Duke | Computer Science
CS+

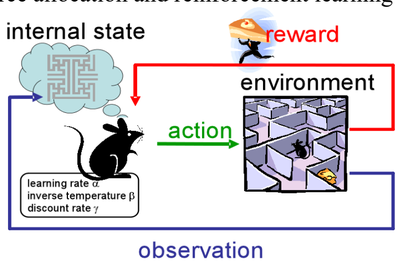
Fair Multi-objective Reinforcement Learning

Muhang Tian, Nianli Peng, Olivia Fan
Mentor: Professor Brandon Fain

Abstract

We designed a fair multi-objective reinforcement learning (**MORL**) algorithm by incorporating ideas from resource allocation and reinforcement learning literature.

It is a **general multi-objective Q learning algorithm** that could be implemented with **nonlinear welfare function**. We tested our algorithm under three simulated environments and analyzed its performance.



Problem Statement

- We are working with a Markov Decision Process (MDP) with multi-objective reward, discounted episodic tasks.
- We choose the Nash Social Welfare (NSW) function as our candidate welfare function, which is defined as the geometric mean of the components of the reward vector.
- We want to maximize expected Nash welfare of total discounted reward in an episode.

Methods

- To solve the problem, we modified Q-learning, an existing reinforcement learning algorithm. Some notable changes we made are
  - The action extraction process takes account of the accumulated rewards.
  - The agent takes action that maximizes the welfare of rewards when updating the Q-table.
- The algorithm has three variations. When updating the Q-table, the agent chooses an action while taking account of
  - Myopic: just the entries of the Q-table.
  - Immediate: the immediate reward of the agent's current action and the entries of the Q-table.
  - Pseudo-SARSA: the accumulated rewards from all the agent's actions and the entries of the Q-table.

Algorithm 1 Modified Q-learning for Multiobjective Reinforcement Learning

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\epsilon > 0$
Initialize $\mathbf{Q}(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $\mathbf{Q}(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Initialize $\mathbf{R_{acc}} = \mathbf{0}$
  Loop for each step of episode:
    Choose

$$A \leftarrow \begin{cases} \arg\max_a [\text{NSW}(\mathbf{R_{acc}} + \gamma \mathbf{Q}(s, a))] & \text{with probability } 1 - \epsilon \\ \text{a random action} & \text{with probability } \epsilon \end{cases}$$

    Take action A, observe $\mathbf{R}, S'$
    $\mathbf{R_{acc}} \leftarrow \mathbf{R_{acc}} + \mathbf{R}$
    $A^* \leftarrow \arg\max_a \text{NSW}[\mathbf{R} + \gamma \mathbf{Q}(S', a)]$
    $\mathbf{Q}(S, A) \leftarrow \mathbf{Q}(S, A) + \alpha[\mathbf{R} + \gamma \mathbf{Q}(S', A^*) - \mathbf{Q}(S, A)]$
    $S \leftarrow S'$
  until S is terminal
The policy is then given by

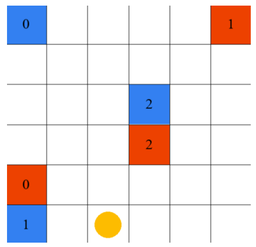
$$\pi(\mathbf{R_{acc}}, s) := \arg\max_a [\text{NSW}(\mathbf{R_{acc}} + \gamma \mathbf{Q}(s, a))]$$

Novel Difficulties

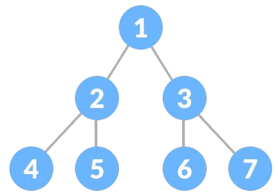
Moving from scalar rewards with linear objective functions to vector rewards with nonlinear welfare functions introduces novel difficulties. Some notable ones include

- The optimal policy may not be stationary even if the environment is stationary, i.e., the optimal action might not be the same even if the agent is in the same state.
- The expected welfare is not welfare of expectations because of the non-linearity of the welfare functions.
- The problem of producing the optimal policy is NP-hard even if there is a small number of states.

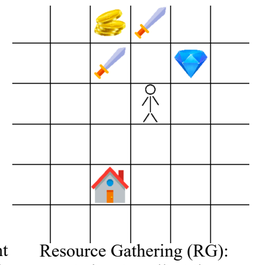
Simulated Environments



Taxi (yellow) learning to deliver multiple passengers from pickup locations (blue) to destinations (red) while trying to be fair



Fruit Tree Navigation (FTN): The agent aims to find paths from the root to leaf nodes where rewards encode the amounts of six different components of nutrition of the fruits.

