Multi-Agent RL with Population-Based Training

Author:
Ivan Nestorov
I.Nestorov@student.tudelft.nl

Supervisor:
Robert Loftin
R.T.Loftin@tudelft.nl

Responsible Professor:
Frans Oliehoek
F.A.Oliehoek@tudelft.nl

1.Introduction

- Context
  - Cooperative games, provide the oportunity to evaluate human-AI collaboration.
- Overcooked[1]
  - Multiplayer game, which requires players to collaborate to prepare and serve dishes.
- Population-Based Training[2] (PBT)
  - Training algorithm, which uses the evolutionary approach. Figure 1, is an illustration of how the PBT is performed.
- Proximal Policy Optimization[3] (PPO)
  - Reinforcement Learning algorithm, used to update policies during training.

2. Research Question

- How does the use of population-based training affect the performance of Multi-Agent Reinforcement Learning algorithms?
- What changes can be made to the PBT to improve the agent's performance when paired with a human player?

3. Methodology

- Run existing experiments for PBT with PPO on simplified environment. The simplified environment can be seen at Figure 2.
- Compare results with the original paper. [5]
- Introduce high-variance agents through custom mutation factors.
- Compare the performance of agents trained with baseline and custom mutation factors.
- Train agents using different population sizes to observe their impact on the learning process.
- Evaluate the performance of agents trained in populations of varying sizes to analyze their effectiveness.

7. References

[1] G.T Games, Overcooked, https://ghosttowngames.com/overcooked/, Published: 2016

[2] M. Jaderberg, V. Dalibard, S. Osindero, et al., "Population based training of neural networks," CoRR, vol. abs/1711.09846, 2017. arXiv: 1711.09846. [Online]. Available: http://arxiv.org/ abs/1711.09846.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," CoRR, vol. abs/ 1707.06347, 2017. arXiv: 1707.06347. [Online]. Available: http:// arxiv.org/abs/1707.06347

[4] https://docs.ray.io/en/latest/tune/examples/pbt_guide.html

[5] M. Carroll, R. Shah, M. K. Ho, et al., "On the utility of learning about humans for human-ai coordination," CoRR, vol. abs/ 1910.05789, 2019. arXiv:1910 . 05789. [Online]. Available: http://arxiv.org/abs/1910.05789

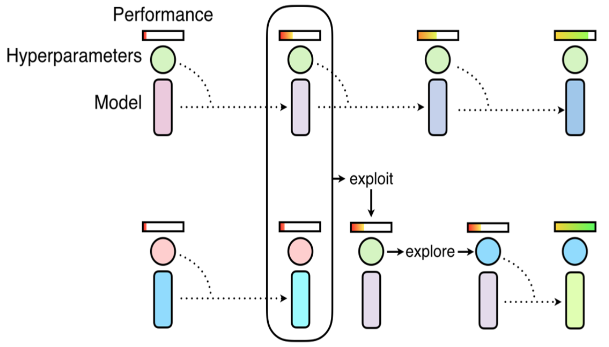


Figure 1. Depiction of the PBT algorithm with two models. Image taken from [4]

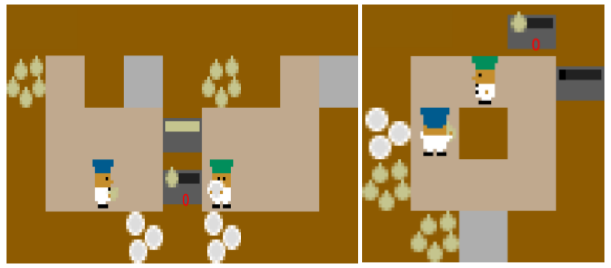


Figure 2. From left to right: Asymmetric Advantages, Coordination Ring. Image taken from [5]

