Unit 1

Date
Page

(1) **AI** :- It is a branch of CS by which we can create intelligent machines which can behave like human, think like human & able to make decisions.

Goals → Replicate human intelligence
→ Solve knowledge intensive task
→ Build machines that require human intelligence.

Advs → High Accuracy
→ High speed
→ useful for risky areas
→ Digital assistant
→ useful for public utility

Disadvs → High cost
→ can't think out of box
→ No feeling, No emotion
→ Use dependency on machines.
→ no creativity

(2) **Approaches to AI**

→ **Reactive Machine** → Most basic form of AI
→ donot store memory/past exp. or future actions
→ focus on current situation
→ Eg. IBM's chess playing super comp.

→ **Limited Memory** → can store memory for short time
→ Eg. self driving car.

→ **Theory of Mind** → where bots will understand & react to human emotions, thoughts.
→ Not developed, under research.

→ **Self Awareness** :→ future of AI, super intelligent machines
→ will have own consciousness, sentiment & self awareness.
→ smarter than human.
→ hypothetical, does not exist.

Sangam®

(3) Applica's of AI

- Astronomy : universe related probs.
- Healthcare : better & faster diagnosis.
- Gaming
- Finance : fraud detection
- Data Security : prevent cyber attack.
- Social Media : identify trends
- Travel & Transport : suggesting travel options.
- Automotives : Self driving car.
- Robotics.
- Agriculture :- crop monitoring
- Education
- E-commerce.

(4) Problem Solving in AI

- Goal formulation :- It organises the steps required to formulate 1 goal out of multiple goals as well as actions to achieve that goal.
- Prob. formulation :- It decides which actions should be taken to achieve the formulated goal.

→ Components of Prob-formulation

- Initial State : Starting state of agent towards its goal
- Actions : Possible actions available to the agent.
- Transition Model :- It describes what each act does.
- Goal Test :- Determines if the given state is a goal state
- Path Cost :- Cost to follow a given path.

An optimal sol^n has the lowest path cost among all the sol^ns.

Sangam®

Date
Page

Natural Language Processing :- Branch of AI that enables comps. to comprehend, generate & manipulate human language like English etc.

Components → Natural Language Understanding (NLU)
Mapping given i/p in natural lang. into useful representations.

Natural Lang. Generation (NLG)
producing meaningful phrases & sentences in the form of natural lang. from some internal representation

Steps in NLP

- Lexical Analysis : dividing whole chunk of text into para, sentences & words.
- Syntactic Analysis :- analyzing grammar & order of words such that it shows the relationship among the words.
- Semantic Analysis :- Text is checked for meaningfulness
- Discourse Integration :- Brings out the meaning of immediately succeeding sentence.
- Programmatic Analysis :- Deriving those aspects of lang. which require real world knowledge

Appns :- chatbot
- Improve Search
- Social Media Analytics.
- Search Engine optimization
- Market insights.

Sangam®

(6) Automated Reasoning : area of CS that is concerned with applying reasoning in the form of logic to computing systems.
→ deductive reasoning in maths & logic.
Appⁿs : → Proving theorems
→ checking proofs of
→ Design circuits.

~~Visual P~~
(7) Visual Perception :- ability of comps: to see & interpret images.
⇒ It includes :-
1) acquiring images using image sensors,
2) processing the image digitally ;
3) extracting meaningful info.
4) representing info. as models.
5) converting the model representation into numerical or symbolic info. interpretable by a comps.

(8) Knowledge in AI :- It means to have some info. saved in the memory from the prior experience of sth like abty, skills, objects or facts.
Role of Knowledge in AI :- creating AI
→ demonstrating intelligent behaviour in AI agents.

