**Batch: MSC-AI Part-1 Roll No.: 15**

**Experiment:- 12**

**Grade: AA / AB / BB / BC / CC / CD /DD**

|  |
| --- |
| **Title:**   **Write a program to implement Real-Time Dynamic Programming (RTDP)** |

# Objective:

Students should write

# 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://towardsdatascience.com/introduction-to-reinforcement-learning-rl-part-4-dynamic-programming-6af57e575b3d
* https://github.com/instance01/RTDP

# Resources used:

# Theory:

Real-Time Dynamic Programming (RTDP) is a technique used in artificial intelligence and control systems to solve decision-making problems in real-time. It is a type of reinforcement learning algorithm that uses dynamic programming to find optimal solutions to decision-making problems while the system is running.

The basic idea of RTDP is to incrementally build a search tree of possible states and actions, where each node in the tree represents a state and each edge represents an action. The algorithm uses an iterative process to improve the estimates of the values of each state and the optimal policy to follow.

The RTDP algorithm starts with an initial estimate of the value of each state, and then uses a series of iterative steps to improve these estimates. At each step, the algorithm selects a state to evaluate, and then evaluates the values of all the states that can be reached from that state by taking a single action. The algorithm then updates the value estimate of the selected state based on the values of the states that can be reached from it.

RTDP is particularly useful for solving decision-making problems in real-time, where the optimal solution must be found quickly and continuously updated as new information becomes available. It has been used in a variety of applications, including robotics, game playing, and control systems for manufacturing and transportation.

Real-Time Dynamic Programming (RTDP) is a type of reinforcement learning algorithm that combines dynamic programming with online search to find optimal solutions to decision-making problems in real-time. RTDP is particularly useful for solving problems that have large state spaces, where traditional search algorithms such as breadth-first search and depth-first search are not practical due to their high computational complexity.

In RTDP, the goal is to find the optimal policy, which is a mapping of states to actions that maximizes the expected cumulative reward over time. The algorithm starts with an initial estimate of the value function, which represents the expected cumulative reward from each state. The value function is defined recursively as follows:

V(s) = max\_a { r(s,a) + γ ∑\_s' P(s'|s,a) V(s') }

where s is a state, a is an action, r(s,a) is the immediate reward for taking action a in state s, P(s'|s,a) is the probability of transitioning to state s' from state s after taking action a, γ is a discount factor that controls the trade-off between immediate and future rewards, and V(s') is the value of state s'.

The RTDP algorithm iteratively refines the value function by performing a series of online search steps. At each step, the algorithm selects a state to evaluate, and then expands the search tree by evaluating the values of all the states that can be reached from that state by taking a single action. The algorithm then updates the value estimate of the selected state based on the values of the states that can be reached from it. The search tree is pruned as soon as the algorithm determines that a state has a low probability of being reached.

RTDP can be seen as a form of anytime algorithm, which means that it can return a solution at any time during its execution and that the quality of the solution improves as the algorithm runs for a longer time. RTDP is particularly useful for solving decision-making problems in real-time, where the optimal solution must be found quickly and continuously updated as new information becomes available. It has been used in a variety of applications, including robotics, game playing, and control systems for manufacturing and transportation.

The implementation of RTDP can be divided into two main phases: initialization and iteration.

1) Initialization:

1.1) Initialize the value function V(s) for all states s to a reasonable estimate.

1.2) Set the current state to the initial state.

2) Iteration:

2.1) Generate a random number between 0 and 1. If the number is less than a specified threshold ε, select a random action. Otherwise, select the action that maximizes the expected cumulative reward.

2.2) Execute the selected action and observe the resulting state and immediate reward.

2.3) Update the value function estimate for the current state using the Bellman update equation.

2.4) If the observed state is a terminal state, go back to step 1 with the initial state. Otherwise, set the current state to the observed state and go back to step 1.

The threshold ε is used to control the exploration-exploitation trade-off. When ε is high, the algorithm is more likely to select a random action, which promotes exploration. When ε is low, the algorithm is more likely to select the action that maximizes the expected cumulative reward, which promotes exploitation.

In RTDP, the value function estimate is updated using the Bellman update equation, which is defined as follows:

V(s) = max\_a { r(s,a) + γ ∑\_s' P(s'|s,a) V(s') }

where s is the current state, a is the selected action, r(s,a) is the immediate reward for taking action a in state s, P(s'|s,a) is the probability of transitioning to state s' from state s after taking action a, γ is a discount factor that controls the trade-off between immediate and future rewards, and V(s') is the value of state s'.

The Bellman update equation is a recursive equation that computes the value of a state based on the values of its successor states. The algorithm iteratively applies the Bellman update equation to refine the value function estimate until it converges to the optimal value function.

# Implementation (Code):

**Kindly find code and output in E12\_RTDP.ipynb**

# Output Screenshots:

# Conclusion (Students should write in their own words):

# Overall, the RTDP algorithm is a powerful and flexible reinforcement learning technique that has been successfully applied in a variety of real-time decision-making problems. The performance of the algorithm depends on the quality of the value function estimate, the exploration-exploitation trade-off, and the computational resources available.

# Applications:

# Real-Time Dynamic Programming (RTDP) has been applied to a variety of decision-making problems in different domains. Some examples of applications of RTDP are:

# 1) Robotics: RTDP has been used to control the movement of robots in real-time, such as path planning, obstacle avoidance, and grasping. RTDP allows the robot to make decisions based on its current state and the expected cumulative reward, which leads to more efficient and effective movements.

# 2) Game playing: RTDP has been applied to games such as chess, checkers, and backgammon. The algorithm can search the game tree in real-time and make optimal decisions based on the expected cumulative reward. This leads to stronger game-playing performance and has been used in various game-playing competitions.

# 3) Control systems: RTDP has been used to control systems in manufacturing, transportation, and other industries. For example, RTDP can be used to optimize the flow of materials in a factory or to control the movement of vehicles in a transportation network. The algorithm can make decisions in real-time based on the current state of the system and the expected cumulative reward.

# 4) Finance: RTDP has been used to optimize investment decisions in finance. The algorithm can make decisions in real-time based on the current market conditions and the expected cumulative reward. This leads to more efficient and effective investment strategies.

# 5) Natural language processing: RTDP has been applied to natural language processing tasks such as dialogue management and language generation. The algorithm can make decisions in real-time based on the current dialogue state and the expected cumulative reward, which leads to more effective and engaging conversations.

# Overall, RTDP is a powerful algorithm that can be applied to a wide range of real-time decision-making problems in various domains. The flexibility and adaptability of the algorithm make it a popular choice for researchers and practitioners in different fields.