403*.
- Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., McMullin, B., Packard, N., Rasmussen, S., Virgo, N., Agmon, E., Clark, E., McGregor, S., Ofria, C., Ropella, G., Spector, L., Stanley, K. O., Stanton, A., Timperley, C., Vostinar, A., and Wiser, M. (2016). Open-ended evolution: Perspectives from the oee workshop in york. *Artificial Life*, 22(3):408–423. PMID: 27472417.
- Weisstein, E. W. (2003). Frobenius norm.