Bhumisha Parchani
D15C
38

04/05

AIDS Assignment 1.

Q.1 What is AI? Considering the COVID-19 pandemic situation, how AI helped to survive and renovated our way of life with different application?

→ Artificial Intelligence (AI) refers to the ability of machines and systems to simulate human intelligence, including tasks like learning, reasoning, problem-solving and decision making. It leverages techniques like machine learning and NLP to analyze data, identify patterns and improve performance over time.

Role of AI during the Pandemic:

- Healthcare - AI was used to quickly analyze medical imaging (eg. CT scans) for diagonosing COVID-19 cases and predicting severity.
- Remote Work & Education - The pandemic drove widespread adoption of AI in virtual learning and remote work tools.
- Retail - Retail stores are using computerised models to map out their stores and track inventory. This is in a response to the need given the rush to buy specific items at various stages of the pandemic.

Q.2 What are AI agents terminology, explain with examples.

→ AI agents are autonomous systems that perceive their environment, make decisions and take actions to achieve specific goals.

AI Agent Terminology:

1. Agent
   - An AI system that interacts with the environment and takes action.
   Example - A self-driving car that perceives traffic signals and adjusts speed accordingly.
2. Environment
   - The surroundings in which the AI agent operates.
   Ex - For a chess-playing AI, the chessboard and opponent are part of its environment.
3. Percepts
   - The information received by agent via sensors.
   Ex: A robot vacuum detecting obstacles using cameras and infrared rays.
4. Actions
   - The moves an agent makes in response to percepts.
   Ex: A chatbot responding to a user's query.
5. Percept sequence
   - The entire history of percepts received by an agent
   Ex: A recommendation system tracking user's past preferences.
6. Performance measure
   - Determine success of an agent
   Ex - A self-driving car's performance measure will be reaching the destination safely and in a timely manner

3. How AI technique is used to solve 8 puzzle problem?

→ Initial state:

1	2	3
4	0	6
7	5	8

Goal state:

1	2	3
4	5	6
7	8	0

Misplaced tiles : $h(n) = 2$

Steps to solve 8-puzzle problem by A^* :

1. Initialize a priority queue
2. Insert the initial state with $f(n) = g(n) + h(n)$
3. While queue not empty:
   - Remove state with lowest $f(n)$
   - If state = goal, return solution
   - Generate valid moves (Up, Down, Left, Right)
   - Compute $g(n)$ and $h(n)$ for new states
   - Insert new states into queue
4. Repeat until goal is reached.

Example execution.

Step 1 :	Step 2 :	Step 3 : Goal
1 2 3	1 2 3	1 2 3
4 0 6	4 5 6	4 5 6
7 5 8	7 0 8	7 8 0
(h=2)	(h=1)	(h=0)

Q.4 What is PEAS descriptor? Give PEAS descriptor for following:

→ PEAS stands for performance measure, Environment, Actuators and sensors.

P → Criteria to evaluate the agent's success

E (Environment) → surrounding where agent operates

A (Actuators) → components that allow agent to take actions

S (sensors) → components that help agent perceive its environment

1. Taxi Driver

P - Reaching destination, fuel efficiency

E - Roads, traffic, pedestrians

A - Steering wheel, accelerator, brakers

S - Camera, GPS, speedometer

2. Medical Diagonosis System

P - Accuracy of diagonosis, patient satisfaction

E - Medical records, test results, symptoms

A - Prescription generation, Report generation

S - Patient data input, lab test results

3. Music composer

P - Quality of composition, user satisfaction

E - Musical genres, musical theory constraints.

A - Music synthesizers, Instruments, Generating musical notes

S - Composition structure, user feedback

4. Aircraft Autolander

P - Smooth and safe landing

E - Weather, runway condition, air traffic

A - Flaps, landing gear, throttle

S - Altitude sensor, GPS, wind direction sensor.

5. Essay Evaluator

P - Accurate grading, feedback quality

E - Grammar rules, submitted essays

A - displaying grades & feedback

S - Text input, spelling and grammar checkers

6. Robotic sentry Gun for keck lab.

P - Correctly identifying threats, accuracy

E - Intruders, authorized personnel

A - camera movement, alarm system, firing mechanism

S - Motion sensors, facial recognition.

