Date __ / __ / ____ Saathi

Artificial intelligence and machine learning (18CS71)

Module-1

What is artificial intelligence? problems, problem spaces and Search, Heuristic Search techniques

Artificial intelligence

According to the father of Artificial intelligence, John McCarthy -> it is the Science and engineering of making intelligent machines, especially intelligent Computer Programs.

(OR)

It is a branch of Computer Science by which we Can Create intelligent machines which Can behave like a human, think like humans and able to make decisions.

History of AI

1950 - "Are there imaginable digital Computers Can do well in imitation game"

1956 - John McCarthy Coined the term artificial intelligence (He is also known as father of artificial intelligence)

1959 - First artificial intelligence lab established in MIT

1975 - First powerful use-case of AI (Stanford university developed MYCIN in 1975 which helped physicians to address the bacterial infection)

1997 - First Computer to beat a chess Champion

Page No.

Scanned with CamScanner

Date __/__/__

2002 - Robots replacing humans
(Amazon made a big decision to replace human editor with basic AI systems)

2011 - Apple introduced Siri to the world

2016 - AI with next level capability
(Google alphaGo AI machine beats Go world Champion)

Benefits of AI

1. No human error

AI reduces errors by making precise decisions based on the past information they accumulate over time, implementing specific algorithms

2. 24*7 availability

Machines don't need a break, they can work 24*7. In november 2020, Google introduced Contact Center AI for Companies to enhance Customer experience. This is a great instance of an AI-based helpline system for Companies to address Customer queries and resolve issue Continually.

3. Unbiased decisions

AI thinks more practically and has a rational approach. A big advantage of AI is that it does not have biased views, thus ensuring an accurate decision-making process.

Saathi

Date __/__/__

4. Quick decision-making

AI machines can make decisions faster compared to humans, it makes decisions without any emotions and biased views. This ensures result-oriented decision making.
Eg: IBM's deep blue supercomputer makes decisions based on all the possibilities from the opponents side however human brain cannot do this.

5. Healthcare applications

AI techniques are highly utilized in the medical field. for instance, AI machines in healthcare helped doctors to evaluate the patient's health-related data and risk factor as well.

Types of Artificial intelligence

1. Artificial narrow intelligence (ANI)
2. Artificial General intelligence (AGI)
3. Artificial Super intelligence (ASI)

1. Artificial narrow intelligence (ANI)

Artificial narrow intelligence systems are designed and trained to complete one specific task and often termed as weak AI/narrow AI

Eg: The chatbots that answer questions based on user input, Voice assistants like Siri, alexa and Cortana, facial recognition systems.

2. Artificial General intelligence (AGI)

It's defined as AI which has a human-level of cognitive function across a wide variety of domains such as language processing, image processing, computational functioning and reasoning

Scanned with CamScanner

Saathi

Date ___/___/___

An AGI System would need to Comprise of thousands of Artificial narrow intelligence system working in tandem, Communicating with each other to mimic human reasoning

3) Artificial Super intelligence (ASI)
An artificial super intelligence (ASI) system would be able to surpass all human Capabilities

Strong and weak Artificial intelligence
Terms Coined by john Searle

Strong AI	Weak AI
• It is wider application with a more vast Scope	• It is a narrow application with a limited Scope
• This application has an incredible Human-level intelligence	• This application is good at Specific tasks
• It uses clustering and association to Process data	• It uses Supervised and Unsupervised learning to Process data
• Eg: Advanced robots	• Eg: Siri, Alexa

AI Problems
AI is used to Solve the below problems
1) Game Playing
2) Theorem Proving
3) Commonsense reasoning
4) Perception (vision and Speech)
5) Natural language Processing

Scanned with CamScanner

Saathi

Date __/__/____

Task domains of Artificial intelligence

1. Mundane tasks
2. Formal tasks
3. Expert tasks

Mundane (ordinary) tasks	Formal tasks	Expert tasks
Perception . Computer Vision . Speech, Voice	Mathematics . Geometry . Integral Calculus . Logic etc	Engineering . Fault finding . Design . Monitoring
Natural language Processing . Understanding . language translation . language Generation	Games . Go . Chess . Checkers	Scientific analysis
Common Sense reasoning	Verification	Financial analysis
Robotics	Theorem Proving	Medical diagnosis

The following questions are to be Considered before we Can Step forward

1. what are our underlying assumptions about intelligence?
2. what kinds of techniques will be useful for Solving AI Problems?
3. At what level of detail, if at all, we trying to model human intelligence?
4. How will we know when we have succeeded in building an intelligent Program?

Page No.

Scanned with CamScanner

Date __/__/__

Underlying assumption of AI

1) Based on the following statement : "Any Physical Symbol system (Computer) has the ability for Common "Quick-witted Actions"

2) The level of the model
   . Model the Program that Computer Could easily Solve
   . Model Human Performance

3) Criteria for Success

AI technique

Artificial intelligence research during the last three decades has Concluded that intelligence requires knowledge.
. In real world, the knowledge has some Unwelcomed Properties

1 It is huge
2 It is difficult to Characterize Correctly
3 It is Constantly Varying
4 It is Complicated
5 It differs from data by being Organized in a way that Corresponds to its application

An AI technique is a method that Exploits knowledge that is represented so that:

1 The knowledge Captures generalizations that share Properties, are grouped together, rather than being allowed Separate representation

Page No.

Scanned with CamScanner

Saathi

Date __/__/____

2 It Can be Understood by people who must provide it - Even though for many programs bulk of the data Comes automatically from readings

3 It should be easily modifiable to Correct errors

4 It Can be widely used even if it is incomplete (or) inaccurate

5 In many AI domains, how the people understand the same People must supply the knowledge to a Program

Example 1:- Tic-Tac-Toe

Tic-Tac-Toe is also known as noughts and Crosses (or) Xs and Os

- The game involves two players placing their respective Symbols in 3X3 grid
- The player who manages to place three of their symbols in horizontal/vertical/diagonal wins the game.
- If either player fails to do so the game ends in draw

O		X
	X	O
X	O	

= Win

O	O	X
X	X	O
O	X	X

= Draw

Here one of the player Chooses 'O' and the other Chooses 'X' to mark their respective Cells

Example:

Heuristic = It is a technique designed to solve a problem quickly

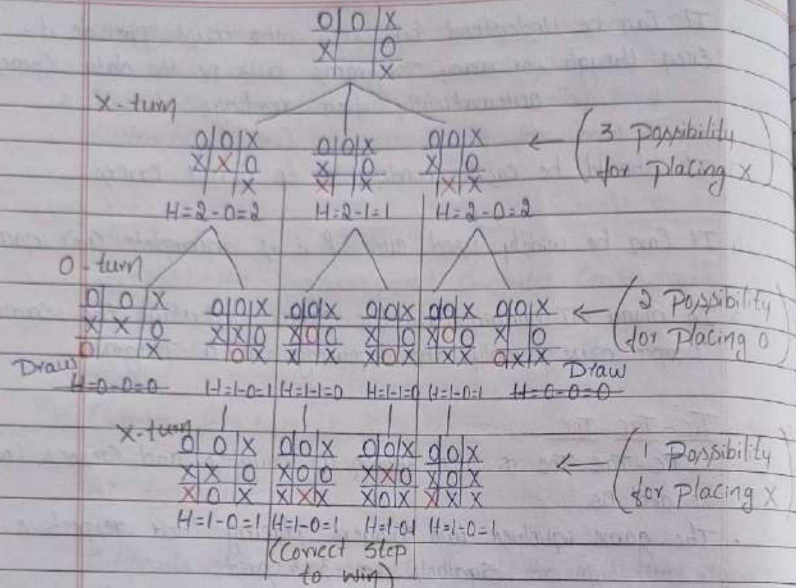
H = X's possibility - O's possibility

H=0 -> Draw

H=1 -> Win

Page No.

