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:

1. The action extraction process takes account of the accumulated rewards.
2. The agent takes action that maximizes the welfare of rewards when updating the Q-table.
3. 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 $\epsilon > 0$.

1. Initialize $Q(s,a)$, for all $s \in S$, $a \in A(s)$, arbitrarily except that $Q(terminal,) = 0$.
2. Loop for each episode:
   - Initialize $S$.
   - Initialize $R_{t=0}$
   - Loop for each step of episode:
     - Choose:
       - $A = arg\max_{a}[NSW(R_{t=0} + \gamma Q(s,a)) + \text{probability } 1 - \epsilon]$.
       - A random action with probability $\epsilon$.
   - Take action $A$, observe $R, S'$.
   - Update $R_{t+1} = R_{t} + \alpha R_{t} + \gamma Q(s', a') - Q(s, a)$.
   - $S \rightarrow S'$.
   - The policy is then given by
     - $\pi(R_{t=0}, s) = arg\max_{a}[NSW(R_{t=0} + \gamma Q(s,a))]$.

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

1. 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.
2. The expected welfare is not welfare of expectations because of the non-linearity of the welfare functions.
3. The problem of producing the optimal policy is NP-hard even if there is a small number of states.

Simulated Environments

In Taxi, we:

1. Trained our agent with myopic action selection, in discounted continuous task.
2. 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.

In FTN, we:

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

In RG, we:

1. Evaluated the agent with 5000 steps over resources of equal weights and unequal weights with scaling, and recorded accumulated rewards for each resource type.
2. All three variations of the modified Q-learning algorithm seem to yield roughly the optimal solution (all within 1%).
3. 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.

Results

In Taxi, we:

1. Trained our agent with myopic action selection, in discounted continuous task.
2. 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.

In FTN, we:

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

In RG, we:

1. Evaluated the agent with 5000 steps over resources of equal weights and unequal weights with scaling, and recorded accumulated rewards for each resource type.

References