5. Categorize a shopping bot for an offline bookstore according to each of the six dimension.

→ 1. Partially observable
   - bot may not have full visibility of store's inventory
2. Stochastic
   - Outcomes uncertain due to customer behaviour, stock availability
3. Sequential
   - Bot will suggest books based on previous customer queries
4. Dynamic
   - Bookstore environment changes like books getting sold, new stock
5. Discrete
   - Bots process finite number of actions
6. Multi-agent
   - Bot interacts with customers and store employees, each have their own goals.

6. Differenciate between Model-Based and Utility Based Agents

→

Model-based agent	Utility-based agent
1. Uses an internal model of environment to make decisions.	1. Chooses actions based on utility function that measure performance.
2. Decisions based on past & present percepts.	2. Selects action based on maximizing utility
3. Can be goal-based but doesn't necessarily optimize for best outcomes	3. Optimizes for highest possible utility, ensuring better performance
4. Ex - Robot vacuum using a map to navigate	4. Ex - self-Driving car choosing safest & fastest route

7. Explain the architecture of a knowledge based agent and learning agent.

→ <u>Knowledge-based agent</u>

A knowledge-based agent (KBA) uses stored knowledge to make decisions. It consists of

- Knowledge Base → Stores facts, rules and logic
- Inference Engine → Uses reasoning to derive conclusions
- Perception (sensors) → Gathers new information from environment
- Action Mechanism → Performs appropriate actions based on reasoning

Eg. AI in medical diagnosis uses past cases and symptoms to diagnose diseases.

<u>Learning agent</u>

A learning agent improves its performance over time. It consists of:

- Learning Element → Updates knowledge based on experience
- Performance Element → Decides actions based on current knowledge
- Critic → Provides feedback by evaluating actions
- Problem Generator → suggests new actions to improve learning

Eg. A self-driving car learns from traffic patterns and adjusts driving behaviour.

8. Convert the following to predicates:

a. Anita travels by car if available otherwise travels by bus.

CarAvailable → TravelsByCar (Anita)

¬CarAvailable → TravelsByBus (Anita).

b. Bus goes via Anderi and Goregaon.

GoesVia (Bus, Andheri) ∧ GoesVia (Bus, Goregaon)

c. Car has puncture so is not available.

Puncture (Car)

Puncture (car) → ¬ CarAvailable

Will Anita travel via Goregaon? Use forward reasoning.

From (c)

Puncture (Car) is true

As Puncture (car) → ¬ CarAvailable

From (a)

¬ CarAvailable, we use ¬CarAvailable → TravelsByBus(Anita)

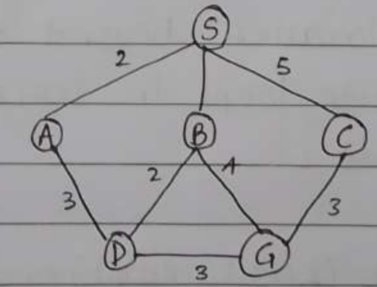
From (b)

GoesVia(Bus, Goregaon)

Since Anita travels by bus, she follows this route

Thus, Anita will travel via Goregaon.

Q.9 Find route from S to G using BFS



→ To find route from S to G using BFS, we systematically explore all nodes level by level starting from source node (S) until we reach destination node (G).

1. Start at S

Queue = [S]

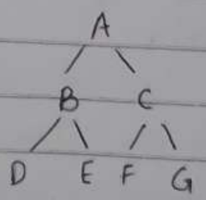
2. From S, we go to it neighbours : A, B, C
Queue = [A, B, C]
3. Dequeue A and explore its neighbours.
Queue = [B, C, D]
4. Dequeue B and explore its neighbours
Queue = [C, D, G]
5. Dequeue C and queue neighbours
Queue = [D, G]
6. ~~Exp~~ Dequeue D
Queue = [G]
7. Dequeue G
As G is our destination, BFS stops here.