Types of Knowledge → Declarative (know about sth)
→ Procedural (how to do sth)
→ Meta (knowledge about other types of knowledge)
→ Heuristic (knowledge of some expert in a field)
& rule of thumb, based on previous exp / awareness of approaches, good to work but not guaranteed
→ Structural (basic knowledge to prob. solving)

Sangam®

Date
Page

(9) Heuristic Algo-

- based on intution, guess, explora^n.
- yield suboptimal result
- can't be proven mathematically
- don't yield same ans. everytime

Sol^n guaranteed Algo.

- based on finite set.
- yield optimal result.
- can be proven
- give same ans.

(10) Soft Computing :- Refers to a grp. of computational techniques that are based on AI & natural selec^n.

- It provides cost effective sol^n to the complex real life probs. for which hard computing sol^n doesn't exist.

Appⁿs → Gaming Products like Poker, Checker
↳ Microwave, washing machine, Fridge,
↳ Heater, AC.
↳ Robotics.
↳ & Image Processing, Data Compression

Techniques of Soft Computing

↳ Fuzzy logic (FL)
↳ Neural Nw (ANN)
↳ Genetic Algos (GA)

Longman®

AI (Unit-2)

Date
Page

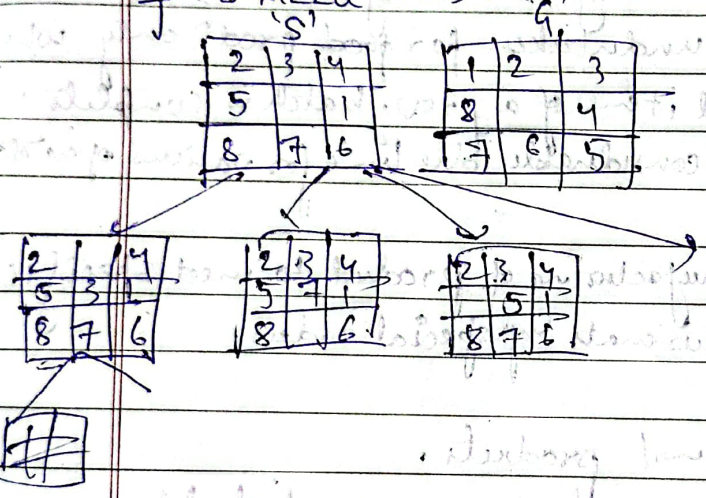
Representing probs in state space :

→ A state space is defined as a set of all possible states of a prob.
→ A state space search representation allows for the formal def^n of a prob. that makes move from the initial to goal state.
→ (Precise) def^n of prob. & agent can properly (analyze) each move.

S: { S, A, Action(s), Result(s, a), Cost(s, a) }

- S: Start, Goal
- A: all possible actions (up, down, right, left)
- Action(s): Action taken in state 's'
- Result(s, a): Resultant state after taking and 'a' on state 's'
- Cost(s, a): cost of taking action 'a' in state 's'.

Eg. 8 Puzzle prob.



Uninformed Searching	Informed Searching
1) Search without info.	→ Search with info.
2) No knowledge	→ Use knowledge to find steps to sol^n (requisite knowledge).
3) Time Consuming	→ Quick sol^n (may have to compromise with optimality).
4) More complex (Time, Space)	→ Less complex
5) DFS, BFS etc.	→ A*, Heuristic DFS, Greedy etc. (estimated value).

NOTE: Heuristic knowledge/Heuristic f^n (h(n)) is used when we have all prob. with exponential growth.

Eg.. Travelling Salesman prob.
We use the nearest neighbour concept as heuristic know

Date / Page

Production System :- Helps in structuring AI progs in a way that facilitates describing & performing the search process.

PS consist of:
- Set of Rules : if-then rules
- Knowledge base : stores info. corr. to particular task.
- Control Strategy : Order in which the rules need to be compared to the db so that conflict can be resolved in min. time.
- Rule Applier :- Applies the rules over control strategy.

Steps to solve the prob :-
- First reduce prob. so that it can be shown in a precise statement.
- Prob can be solved by searching a path through space. [Start → Goal]
- Solving process can be modelled.

