Augmentative Topology Agents For Open-ended Learning

Muhammad Umair Nasir, Michael Beukman, Steven James, Christopher Cleghorn
University of the Witwatersrand
School of Computer Science and Applied Mathematics
Johannesburg, South Africa
{umairnasir1,michael.beukman1}@students.wits.ac.za,{steven.james,christopher.cleghorn}@wits.ac.za

ABSTRACT

We tackle the problem of open-ended learning by introducing a method that simultaneously evolves agents while also evolving increasingly challenging environments. Unlike previous open-ended approaches that optimize agents using a fixed neural network topology, we hypothesize that generalization can be improved by allowing agents' controllers to become more complex as they encounter more difficult environments. Our method, Augmentative Topology EPOET (ATEP), extends the Enhanced Paired Open-Ended Trailblazer (EPOET) algorithm by allowing agents to evolve their own neural network structures over time, adding complexity and capacity as necessary. Our empirical results demonstrate that ATEP produces general agents capable of solving more environments than fixed-topology baselines. We also investigate mechanisms for transferring agents between environments and find that a species-based approach further improves the performance and generalization of agents.

CCS CONCEPTS

• **Computing methodologies** → **Lifelong machine learning**; • **Theory of computation** → **Evolutionary algorithms**.

KEYWORDS

open-ended learning, lifelong learning, neural networks, neuroevolution, reinforcement learning

ACM Reference Format:
Muhammad Umair Nasir, Michael Beukman, Steven James, Christopher Cleghorn. 2023. Augmentative Topology Agents For Open-ended Learning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590576

1 INTRODUCTION

Machine learning has successfully been used to solve challenging problems, such as Atari [5] and chess [6]. While impressive, these approaches still largely follow a traditional paradigm where a human specifies a task that is subsequently solved by the agent. In most cases, this is the end of the agent's learning—once it can solve the required task, no further progression takes place.

In contrast, the subfield of *open-ended learning* seeks to develop agents capable of learning indefinitely [9]. To achieve this, a common paradigm is to allow both the agents and their environments to change, evolve and improve over time [1, 11].

One algorithm that adopts this approach is Enhanced Paired Open-Ended Trailblazer (EPOET), which creates environments endlessly while evolving agents to solve each of them [12]. The authors use a modified version of the `2D Bipedal Walker Hardcore` environment from OpenAI Gym [2] as a benchmark. The first environment is a flat surface and, as evolution progresses, the environments become harder with the addition of more obstacles. POET also transfers agents across environments, which allows agents to leverage experience gained in one environment to solve another. An Environment-Agent (EA) pair is eligible to reproduce when the agent crosses a preset reward threshold on this environment. The next generation of environments is formed by mutating the current population and selecting only those environments that are neither too easy nor too hard. Finally, environments are ranked by novelty, and only the most novel children pass through to the next generation.

EPOET also extends prior work [11] along several fronts. First, EPOET uses compositional pattern-producing networks (CPPNs) [7] to mutate the environment, rather than mutating obstacles. Secondly, Wang et al. [12] introduce an *Environment Characterization* score, named *Performance of All Transferred* Agents EC (PATA-EC), as a domain-general environment characterization score. PATA-EC is a normalized score that describes the environment's behaviour through all active and archived agents' behaviour on the environment. The Euclidean distance between these scores can be used to determine novel environments in a domain-independent fashion. Finally, EPOET introduces a more selective transfer mechanism. As a result, only very promising agents undergo transfer.

However, the problem with these systems in general (and EPOET in particular) is that the agents have *fixed* neural network topologies. This means that the agent has a fixed capacity, which could cause learning to plateau—as observed by Wang et al. [12]—once the environments become too complex.

To tackle this problem, we apply NeuroEvolution of Augmenting Topologies (NEAT) [8] to EPOET. NEAT is a genetic algorithm that evolves the structure and weights of neural networks, instead of just evolving the weights within a fixed structure. Using NEAT allows the agents' network topologies to increase in complexity over time, leading to indefinite learning potential. In this work, we demonstrate that our method, Augmentative Topology EPOET (ATEP), can improve performance compared to the fixed-topology version.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590576

2 PROPOSED METHOD

In this section, we discuss the basic building blocks of our algorithm and the different variants we experiment with. ATEP combines EPOET with NEAT to allow the agents' network topologies to evolve. This means that the algorithmic steps are very similar to EPOET, and the main differences are (1) the optimizer used: we use NEAT to optimize the variable-topology agents whereas EPOET used Evolution Strategies to optimize fixed-topology agents; and (2) the transfer mechanism, discussed later in this section. The detailed flow of ATEP is illustrated in Figure 1.

