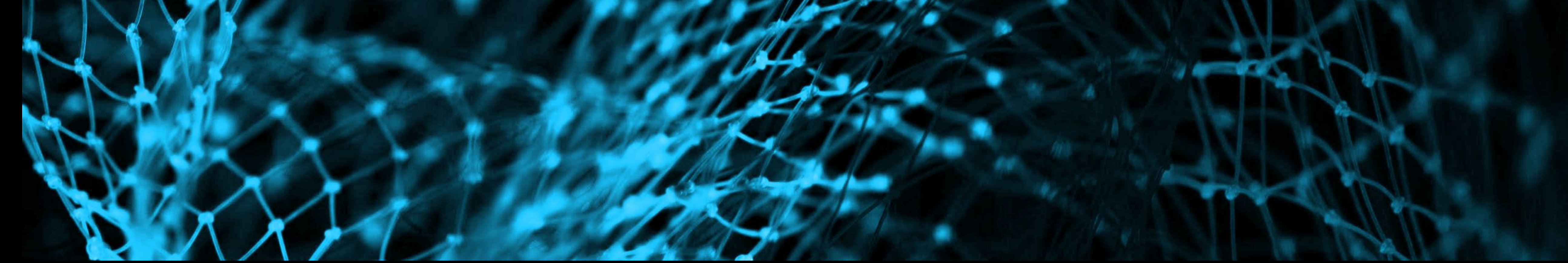
MORL



Jeremiah D Greer, Frazier N Baker, Fred Annexstein

Purpose

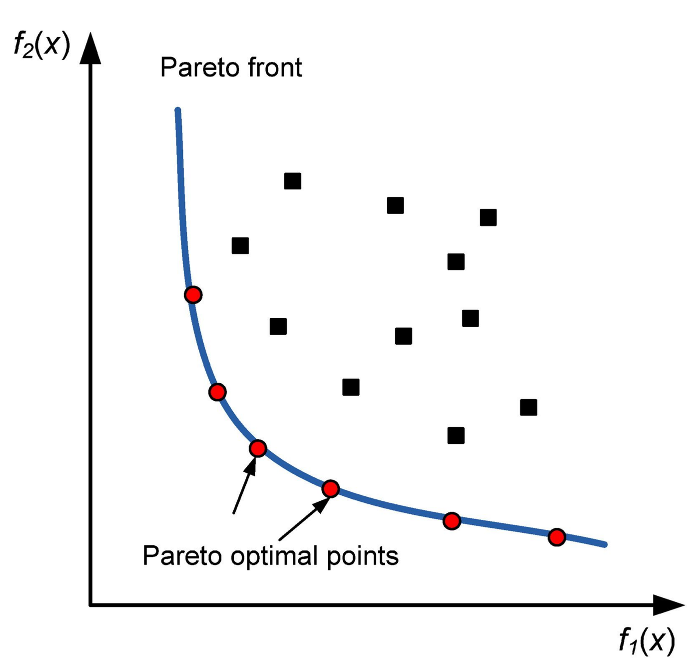
Goal: Use Filter Methods to find a novel way to solve Multi-Objective Reinforcement Learning Problems

What is Reinforcement Learning?

- Reinforcement Learning: a program learning what action(s) to perform in a given state based on its experience
- Many complex problems require the program to learn multiple things (i.e. multiple functions).
- Normally a linear combination of the rewards is used
- This requires tuning reward coefficients and doesn't always provide the best results
- We seek to utilize filter methods[1] (a form of multi-objective optimization) to provide a better method of solving these problems

