Artificial Intelligence\_1-5\_Unit-I

Introduction to Artificial Intelligence: Artificial Intelligence (AI) is a branch of computer science that aims to create intelligent machines capable of simulating human behaviour and cognition.

**Artificial Intelligence Techniques**: AI techniques encompass various methods such as machine learning, neural networks, natural language processing, expert systems, and robotics to replicate human intelligence in machines.

**Intelligent Agents**: Intelligent agents are autonomous entities that perceive their environment and act upon it to achieve specific goals.

**Nature of Agents**: Agents exhibit autonomy, reactivity, proactiveness, and social ability in interacting with their environment.

**Types of Agents**: Agents can be categorized into simple reflex agents, model-based agents, goal-based agents, utility-based agents, and learning agents based on their characteristics and functionalities.

**Learning Agents**: Learning agents utilize algorithms and data to improve their performance over time through experience and adaptation.

**Advantages and Limitations of AI**: AI offers benefits like automation, efficiency, and decision support, but it also poses challenges such as ethical concerns, job displacement, and biases in algorithms.

**Applications of AI**: AI finds applications in various fields including healthcare, finance, transportation, gaming, cybersecurity, and customer service, among others, to enhance productivity and decision-making processes.

**Problem-solving techniques in AI** involve various methods to find solutions in complex environments. State space search is a fundamental approach where the problem is represented as a set of states, and the search involves transitioning between these states until a solution is found. Control strategies dictate how the search process is managed, determining factors such as the order in which states are explored.

**Blind search algorithms**, such **as Depth-First Search (DFS) and Breadth-First Search (BFS)**, are common approaches to traverse the state space without using any domain-specific knowledge. DFS explores as far as possible along each branch before backtracking, while BFS explores neighbouring states before moving deeper into the search tree. These techniques are essential for exploring unknown environments efficiently.

**Depth-First Search (DFS):**

Imagine you're exploring a maze by always taking the first available path until you hit a dead-end. Then, you backtrack to the last intersection and try another path.

DFS's time complexity depends on how deep the maze goes and how many paths each intersection has.

Depth-First Search (DFS):

* V is the number of vertices (nodes) in the graph or search tree.
* E is the number of edges in the graph or search tree.
* Time complexity: O (V + E)

**Breadth-First Search (BFS):**

Picture you're exploring a maze level by level. You start from the entrance, explore all paths on the current level, then move to the next level.

BFS's time complexity depends on how wide the maze is and how far the goal is from the starting point.

* V is the number of vertices (nodes) in the graph or search tree.
* E is the number of edges in the graph or search tree.
* Time complexity: O (V + E)

**Heuristic search** utilizes domain-specific knowledge (heuristics) to guide the search process towards more promising paths, typically towards the goal state, thus improving efficiency compared to blind search algorithms.

Problem characteristics influence the choice of problem-solving techniques. For example, problems with well-defined states and transitions are suitable for state space search, while problems with complex constraints may require constraint satisfaction techniques.

**Production system characteristics** involve rules (productions) that trigger actions based on the current state and conditions. Production systems are commonly used in AI for problem-solving and decision-making tasks.

Generate and test is a problem-solving technique where potential solutions are generated and then tested against the problem constraints until a satisfactory solution is found.

**Hill climbing** is an iterative optimization algorithm that aims to find the best solution by continuously moving towards increasing values of an objective function. However, it can get stuck in local optima.

**Best-first search** explores the most promising states first based on some evaluation function, often using heuristics to guide the search towards the goal.

**A\* search** is an informed search algorithm that combines the advantages of both breadth-first and greedy best-first search. It uses a heuristic function to estimate the cost from the current state to the goal and chooses the path with the lowest estimated total cost.

* Time complexity: O (bd)
* b is the branching factor (maximum number of successors of any node).
* d is the depth of the shallowest goal node.

**Ao\* search is an extension of A\* search** that deals with uncertain environments by considering multiple possible outcomes at each state and selecting actions that maximize expected utility.

**Constraint satisfaction** problems involve finding values for variables that satisfy a set of constraints. Techniques like backtracking, constraint propagation, and local search are commonly used to solve such problems efficiently.