We first use NEAT to evolve a population of agents for each environment. The valid environments (those that pass the minimal criterion) then reproduce to create a new generation of (slightly harder) environments. We then take the environment that is the most novel, as measured by the Euclidean distance between the *PATA-EC* scores), and create a new environment-agent pair. The transfer eligibility of these environments is then evaluated, and if there are valid transfers available, we can move agents between environments.

In EPOET, the transfer is performed as follows: we compare the fitness of the candidate agent to the fitness of the target agent over the previous 5 generations. If the candidate's fitness is greater than all previous 5 fitness scores, we fine-tune it on the target environment and again compare it against the best fitness from the previous 5 generations. If the candidate outperforms the target agent in both cases, we transfer the candidate and replace the target.

For ATEP, we experiment with two different transfer mechanisms. The first approach, termed *Fitness-Based Transfer ATEP* (FBT-ATEP), is inspired by EPOET's mechanism. In this case, we compare the best genome in the candidate population to the best genome from the target population. We then perform the same checks as EPOET, and if both are passed, we replace the entire target population with the candidate population.

For the second transfer mechanism, we use the speciation inherent in NEAT to influence transfer. Specifically, we check if the best genome in the candidate population is within a δ threshold of any target environment's best genome. This threshold is given by $\delta = c_1 E/N + c_2 D/N + c_3 W$, where c_1, c_2, c_3 are importance coefficients, N is the number of genes in the larger genome, E and D are the numbers of excess and disjoint genes respectively, and W is the average weight difference of similar genes.

If this is the case, we transfer the candidate species and replace the target species with it. This approach, called *Species-Based Transfer ATEP* (SBT-ATEP), avoids the need to compare fitness scores and has its own advantages, which we discuss in the next section. Algorithm 1 illustrates the pseudocode for Species-Based Transfer. Finally, we also consider random transfer (RT-ATEP) and no transfer (NT-ATEP) to investigate whether the transfer mechanisms have a large impact on the results.

Algorithm 1: Species-Based Transfer

```
Input : Candidate population's best individual I_c. A
        function find_delta(.) that calculates delta score
        and δ_threshold.
Let M = All environments \{Candidate environment}
foreach m ∈ M do
    I_m = best individual of environment m
    δ_ct = find_delta(I_c, I_m)
    if δ_ct ≤ δ_threshold then
        Delete target species
        Transfer candidate species to target population
    else
        Transfer is not permitted
    end
end
```

3 EXPERIMENTS AND RESULTS

To reduce the amount of compute required for our experiments, we change one aspect of the original EPOET paper by reducing the number of active environment-agent pairs from 40 to 20. We make this change to both EPOET and ATEP to ensure a fair comparison. We consider two baselines: the first, denoted as EPOET40x40, is EPOET with the original controller consisting of two hidden layers with 40 nodes each. The second baseline, EPOET20x20, is a controller with two layers of 20 nodes each. This also allows us to evaluate the effect of having a small fixed topology, a comparatively larger fixed topology, and a variable topology.

All results are averaged over 5 iterations owing to the expensive computational load, with each algorithm requiring approximately 50k to 200k CPU hours for a single run. EPOET20x20 required the least amount of computation, while SBT-ATEP required the most. Each algorithm was run in parallel on a cluster consisting of 264 cores, with the runtime ranging between 10 and 30 days.

3.1 Investigating Open-endedness

To quantify open-endedness, we use the *All New and Novel Environments Created and Solved* (ANNECS) score [12] to measure the progress of the overall system. A high ANNECS score indicates that the system is not simply creating easily-solvable environments, but also creates more challenging environments. We consider this metric as our most important score to judge which algorithm performs better in complex environments.

Figure 2 shows the ANNECS score as a function of training time. We see a significant difference between EPOET20x20 and both the FBT-ATEP and SBT-ATEP, indicating that the small network results in fewer solved environments. EPOET40x40 performs substantially better than EPOET20x20, and is competitive with ATEP early on during training. The rate of increase in ANNECS, however, does decrease after about 13k iterations, whereas ATEP increases at a consistent rate. This substantiates our hypothesis that fixed topology agents will start stagnating at some level of environment complexity, due to capacity issues. While we can improve the results by increasing the size of the network, this will merely delay the onset of the performance plateau.