4. Results

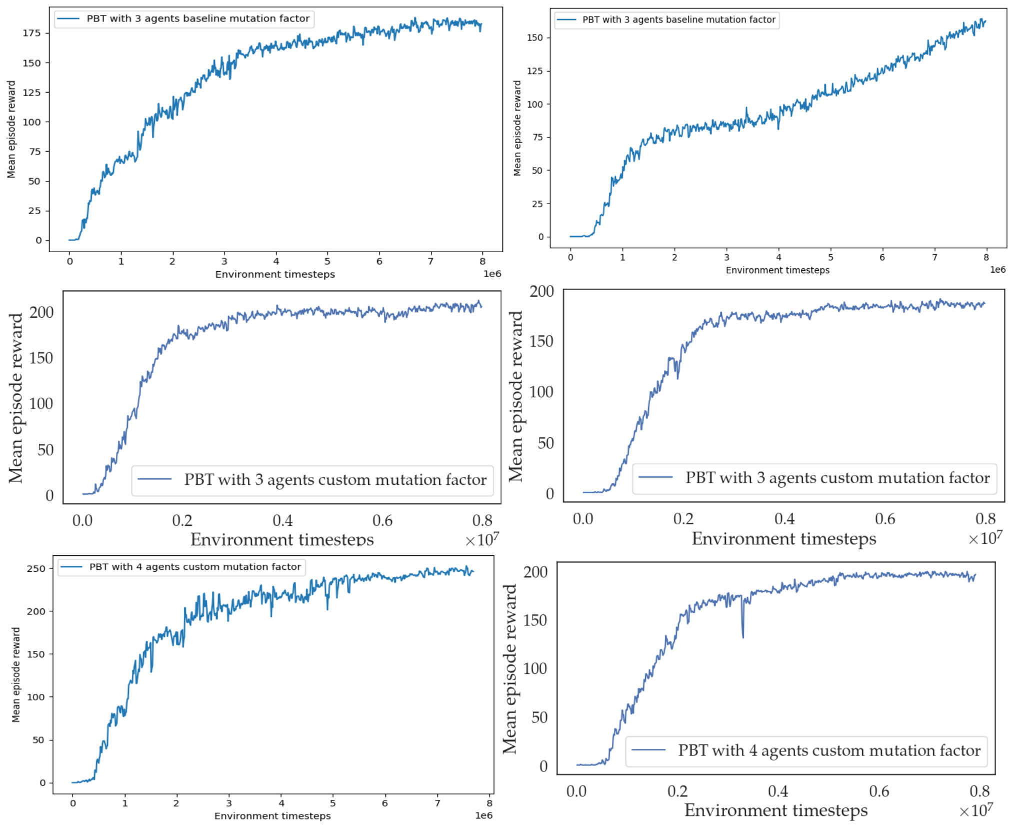
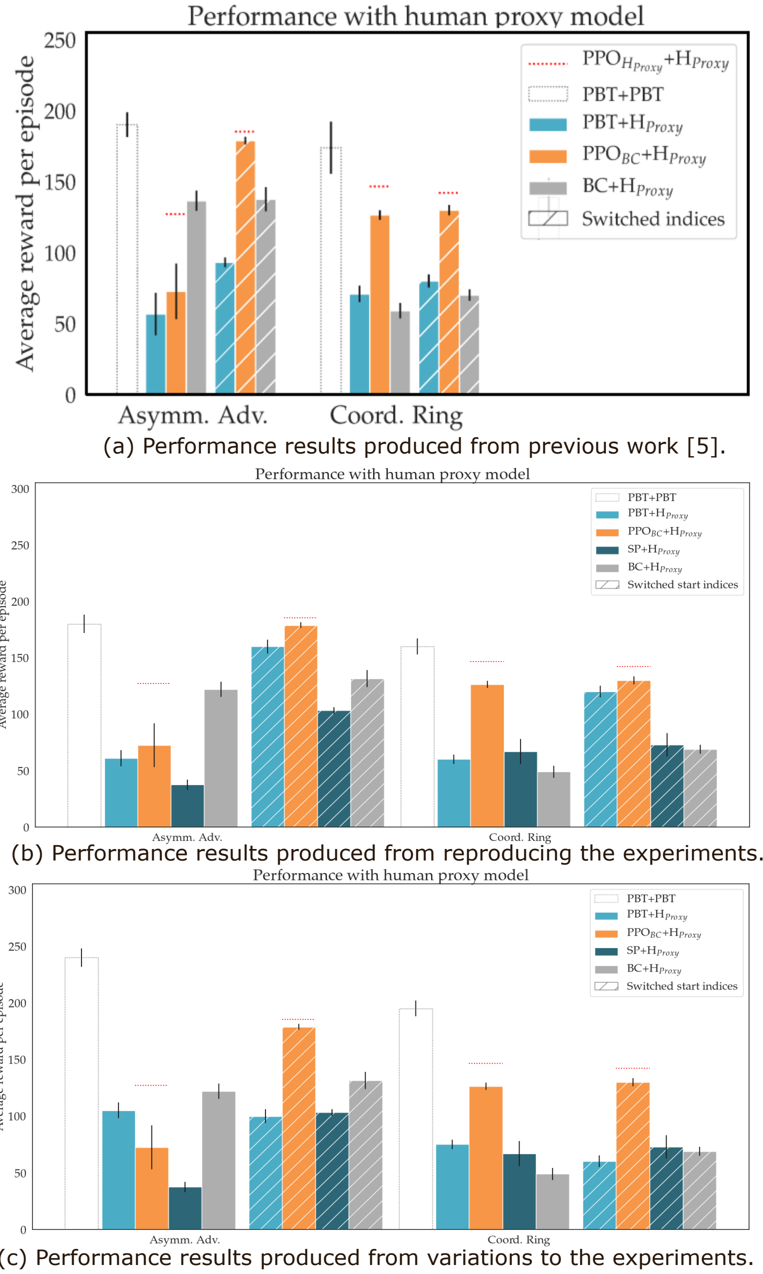


