**Experiment :- 11**

|  |
| --- |
| **Title:**  **Write a program to implement Rollout algorithm** |

# 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://medium.com/chiukevin0321/motion-planning-for-self-driving-cars-week-5-6-4de794bcad66
* https://github.com/alirezaig/RolloutPower

# Resources used:

# Python

# Theory:

# The Rollout Algorithm is a Monte Carlo Tree Search (MCTS) technique commonly used in artificial intelligence and game playing. It is used to estimate the expected outcome of a game by simulating a large number of possible gameplays.

# In the Rollout Algorithm, the game tree is traversed starting from the current state of the game, and a random playout is simulated until the end of the game. The result of the playout is then used to update the statistics of the nodes in the game tree.

# The playouts are usually performed by a simple heuristic or evaluation function, which estimates the outcome of the game based on the current state. The heuristic is often a very basic one, but can be refined with more advanced methods depending on the complexity of the game.

# By simulating a large number of playouts, the Rollout Algorithm can estimate the probability of winning from a given position. This information can then be used by the MCTS algorithm to determine the best move to make from that position.

# The Rollout Algorithm is widely used in game playing AI, especially in games with a large branching factor, such as Go or Chess.

# The Rollout Algorithm is a Monte Carlo Tree Search (MCTS) technique commonly used in artificial intelligence and game playing. It is used to estimate the expected outcome of a game by simulating a large number of possible gameplays.

# The Rollout Algorithm is typically used in conjunction with other MCTS techniques such as tree policy and backpropagation. Here's a step-by-step explanation of how the Rollout Algorithm works within the context of MCTS:

# 1) Tree policy: Starting from the root node of the game tree, the algorithm traverses the tree using a tree policy to select the best child node to explore. The tree policy typically balances exploration and exploitation, favoring nodes with high potential for reward and/or low visit count.

# 2) Expansion: Once the tree policy has identified a child node to explore, the algorithm creates a new node in the tree for that state and adds it as a child of the selected node. This new node represents a potential future state of the game.

# 3) Simulation/Rollout: At this point, the Rollout Algorithm comes into play. The algorithm performs a simulation of the game starting from the new node that was just added to the tree. The simulation involves playing out the rest of the game using a heuristic or evaluation function to estimate the expected outcome of the game. The outcome of the simulation is a score, which is typically a value between 0 and 1 representing the probability of winning from the current state.

# 4) Backpropagation: After the simulation is complete, the algorithm updates the statistics of all nodes along the path from the root node to the newly added node based on the outcome of the simulation. Specifically, the visit count and reward value of each node are updated to reflect the new simulation result. This process is called backpropagation.

# 5) Repeat: Steps 1-4 are repeated for a large number of iterations until a stopping criterion is met (e.g. a time limit, a maximum number of iterations, or a convergence criterion). At the end of the search, the algorithm selects the move that leads to the child node with the highest estimated reward value, which is typically the child node with the highest win rate or expected value.

# The Rollout Algorithm is a key component of the MCTS approach to game playing, as it enables the algorithm to efficiently explore a large space of possible game states and select the most promising move based on the expected outcome of the game.

# simple example of the Rollout Algorithm:

# Let's say we have a game where two players, A and B, take turns rolling a single six-sided die. The first player to roll a 6 wins the game.

# Starting from the initial state where neither player has rolled the die yet, the Rollout Algorithm would perform the following steps:

# 1) Traverse the game tree to the current state, where neither player has rolled the die yet.

# 2) Perform a random playout by simulating the roll of the die for both players until one of them rolls a 6.

# 3) Update the statistics of the nodes in the game tree based on the result of the playout. If player A wins the playout, then the statistics of the node corresponding to player A's move will be updated to reflect the win, and vice versa for player B.

# 4) Repeat steps 2-3 for a large number of playouts.

# 5) Use the statistics of the nodes in the game tree to determine the best move to make from the current state.

# For example, if the Rollout Algorithm determines that player A wins the game from the current state with a probability of 60%, and player B wins with a probability of 40%, then player A would choose to roll the die in the hope of winning.

# Implementation (Code):

**Kindly find code and output in E11\_Rollout.ipynb**

# Output Screenshots:

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

# In conclusion, the Rollout Algorithm is a Monte Carlo Tree Search technique that is used to estimate the expected outcome of a game by simulating a large number of possible gameplays. The algorithm performs a simulation of the game starting from a new node in the game tree and uses a heuristic or evaluation function to estimate the expected outcome of the game. The outcome of the simulation is a score that is used to update the statistics of nodes in the game tree through backpropagation. This process is repeated for a large number of iterations until a stopping criterion is met, and the algorithm selects the move that leads to the child node with the highest estimated reward value. The Rollout Algorithm is a key component of the MCTS approach to game playing and enables the algorithm to efficiently explore a large space of possible game states and select the most promising move based on the expected outcome of the game.

# Applications:

# The Rollout Algorithm has a variety of applications in artificial intelligence and game playing, including:

# 1) Board games: The Rollout Algorithm is commonly used in games such as Chess, Go, and Checkers to estimate the expected outcome of a game and select the best move.

# 2) Video games: The Rollout Algorithm can be used to generate realistic behaviors for non-player characters (NPCs) in video games, making them more challenging and engaging for players.

# 3) Robotics: The Rollout Algorithm can be used to plan and optimize the actions of robots in real-world environments, such as manufacturing plants or warehouses.

# 4) Financial modeling: The Rollout Algorithm can be used to simulate the performance of financial portfolios and optimize investment strategies.

# 5) Transportation planning: The Rollout Algorithm can be used to simulate traffic patterns and optimize transportation networks, such as bus or subway routes.

# Overall, the Rollout Algorithm is a versatile and powerful technique that can be applied to a wide range of problems in various fields, making it an essential tool for many researchers and practitioners in artificial intelligence and game playing.