FBT-ATEP outperforms EPOET40x40, although it also slows down slightly as time progresses. This is potentially due to replacing the entire target population with the transferred population, which may eliminate all useful skills learned by the target population. SBT-ATEP, on the other hand, only replaces a single species that is close to the candidate species, leaving the rest of the population intact. We also find that SBT-ATEP has negligible performance plateaus

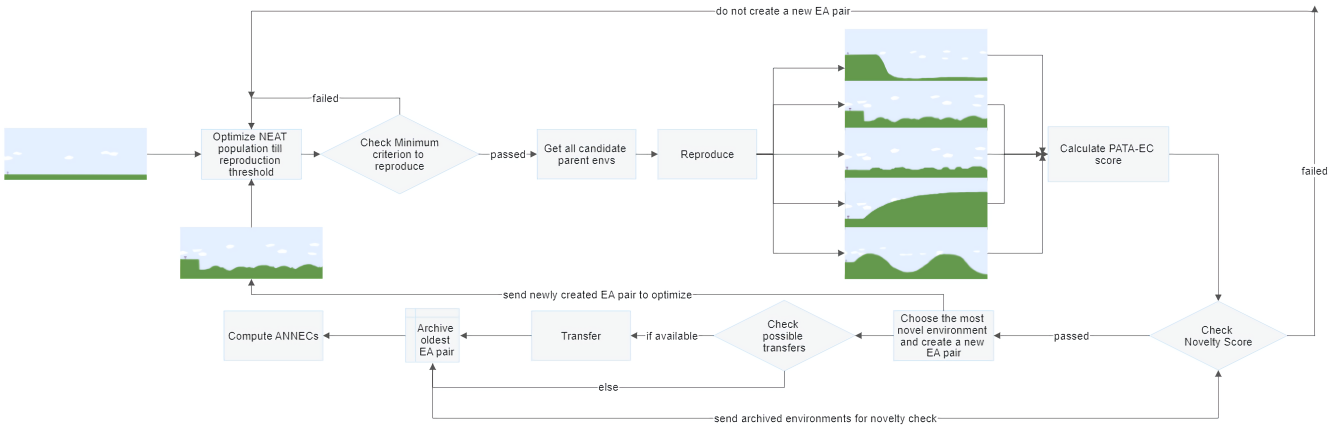


Figure 1: A flowchart illustrating the ATEP framework. For both EPOET and ATEP, each environment is associated with an agent, represented by an ES population for EPOET and a NEAT population for ATEP.

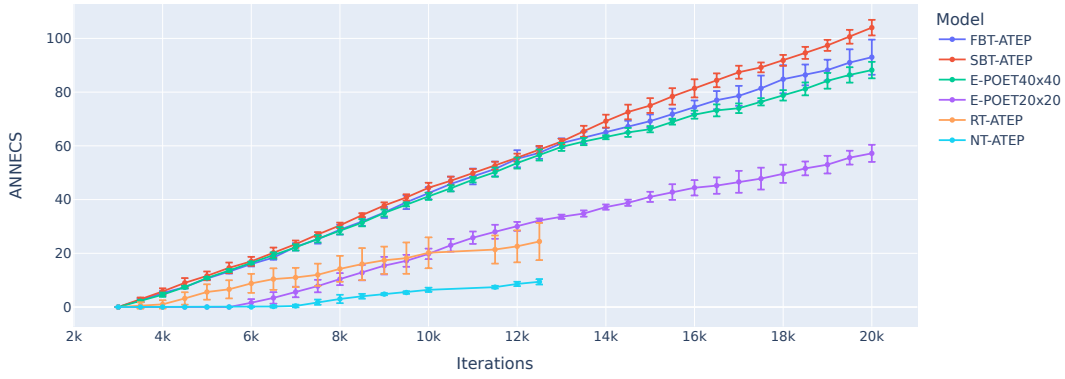


Figure 2: Accumulated Number of Novel Environments Created and Solved (ANNECS). The results are averaged over 5 seeds with the lines and error bars representing the mean and standard deviation respectively. To save compute, we halt the NT- and RT-ATEP experiments early, as it is clear that they perform poorly.

during the run of our experiments in solving environments. Further, even though it performs similarly to FBT-ATEP and EPOET40x40 early on during training, it starts to outperform these in the second half of the experiment. This, as we show in the supplementary material, is partly due to SBT-ATEP exploring more actions. We finally note that the variations using no transfer (NT-ATEP) or random transfer (RT-ATEP) perform poorly, indicating that intelligent transfer mechanisms are necessary.

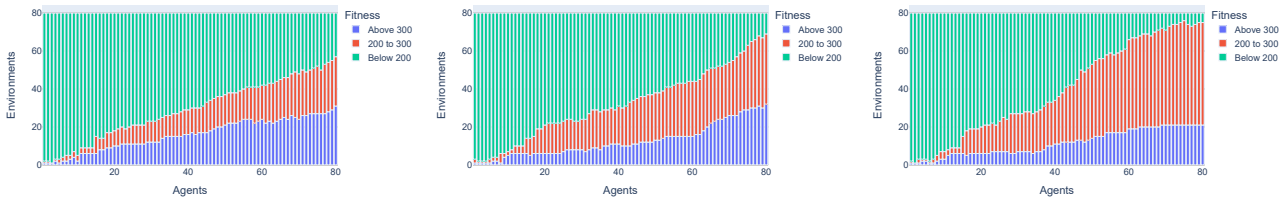
Although ATEP outperforms EPOET, it is more computationally expensive, as measured by the number of function evaluations. One function evaluation means one individual being evaluated on an environment. SBT-ATEP has the most function evaluations since once a species transfers from one population to another, it becomes highly probable that it can transfer in the opposite direction because it may now be within the $\delta_{threshold}$ range. This increases the population size, resulting in more function evaluations.

