Name- Shreya Pawiktassery
TE comps A
Roll no- 9562

Postlab-7

Q.1. Advantages and disadvantages of state space search

→ Advantages:

1) completeness - State space search algorithms are generally complete, meaning they will find a solution if one exists.
2) Optimality - Some state space search algos, like A* search, guarantee finding the optimal solution when certain conditions are met.
3) Adaptability - State space search algos. can be adapted & customized for different problem domains.
4) Scalability - With proper optimization & pruning techniques, & it can efficiently handle large search spaces.
5) Versatility - It can be applied to a wide range of problems, including pathfinding, planning, etc.

Disadvantages:

1.) Exponential growth - In some cases, the search space may grow exponentially, leading to computational inefficiency and & infeasibility.
2) Heuristic dependency - Performance heavily relies on the quality of heuristics used, which may not always be available or easy to define accurately.
3) Time complexity - It can be high, especially for problems with complex state transitions and goal conditions.

Q2 Advantages and disadvantages of Hill climbing

→ Advantages :

1) Simplicity: Hill climbing is easy to understand and implement
2) Efficiency: It can work well for simple optimization problems with smooth continuous landscapes.
3) Memory efficiency: Requires minimal memory overhead as it only needs to keep track of the current state and its neighbours.
4) Local optima avoidance: Can escape local optima by making incremental improvements
5) Online learning: Suitable for online learning scenarios where real time decisions are required.

Disadvantages :

1) Local optima: Tends to get stuck in local optima, failing to find the global optimum.
2) Plateaus: May struggle on plateau regions where there is little or no gradient information.
3) Initial solution dependency: Highly dependent on the initial solution and can converge to suboptimal solutions if the initial guess is poor.
4) Limited exploration: Lacks global exploration capability, leading to limited coverage of the search space.
5) Non-determination - Behaviour can vary significantly depending on the choice of step size and selection strategy, making it less reliable in certain cases.

Q3) Variations of Hill climbing approaches.

→ 1) Simple Hill Climbing: In this variant, at each step, the algorithm considers all neighbouring states and moves to the one that maximally improves the objective function. This approach aims to reach the peak of the current hill as quickly as possible.

2) First-choice Hill climbing: Instead of examining all neighbouring states, this variant randomly selects one neighbouring state and moves to it if it improves the objective function. If the selected neighbour does not lead to an improvement, another random neighbour is chosen. This approach is efficient.

3) Random-Restart Hill Climbing: This approach involves performing multiple hill climbing searches from different initial states. After reaching a local optimum, the algorithm restarts from a randomly selected initial state and repeats the process.

4) Simulated Annealing: It is a probabilistic variant of hill climbing that allows for "downhill" moves with a certain probability. At each step, the algorithm accepts worse solutions with a decreasing probability allowing it to escape local optima and explore the search space more extensively.

4) Solve the Block World problem by using the STRIPS method

→ The Block world problem involves moving blocks from an initial configuration to a goal configuration using a robot arm. STRIPS is a method of solving this kind of problem

1) Define initial state: Specify the initial configuration of blocks and their positions.
2) Define the goal state: Specify the desired configuration of blocks
3) Define actions: Identify the actions the robot arm can perform (eg. move a block from one position to another, stack a block on top of another)
4) Define preconditions and effects: For each action, specify the conditions that must be true for the action to be executed and the changes that occur when the action is executed.
5) Apply a planning algorithm: Use a planning algorithm like STRIPS to find a sequence of actions that transforms the initial state into the goal state.
6) Execute the plan: Execute the sequence of actions generated by the planning algorithm to move the blocks to the goal state.