Route from S to G : S → B → G

Q.11 What do you mean by depth limited search? Explain Iterative Deepening search with example.

→ Depth Limited Search (DLS) is an uninformed search algorithm that modifies DFS by introducing a depth limit L, preventing exploration beyond the predefined level. This prevents infinite loops in infinite graphs but risks missing goals beyond L.

Iterative Deepening search (IDS) combines DLS with BFS by incrementally increasing the depth limit.
Example :



Goal = G

Iteration 1 : Depth limit = 0

Nodes visited : A

Result : Goal not found.

Iteration 2 : L = 1

Nodes visited : A → B → C

Goal not found

Iteration 3 : L = 2

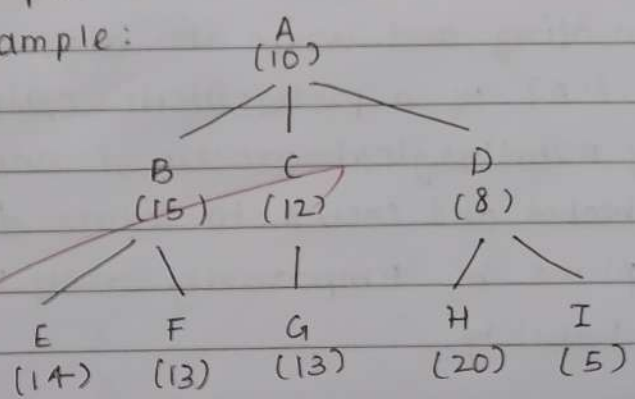
Nodes visited : A → B → D → E → C → F → G

Goal G found at L = 2.

Q.12 Explain Hill climbing and its drawbacks in detail with example. Also state limitations of steepest-ascent hill climbing.

→ Hill climbing is a local search optimization algorithm, which moves toward better neighboring solutions until it reaches a peak.

Example:



Goal = G.

Steps:

Start at root node A(10)

Compare its children B, C and D

Move to child with highest value i.e. B(15)

Repeat for B's children E and F

~~Terminate at E(14)~~

~~The algorithm stops at E(14) not reaching the goal G.~~

Drawbacks:

1. Local Maxima - The algorithm greedily selects the best immediate child and can thus get stuck on local maxim
2. Plateaus - If siblings have equal values, the algorithm can't decide the next step and gets stuck
3. Ridges - Narrow uphill paths require backtracking which Hill climbing algorithm does not support.

Limitations of Steepest-Ascent Hill Climbing:

1. Computationally Expensive - Evaluates all neighbors before selecting the best
2. Can get stuck - It can still get stuck in local maxima plateaus or ridges.
3. No Global Optimality - It only focuses on immediate improvements.

Q.13. Explain simulated annealing and write its algorithm.

→ Simulated annealing (SA) is a probabilistic optimization algorithm inspired by metallurgical process of annealing. where materials are heated and cooled to reduce defects. It escapes the local optima by temporarily accepting worse solution with a probability.

Algorithm:

1. Initialize
   - Set an initial solution and define an initial temperature T.
2. Repeat until stopping condition
   - Generate a new neighbor solution
   - Compute change in cost ($\Delta E = E_{new} - E_{current}$)
   - If new solution is better i.e. $\Delta E > 0$, accept it.
   - If worse, accept it with probability. $P = e^{-\Delta E/T}$

- Decrease temperatur T (cooling schedule)

3. Return best solution.

Example:

Travelling salesman Problem

Swap two cities in a route. Accept a longer route early (high T) but reject it later (low T).

Q.14 Explain A^* algorithm with an example.