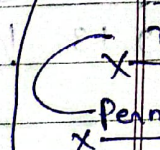
Advs. → Structuring AI Prob.] → excellent tool.
→ Highly modular] → rules - add, remove, change
→ Rules are expressed in natural form] → easy to understand

Characteristics :-

→ Monotonic PS :- App^n of a rule never prevents later app^n of another rule] Rules are independent.

→ Non-Monotonic PS :- is one in which this is not true.

→ Partially Commutative PS :- If app^n of a particular sequence of rules transforms state X to Y, then allowable permutations of those rules also transforms state X to Y.

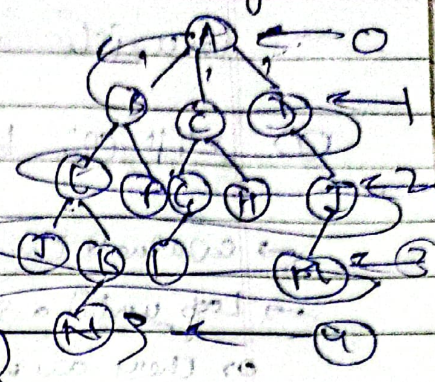


→ Commutative : Both monotonic & partially commutative.

Date
Page

We only have prepeal knowledge
and we don't have domain level knowledge.

BFS



Uninformed Search Tech.
FIFO (Queue)
Shallowest (Nearest) Node.
Complete (surely finds an element)
optimal (Shortest Path)
Time Complexity : O(V+E) + $O(b^d)$
b → branch factor (no. of children of each node)
d → depth

DFS

Uninformed Search Tech.
Stack (LIFO).
Deepest Node
Incomplete (may get trapped in a loop or ∞ search space)
Non-optimal
Time Complexity : $O(b^d)$

Heuristics in AI

→ It is a tech. designed to solve a prob. quickly.
→ It provides good solⁿ but doesn't guarantee to provide optimal solⁿ.
How to calc. Heuristic fⁿ
↳ Euclidean distance
↳ Manhattan distance
↳ No. of misplaced tiles, etc.

Sangam®

Heuristic Search Tech.

① Hill Climbing Algo (Local search along greedy approach, no backtrack)

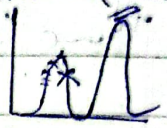
→ Evaluate the initial state

→ Loop until a soln is found or there are no operators left.:

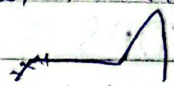
-) Select & apply a new operator.
-) Evaluate the new state:
  - ⊛ if goal then quit.
  - ⊛ if better than current state then it is new current state

Probs. in Hill Climbing

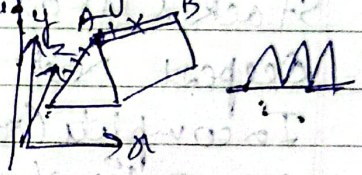
1) Local Maxima Stop

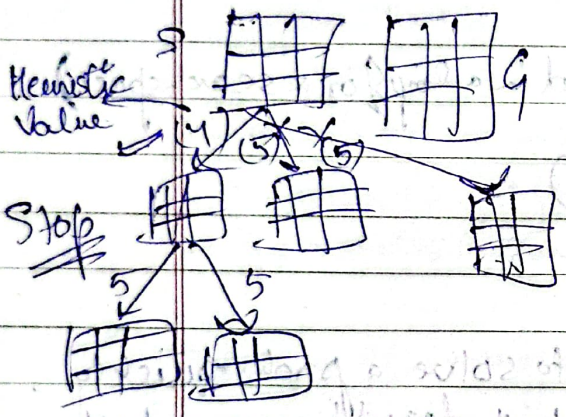


2) Plateau / Flat Maxima Stop



3) Ridge (no direc" change)





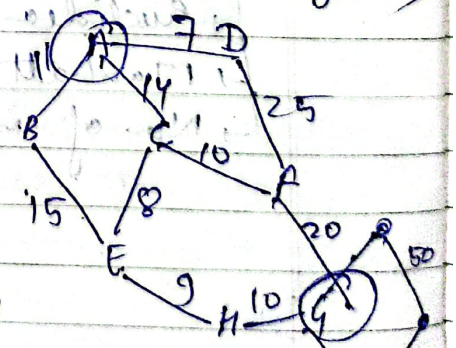
② Best First Search (Informed, Heuristic)

~~get~~ good soln, but no guarantee of optimal.

