Collaborative Interactive Evolution in Minecraft

Pablo González de Prado Salas
IT University of Copenhagen, Denmark
pago@itu.dk

Sebastian Risi
IT University of Copenhagen, Denmark
sebr@itu.dk

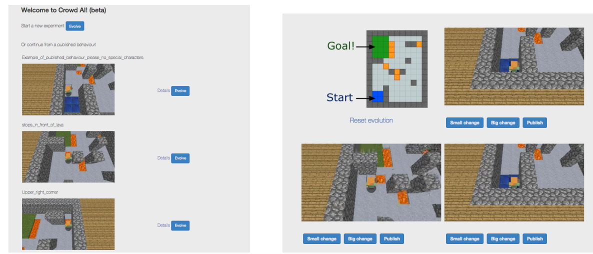
ABSTRACT

Interactive Evolution (IE) has shown promise in domains where objective-driven approaches are problematic. Collaborating by branching from the evolutionary stepping stones discovered by others (CIE) has allowed users to evolve a large variety of artefacts. In this paper we explore the potential of collaboratively evolving behaviours in Minecraft through an online CIE system. Results suggest that it is indeed possible to solve a maze navigation task in Minecraft collaboratively, which is difficult to achieve individually. We hope that the released framework will allow others to build more complex domains, solving a series of more and more challenging tasks through many collaborating humans.

KEYWORDS

Evolutionary computation; interactive evolutionary computation; collaborative IEC; neuroevolution

ACM Reference Format:
Pablo González de Prado Salas and Sebastian Risi. 2018. Collaborative Interactive Evolution in Minecraft. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205723



(a) Behaviour Library **(b) IEC interface**

Figure 1: CrowdAI Interface.

1 INTRODUCTION

In interactive evolutionary computation (IEC) [1] the traditional fitness function is replaced by users iteratively selecting candidate solutions. This approach has shown promise in domains with no clearly defined objective [1, 2]. When combined with neuroevolution (i.e. the evolution of artificial neural networks), IEC can evolve game artefacts [3], music [4], or agent behaviours [5–7].

A problem for IEC is *user fatigue*. While different attempts have been made to alleviate this problem [8, 9], evolving complex behaviours interactively will likely require a large number of users collaborating, as in Picbreeder [10].

In this paper we take a first step towards providing a framework for the collaborative evolution of controllers in Minecraft. Interaction with our framework only requires a web browser, making it accessible to as many participants as possible. Similarly to Picbreeder, users can publish their creations and branch from behaviours discovered by others.

Our experiments build on the Malmo platform [11], that extends the Minecraft video game. While the task in this paper is relatively simple, it demonstrates that collaborative evolution is possible for tasks that are difficult to solve for users by themselves.

2 APPROACH

An overview of our CIEC approach is shown in Fig. 1. Users can start evolution from scratch or elaborate on behaviours evolved by others (a). At each iteration, users choose among the parent behaviour and two offspring, or reset evolution (b). The current version of the project can be found here: https://github.com/pablogps/CrowdAI.

Our experimental environment works on Minecraft using Project Malmo[1]. Users interact with our server through their Internet browser, so they do not need to install Minecraft in their machines. The server is implemented in ASP.NET-MVC, and uses a custom version of SharpNEAT. This architecture aims to be easily extendible and adaptable to different applications and evolutionary algorithms (an example of this are the fitness-based experiments reported in this paper). Users interested in extending the framework (e.g. with new domains) can do so on a higher-level of abstraction that does not require intimate knowledge of the underlying framework.

Our navigation domain is more complicated than typical set-ups in which agents can rotate [12], enabling basic maze solving strategies like wall-following. We discarded rotation because this forces discomforting camera movements. Instead of developing rules such as "keep wall close to right" agents must develop knowledge of where they are in the maze given the current inputs.

Neural controllers have six sensory inputs and two outputs for movement. Only rotation of ±90 degrees and one-block steps are allowed. Several positions on the map produce very similar inputs, but require contradictory outputs. Therefore, the last part of the maze is very challenging.

3 EXPERIMENTS

We performed two user studies over a period of one week. The experiment was open to any Internet users. The authors participated during the first experiment to collaboratively solve the complex

[1] https://github.com/Microsoft/malmo

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205723

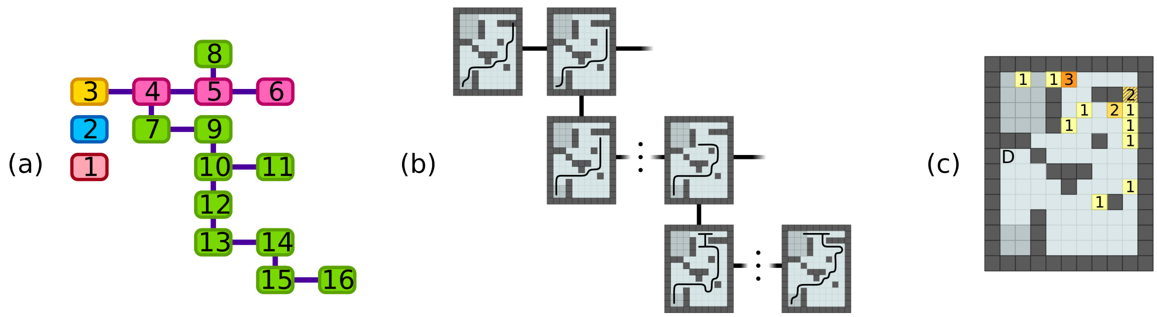


Figure 2: (a) Evolved behaviour lineage. Colours represent different users, and connecting lines indicate branching. Node 16 is the solution. (b) Selected behaviours for nodes from 3 to 16. (c) Heatmap with end position of published behaviours.

