Dynamic preference allocation for multi-objective, multi-agent reinforcement learning

Asad Jeewa $^1$ Anban Pillay $^1$ Jonathan Shock $^2$ Benjamin Rosman $^3$

$^1$University of KwaZulu-Natal $^2$University of Cape Town $^3$University of Witwatersrand

How can agents coordinate their behaviour to achieve their individual and the collective objectives of the system?

**Motivation and Objective**

**Goal:** To train RL agents to balance conflicting objectives in multi-agent environments.

**Motivation:** Robust reinforcement learning algorithms are required to solve complex real-world problems that necessitate coordination among multiple agents as well as reasoning about their collective and individual benefits.

**Problem:** These environments are challenging due to:
- Divergent Rewards
- Temporally and spatially extended, interdependent objectives
- Severe non-stationarity
- Partial observability
- Sparse Rewards

**Case Study: Clean Up**

- Agents are rewarded for eating apples.
- Apples grow at a rate based on the cleanliness of a nearby river.
- The reward for cleaning the river is implicit.
- There is a tension between the short-term individual incentive and the long-term group interest.
- A turn-taking approach that benefits all agents would be an ideal solution.

![Figure 1. Clean Up Environment [1]](image)

**Current Approaches**

![Figure 2. Training performance on Clean Up$^1$](image)

(a) $[1]$ (b) $[2]$

$^1$Prosocial training uses the mean reward of all agents as a training signal. ACB is a actor-critic baseline built on top of A3C $[1]$. RPM is a novel MARL approach that randomly samples policies from a buffer $[2]$. Further details can be obtained from the papers.

- Numerous baselines fail to perform meaningful learning.
- Prosocial approaches have exhibited some success but such methods struggle with credit assignment $[1]$.

**Proposed Approach**

We propose:
- Making the reward function explicit for all objectives.
- Vectorising the reward function to handle-trade-offs and incorporate preferences into the learning process $[4]$.
- Training a universal policy that is parametriised by preferences $[5]$.
- Introducing a high-level controller that dynamically allocates preferences to agents.
- Semi-sequential training for handling non-stationarity.

**Future Work**

- Dividing agents into teams
- Making the agents heterogeneous
- Modelling the behaviour of agents in the environment

**References**