What are Filter Methods

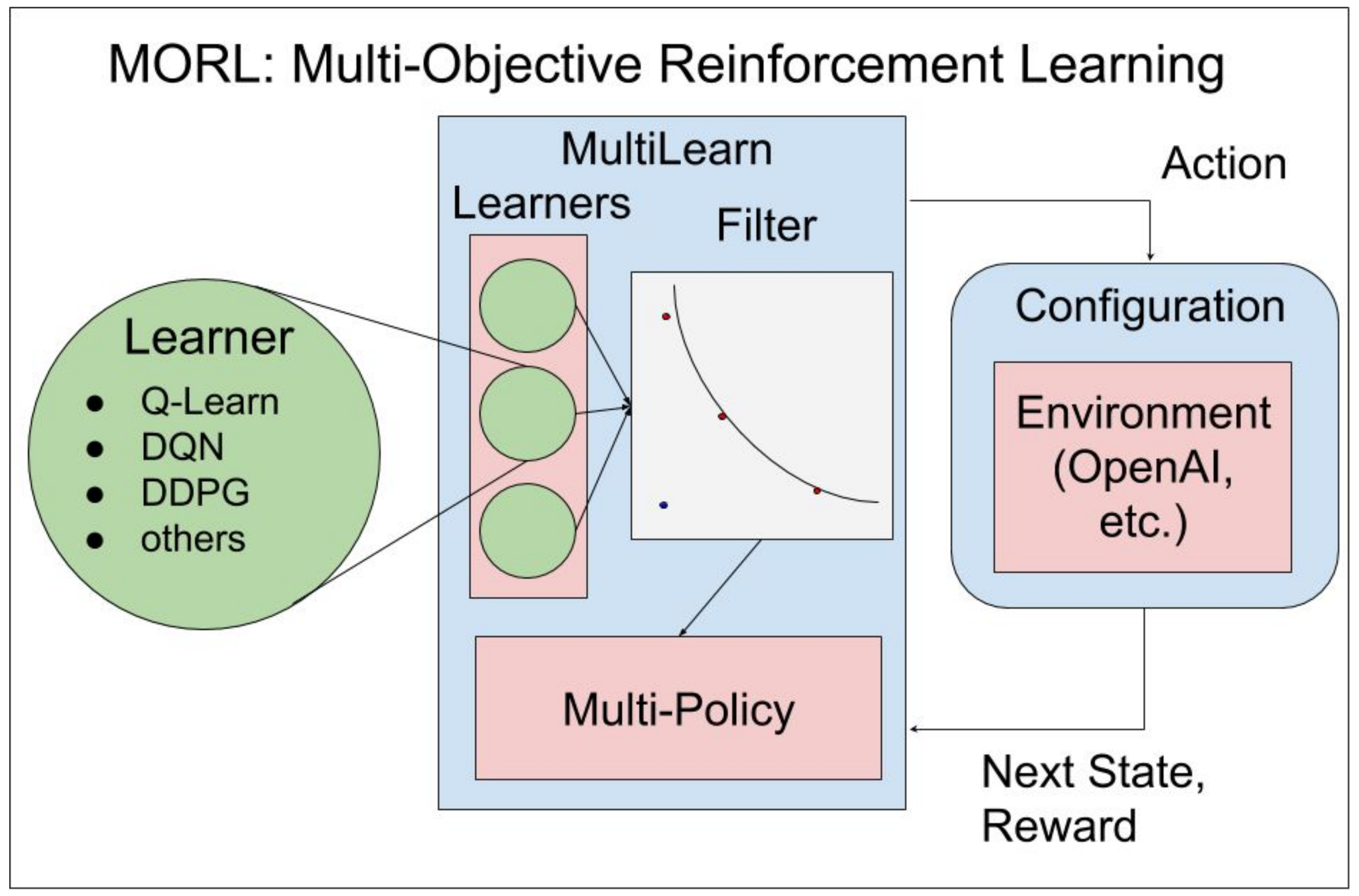
- Method of optimizing multiple functions
- Often cannot find a single optimal point for multiple functions
- Instead of choosing one, filters collect all points optimal for at least one of the functions
- For a pareto filter, these points are called "Pareto Optimal"[2]
- Gives multiple "optimal" points to choose from