→ Let 'OPEN' be a priority queue containing initial state.

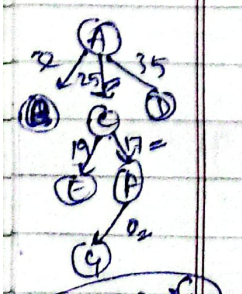
→ Loop:

-) if OPEN is empty, return failure
-) Node ← Remove-First (open):
  - ⊛ if Node is goal, then return the path from initial to ~~goal~~ Node
  - ⊛ else generate all successors of Node & put the newly generated node into OPEN acc'ing to their f' values



Estimated Straight Line Dist.

A → G	= 40
B → G	= 32
C → G	= 25
D → G	= 35
E → G	= 19
F → G	= 17
H → G	= 10
G → G	= 0



Example

Acc'ing to heuristic, A → C → F → G = 14 + 10 + 20 = 44

w/o heuristic, A → C → E → H → G = 41

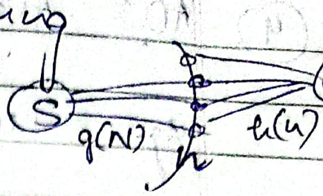
Admissible (guarantees ~~optimal~~ sol^n)

Date ______
Page ______

(3) A* Algo → Informed searching

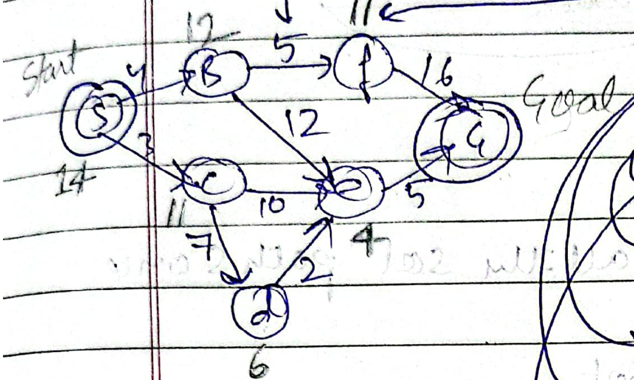
$TC = SC = O(b^d)$

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



g(N): Actual cost from start node to 'n'

h(N): Estimation cost from n to goal node.



S→B = 4+12 = 16

S→C = 3+11 = 14

SC→d = 3+7+6 = 16

SC→e = 3+10+4 = 17

~~Scd~~

~~SB→f = 5+4+11 = 20~~

~~SB→e = 4+12+4 = 20~~

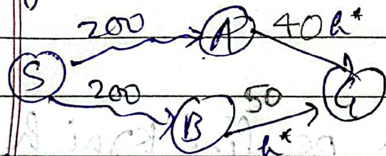
~~Scd→e = 3+7+2+4 = 16~~

Scde→G = 3+7+2+5 = (17)

$h(n) \leq h^*(n)$ ⇒ underestimation (h(n): estimated, $h^*(n)$: Actual)

$h(n) \geq h^*(n)$ ⇒ overestimation

NOTE: We will always get the optimal ans in the case of underestimation.



g(A) = 200

g(B) = 200

Case I: Overestimation	Case II: Underestimation
h(A) = 80, h(B) = 70] > h*	h(A) = 30, h(B) = 20] < h*

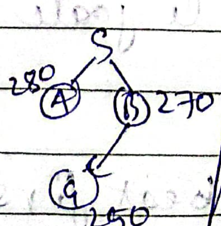
Case I:

f(A) = 200+80 = 280

f(B) = 200+70 = 270

f(G) = g(G) + h(G)

