Multi-Objective Reinforcement Learning Application in the View of Agent AI Safety

Manu Hajari\textsuperscript{1}, V. Surekha\textsuperscript{2}, K Chandra Mouli\textsuperscript{3}

\textsuperscript{1}Department of Computer Science and Engineering (AIML), Gokaraju Rangaraju Institute of Technology and Science, Hyderabad, Telangana, India.

\textsuperscript{2}Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Technology and Science, Hyderabad, Telangana, India.

\textsuperscript{3}Department of Computer Science and Engineering (AIML), CMR Institute of Technology, Hyderabad, Telangana, India.

Received: 22.01.2024 ● Accepted: 26.03.2024 ● Published Online: 29.03.2024

Abstract: Reinforcement learning is frequently used to make single decisions using a single decision-based algorithm, where a single objective function is solved, for example, in an agent playing chess, the movement of any player is observed from previous action by considering whether it has received positive or negative reward based on this it will proceed. Nowadays, the requirement of making sequential decisions is complex for this reason. Learning involves using two or more objective functions to address a challenge in multi-objective reinforcement. As an illustration, consider a client purchasing a car and maximising conveniences like minimising fuel use, keeping costs low, and fitting additional luxury items in the automobile. Multi-objective reinforcement learning is applicable in decision-making sequential problem-solving applications. Still, at the same time, we need to avoid the risk that Artificial intelligence agents will be having when problem-solving.

\textsuperscript{*}Correspondence: Assistant Professor, Department of Computer Science and Engineering (AIML), Gokaraju Rangaraju Institute of Technology and Science, Hyderabad, Telangana, India. Email:manu.hajari@gmail.com

\textcopyright{} 2024 by the Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License CC BY-NC-ND 4.0.
using MOO-based algorithms. In this paper, we discuss how we use Multi-Objective Optimization (MOO)-
based algorithm to low-impact agents for artificial intelligence (AI) safety.

**Key words:** Markov Decision Processes, Multi-Objective, Reinforcement Learning, Artificial Intelligence.

1. **INTRODUCTION**

Through performing actions and observing the outcomes of those activities, an agent learns to respond
to its environment via reinforcement learning, a feedback-based machine learning approach. Each
positive activity results in positive feedback for the agent, while each poor action results in negative
feedback or a punishment for the agent. In contrast to unsupervised learning methods, reinforcement
learning uses feedback to help agents learn on their own from data that has not been labelled.

Reinforcement learning functions in a hit-and-miss fashion [1]. A positive reward is given if the
hit transits to a good action, while a negative reward is given if the hit transits to a poor action. A
computer agent learns to carry out a job using this sort of machine learning approach through several
iterations of trial and error. Interactions with a dynamic environment. With this kind of learning, the
agent may complete the job without interference from humans and without being explicitly taught to do
so by making a series of decisions that maximise a reward measure. A subset of machine learning called
reinforcement learning involves agents acting with the aim of maximising their cumulative rewards [2].

An agent learns continually in an interactive environment based on its own actions and experiences
by employing a series of trial-and-error procedures. Finding an appropriate action model that would
raise the agent’s overall cumulative reward is the only goal. Here are three methods for putting
reinforcement learning algorithms into practise [3]. Maximising a value function is the core goal of the
value-based approach. In this case, a policyholder expects a long-term continuation of the existing
situation. Policy-Based – In policy-based, you can come up with a strategy that helps to accumulate
maximum rewards in the future through possible actions performed in each state [4]. Two types of
policy-based methods are deterministic and stochastic. Model-Based – In this method, we need to
create a virtual model for the agent to help in learning to perform in each specific environment type
of Reinforcement Learning.

There are two types:

Positive reinforcement is termed as when an event occurs due to specific behavior, and increases the
strength and frequency of the behavior. Mathematical frameworks called Markov Decision Processes
(MDPs) are used to map RL solutions. A collection of parameters that consists of a reward (R), a
model (T), a set of feasible actions (A) in each state (S), a set of finite states (S), and a policy (.). The
results of deploying an action to a state depend on current actions and states rather than on past actions or states. It consists of States: S, Model: T(S, an S') P(S'— S, a), Actions: A(s), A,
Reward: R(S), R(S, a), R(S, a, S'), Policy: (S) − > a − *. In the real world, agents interact with the
environment and perform the action through learning [5]. When these effects are measured in terms
of a single scalar objective, such problems can be modeled as a Markov decision process. However
many real-world problems require multiple objective problems to be solved leading to the Multiple Objective Markov decision process model to be used [6]. Multi-Objective Markov Decision Processes (MOMDPs) accommodate this by allowing vector-valued rewards since we use MOMDP for satisfying multiple objective functions depending on certain application use single scalar MDP or use vector-valued policies returns while achieving this like if an agent wants to perform multiple actions to meet multiple objectives, for example, we have a robot performing household chores like robot carrying a vase. It should carry it carefully without breaking it and place it on the table, another robot cooking omelet, need to prepare this it has to break the egg [7].

2. MULTI OBJECTIVE MARKOV DECISION PROCESS
Reinforcement Learning for multiple objective functions can be modeled in different ways, we implement using MOMDP, and MOMDP [8]. In many modeling domains, however, there is no unique objective to be optimized, but multiple, potentially dependent, and conflicting objectives [9]. For example, in designing a computer system, one is intent not only in maximizing performance but also in minimizing power. Similarly, in an inventory management system, one wish to optimize several potentially dependent costs for maintaining each kind of product, and in AI planning, one wishes to find a plan that optimizes several distinct goals. The usual MDP model is insufficient to express these natural problems [10]. Single-objective optimization (SOO) can be expressed as optimizing a problem by using a single objective function. In contrast, multi-objective optimization (MOO) utilizes two or more objective functions to solve a problem. Numerous SOO algorithms have been considered for the modeling of linear and non-linear systems [11]. Multi-objective reinforcement learning (MORL) is a abstraction of standard reinforcement learning where the scalar reward signal is put out to multiple feedback signals, in essence, one for each objective. MORL is the operation of learning policies that Minimize cost of car while buying and maximizing the comfort, and also maximizing performance whilst minimizing fuel utilization and emission of pollutants of a vehicle are examples of multi-objective optimization problems having two and three objectives, respectively. Optimize multiple criteria all together.