Scanned with CamScanner



Imp

Approach-1

Data structure

Board - Consider a board having nine elements vector

- Each element will contain -> 0 for blank
  1 for X
  2 for O
- first player who so ever is always plays X
- Computer may play as X or O player

Move table

Move table is a vector of 3^9 elements, each element of which is a nine element vector representing board position

- Total of 3^9 (19,683) element in move table

Scanned with CamScanner

Date __/__/__ Saathi

Algorithm

The algorithm makes moves by pursuing the following:

1. View the vector as a ternary number. Convert it to a decimal number
2. Use the decimal number as an index in movetable and access the vector
3. Set Board to this vector indicating how the board looks after the move

Disadvantages

1. It takes a lot of space to store the move table
2. Lot of work to specify all the entries in move table
3. Poor Extensibility

Approach - 2

Data structure

Board - A nine-element vector representing the board, but instead of using the numbers 0,1,2

The following Conventions are used

2 - indicates blank
3 - X
5 - O

Turn - An integer indicating, 1 - first move
9 - last move

Algorithm

The algorithm Consists of three actions:

- Make2 -> which returns 5 if the Centre square is blank, otherwise it returns any blank non-Corner square (i.e., 2, 4, 6, 8)
- Posswin(P) -> returns 0 if player Cannot win on the next move

Page No.

Scanned with CamScanner

Date __/__/__

- Otherwise returns the number of the square that gives a winning move.
  if $3*3*2 = 18$ gives a win for X
  else if $5*5*2 = 50$ gives a win for O

- Go(n) -> Makes a move to square n setting board[n] to 3 or 5

Turn=1	Go(1)
Turn=2	if Board[5] is blank, Go[5], else Go(1).
Turn=3	if Board[9] is blank, Go(9), else Go(3).
Turn=4	if Posswin(X) is not 0, then Go(Posswin(X)), else Go(Make2)
Turn=5	if Posswin(X) is not 0 then Go(Posswin(X)), else if posswin(O) is not 0 then Go(posswin(O)), or Go(7), or Go(3)
Turn=6	if Posswin(O) is not 0 then Go(posswin(O)), else if posswin(X) is not 0, then Go(posswin(X)), else Go(Make2)
Turn=7	if Posswin(X) is not 0 then Go(posswin(X)), else if Posswin(O) is not 0, then Go(posswin(O)), else go anywhere that is blank
Turn=8	if Posswin(O) is not 0 then Go(posswin(O)), else if Posswin(X) is not 0, then Go(Posswin(X)), else go anywhere that is blank
Turn=9	Same as Turn=7

Disadvantages

- Not Efficient in terms of time
- Several Conditions are Checked before each move
- It is memory Efficient
- Still Cannot Generalize to 3-D

Page No.

Scanned with CamScanner

Date __/__/____

Approach-2

Same as approach 2 except for one change in the representation of the board.

Magic Square:

8	3	4
1	5	9
6	7	2

Board is considered to be a magic square of size 3x3 where each row, column and diagonals add to 15

Strategy

- First check if computer can win
- if not then check if opponent can win
- if so, then block it and proceed further

Steps involved in the play are:

1. player 1 - blocks numbered as 8 (Human)
2. player 2 - blocks numbered as 5 (computer)
3. player 1 - blocks numbered as 3
4. player 2 - strategy used => compute sum of blocks played by player 1
   i.e., sum of two pair owned by player
   $S = 8+3 = 11$
   Compute - $D = 15 - S$
   $D = 15 - 11 = 4$ (so blocks numbered as 4)
5. player 1 - blocks numbered as 6
6. player 2 - strategy used => checks the possibility of winning
   $S = 5+4 = 9$
   $D = 15-9 = 6$ (cannot win)
   Now checks the player 1 possibility of winning
   $S = 8+6 = 14$
   $D = 15-14 = 1$ (so blocks numbered as 1)

Page No.

Scanned with CamScanner

Saathi

Date __/__/____

7 player 1 - Assume player 1 blocks numbered as 2
8 player 2 - checks its pair (5,1) and (4,1)
$D = 15 - 6 = 9$
Block 9 is free, so player 2 block numbered as 9 and win the game

Disadvantages

- Requires more time
- Has to search all possible move sequences before making each move

Approach-3

Data structure

Board - The structure of the data consists of Board which contains a nine element vector, a list of board positions that could result from the next move and a number representing an estimation of how the board position leads to an ultimate win for the player to move

Algorithm -

To decide the next move, look ahead at the board positions that result from each possible move. Decide which position is best, make the move that leads to that position, and assign the rating of the best move to the current position

To decide which of a set board positions is best, do the following for each of them:-

1 See if it is a win. if so, call it the best by giving it the highest possible rating
2 otherwise, consider all the moves the opponent could make next. see which of them is worst for us. Assume the opponent will make that move. whatever rating that move has, assign it to the node we are considering

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

3. The best node is then the one with the highest rating

Disadvantages

- This program require much more time than either of the others since it must search all possible move sequences before making each move
- Extendable

Example 2: Question Answering

Let us Consider Question answering system that accept input in English and Provide answers also in english
Difficulty -> is to decide whether the answer obtained is Correct (or) not

Let us Consider the following Situation

Text

Mary went shopping for a new Coat. She found a red one she really liked. when she got it home, she discovered that it went Perfectly with her favorite dress

Question

1. what did mary go shopping for?
2. what did mary find that she liked?
3. Did mary buy anything?

Approach-1

Data Structures

Question Patterns: A set of templates that match Common questions and Produce patterns used to match against inputs Templates and Patterns (text patterns) are used so that a template that matches a given question is associated with

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

the Corresponding pattern to find the answer in the input text

for Eg, the template "who did x y" generates "x y z" if a match occurs and z is the answer to the question.

