CSE440 – Artificial Intelligence
Project Update: Group - 7

The Three Missionaries and Three Cannibals problem is a classic puzzle that involves three missionaries and three cannibals on the left bank of a river, along with a boat capable of carrying at most two people. The objective is to safely transport all individuals from the left bank to the right bank without ever leaving a group of missionaries in a position where they are outnumbered by cannibals (either on the banks or in the boat).

To create an intelligence solver for this problem, we can break this problem into states and constraints. The initial state is (m, c, b) = (3, 3, 1) where m is missionaries on the left bank, c is Cannibals on the left bank and b = 1 means the boat is on the left bank. The goal state is (0, 0, 0) meaning that all missionaries and cannibals have crossed over to the right side and the boat is at the right side also. We can add constraints to check if there are more cannibals then missionary at any side or if the boat has changed position by carrying at least one passenger.

There are two optimal solutions to this problem, both with the same cost (11 crossings):

(3, 3, 1) → (3, 1, 0) → (3, 2, 1) → (3, 0, 0) → (3, 1, 1) → (1, 1, 0) → (2, 2, 1) → (0, 2, 0) → (0, 3, 1) → (0, 1, 0) → (1, 1, 1) → (0, 0, 0)

(3, 3, 1) → (3, 1, 0) → (3, 2, 1) → (3, 0, 0) → (3, 1, 1) → (1, 1, 0) → (2, 2, 1) → (0, 2, 0) → (0, 3, 1) → (0, 1, 0) → (0, 2, 1) → (0, 0, 0)

The methods that we implemented to solve this problem are:

1. Reinforcement learning: This method solves the problem by considering rewards and penalty. We defined the initial state, goal state and the set of actions. Reaching the goal state results in a positive reward, while reaching an invalid state results in a negative reward. There are three hyper-parameters in the Q learning model that we used: α - Learning Rate, (γ) - Discount Factor and (ε) - Exploration Rate. We train the model to run episodes and find out the optimal path to reach the goal state.

2. A* search: The A*search method utilizes a priority queue to explore the most promising states first, keeping track of the cost so far and the estimated total cost to reach the goal. The heuristic function estimates the remaining number of crossings required to reach the goal.

3. Markov Decision Process: MDP employs value iteration to find an optimal policy. The algorithm calculates rewards for reaching the goal state and iteratively updates the value function until convergence. It then extracts the optimal policy for each state and constructs the solution path.

4. Breadth First Search: BFS uses a queue to explore a graph of all possible states by traversing level by level exploring each node in the process.

5. Depth First Search: DFS uses a stack to explore the graph of all possible states by traversing a single node in each level to find the goal state. It backtracks if a goal state cannot be found in a path.

6. Bidirectional Search: This method uses two queues track the front and back searches, while dictionaries store visited states. When a common state is found, the reconstruct function combines the paths from both directions, yielding the complete solution path.