DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE
ARVIND GAVALI COLLEGE OF ENGINEERING, SATARA
EVEN SEM 2025-26
CA1 Examination

Course: B. Tech. **Class: B.Tech** **Semester: VII**
Branch: C.S.E. **Subject Code & Name: BTCOC701 Artificial Intelligence**
Max Marks: 30 **Date:** **Duration: 01:30 Hrs**

Instructions to the Students:
(**1**) Illustrate your answers with neat sketches wherever necessary.
(2) Figures to the right indicate full marks.
(3) Assume suitable data if necessary.
(4) Preferably, write the answers in sequential order.

Q.1 Objective type questions. (All questions are compulsory)

1. Which of the following best describes the “State of the Art” in AI? CO1 1 Mark
a) AI is only theoretical and lacks practical applications
b) AI is widely used in everyday life, such as speech recognition and self-driving cars
c) AI can fully replicate human consciousness d) AI has no impact on business or society

2. An intelligent agent is defined as: CO1 1 Mark
a) A software program with high processing speed b) An entity that perceives its environment and acts upon it
c) A system designed only for data storage d) A robot that performs physical tasks only

3. Which of the following is NOT a component of an agent?
a) Sensors b) Actuators c) Memory d) Compiler CO1 1 Mark

4. Informed search strategies differ from uninformed search strategies because they:
a) Do not use any additional information
b) Use heuristics to guide the search process
c) Always guarantee an optimal solution without computation
d) Only work in deterministic environments CO2 1 Mark

5. A heuristic function h(n) is used to: CO2 1 Mark
a) Measure the exact cost from the start to a node b) Estimate the cost from the current node to the goal
c) Always replace the cost function g(n) d) Eliminate the need for search

6. Which of the following is NOT an example of a Constraint Satisfaction Problem (CSP)? CO2 1 Mark
a) Sudoku puzzle b) Map coloring problem c) Missionaries and Cannibals problem d) Cryptarithmetic puzzle

Q .2 Solve Any two of the following.

A. State the relationship between Agent and Environment. CO1 6 marks

An **Agent** is anything that can perceive its environment through sensors and act upon that environment through actuators. The **Environment** is everything external to the agent with which it interacts.

The relationship is a **continuous perception–action cycle**:

1. **Perception**:
- The agent receives input about the environment through **sensors**.
- Example: A self-driving car’s sensors detect traffic signals, nearby vehicles, and road conditions.

2. **Decision/Processing**:
- The agent’s **internal structure** (rules, goals, utility, or learning) decides what action to take based on percepts.
- Example: The car decides whether to stop, accelerate, or change lanes.

3. **Action**:
- The agent affects the environment through **actuators**.
- Example: The car’s steering, brakes, and accelerator execute the decision.

4. **Feedback Loop**:
- The environment changes because of the action.
- The agent then perceives the updated environment again, continuing the cycle.

Key Points in Agent–Environment Relationship
 Mutual Dependence:

The agent needs the environment to sense and act.
 The environment is influenced/changed by the agent's actions.
 Environment Characteristics: The nature of the environment (fully/partially observable, deterministic/stochastic, episodic/sequential, static/dynamic, discrete/continuous, single/multi-agent) affects how an agent must be designed.
 Performance:
 The success of an agent is judged by its **performance measure**, which depends on how well it interacts with and adapts to the environment

B. Elaborate Artificial intelligence with suitable example along with it's application. CO1 6 Marks

Artificial Intelligence (AI) is a branch of computer science that aims to create systems capable of **thinking, reasoning, learning, and making decisions** similar to humans. It combines techniques from **mathematics, statistics, cognitive science, computer engineering, and data science** to enable machines to mimic human-like intelligence.

Core Components of AI

AI is not a single technology but a collection of interrelated fields.
 Machine Learning (ML): Algorithms that learn from data to improve performance over time without explicit programming. *Example: Email spam filtering, fraud detection.*
 Deep Learning (DL): A subset of ML that uses **artificial neural networks** to recognize complex patterns. *Example: Face recognition, self-driving cars.*
 Natural Language Processing (NLP): Enables machines to understand and respond in human language. *Example: ChatGPT, Google Translate, voice assistants.*
 Computer Vision: Allows machines to interpret and analyze visual data like images and videos. *Example: Medical image analysis, facial recognition, surveillance.*
 Robotics: AI-powered robots perform physical tasks intelligently. *Example: Industrial robots, delivery drones, humanoid robots.*
 Expert Systems: Rule-based systems that mimic human experts in specific domains. *Example: Medical diagnostic systems.*

Applications of AI

AI touches almost every aspect of modern life:
 Healthcare: Disease prediction, drug discov