Text, Question ⇒ The given text and question both stored as Strings

Algorithm

- Compare the template against the questions and store all successful matches to produce a set of text patterns
- Pass each of these text patterns through substitution process that generates alternative forms of verbs and produce an Expanded set of text patterns
- Apply each of these text patterns to text, and Collect the resulting answers

Examples

To Question-1 – we use the template "what did x y" which generates mary go shopping for z and after substitution we get mary goes shopping for z and mary went shopping for z (z = new Coat)

To Question-2 – Requires large number of templates and also a scheme to allow the insertion of 'find' before 'that she liked', the insertion of 'really' in the text, and substitution of 'she' for 'Mary' gives the answer 'a red one'

To Question-3 - Cannot be answered

Drawbacks

Does not match the Criteria we set for intelligence

Page No.

Scanned with CamScanner

Date __/__/__ Saathi

Approach-2

Data structure

- A structure called English consists of a dictionary, grammar and appropriate semantic interpretations.
- This data structure provides the knowledge to convert english text into storable internal form and also converts response back into English.
- The structured representation of the text is a processed form and defines the context of the input text by making explicit all references such as pronouns
- knowledge representation system used here is slot and filler system

Eg:- "She found a red one she really liked"

Event 2	Event 2
instance: finding	instance: liking
tense: past	tense: past
agent: Mary	Modifier: much
object: Thing 1	object: Thing 1

Thing 1
instance: Coat
Colour: red

Algorithm

- Convert the question to a structured form using English knowledge then use substring (like 'who' (or) 'what') of the structure, that should return as answer.
- Match this structured form against structured text
- The structured form is matched against the text and the requested segments of the question are returned

Examples

Question 1 - Generate answer "new Coat"
Question 2 - Generate answer "red Coat"
Question 3 - Cannot be answered

Page No.

Scanned with CamScanner

Date __/__/__

Drawback

English knowledge base is a Complex task

Eg:- 1 Mary walked up to the salesperson: she asked where the toy department was

2 Mary walked up to the salesperson: she asked her if she needed any help

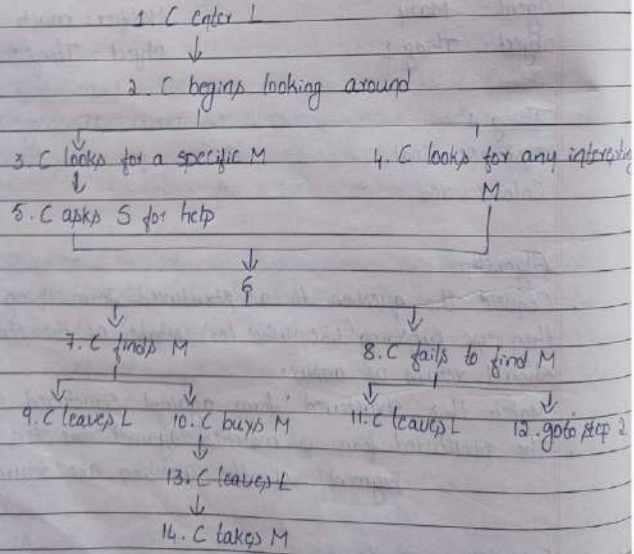
Approach 3

Data structure

world model - Contains knowledge about objects, actions and situations that are described in the input text.

This structure is used to create integrated text from input text

Diagrammatic representation of Shopping Script :-



Page No.

Saathi

Date __ / __ / ____

Shopping Script

roles: C(customer), S(salesperson)
props: M(Merchandise), D(dollars)
location: L(a store)

Algorithm:

- Convert the question to a structured form as before but use the world model to resolve ambiguities that may occur
- The structured form is matched against the text and the requested segments of the question are returned.

Examples
Question 1 - Generates answer "new Coat"
Question 2 - Generates answer "red Coat"
Question 3 - Can be answered ('M' is barrowed red Coat home. 'Mary buys a red Coat' Comes from Step 10 and the integrated text Generates that She bought a red Coat)

Most important terms in AI techniques.

Search - Provides a way of Solving problems for no more direct approach is available as well as framework into which any direct techniques that Can be embeded

Use of knowledge - Provides a way of Solving Complicated problems by manipulating the structures of the objects that are Concerned

Abstraction - finds a way of Separating important features and notifications from the Unimportant ones

Page No: ____

Scanned with CamScanner

Saathi

Date __/__/____

Level of model

To build programs that perform tasks the way people do can be divided into two classes

→ First class → They are problems that computer could easily solve, do not really fit our definition of an AI task

- These programs can be useful tools for psychologists who wants to test theories of human performance

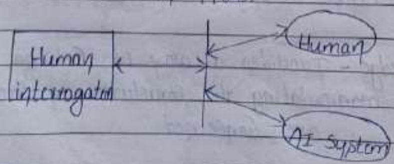
→ Second class → That attempt to model human performance, do things that clearly fall within our definition of AI task

The following reason on the human performance

1. To test psychological theories of human performance
2. To enable computer to understand human reasoning
3. To enable people to understand computer reasoning.
4. To exploit that knowledge we can glean from people

Criteria for success

Turing proposed operational test for intelligent behavior in 1950.



- To construct this test we need two people and the machine. one person plays the role of interrogator

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

- The interrogator will be in the Separate room from the Computer and the other person.
- The interrogator Can ask the question of either the person (or) the Computer by typing them

We need four Steps

- Define the problem precisely (Determine the initial and final Situation of the problem)
- Analyze the problem
- Isolate and represent the task knowledge that is necessary to Solve the problem
- Choose the best problem solving technique and apply it to the particular problem

Defining the problem as a State Space Search

Problem Solving = Searching for a goal State

- It is a Structured method for Solving an Unstructured problem. This approach Consists of number of States.
- The Starting of the problem is "initial State" of the problem. The last point in the problem is Called a "Goal State" (or) "Final State" of the problem
- State Space is a set of legal positions, Starting at the initial State, using the Set of rules to move from one State to another and attempting to end up in a goal State

Page No.

Scanned with CamScanner

Saathi

Date ___/___/____

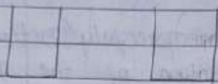
Imp

Water jug Problem

You are given two jugs, a 4-gallon one and a 3-gallon one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2-gallons of water into the 4-gallon jug?

Current state = (0,0)

Goal state = (2,0)



4-Gallons jug (Represent it with x) 3-Gallons jug (Represent it with y)

1) Initially both the jugs are empty => ($x=0, y=0$)

2) Fill jug y with 3 Gallons of water => ($x=0, y=3$)

3) Transfer the water from y to x => ($x=3, y=0$)

4) Fill jug y with 3 Gallons of water again => ($x=3, y=3$)

5) Transfer 1 Gallon of water from 3 Gallon jug(y) to 4 Gallon jug(x) => ($x=4, x=2$)

6) Empty the jug x => ($x=0, y=2$)

7) Transfer the 2 Gallons of water from 3 Gallon jug(y) to 4 Gallon jug(x) => ($x=2, y=0$) (Reached Goal)

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

Production rules for water jug Problem

Rule	State	Process
1	$(x, y / x < 4)$	$(4, y)$ {fill 4-Gallon jug}
2	$(x, y / y < 3)$	$(x, 3)$ {fill 3-Gallon jug}
3	$(x, y / x > 0)$	$(x-d, y)$ {pour Some water out of 4-Gallon jug}
4	$(x, y / y > 0)$	$(x, y-d)$ {pour Some water out of 3-Gallon jug}
5	$(x, y / x > 0)$	$(0, y)$ {Empty 4-gallon jug}
6	$(x, y / y > 0)$	$(x, 0)$ {Empty 3 gallon jug}
7	$(x, y / x + y \geq 4$ and $y > 0)$	$(4, y-(4-x))$ {pour water from 3 gallon jug to 4-gallon jug until 4-gallon jug is full}
8	$(x, y / x + y \geq 3$ and $x > 0)$	$(x-(3-y), 3)$ {pour water from 4-gallon to 3-Gallon jug until 3-Gallon jug is full}

Page No.