→ A^* is a best first search algorithm used in pathfinding and graph traversal. It uses the following formula

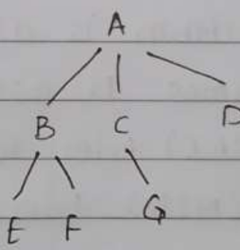
$$f(n) = g(n) + h(n)$$

$g(n) \rightarrow$ cost to reach node n from start

$h(n) \rightarrow$ heuristic estimate of cost to reach from goal to n

$f(n) \rightarrow$ total estimated cost

Example: Goal: G



Node	g (A, n)	h (n, G)
A	0	6
B	1	4
C	2	2
D	4	7
E	3	5
F	5	3
G	6	0

Steps:

1. Start at root node A

$$f(A) = g(A) + h(A) = 0 + 6$$
$$= 6.$$

2. Expand neighbors : B, C, D.

$f(B) = 1 + 4 = 5$

$f(C) = 2 + 2 = 4$

$f(D) : 4 + \cancel{1}7 = 11$

3. Choose lowest value that is C ($f(C) = 4$)

4. Expand neighbors of C : G.

$f(G) = 2 + 4 + 0 = 6.$

5. Goal reached at G with total cost 6.

Advantages →

- efficient for finding shortest paths in weighted graphs
- balances exploration by considering both $g(n)$ & $h(n)$

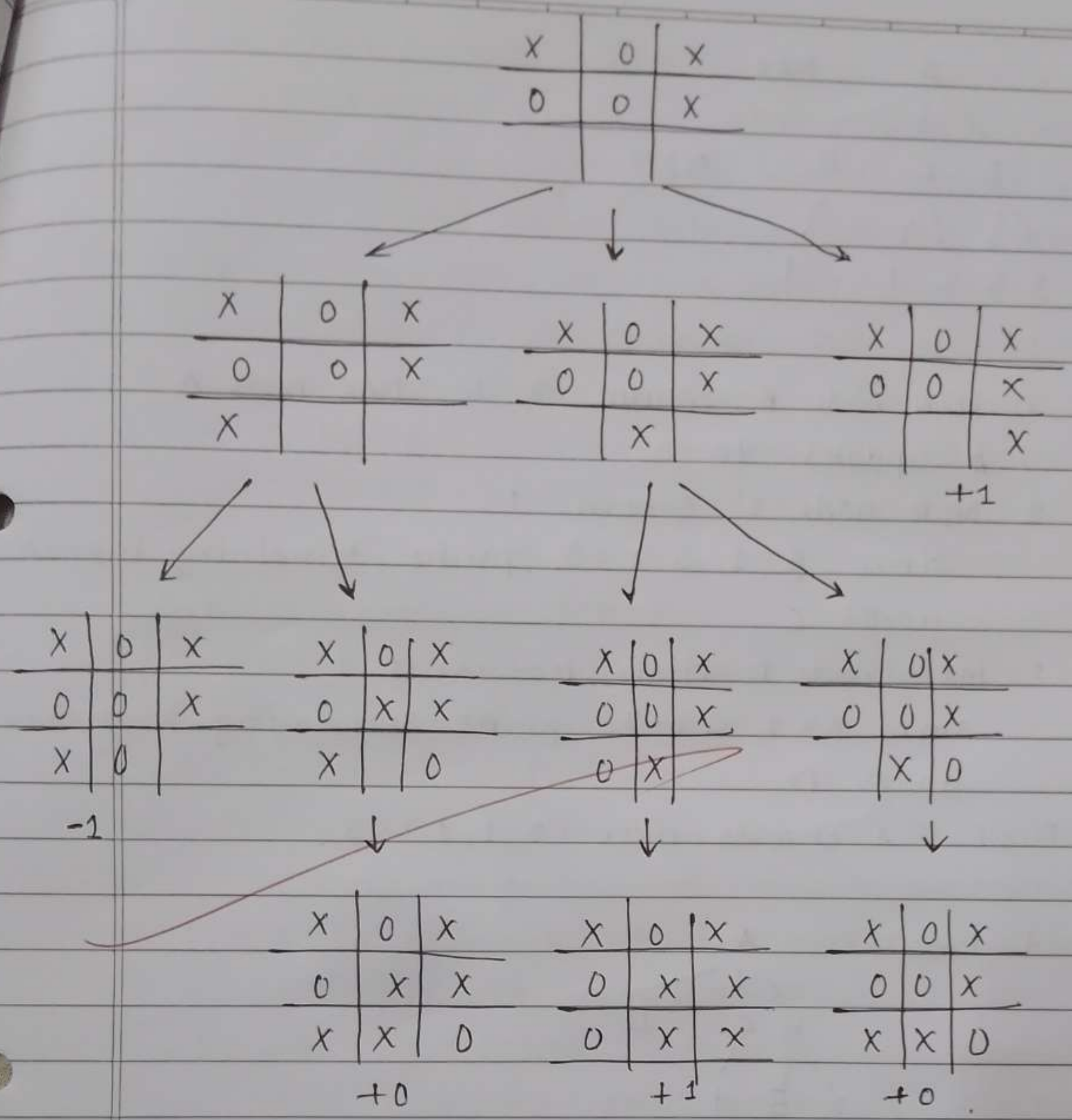
Q.15 Explain Min-Max Algorithm and draw game tree for Tic Tac Toe Game.