https://ascelibrary.org/doi/full/10.1061/%28ASCE%29AS.1943-5525.0000464

Technologies Used

Algorithm

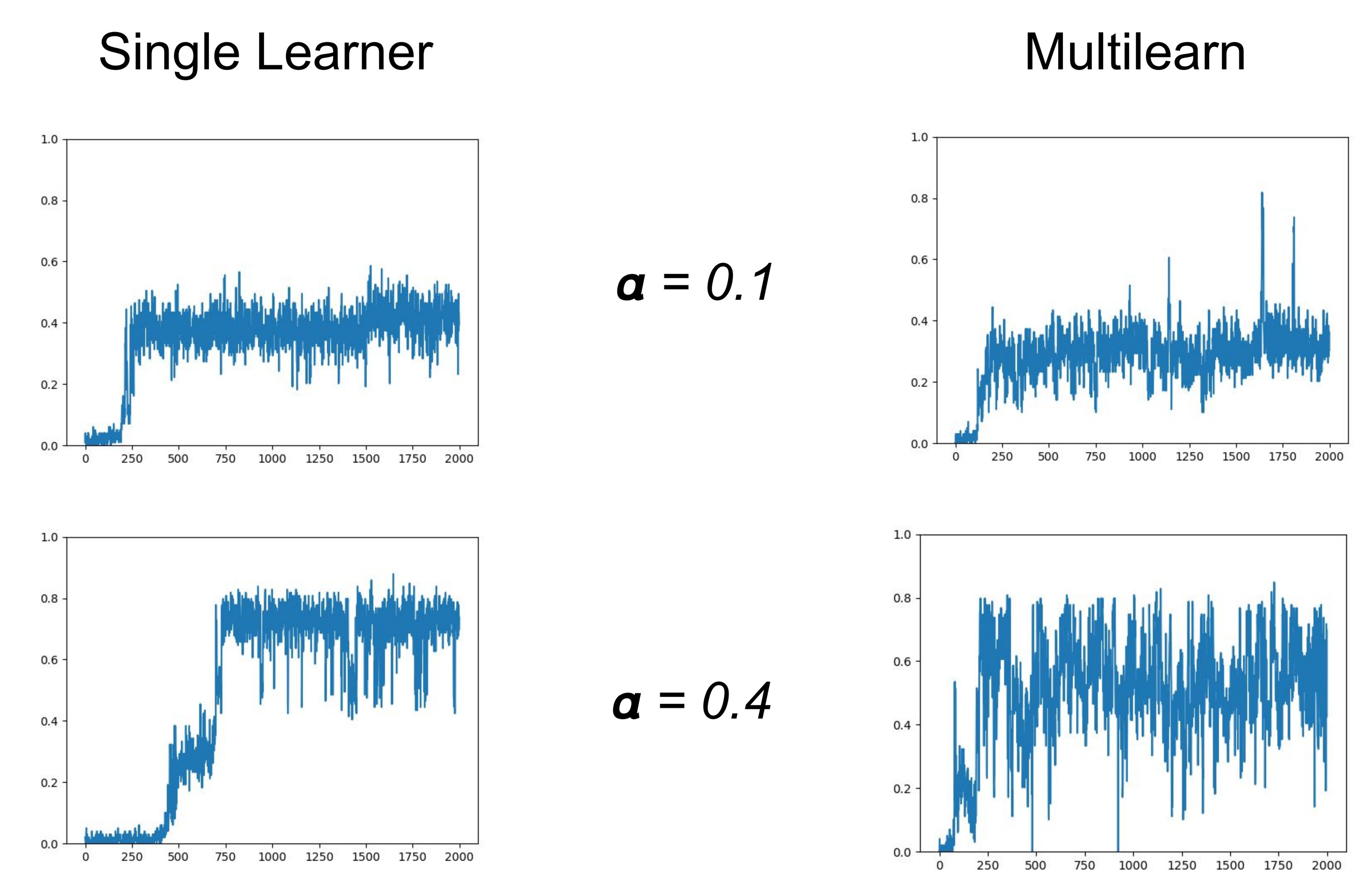


1. Multilearn instantiates a collection of identical Learners
   a. Q-Learn by default, can be anything
   b. Allows for DQN[3], DDPG[4], etc
2. Multilearn forwards state to the learners and receives back utilities from each of them for each possible action
3. Multilearn builds a filter using these utilities and randomly selects an action from the filter
   a. Further tested by altering action selection
4. Multilearn sends this action to the environment and receives a reward and new state
5. Mutlilearn forwards these to its Learners and the learners update themselves accordingly

Deliverables

- Literature Review - December 2017
- CLI Tool - January 2018
- Experimental Results - March 2018
- GUI Tool - March 2018
- Paper - Summer 2018

Benchmarking



Observations:

- Multilearn "learns" certain behaviors more quickly than a single learner with an aggregate function
- Multilearn's performance is more erratic than a single learner (most likely due to random action selection from learners)

Future Directions

- Project is parallelizable, allowing it to scale up to any number of learners and reward functions (Apache Spark)
- Experiment with other methods of multi-learn action selection (such as voting methods similar to bagging)
- Further testing should be done on other environments

References

1. R Fletcher, S Leyffer, and PL Toint, "A brief history of filter methods," Preprint ANL/MCS, 2006.
2. J Branke, K Deb, and K Miettinen, *Multiobjective optimization: Interactive and evolutionary approaches*. Springer Science & Business Media, 2008.
3. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S. Human-level control through deep reinforcement learning. Nature. 2015 Feb;518(7540):529.
4. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971, 2015 Sep 9.

University of CINCINNATI