3. PARAMETERS IN MULTI OBJECTIVE
For example, in robotic locomotion, we aimed to maximize forward velocity but also minimize joint torque and effect with the ground. The subpart of reinforcement learning that handle with multiple objectives, i.e., a vector reward function rather than a scalar, is known as Multi-objective reinforcement learning (MORL) In the MORL domain, there are two standard methods that are usually taken:
1) The single-objective. Single-objective experienced to use a scalar objective function that is a weighted sum or a function of all the objectives. In this regard, it is sometimes ordinary to order or rank the objectives for selecting the appropriate weights; and also order or rank the solutions produced.
2) The alternative Pareto, Pareto strategy attempts to find multiple solutions to the MORL problem
that offer trade-offs among the different types of objectives. In other term, these multiple solutions, also called Pareto solutions, are non-superior or non-dominating over each other. The utility of a user is taken from a single execution of a policy. In this case, to make use of multi-objective reinforcement learning, the expected utility of the returns must be optimized. Numerous scenarios exist where a user’s preferences over objectives (also called as the utility function) are unknown or hard to specify. In such scenarios, a set of optimal policies must acquire a knowledge. However, settings where the expected utility must be maximized have been mainly overlooked by the multi-objective reinforcement learning community and, as a result, a set of optimal solutions has yet to be defined [6].

3.1. Policies based agent environment, space in order to get MEU long term reward

3.1.1. Definition of policies

When making decisions in the real world, decision makers must make trade-offs between multiple, often differ, objectives. In many real-world settings, a policy is only executed single time. A value function is a representation of the relationship between an environment observation and the value (i.e., the anticipated total long-term reward) of a strategy. The tactic of an agency is a policy. It is more accurate to define Markov Decision Process (MDP) as a tuple first. 

\[ \begin{align*}
S &= \text{set of states}, \\
A &= \text{set of actions}, \\
P &= \text{(Probability of transitioning into a state for each state currently in existence and for each action) state transition probability matrix}, \\
R &= \text{Given a state and an action, a reward function}, \\
y &= \text{discount factor, between 0 and 1.}
\end{align*} \]

A policy is thus an estimation of the probability distribution of actions given states. Every action has that chance of happening when an agent is in a certain condition. The policy may occasionally be stochastic rather than deterministic. In this scenario, the policy provides a probability distribution across a collection of actions rather than a single unique action, \( a \). Any RL algorithm’s general objective is to discover the best course of action for achieving a certain objective.

3.1.2. Maximum Expected Utility (MEU)

According to the maximisation of expected utility principle, a rational actor should make the decision that would maximise their expected utility. The MEU principle is a strategy for intelligent behaviour in other words. Consider the following example to illustrate. I want to know whether I need to bring an overcoat. As a result, if it rains and I don’t have an overcoat, my utility is 0, but it is 15 if I have. If it doesn’t rain and I don’t have an overcoat, my utility is 18 units; if I do, it’s 15 units. There is a 50% probability of rain every day. The Illustrating the example of Maximum Expected Utility is shown in Figure 1.
In the above example, we calculated the EU for both cases, and according to MEU, the EU of the left part is higher; therefore, I should carry the overcoat.

- In order to get low impact Agent AI safety using multi objective reinforcement learning we need to train our agents based on the requirements of what optimal set of actions chosen, mapping the state of the environment to the set of action policies. The Reinforcement learning Environment Model is shown in Figure 2.

Agents that pick their actions only based on actors do so by using a direct policy representation. They are also referred to as "policy-based" agents. The policy may be stochastic or deterministic. These agents can manage flows of action spaces and are generally simpler than other types, albeit the training process may be susceptible to noisy measurement and converge on local minima. In order to properly criticise the actor, the critic also gains knowledge of the value function from the rewards. These agents often have the ability to transport both discrete and continuous action spaces. The following categories of performers and critics are supported by reinforcement learning: Those who compute a policy’s predicted cumulative long-term reward based on an observation are known as V(S—V) critics.

4. CONCLUSION
In this paper we aim to low impact agent AI safety using multi objective reinforcement learning approaches. to achieve this many different models are available based on the type of agent, agent environment, space, observation, task. however here in this paper we presented the basic overview
needed to achieve this by making agents to learn through policies or through learning reinforcement
learning algorithm. Basically MORL model/method is used to get optimal solution with long term
reward, reward can be maximized by using MEU maximum expected utility and RL continuously up-
dates policies parameters based on actions, observations, and incentives, the learning system constantly
modifies the policy settings. The learning algorithm seeks to identify the best course of action that
maximises the task’s projected cumulative long-term payoff. Therefore, MORL utilised to get vector
solutions rather than scalar ones in order to reduce the influence AI safety by taking into account all
the factors listed in this section.

References

multiple objectives. In STACS 2006: 23rd Annual Symposium on Theoretical Aspects of Computer Science,

multi-objective mdps. In International Conference on Tools and Algorithms for the Construction and

[3] Huan Xu and Shie Mannor. Probabilistic goal markov decision processes. In IJCAI Proceedings-