→ The Minimax Algorithm is a decision making algorithm used in two-player games. It assumes

- One player (MAX) tries to maximize the score
- Other player (MIN) tries to minimize the score
- Game Tree represents all possible moves.

Algorithm

1. Generate Game tree
   → All possible moves from current state
2. Assign scores
   Terminal states (win / loss / draw), get +1, -1, 0
3. MAX picks highest value from children
   MIN picks lowest value
4. Repeat until Root Node is evaluated



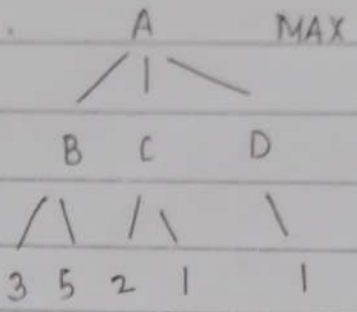
16. Explain Alpha beta pruning algorithms for adversarial search with example.

→ Alpha beta pruning is an optimization technique for the minimax algorithm used in adversarial search purposes like game playing AI. It reduces computational complexity by pruning branches of game tree that do not influence the final decision.

α - best value maximizing player can guarantee

β - best value minimizing player can guarantee.

Example .



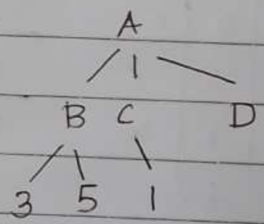
Steps : 1. MIN node B returns 3 to MAX node A
A updates to 3

2. MIN node C return 1
since $\beta = 1 \leq \alpha = 3$, prune remaining branches under C

3. MIN node D finds leaf value 1
since $\beta = 1 \leq \alpha = 3$, prune remaining branches under D

Final : A chooses max (3, 1, 1) = 3.

Pruned Tree :



17. Explain WUMPUS world environment giving its PEAS description. Explain how percept sequence is generated.

→ PEAS for WUMPUS world problem

1. P — • Maximize Rewards
+1000 for collecting gold & exiting grid
• Minimize penalties
-1000 for falling into a pit or being eaten by WUMPUS.
-1 point for each action taken
-10 points for using an arrow

2. E - • Grid Layout (4 x)
containing pits, wumpus, gold, walls, breeze
• Partially observable. - agent cannot see entire grid and must rely on sensory inputs

3. A - Move left, right, forward,
Grab (to collect gold, shoot (to eliminate wumpus)

4. S - Breeze : indicates pit is adjacent
Stench : indicates wumpus in adjacent cell
Glitter : indicates gold
Bump : indicates a wall has been encountered.

<u>Generating percept sequence</u>

1. Initial position - The agent starts at a defined position typically (1, 1)
2. Movement & Perception :
   • As agent moves from one cell to another, it uses sensors to gather info about surroundings
   • For eg. if it detects a breeze, it means there is a pit in adjacent cell.
3. Creating percept sequence
   Each time the agent moves, it records its percepts as sequence.
   For eg : After moving to (1, 2) : [None, Breeze, None]
   (no stench or glitter
   After moving to (2, 1) : [Stench, Breeze, None]
   (indicating nearby dangers)
   This sequence continues as agent explores more cells.
4. Decision making
   The agent uses these percept sequences to make decisions about its next actions based on logical reasoning and inference from previous observations.

18. Solve following crypto-arithmetic problems.

SEND + MORE = MONEY.

