**Experiment 01**

|  |
| --- |
| **Title: Simulate the Markov Decision Process (MDP)** |

**Objective:**

Student Needs to Use the MDP toolbox

Get familiarized with all the modules/ Functions present in it.

Apply the MDP for their problem statement

**Books/ Journals/ Websites referred:**

* Markov Decision Processes in Artificial Intelligence MDPs, Beyond MDPs and Applications, Edited by Olivier Sigaud, Olivier Buffet, Wiley Publications, 2010
* https://pymdptoolbox.readthedocs.io/en/latest/api/mdptoolbox.html

**Resources used:** Markov Decision Process (MDP) Toolbox for python

**Theory:**

The MDP toolbox provides classes and functions for the resolution of descrete-time Markov Decision Processes.

Available modules with MDP toolbox

**example**

Examples of transition and reward matrices that form valid MDPs

**mdp**

Makov decision process algorithms

**util**

Functions for validating and working with an MDP

**Problem Consideration:**

In the MDP Toolbox, the Forest MDP is an example of a discrete-state and discrete-action MDP that is commonly used to demonstrate various MDP algorithms. The Forest MDP represents a simplified version of a forest management problem, where a forest manager must decide how much to harvest each year in order to maximize long-term profits.

Generate a MDP example based on a simple forest management scenario.

This function is used to generate a transition probability (A × S × S) array P and a reward (S × A) matrix R that model the following problem. A forest is managed by two actions: ‘Wait’ and ‘Cut’. An action is decided each year with first the objective to maintain an old forest for wildlife and second to make money selling cut wood. Each year there is a probability p that a fire burns the forest.

Here is how the problem is modelled. Let {0, 1 . . . S-1 } be the states of the forest, with S-1 being the oldest. Let ‘Wait’ be action 0 and ‘Cut’ be action 1. After a fire, the forest is in the youngest state, that is state 0.

*Output of the Markov Decision process:-*

*Policy:* The optimal policy, which specifies the best action to take in each state of the MDP. The policy can be represented as a mapping from states to actions, or as a vector of action probabilities.

*Value function:* The value function, which specifies the expected cumulative reward from each state under the optimal policy. The value function can be represented as a mapping from states to values, or as a vector of values.

*Optimal value:* The optimal value, which is the expected cumulative reward from the start state under the optimal policy. This provides a measure of the quality of the optimal policy.

The Forest MDP can be solved using a variety of algorithms, such as value iteration, policy iteration, or Q-learning. The optimal policy can be used to determine the optimal harvest level at each age of the forest. Overall, the Forest MDP is a useful example of a discrete-state and discrete-action MDP that is widely used in the MDP literature.

**Implementation Code:**

**Output Screenshot:**

**Please, Find the Implementation of Code and Output in below google colab link:-**

<https://colab.research.google.com/drive/1hTldkEJBFOoOiwVj-nT1ZXcgHTDNCuX0#scrollTo=6pMSGugRDDl8>

**OR**

**Kindly find code and output in E1\_MDP.ipynb**

**Conclusion (Students should write understanding of MDP):**

# MDP stands for Markov Decision Process, which is a mathematical framework for modeling decision-making in situations where the outcome depends on both the current state and the action taken. MDPs are widely used in fields such as operations research, artificial intelligence, control theory, and economics.

# In an MDP, the system is modeled as a set of states, actions, and transitions between states, and a reward function that specifies the desirability of each state-action pair. The system is assumed to satisfy the Markov property, which means that the probability of transitioning to a future state depends only on the current state and action, and not on the past history.

# The goal in an MDP is to find a policy, which specifies the best action to take in each state, in order to maximize the expected cumulative reward over time. This can be done using a variety of algorithms, such as value iteration, policy iteration, Q-learning, and actor-critic methods.

# Application (MDP):

# MDPs have applications in a wide range of fields, such as robotics, finance, healthcare, energy management, and environmental management. They provide a powerful framework for modeling decision-making under uncertainty, and for designing optimal decision strategies that can improve outcomes and reduce costs.