Computer Vision: Allows machines to interpret and analyze visual data like images and videos. *Example: Medical image analysis, facial recognition, surveillance.*
 Robotics: AI-powered robots perform physical tasks intelligently. *Example: Industrial robots, delivery drones, humanoid robots.*
 Expert Systems: Rule-based systems that mimic human experts in specific domains. *Example: Medical diagnostic systems.*

Applications of AI

AI touches almost every aspect of modern life:
 Healthcare: Disease prediction, drug discovery, robotic surgeries.
 Finance: Stock market prediction, fraud detection, credit scoring.
 Education: Smart tutors, adaptive learning, plagiarism detection.
 Transportation: Self-driving cars, traffic management, logistics.
 Agriculture: Crop monitoring, soil testing, drone spraying.
 Military & Defense: Autonomous drones, surveillance systems.
 Entertainment: Personalized recommendations, AI-generated music/art.

Benefits of AI

 Automation of repetitive tasks.

 Improved accuracy and efficiency.
 Cost reduction and productivity increase.
 Enhanced decision-making with big data insights.

Future Scope of AI

AI is one of the fastest-growing technologies, and its future scope is vast across multiple domains. Some key areas are:

1. **Healthcare**
o AI-based systems for early diagnosis, personalized treatment, and drug discovery.
o Robotic surgeries and medical image analysis.
2. **Education**
o Personalized learning platforms using AI tutors.
o Smart content, automated grading, and virtual classrooms.
3. **Business & Industry**
o Predictive analytics for decision-making.
o Automating repetitive tasks, enhancing productivity, and reducing costs.
4. **Transportation**
o Development of fully autonomous vehicles and AI-powered traffic management.
o Smart logistics and supply chain optimization.
5. **Agriculture**
o Smart farming using AI for crop monitoring, soil analysis, and pest detection.
o Drone-based agricultural solutions.
6. **Cybersecurity**
o AI-powered threat detection and prevention systems.
o Fraud detection in financial transactions.
7. **Daily Life Applications**
o Smart homes, personal assistants, and lifestyle management.
o AI in entertainment (gaming, media personalization).
8. **Scientific Research & Space Exploration**
o AI algorithms for analyzing large datasets from experiments.
o Autonomous robots and rovers in space missions.

C. Define what an intelligent agent is and describe its components. Provide example of an intelligent agent & explain how it interacts with its environment. CO1 6 Marks

Definition of an Intelligent Agent

An **intelligent agent** is an autonomous entity that **perceives** its environment through sensors, **reasons/decides** based on perceptions and knowledge, and **acts** upon the environment through actuators in order to achieve specific goals.

Components of an Intelligent Agent

Every intelligent agent consists of the following core components:

1. **Sensors**
o Devices or modules that perceive the environment.
o Example: Cameras, microphones, GPS in a robot; keystrokes and mouse input in a software agent.
2. **Percepts**
o The inputs received from the environment through sensors.
o Example: "Traffic light is red," "Battery level is low."
3. **Agent Function / Decision-Making Unit**
o Maps percepts (or percept history) to actions.
o Can be implemented using:
 Condition–action rules (reflex)
 State models (model-based)
 Goal reasoning (goal-based)
 Utility maximization (utility-based)
 Learning algorithms
4. **Actuators**
o Components used to take action on the environment.

Example: Motors, wheels, speakers, robotic arms in hardware; sending messages or executing code in software.
5. **Performance Measure**
o A criterion for judging how successful the agent is in achieving its objectives.
o Example: For a vacuum cleaner → cleanliness of the floor, efficiency, battery usage.

Example of an Intelligent Agent
Example: Autonomous Vacuum Cleaner (Robot Vacuum like Roomba)
 Sensors:
o Infrared sensors (detect obstacles)
o Dirt detection sensors
o Battery sensors
 Percepts:
o "There is dirt on the floor."
o "Obstacle ahead."
o "Battery is low."
 Decision-Making:
o If dirt detected → move to that spot and clean.
o If obstacle detected → change direction.
o If battery low → return to charging dock.
 Actuators:
o Wheels (to move)
o Suction motor (to clean)
o Brushes
 Performance Measure:
o Percentage of floor cleaned, minimum collisions, and optimal battery usage.

How it interacts with the Environment
1. The environment (home floor) presents conditions like dirt patches, obstacles, and charging station location.
2. The vacuum's **sensors** perceive these conditions (detects dirt, sees obstacle, checks battery).
3. The **decision-making unit** processes inputs and decides actions (move forward, turn, clean, return to dock).
4. The **actuators** execute the action (motor turns, suction cleans).
5. The environment changes (floor becomes cleaner, vacuum moves location).
6. The cycle repeats continuously → forming a **perception–action loop**.