3.2 Investigating Generalization

We next evaluate the generalizability of our open-ended agents, as prior work [10] has shown that these agents have the potential to generalize to different environments. We first test the generalization capabilities of agents produced by a given algorithm on all of the environments created by that same algorithm. We take into account 80 environments that were solved by the model itself and observe how each agent performs on all of them.[1] We split the results into three categories: environments with fitness scores above 300, between 200 and 300, and below 200. Scores below 200 indicate that the environment has not been solved by the agent.

The results in Figure 3 indicate that early-stage agents perform worse, while late-stage agents are shown to have generalization abilities on previously unseen landscapes. The transfer mechanism plays a key role in this generalization, as it exposes agents to more

[1]We exclude EPOET20x20 as it fails to solve 80 environments across the entire run.



(a) EPOET40x40 (b) FBT-ATEP (c) SBT-ATEP

Figure 3: Figures showing generalization capabilities. Figures (a), (b) and (c) show agents of 80 solved environments being tested on all 80 environments, for EPOET40x40, FBT-ATEP and SBT-ATEP respectively. The Y-axis shows the percentage of environments in each category, with means reported over 30 runs.

environments. Despite not having seen all environments, late-stage agents generalize significantly better. SBT-ATEP demonstrates the best generalization, with the lowest proportion of unsolved environments when compared to EPOET40x40 and FBT-ATEP.

To test generalization to new, unseen environments, we next take the 20 latest environments from each algorithm. For each environment, we take the latest agent that could solve this environment from the approach under consideration. Each of these agents is now evaluated on the selected environments from *all other approaches*. Figure 4 shows the performance of each method when evaluated on the 60 other environments, where we observe that SBT-ATEP outperforms all other models, with only 10% of the environments remaining unsolved.

4 CONCLUSION AND FUTURE WORK

This work investigated the effect of having an augmentative topology agent on an open-ended learning algorithm's performance. We hypothesized that using a fixed topology would result in agents that exhibit delays in solving an environment after a certain point in environment complexity. We showed that this is indeed the case, and addressed this limitation by introducing ATEP, which allows the network topology of the agents to change and add complexity as necessary. We demonstrated that this approach outperforms existing methods in terms of the ANNECS and generalizability.

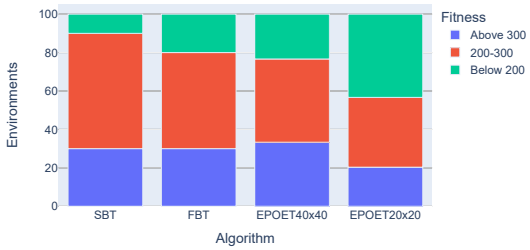


Figure 4: Each algorithm's agent evaluated on the 20 latest environments created by all other algorithms. The Y-axis shows the percentage of environments in each category. Means over 30 runs are reported.

Promising future directions include incorporating novelty search [3], or exploring alternatives to NEAT such as neurogenesis [4]. Our work demonstrates that transfer mechanisms have a large overall impact, and while we compared simple approaches such as FBT and SBT, more advanced approaches could yield further performance improvements. For instance, we could combine both FBT and SBT in a weighted manner, or transfer only a certain percentage of a species or population. Ultimately, we hope that this new approach furthers research into open-ended algorithms that continue to learn over time and can adapt to an ever-changing environment.

ACKNOWLEDGMENTS

The authors acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project.

REFERENCES

[1] J. C. Brant and K. O. Stanley. Minimal criterion coevolution: a new approach to open-ended search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 67–74, 2017.
[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016.
[3] J. Lehman, K. O. Stanley, et al. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.
[4] K. Maile, E. Rachelson, H. Luga, and D. G. Wilson. When, where, and how to add new neurons to ANNs. *arXiv preprint arXiv:2202.08539*, 2022.
[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
[6] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
[7] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.
[8] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
[9] K. O. Stanley, J. Lehman, and L. Soros. Open-endedness: The last grand challenge you've never heard of. *O'Reilly Online*, 2017. URL https://www.oreilly.com/ideas/open-endedness-the-last-grand-challenge-youve-never-heard-of.
[10] O. E. L. Team, A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.
[11] R. Wang, J. Lehman, J. Clune, and K. O. Stanley. Paired open-ended trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.
[12] R. Wang, J. Lehman, A. Rawal, J. Zhi, Y. Li, J. Clune, and K. Stanley. Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. In *International Conference on Machine Learning*, pages 9940–9951, 2020.