= 250 + 0 = 250



Case II:

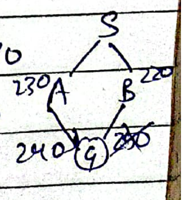
f(A) = 200+30 = 230

f(B) = 200+20 = 220

f(G) = g(G) + h(G)

= 250+0 = 250

f(G) = 240+0 = 240

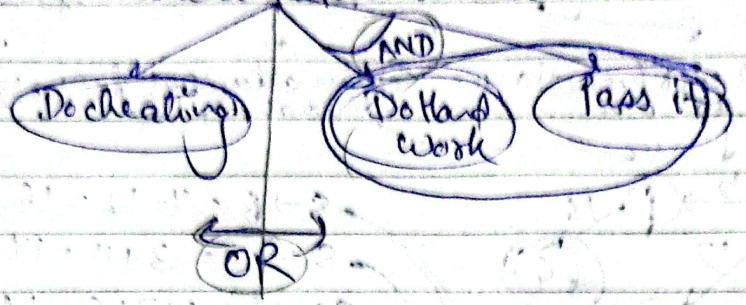


Sangam®

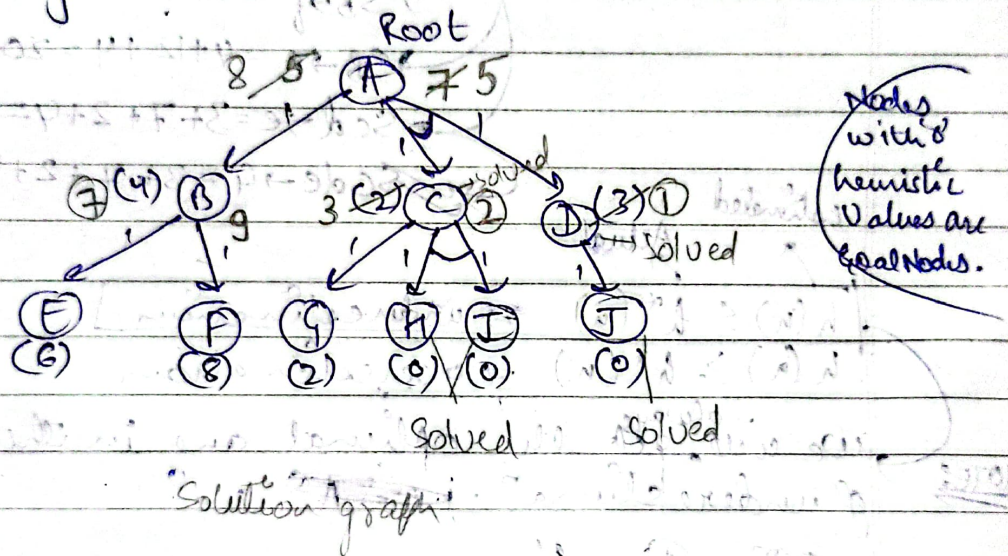
→ Admissible (always gives soln)

(4) AO* Algo (AND/OR) → Prob decomposition (breakdown into smaller pieces)

eg. Want to Pass in Exam



AO* doesnot explore all the soln paths once it gets a soln.



How AO* works ?

Aapko pehele ek level pe find krni h new heuristic value, fir dusre level pe find krni h new heuristic value, fir is value ke basis pe hum upar ke levels ki heuristic values update krte hn or root tak le jaate hn.

A* v/s AO*

Both work over best first search.

Both are informed search in which heuristic values are given.

AO* doesnot guarantee optimal soln.

A* always gives optimal soln when we work on underestimation.