Scanned with CamScanner

Saathi

Date ___/___/___

9	$(x, y / x+y \leq 4$ and $y > 0)$	$(x+y, 0)$ {Pour all water from 3 gallon jug to 4-gallon jug}
10	$(x, y / x+y \leq 3$ and $x > 0)$	$(0, x+y)$ {Pour all water from 4 gallon jug to 3-gallon jug}
11	$(0, 2)$	{Pour 2 gallons of water from 3-gallon jug into 4-gallon jug}
12	$(2, y)$	Empty the 2 gallons in the 4-gallon jug on the ground

Solution to the water jug problem:

Gallons in the 4-Gallon jug(x)	Gallons in the 3-Gallon jug(y)	Rule applied
0	0	
0	3	2
3	0	9
3	3	2
4	2	7
0	2	5 or 12
2	0	9 or 11

Page No.

Scanned with CamScanner

Saathi

Date __ / __ / __

Production System

The entire procedure for getting a Solution for AI problem is known as "Production system"

Components

1. Set of rules => Each Consist of a left side the pattern that determines the applicability of rule and right side that describes the operation to be Performed if the rule applied.

2. Knowledge Base -> It Consists of whatever information is appropriate for a Particular task.

3. Control Strategy => It specifies the order in which the rules will be Compared to the database and the way of Solving the Conflicts that arise when several rules match at once
   1) first requirement of a good Control Strategy is that it Cause motion
   (A Control Strategy that does not Cause motion will never lead to a Solution)

   2) Second requirement of a good Control Strategy is that it Should be Systematic

4. A rule applier

Production System Characteristics

Production system is a good way to describe the operations that Can be Performed in a Search for Solution of the Problem

Page No.

Scanned with CamScanner

Saathi

Date ___/___/___

1) Simplicity

It helps to represent knowledge and reasoning in the simplest way possible to solve real-world Problems

2) Modularity

The modularity of a production rule helps in its incremental improvement (enhance the performance of the production system by adjusting the parameters of the rules)

3) Modifiability

It helps in the iterative improvement of the Production System.

Classes of a Production System

1) Monotonic Production System
2) Non-monotonic Production System
3) Partially Commutative Production System
4) Commutative System

1) Monotonic Production System

In the monotonic production system, the rules can be applied simultaneously as the use on one rule does not prevent the involvement of another rule that is selected at the same time.

2) Non-monotonic Production System

Increases the efficiency in solving problems

- The implementation of these systems does not require backtracking to correct the previous incorrect moves

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

3) Partially Commutative production system
It helps to Create a production system that Can give the results even by interchanging the States of rules.

. If set of rules transforms State A into State B, then multiple Combinations of those rules will be Capable to Convert State A into State B.

4) Commutative production system
is a Production system that is both monotonic and Partially Commutative

	Monotonic	Non-monotonic
Partially Commutative	Theorem Proving	Robot navigation
Not partially Commutative	Chemical Synthesis	Bridge

Search Algorithms

Uninformed/Blind
1) Breadth first Search
2) Depth first Search

Informed/Heuristic
1) Generate and test
2) Hill Climbing
3) Best first Search
4) A* Search
5) AO* Search
6) Problem reduction

Page No.

Scanned with CamScanner

Saathi

Date ___/___/___

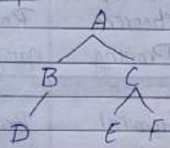
Uninformed / Blind Search algorithms

1) Breadth - first Search
is a Vertex based technique for finding a Shortest path in graph

- It uses a Queue data Structure which follows FIFO

- In BFS, one vertex is Selected at a time when it is visited and marked then its adjacent are visited and Stored in the queue

Example



output => A B C D E F

Algorithm:

1 Create a variable Called NODE-LIST and set it to the initial node

2 Until a goal State is found or NODE-LIST is Empty
  a) Remove the first element from NODE-LIST and call it E. if NODE-LIST was Empty, quit
  b) For each way that each rule Can match the state described in E do
    1) Apply the rule to generate new state.
    2) if the new state is a goal state, quit and return this state
    3) Otherwise, add the new state to the end of NODE-LIST

Page No.

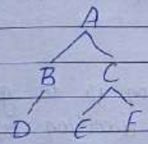
Saathi

Date __/__/____

2) Depth-first Search
is a edge based technique

It uses the stack data structure, performs two stages, first visited vertices are pushed into stack and second if there is no vertices then visited vertices are popped.

Example



Output => A B D C E F

Algorithm:

1. If the initial state is a goal state, quit and return success
2. Otherwise, do the following until success or failure is signaled:

a) Generate a successor, E, of the initial state. if there are no more successors, signal failure.
b) Call depth-first search with E as the initial state
c) If success is returned, signal success otherwise continue in this loop.

Page No.

Scanned with CamScanner

Saathi

Date ___/___/___

Heuristic Search

Heuristic:

It is a technique that improves the efficiency of Search process, possibly by sacrificing claims of Completeness

Heuristic function:

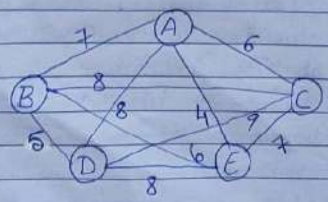
It is a function applied to a state in Search space to indicate a likelihood of success if that state is Selected

The purpose of heuristic function is to guide the Search in the most Profitable direction by suggesting which path to follow first when more than one is available (best promising way)

Travelling Salesman Problem

A salesman has a list of cities, each of which he must visit exactly once. There are direct roads between each pair of cities on the list. Find the route the Salesman should follow for the shortest possible round trip that both starts and finishes at any one of the cities

Consider an Example:-



Page No.