Figure 3. The learning curves of a PBT agent are presented, showing the average sparse reward per episode (mean of 100 episodes) throughout the training process over 400 horizon timesteps. The left column displays the results on the Asymmetric Advantages layout, while the right column represents the results on the Coordination Ring layout.



(a) Performance results produced from previous work [5].

(b) Performance results produced from reproducing the experiments.

(c) Performance results produced from variations to the experiments.

Figure 4. The performance results of a PBT agent, when paired with the human proxy, are compared to the performances of other agents also matched with the proxy. Performance for each layout is evaluated based on the average sparse reward per episode (mean of 100 episodes) over 400 horizon timesteps.

5. Results Analysis

- Reproduction Experiment Findings:
  - PBT underperforms when paired with a human proxy.
  - PBT outperforms self-play
  - Results confirm the conclusions derived in the previous research [5].
  - PBT exhibits poor sample efficiency in layouts with a high risk of agent collision.

- Variations Experiment Findings:
  - Improves sample efficiency for layouts with low risk of agent collison.
  - Sample efficiency stays the same for various population sizes in layouts with high risk of agent collison.
  - There is a significant performance improvement for agents in layouts with low collision risks, while the performance boost is minimal for layouts with high collision risks.

6. Conclusion and future work

- RQ1 Answer: When combined with the human proxy, PBT shows improvement over self-play but underperforms when compared to agents trained on human data.
- RQ2 Answer: By incorporating custom mutation factors and increasing the population size, PBT improves sample efficiency for specific layouts. However, additional research is needed to assess its effects on final performance.
- Limitations:
  - The study is limited by computational constraints and the availability of research time.
- Future Work:
  - Future research should continue to explore the effects of increasing population diversity and size within the PBT framework.
  - Investigate how incorporating Behavior Cloning(BC) agents into the PBT population influences the final performance.