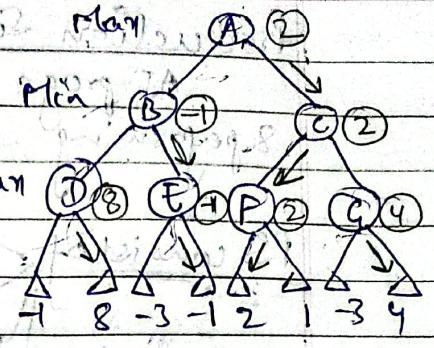
→ pts. for winning or loosing.
Max → user
Min → comp.

Date
Page

Game Playing Algo

(1) Minimax Algo.

→ Backtracking Algo.
→ Best move strategy used.
→ Max will try to maximize its utility (Best move).
→ Min will try to minimize utility (worst move).



Game tree.

branch factor

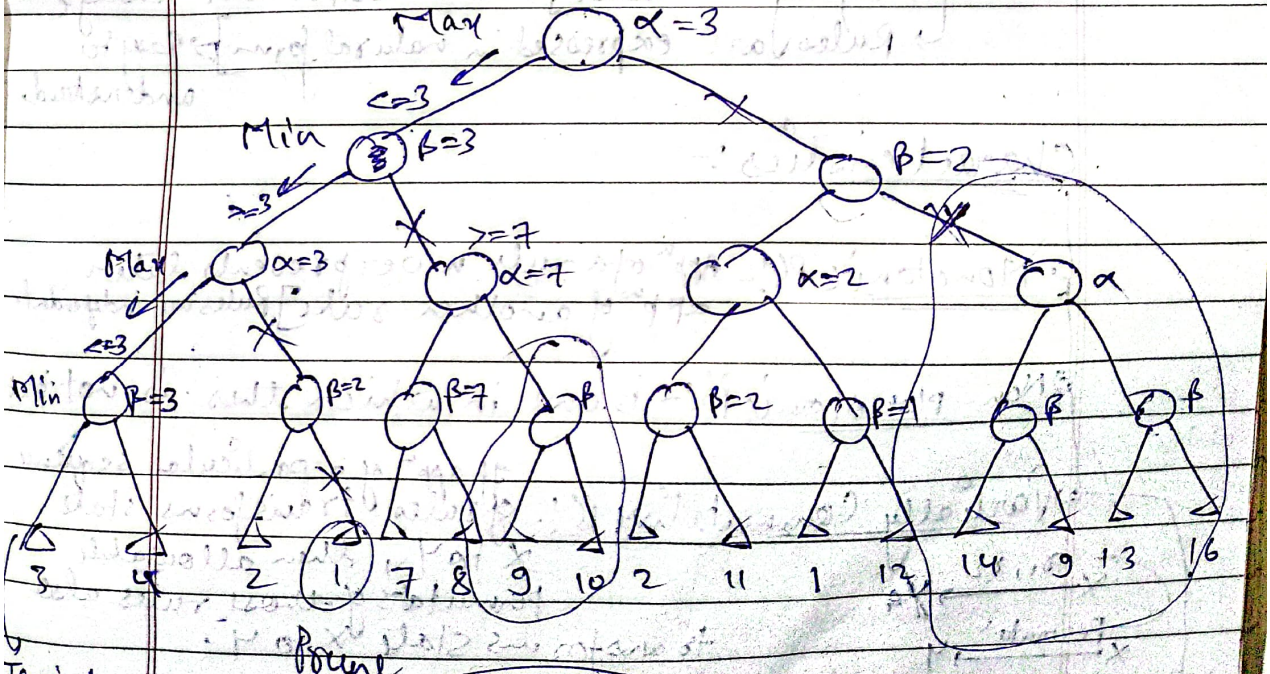
$TC = O(b^d)$ ← depth/ply

∴ we don't use Minimax in all games bcoz for games like Chess, its space tree might become very ~~long~~ large.

(2) Alpha-Beta Pruning (α-β)

→ for min node
→ for max node

→ Modification over minimax.
→ Cut off search by exploring less no. of nodes.
→ Once we find a best path, then we prune (cut-off) rest of the paths, we won't explore them.



$TC = O(b^{d/2})$

Sangam®