Scanned with CamScanner

Date __/__/__

Solution

Solve travelling Salesman problem using branch and bound technique

	A	B	C	D	E
A	-	7	6	8	4
B	7	-	8	5	6
C	6	8	-	9	7
D	8	5	9	-	8
E	4	6	7	8	-

Step 1:- Row minimization

	A	B	C	D	E
A	-	3	2	4	0
B	2	-	3	0	1
C	0	2	-	3	1
D	3	0	4	-	3
E	0	2	3	4	-

How it's done
(Eg:- 7-4=3 (Row A)
7-5=2 (Row B))

Step 2:- Column minimization

	A	B	C	D	E
A	-	3	0	4	0
B	2	-	1	0	1
C	0	2	-	3	1
D	3	0	2	-	3
E	0	2	1	4	-

(for column minimization Consider the result you got in previous step)

Step 3:- Calculate the penality of zero

Page No.

Saathi

Date __/__/____

	A	B	C	D	E
A	-	3	0^1	4	0^1
B	2	-	1	0^4	1
C	0^1	2	-	3	1
D	3	0^4	2	-	3
E	0^1	2	1	4	-

(add minimum value from the row and column to zero)

Consider the max penality for "zero" ⇒ B → D

Step 4:- Exclude the path which gave max penality

	A	B	C	E
A	-	3	0	0
C	0	2	-	1
D	3	-	2	3
E	0	2	1	-

Step 5:- Row and Column minimization

	A	B	C	E
A	-	1	0	0
C	0	0	-	1
D	1	-	0	1
E	0	0	1	-

(R & C minimization done coz to find zero penality we require each row & column to have zero)

Step 6:- Calculate the penality of zero

	A	B	C	E
A	-	1	0^0	0^1
C	0^0	0^0	-	1
D	1	-	0^1	1
E	0^0	0^0	1	-

Consider the max penality for "zero" ⇒ A → E

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

step 7:

	A	B	C
C	0^1	0^0	-
D	1	-	0^2
E	-	0^0	1

Consider the max penality for "zero" => D->C

Step 8:

	A	B
C	0	0
E	-	0

E->B
C->A

=> B->D, A->E, D->C, E->B, C->A

optimal path is:

$\boxed{A \to E \to B \to D \to C \to A}$

B->D = 5
A->E = 4
D->C = 9
E->B = 6
C->A = 6

Cost :-

$\boxed{30}$

Difficulty in TSP

- If there are N Cities, then number of different paths among them is 1.2.... (N-1) or (N-1)!

Scanned with CamScanner

Saathi

Date __/__/____

AI Problem Characteristics

1 Is the problem decomposable into easier sub problems?

2 Can solution steps be ignored or at least undone if unwise?

3 Is the problem universe predictable?

4 Is a good solution obvious?

5 Is the desired solution a state or a path to a state?

6 Is a large amount of knowledge required absolutely (or) only to constraint the search?

7 Can a computer give the problem solution or interaction of human is required?

1 Is the problem decomposable?

Characteristics to be tested:
- Problem is broken into smaller problems.
- Solve each small problem using few specific rules

Example:

- In integration problem of Calculus - at each step check to see if small problem is immediately solvable if not check to see if the problem can be further decomposed and call itself recursively
- Solving problems in the blocks world.

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

2 Can Solution steps be ignored (or) undone?

Classes of problems:

1. Ignorable - Solution steps can be ignored. eg., theorem proving.
2. Recoverable - Solution steps can be undone. eg., 8-puzzle
3. Irrecoverable - Solution steps can be undone. eg., Chess

Characteristics:

- Ignorable problems use simple structure that never backtracks
- Recoverable problems use backtracking to recover mistakes
- Irrecoverable problems need planning since more effort is required

3 Is the problem's universe predictable?

- Certain outcome - Eg:- 8-puzzle
  For solving certain outcome problems - open loop approach (without feedback) will work fine
- Uncertain outcome - Eg:- Bridge
  For solving uncertain outcome problem - planning can best generate a sequence of operators that has a good probability of leading to a solution

Page No.

Scanned with CamScanner

Saathi

Date __/__/____

4 Is a good Solution absolute or relative?

Classes

Any path problem and best path problems

Characteristics

- Any path problems are easier to solve
- for any path problem heuristics can be used to suggest good paths to explore
- for best path problem no heuristic that could possibly miss the best solution can be used and a more exhaustive search is performed.

5 Is the solution a state or a path?

- Consideration 1: Inference from the statement
  "The bank president ate a dish of pasta salad with the fork"

Inference: pasta salad was a dish, pasta salad was eaten, pasta salad consists of pasta.

Solution: Statement

- Consideration 2: path problem
  "Water jug problem": Goal (2,0)

path taken to reach goal state from initial state

Solution: path taken from (0,0) - (2,0)

6 Is a large amount of knowledge required absolutely (or) only to constraint the search?

Page No.

Scanned with CamScanner

Saathi

Date __/__/____

- Some problems require very little knowledge -
  Eg:- Chess (Rules to determine legal moves and simple Control mechanism to implement an appropriate search)

- Some problems require huge amount of knowledge -
  Eg:- Newspaper Story understanding (this requires a variety of facts and Complexity of Constructs)

7 Can a Computer give the problem solution or interaction of human is required?

Solutions -

Solitary - Computer given problem description produces answer with no immediate Communication and demand for Explanation of the reasoning

Conversational - there is immediate Communication between a person and Computer for assistance (or) additional information to Computer

8-puzzle

The 8-puzzle is a square tray in which are placed, eight square tiles. The remaining ninth square is uncovered. Each tile has a number on it. A tile that is adjacent to the blank space can be slid into that space. A game consists of a starting position and a specified goal position. The goal is to transform the starting position into the goal position by sliding the tiles around.

2	8	3
1	6	4
7		5

Start

1	2	3
8		4
7	6	5

Goal

Page No.