Q. 3 Solve Any two of the following.

A. Explain A* algorithm for the shortest path. CO2 6 Marks

A* is a popular **informed search algorithm** in Artificial Intelligence. It combines the advantages of **Uniform Cost Search (UCS)** and **Greedy Best-First Search** by considering both:
 g(n): Cost from the start node to node *n*.
 h(n): Heuristic estimate of the cost from *n* to the goal.
 f(n) = g(n) + h(n): Estimated total cost of the cheapest solution through *n*.

At each step, A* expands the node with the **lowest f(n)**.

```
function A_STAR(start, goal):
open_list ← priority queue ordered by f(n) = g(n) + h(n)
open_list.insert(start, f(start))
came_from ← empty map
g[start] ← 0
while open_list is not empty:
current ← node with lowest f(n) in open_list
if current == goal:
return RECONSTRUCT_PATH(came_from, current)
remove current from open_list
for each neighbor of current:
tentative_g = g[current] + cost(current, neighbor)
if neighbor not in g OR tentative_g < g[neighbor]:
came_from[neighbor] = current
g[neighbor] = tentative_g
f[neighbor] = g[neighbor] + h(neighbor)
if neighbor not in open_list:
open_list.insert(neighbor, f[neighbor])
return "No path found"
```

Observations about A*

1. **Completeness**

o A* is complete (it will always find a solution if one exists) **as long as step costs are positive**.

2. **Optimality**

o A* finds an **optimal solution** if the heuristic **h(n) is admissible** (never overestimates the true cost).

o If **h(n) is also consistent (monotonic)**, A* is not only optimal but also efficient.

3. **Efficiency**

o A* expands fewer nodes than UCS or BFS because the heuristic guides the search toward the goal.

o But in the worst case (poor heuristic), it may still expand many nodes.

4. **Heuristic Quality Matters**

o If **h(n) = 0**, A* behaves like UCS (optimal but slow).

o If **h(n) is exact**, A* expands only the nodes on the optimal path (best case).

o If **h(n) overestimates**, A* may lose optimality.

5. **Time & Space Complexity**

o Time: O(bd)O(b^d)O(bd) in the worst case (like BFS), but usually much faster with a good heuristic.

o Space: Requires storing all generated nodes (can be large).

6. **Practical Usage**

o Widely used in **pathfinding** (maps, games, robotics).

o Common heuristics:

- Manhattan Distance (grid worlds).
- Euclidean Distance (maps).
- Domain-specific admissible heuristics.

Example

Suppose we want to move from Start (S) to Goal (G) in a grid.

- **g(n):** Number of steps taken so far.
- **h(n):** Manhattan distance (|x1 – x2| + |y1 – y2|).
- A* will select nodes by minimizing f(n) = g(n) + h(n), ensuring both *shortest so far* + *closeness to goal*.

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE
ARVIND GAVALI COLLEGE OF ENGINEERING, SATARA
EVEN SEM 2025-26
CA1 Examination

B. What is alpha-beta pruning? Explain the function of alpha-beta pruning. CO2 6 Marks

Alpha-Beta pruning is an optimization technique for the **Minimax algorithm**, which is used in **two-player games** (like chess, tic-tac-toe, checkers).

- The goal of minimax is to choose the best possible move assuming the opponent also plays optimally.
- However, minimax explores **all nodes** in the game tree → this can be very costly.
- **Alpha-Beta pruning** reduces the number of nodes evaluated by the minimax algorithm without affecting the final result.

It "prunes" (cuts off) branches of the tree that cannot possibly influence the final decision.

Alpha-beta keeps track of two values during the search:

1. **Alpha (α):**

o The best (maximum) value that the **Maximizer** has found so far.

o Initially $\alpha = -\infty$

2. **Beta (β):**

o The best (minimum) value that the **Minimizer** has found so far.

o Initially $\beta = +\infty$

Rules:

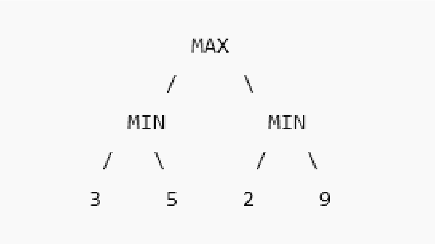
- If at any point **$\alpha \geq \beta$**, we stop exploring further down that branch (pruning).
- This means the current player already has a better option available, so exploring further is unnecessary.

Function of Alpha-Beta Pruning

1. **Improves efficiency:** It reduces the number of nodes evaluated compared to plain minimax.
2. **Does not affect correctness:** The result is the same as minimax, just faster.
3. **Saves time and memory:** By pruning branches, it allows searching deeper levels of the game tree.

Example

Imagine a game tree with **Max** at the root:



```
           MAX
          /   \
       MIN     MIN
      /  \     /  \
     3    5   2    9
```

- Minimax would evaluate **all 4 leaf nodes** (3, 5, 2, 9).
- With **Alpha-Beta**:

o Left MIN gives value = 3 (since MIN chooses the minimum of 3 and 5).

o At right MIN: as soon as we see 2, we know MAX will **never choose 9** (because 3 from the left is already better for MAX than anything larger).