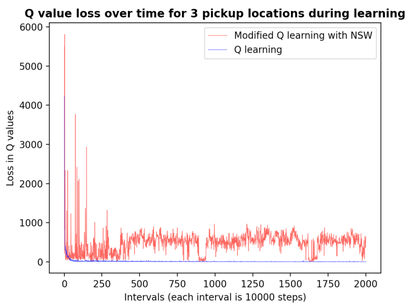


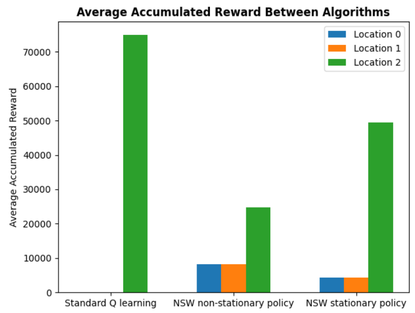
Resource Gathering (RG): agent aims to collect three different types of resources (swords, gold, gem) while maintaining balance.

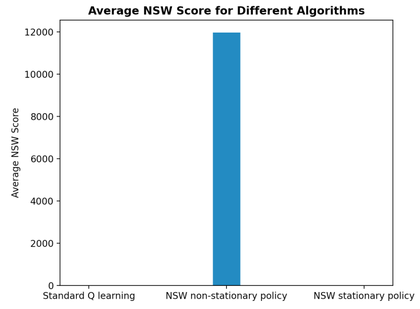
Results

In ***Taxi***, we

- Trained our agent with myopic action selection, in discounted continuous task
- Evaluated our trained agent with 10000 steps over 50 runs for three algorithms (linear scalarization, nonlinear scalarization with stationary policy, and nonlinear scalarization with non-stationary policy) and recorded accumulated reward at each location, their convergence performance while learning, and NSW score







Results observation

- Nonlinear scalarization with non-stationary policy performs best in terms of fairness (measured by NSW)
- Linear scalarization with stationary policy algorithm performs poorly, only favoring location 2, which is the easiest option in terms of successful delivery.
- However, nonlinear scalarization with non-stationary policy perform worse in terms of convergence

In ***FTN***, we

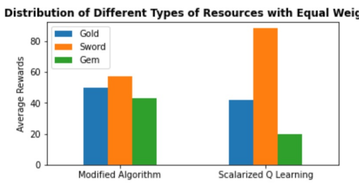
- Trained the agent using non-stationary learning update with the three variations of the modified Q-learning algorithm
- Asked the agent to choose one to five fruits at a time using the same non-stationary action extraction process.
- The agent's selections were then compared against the global optimal solution in terms of the Nash social welfare of the accumulated rewards.

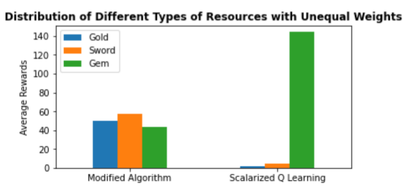
	1	2	3	4	5
Myopic	3.903	7.806	11.750	15.729	19.627
Immediate	3.903	7.810	11.740	15.703	19.626
Pseudo-SARSA	3.903	7.809	11.814	15.792	19.714
Global optimal	3.903	7.848	11.841	15.796	19.731

- All three variations of the modified Q-learning algorithm seem to yield roughly the optimal solution (all within 1%).
- The key to the performance improvement of the algorithm lies in the non-stationary policy extraction process that not only takes account of the entries in the Q-table but also the accumulated rewards.

In ***RG***, we

- Trained the agent with non-stationary action selection in discounted continuous task
- Evaluated the agent with 5000 steps over resources of equal weights and unequal weights with scaling, and recorded accumulated rewards for each resource type





- NSW agent achieved balanced distribution between the three types of resources while the scalarized agent tends to favor resource closest to the homebase
- NSW algorithm is immune to scaling of the item weights, whereas the scalarized version is affected drastically favoring the resource with inflated weight

Conclusion

- In this project, we introduced the novel problem of producing policies that maximize the expectation of welfare of accumulated rewards in the context of multi-objective Markov decision processes.
- We proposed adaptations of the Q-learning algorithm to solve relatively small-scale problems and provided an extensive empirical validation.
- As future work, we may consider other fair welfare functions and extend our algorithm, which works only in tabular settings, to function approximations using deep neutral networks.

References

- Siddique, Umer, Paul Weng, and Matthieu Zimmer. "Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards." *International Conference on Machine Learning*. PMLR, 2020
- Fain, Brandon, Kamesh Munagala, and Nisarg Shah. "Fair allocation of indivisible public goods." *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018.
- Caragiannis, Ioannis, et al. "The unreasonable fairness of maximum Nash welfare." *ACM Transactions on Economics and Computation (TEAC)* 7.3 (2019): 1-32.