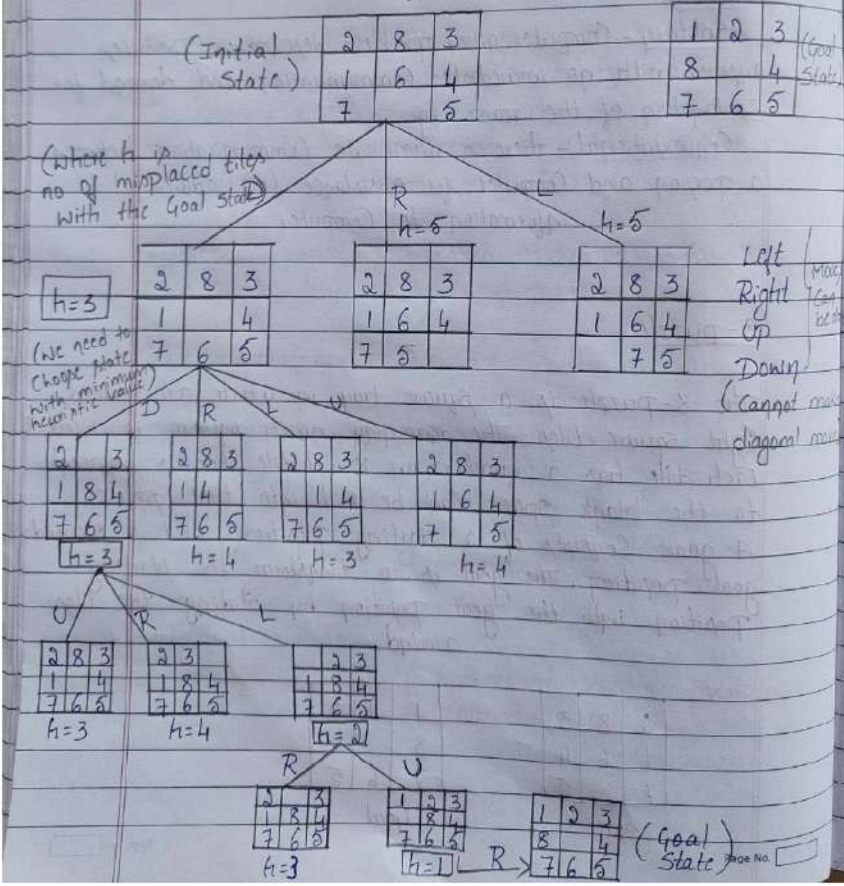
Scanned with CamScanner

Date ___/___/______

8- puzzle problem Can be solved by Using heuristic function at the same time it Can be solved without heuristic

- Without heuristic (uninformed Search) - we are Exploring each and Every step until we reach the goal State

- With heuristic (Informed search) - Need not to Explore all the step only search where it Can find the goal state.



Page No.

Saathi

Date __/__/__

Additional Problems

1. The Missionaries and Cannibals Problem

Three missionaries and three Cannibals find themselves on one side of a river. They have agreed that they would all like to get to the other side. But the missionaries are not sure what else the Cannibals have agreed to. So the missionaries want to manage the trip across the river in such a way that the number of missionaries on either side of the river is never less than the number of Cannibals who are on the same side. The only boat available holds only two people at a time. How Can Everyone get across the river without the missionaries risking being eaten?

Solution

To solve this problem we will make the following assumptions:

1. Number of Cannibals should be lesser than the missionaries on either side.
2. Only one boat is available to travel
3. Only one (or) maximum of two people Can go in the boat at a time
4. There is no restriction on the number of trips that Can be made to reach of the goal.

Page No.

Scanned with CamScanner

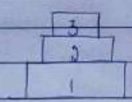
Rules

1	(0,M) one missionary sailing the boat from Bank 1 to Bank 2
2	(M,0) one missionary sailing the boat from Bank 2 to Bank 1
3	(M,M) Two missionaries sailing from Bank 1 to Bank 2
4	(M,M) Two missionaries sailing from Bank 2 to Bank 1
5	(M,C) one missionary and one Cannibal sailing from Bank 1 to Bank 2
6	(C,M) one missionary and one Cannibal sailing from Bank 2 to Bank 1
7	(C,C) Two Cannibals sailing from Bank 1 to Bank 2
8	(C,C) Two Cannibals sailing from Bank 2 to Bank 1
9	(0,C) one Cannibal sailing from Bank 1 to Bank 2
10	(C,0) one Cannibal sailing from Bank 2 to Bank 1

Scanned with CamScanner

River bank-1	River bank-2	Boat position	Rule Applied
MMMCCC	0	Bank-1	Start state
MMCC	MC	Bank-2	5
MMMCC	C	Bank-1	2
MMM	CCC	Bank-2	7
MMMC	CC	Bank-1	10
MC	CCMM	Bank-2	3
MCCM	CM	Bank-1	6
CC	CMMM	Bank-2	3
CCC	MMM	Bank-1	10
C	MMMCC	Bank-2	7
CC	MMMC	Bank-1	10
0	MMMCCC	Bank-2	7

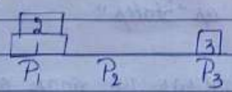
2 The Tower of Hanoi



initial state

Task is to move all the disks to the other post.
- only one disk can be moved at a time
- large disk can't be placed on smaller

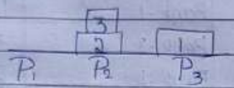
Solution: Step 1:-



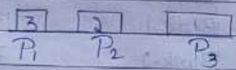
Step 2:

Step 3:

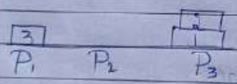
Step 4:



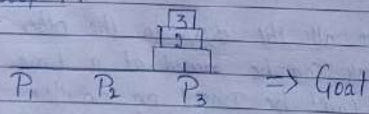
Step 5:



Step 6:



Step 7:



Situations encountered while solving the problem are described as "States"

Set of all possible configurations of rings on the pegs is called "Problem space"

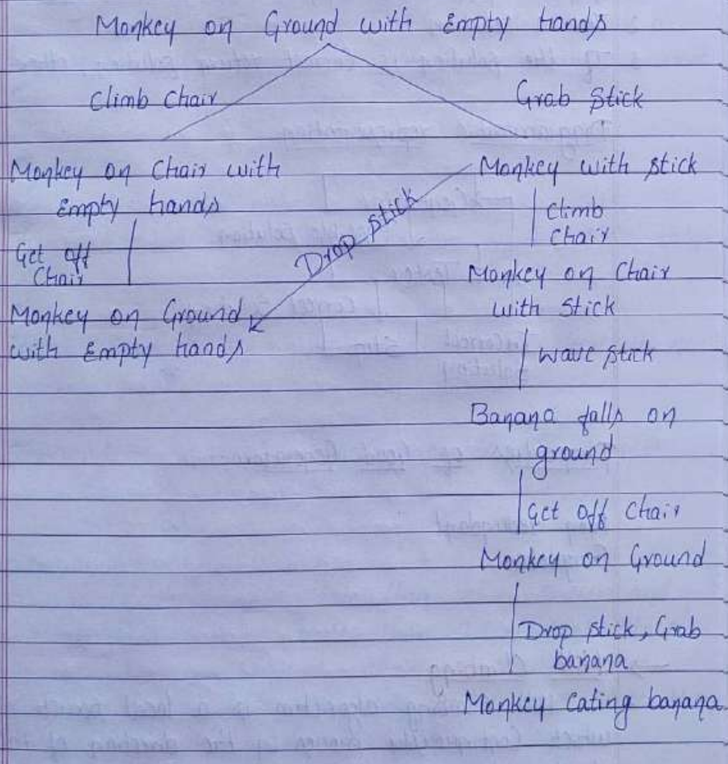
Scanned with CamScanner

Date __/__/____

3 Monkey and Bananas problem

Conditions
- In the room there are Chair and a stick
- If the monkey stands on the Chair and waves the stick, he can knock a banana down and eat.

Actions



Page No.

Date ___/___/______

Heuristic Search technique

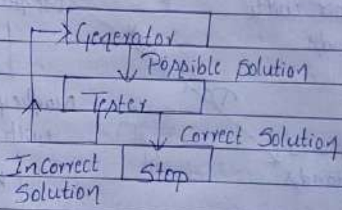
→ **Generate - AND - TEST**

In this technique all the solution are generated and tested for best solution.