C. What is state space searching? What are the main differences between uninformed and informed search strategies? CO2 6 Marks

State space search is a fundamental problem-solving approach in **Artificial Intelligence (AI)**.

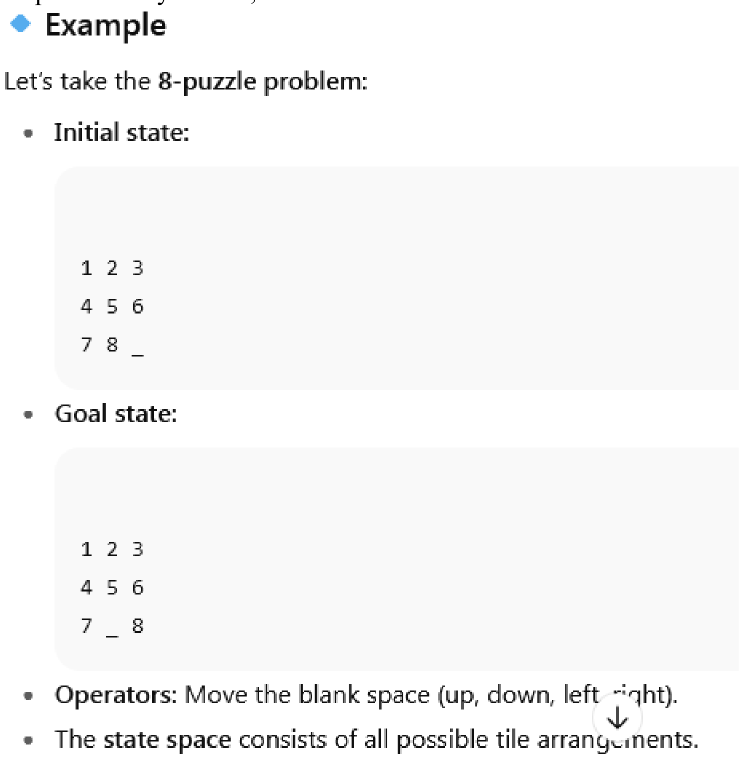
- It involves exploring a set of possible **states** (configurations of the problem) to find a path from the **initial state** to the **goal state**.
- The collection of all these states and the transitions between them is called the **state space**.

State space search = exploring all possible problem states until the goal is found.

Key Concepts

1. **State:** A representation of a possible situation in the problem domain. Example: In the 8-puzzle, each arrangement of tiles is a state.
2. **Initial state:** The starting point of the search.
3. **Goal state:** The desired target configuration.
4. **Operators (Actions):** Rules that define how to move from one state to another. Example: In chess, moving a piece is an operator.
5. **State space:** The set of all possible states reachable from the initial state.
6. **Path:** A sequence of states generated by applying operators.

Types of State Space Search
1. **Uninformed (Blind) Search:**
o No additional information about the problem domain is used.
o Examples: Breadth-First Search (BFS), Depth-First Search (DFS), Uniform Cost Search.
2. **Informed (Heuristic) Search:**
o Uses domain knowledge (heuristics) to guide the search.
o Examples: Greedy Search, A* Search.



🔹 **Example**

Let's take the **8-puzzle problem:**

- **Initial state:**

```
1 2 3
4 5 6
7 8 _
```

- **Goal state:**

```
1 2 3
4 5 6
7 _ 8
```

- **Operators:** Move the blank space (up, down, left, right).
- The state space consists of all possible tile arrangements.

Aspect	**Uninformed Search (Blind Search)**	Informed Search (Heuristic **Search)**
Definition	Searches the state space without any problem-specific knowledge beyond the definition of the problem.	Uses additional knowledge (heuristics) about the problem to guide the search efficiently.
Knowledge used	Only the problem definition (initial state, goal state, operators).	Problem definition + heuristic function $h(n)$ estimating cost to goal.

Search guidance	Explores blindly, systematically covering the state space.	Guided towards promising states based on heuristics.
Efficiency	Less efficient – may expand many unnecessary nodes.	More efficient – expands fewer nodes by focusing on likely paths to the goal.
Examples	- Breadth-First Search (BFS) <br> - Depth-First Search (DFS) <br> - Uniform Cost Search (UCS) <br> - Iterative Deepening Search	- Greedy Best-First Search <br> - A* Search <br> - Hill Climbing <br> - Beam Search
Optimality	Some (like BFS, UCS) can guarantee optimal solution, but often at high cost.	Can guarantee optimality if heuristic is admissible (e.g., A*).
Time & Space Complexity	Generally higher because no guidance (may explore entire state space).	Generally lower because heuristics reduce unnecessary exploration.