→ To solve this problem, we need to assign a unique digit (0-9) to each letter such that the given equation holds true.

1. Set up the Equation

$$\begin{array}{r} S\ E\ N\ D \\ +\ M\ O\ R\ E \\ \hline M\ O\ N\ E\ Y \end{array}$$

2. since M is leading digit, it must be 1 because sum of 2 four-digit numbers cannot exceed 19998.

$\therefore M = 1$

$$\begin{array}{r} S\ E\ N\ D \\ +\ 1\ O\ R\ E \\ \hline 1\ O\ N\ E\ Y \end{array}$$

3. $D + E = Y$ (if carry = 0)

or $D + E = 10$ (if carry = 1)

Tens place :

$N + R + carry = E$

Hundreds place : $E + O + carry = N$

Thousands place : $S + 1 \cancel{\$} = 1 + carry$

$S + carry = 0$

$\therefore S = 9$ since there is no carry

4. Try $\underline{E = 5}$

If $D = 7$ then

$Y = 7 + 5 = 12$ (invalid)

$D = 2$ then

$\underline{Y = 7}$.

5. Y = 7

Assume N = 8

N + R = E

8 + R = 5 (impossible)

∴ $\boxed{N = 6}$

Final: S = 9, E = 5, N = 6, D = 7, O = 0, R = 8, Y = 2

	S	E	N	D		
+	M	O	R	E		
M	O	N	E	Y		

→

	9	5	6	7	
+ 1	0	8	5		
1 0	6	5	2		

Q.19. Consider the axioms:

All people who are graduating are happy

All happy people are smiling

Someone is graduating.

Explain: 1. Represent these axioms in first predicate logic

→ 1. $\forall x$ (Graduating (x) → Happy (x))

2. $\forall x$ (Happy (x) → Smiling (x))

3. $\exists x$ (Graduating (x))

2. Convert each to clause form

1. $\forall x$ (¬ Happy (x) V Smiling (x))

{¬ Happy (x), Smiling (x)}

2. $\forall x$ (¬ Happy (x) V Graduating (x))

{¬ Graduating (x), Happy (x)}

3. Graduating (c)

3. Prove "Is someone smiling?" using resolution technique

$\exists x$ (Smiling (x))

Clause form: C4 = Smiling (y).

$C_1 = \{ \neg Graduating(x), Happy(x) \}$

$C_2 = \{ \neg Happy(y), Smiling(y) \}$

$C_3 =$ Graduating (c)

Resolution between C_3 & C_1

Substitute c for x in C_1:

From Graduating (c)

Happy (c)

Resolving with C_2

Smiling (c)

Graduating (A)

$\neg$graduating (x) $\vee$ happy(x)

Happy (x)

$\neg$happy (x) $\vee$ smiling (x)

smiling (x)

$\neg$ smiling (x)

{ }

→ someone is smiling.

Q.20 Explain Modus Ponen with example.

→ Modus ponen is a fundamental rule of inference in logic. It states that if $P \rightarrow Q$ is true P is true, then Q must also be true. Formula: $\dfrac{P \rightarrow Q,\ P}{Q}$

Ex, if it is raining, then it is soggy ($P \rightarrow Q$)

it is raining (P)

∴ it is soggy (Q).

Q.21 Explain forward and backward chaining algo with example.

→ Forward chaining - It is data driven, inference algorithm that starts with known facts & applies inference rules to derive new facts with until the goal is reached

fact : A, B

Rule : $A \to C$, $B \to D$, $C \wedge D \to I$

Goal : I

Start with A & B

apply $A \to C$ to derive C

$B \to D$ to derive D

$C \wedge D \to I$ to derive I

The goal I reached

Backward chaining :

BC is a goal-driven, starts with goal & works backward to find the facts that support it.

start with goal.

fact : A, B

rule : $A \to C$, $B \to D$, $C \wedge D \to I$

goal : I

find the rule $C \wedge D \to I$

~~sti~~ if C & D are true

use $A \to C$ since A is true, C is true

use $B \to D$ since B is true, D is true

C & D are true, I is true

conclusion : I is reached.