Algorithm

1. Generate possible solution
2. Test a solution
3. If the solution is correct return solution, otherwise quit.

Diagrammatic representation



Properties of Good Generators:-

- Non redundant
- Complete

→ **Hill Climbing**

- Hill Climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation to find the peak of the mountain (or) best solution to the problem

Page No. ____

Saathi

Date __/__/____

- Terminates when it reaches a peak value where no neighbor has a higher value

Types of Hill climbing algorithm :

- Simple Hill Climbing
- Steepest-Ascent hill-Climbing

1. Simple Hill Climbing

It only evaluates the neighbor node state at a time and selects the first one which optimizes current cost and set it as a current state.

Algorithm

Step 1 : Evaluate the initial state, if it is goal state then return success and stop.

Step 2 : Loop until a solution is found (or) there is no new operator left to apply

Step 3 : Select and apply an operator to the current state

Step 4 : Check new state.

a. if it is goal state, then return success and quit

b. Else if it is better than the current state then assign new state as current state.

c. Else if not better than the current state, then return to step 2

Page No.

Scanned with CamScanner

Saathi

Date ___/___/___

2. Steepest - Ascent hill Climbing
is a Variation of simple hill climbing algorithm. This algorithm examines all the neighboring nodes of the current state and selects one neighbor node which is closest to the goal state

Algorithm

Step 1: Evaluate the initial state, if it is goal state then return success and stop, else make current state as initial state.

Step 2: Loop until a solution is found or the current state does not change

a) Let SUCC be a state such that any successor of the current state will be better than it

b) For each operator that applies to the current state:

a. Apply the new operator and generate a new state.

b. Evaluate the new state.

c. If it is goal state, then return it and quit, else compare it to the SUCC.

d. if it is better than SUCC, then set new state as SUCC

e. if the SUCC is better than the current state, then set current state to SUCC

Page No.

Scanned with CamScanner

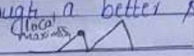
Saathi

Date __/__/__

Problems in hill climbing algorithm:

1 Local maximum

At a local maximum all neighbouring states have values which are worse than the current state. Since hill-climbing uses a greedy approach, it will not move to the worse state and terminate itself.
This process will end even though a better solution may exist.



To overcome local maximum:

Utilize the backtracking technique
(Maintain a list of visited states. if the search reaches an undesirable state, it can backtrack to the previous configuration and explore a new path)

2 plateau

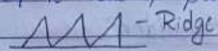
All neighbours have the same value. Hence, it is not possible to select the best direction.

To overcome plateaus:

Make a big jump (Randomly select a state far away from the current state. Chances are that we will land at a non-plateau region)

3 Ridge

Any point on a ridge can look like a peak because the movement in all possible directions is downward. Hence, the algorithm stops when it reaches such a state.



To overcome ridge:

You could use two (or) more rules before testing. it implies moving in several directions at once.

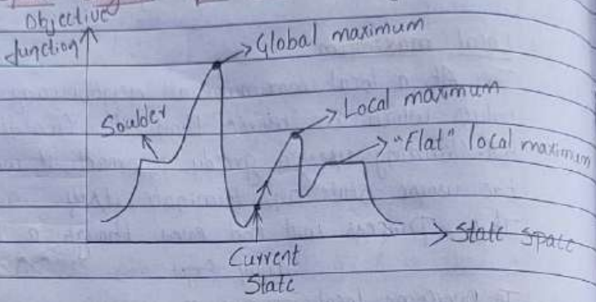
Page No.

Scanned with CamScanner

Saathi

Date ___ / ___ / ___

State-Space diagram for Hill Climbing



1 Global maximum

is the best possible state of state space diagram. Here the value of the objective function is highest.

2 Local maximum

is a state which is better than its neighbouring state, however, it is not the best possible state as there exists a state where objective function value is higher.

3 Current state

where an agent is currently present.

4 Flat local maximum

where all the neighbor states of current states have the same value

5 Shoulder

It is a plateau region which has an uphill edge

Scanned with CamScanner

Date __/__/____

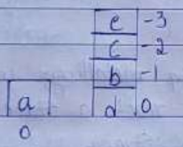
Block world problem

$Hf = -10$ (Initial state), $Hf = 10$ (Goal state)

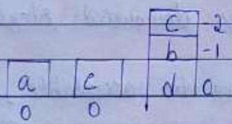
Hf	Initial State	Goal State	Hf
-4	a	c	+4
-3	c	d	+3
-2	c	c	+2
-1	b	b	+1
0	d	a	0

(Hf = Heuristic function)

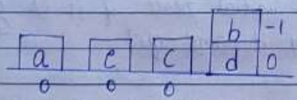
Step 1:



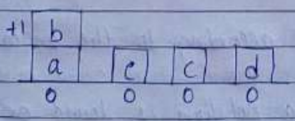
Step 2:



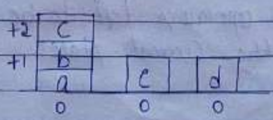
Step 3:



Step 4:



Step 5:



Page No.

Scanned with CamScanner

Saathi

Date __ / __ / __

Step 6:

+3	d		
+2	c		
+1	b		
	a		e
	0		0

Step 7:

+4	e
+3	d
+2	c
+1	b
0	a

H = +10 (Reached Goal)

Simulated Annealing:

Annealing is a process in metallurgy where metals slowly cooled to make them reach a state of low Energy where they are very strong.

- Simulated annealing allows downward steps
- Formula $\Rightarrow p' = e^{-\Delta E/T}$
- Move to worst state may be acceptable, It has annealing schedule

Algorithm:

Step 1: Evaluate the initial state. if it is goal state, then return it and quit

Step 2: Initialize BEST-SO-FAR to the current state

Step 3: Initialize T according to the annealing schedule

Step 4: Loop until a solution is found or until there are no new operators left to be applied in the current state

Page No.

Scanned with CamScanner

Saathi

Date ___/___/___

a) Select an operator that has not yet been applied to the current state and apply it to produce a new state

b) Evaluate the new state. Compute
$\Delta E = $ (value of current) − (value of new state)

→ BEST-FIRST SEARCH (OR GRAPHS)

Best-first search is a technique that decides which node is to be visited next, by checking which node is the most promising one

To implement BFS, it requires two lists
1) OPEN
2) CLOSED

OPEN list - where nodes not yet been examined.

CLOSED list - nodes that have already been examined.

Algorithm

1 Start with OPEN containing just the initial state

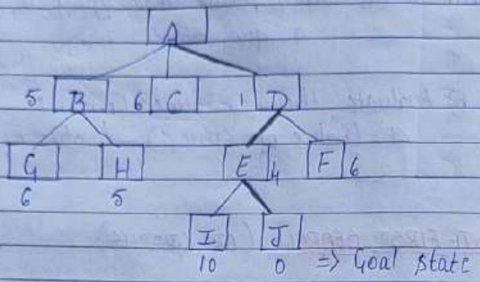
2 Until a goal is found (or) there are no nodes left on OPEN do:
a. Pick them best node on OPEN
b. Generate its successors
c. For each successor do:
i) if it has not been generated before, evaluate it, add it to OPEN, and record its parent
ii) generated before, change the parent if this new path is better than the previous one.