Table 1: Data from the second experiment. Average values and standard deviations per user when not specified otherwise. Generations per node measures the number of generations a user adds to a genome from branching to publication.

Total number of users	27
Total generations evolved	1620
Total published behaviours	15
#Generations	60 ± 90
#Small mutations	40 ± 70
#Large mutations	20 ± 30
#Published behaviours	0.6 ± 1.0
#Generations per node	30 ± 30

maze. The second experiment had a focus on obtaining user data (see Table 1).

Additionally, we performed five fitness-based experiments to test whether the environment is deceptive in the traditional sense. Each of the five experiments involved 211 genomes. In user-driven evolution, 105 generations are required to produce the same amount of genomes. Fitness is: $F = F_{max} - d_x - d_y$ (d is distance to goal).

4 RESULTS

As expected, this task is not solvable with a trivial fitness-based strategy, and thus offers a good testing domain for collaborative IEC. Fitness graphs are omitted since the best individuals from all runs end in tile **D** (Fig. 2c).

Fig. 2a shows a collective behaviour lineage that solves the maze in the first user study. We observe backtracking, where new users re-examine old behaviours and successfully branch from them, which is not typically possible in neuroevolution, or even in IEC. Fig. 2c maps the end positions for the published behaviours, with some tiles being the ending position for many published behaviours; users not only judge distance to the goal, but use additional information to judge how promising behaviours are. Similar observations have been made by Woolley *et al.* [5].

5 CONCLUSIONS

We have presented an IEC approach that allows users to collaboratively breed behaviours in the Minecraft video game. The results suggest that tasks that are potentially too difficult (or extenuating) for a single user to solve can be solved by allowing people to elaborate on partial solutions found by others. In the future, it will be interesting to extend the approach to more complex tasks in the Minecraft world, such as building structures or interacting with game agents. We hope that providing this framework and proof-of-concept experiments will spur development in such direction.

6 ACKNOWLEDGEMENTS

We thank Fundacion Ramon Areces for funding as part of their postdoc fellow-ship program. Special thanks to our early testers, particularly our most dedicated contributors, José Manuel and Ana.

REFERENCES

[1] Takagi, H.: Interactive evolutionary computation: Fusion of the capacities of ec optimization and human evaluation. Volume 89. (2001) 1275–1296
[2] Todd, S., Latham, W.: Evolutionary art and computers academic press (1992)
[3] Risi, S., Lehman, J., D'Ambrosio, D.B., Hall, R., Stanley, K.O.: Petalz: Search-based procedural content generation for the casual gamer. IEEE Transactions on Computational Intelligence and AI in Games **8**(3) (Sept 2016) 244–255
[4] Hoover, A.K., Szerlip, P.A., Stanley, K.O.: Interactively evolving harmonies through functional scaffolding. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM (2011) 387–394
[5] Woolley, B.G., Stanley, K.O.: 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, New York, NY, USA, ACM (2014) 233–240
[6] Löwe, M., Risi, S.: Accelerating the evolution of cognitive behaviors through human-computer collaboration. In: Proc. of the Genetic and Evolutionary Computation Conference 2016. GECCO '16, New York, NY, USA, ACM (2016) 133–140
[7] González de Prado Salas, P., Risi, S.: Interactive evolution of complex behaviours through skill encapsulation. In: European Conference on the Applications of Evolutionary Computation, Springer (2017) 853–869
[8] Bongard, J.C., Hornby, G.S.: Combining Fitness-based Search and User Modeling in Evolutionary Robotics. Proceeding of the 15th Annual conference on Genetic and Evolutionary Computation Conference - GECCO '13 (2013) 159–166
[9] Kamalian, R., Zhang, Y., Takagi, H., Agogino, A.: Reduced human fatigue interactive evolutionary computation for micromachine design. In: Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. Volume 9. (08 2005) 5666–5671 Vol. 9
[10] Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., Stanley, K.O.: Picbreeder: A case study in collaborative evolutionary exploration of design space. Evol. Comput. **19**(3) (September 2011) 373–403
[11] Hofmann, K., Hutton, T., Bignell, D., Johnson, M.: The Malmo platform for artificial intelligence experimentation. In: 25th International Joint Conference on Artificial Intelligence (IJCAI-16), AAAI - Association for the Advancement of Artificial Intelligence (July 2016)
[12] Lehman, J., Stanley, K.O.: Exploiting open-endedness to solve problems through the search for novelty. In: Proceedings of the Eleventh International Conference on Artificial Life (Alife XI, MIT Press (2008)