Page No.

Scanned with CamScanner

Date __/__/__

Example



OPEN list	CLOSED list
	A
~~D~~ B C	A D
~~E~~ B C F	A D E
~~J~~ B C F I	A D E J => Goal state

→ A* Algorithm

Is a searching algorithm that is used to find the shortest path between an initial and a final state

A* algorithm extends the path that minimizes the following function:-

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

- $g(n)$ = Cost of the path from start node to node 'n'
- $h(n)$ = heuristic function
- $f(n)$ = estimated total path to reach goal

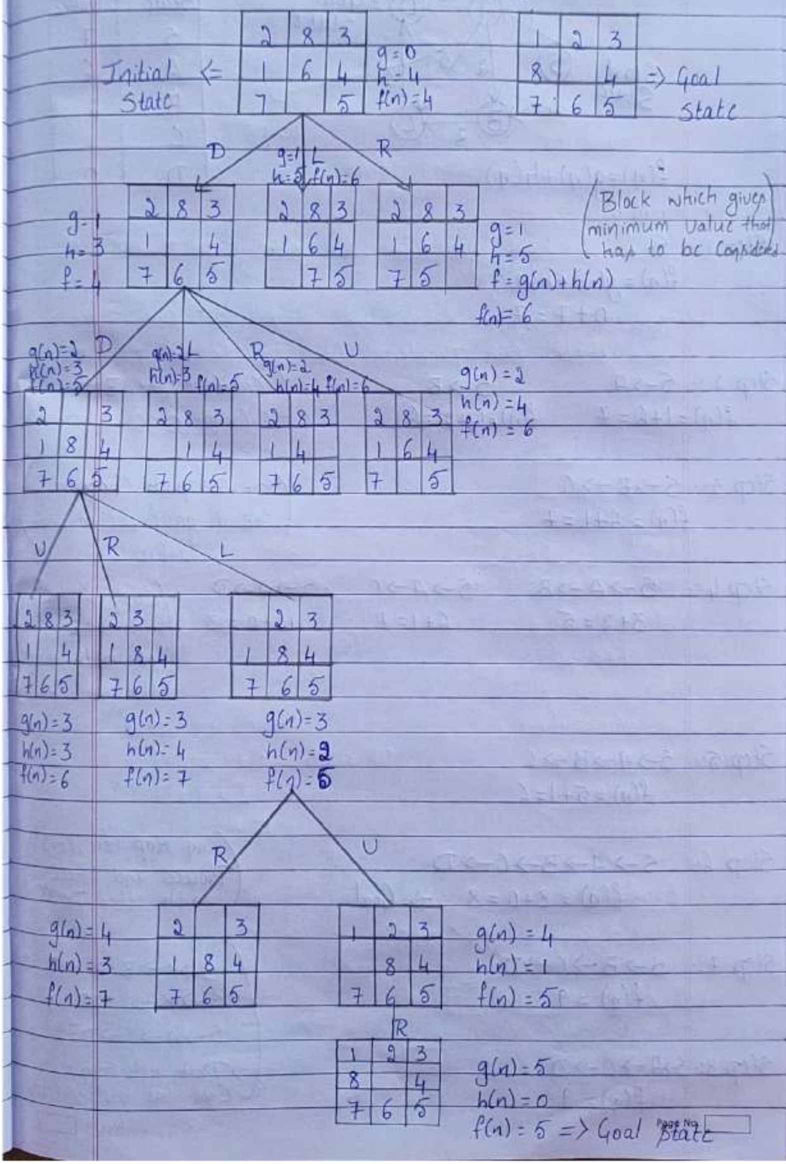
Page No.

Scanned with CamScanner

Saathi

Date __/__/__

Example 1:- Solve 8-puzzle using A*



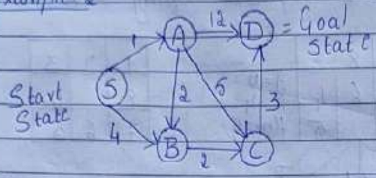
Page No.

Scanned with CamScanner

Saathi

Date __ /__ /____

Example 2



Heuristic value table

State	Heuristic Value
S	7
A	6
B	2
C	1
D	0

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

Step 1 :- S
$f(n) = g(n) + h(n)$
$0 + 7 = 7$

Step 2 :- S→A $f(n) = 1+6 = 7$ S→B $f(n) = 4+2 = 6$ (Calculate for Both Child nodes of S)

Step 3 :- S→B→C
$f(n) = 6+1 = 7$ (S→B, path Choosen Coz it gave me minimum value)

Step 4 :- S→A→B $3+2 = 5$ S→A→C $6+1 = 7$ S→A→D $13+0 = 13$ (Can Choose S→A (or) S→B→C Coz both gave the same min value)

Step 5 :- S→A→B→C
$f(n) = 5+1 = 6$

Step 6 :- S→A→B→C→D
$f(n) = 8+0 = 8$ → Goal (Can stop here, (or) Proceed and Check with other paths)

Step 7 :- S→B→C→D
$f(n) = 9$

Step 8 :- S→A→C→D
$f(n) = 9$

(S→A→B→C→D = 8
S→B→C→D = 9
S→A→C→D = 9
⇒ Path with minimum Cost has to be Considered)

Page No.

Scanned with CamScanner

Date __/__/__ Saathi

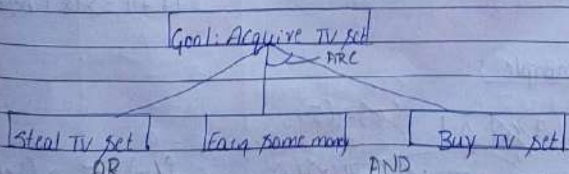
Algorithm

1. Add start node to list
2. For all the neighboring nodes, find the least cost F node
3. Switch to the closed list
   a) if the node is not reachable, ignore it. else
      1) if the node is not on the open list, move it to the open list and calculate f, g, h
      2) if the node is on the open list, check if the path it offers is less than the current path and change to it if it does so.
4. Stop working when
   1) you find the goal
   2) you cannot find the goal going through all possible points

→ Problem reduction - (AND-OR Graphs)

The process of decomposing a complex problem into a set of sub-problems and then integrating all these sub-problems to get a solution of the given complex problem.

Example for AND-OR Graph

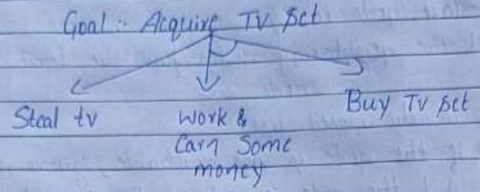


- OR node represents a choice between possible decompositions
- AND node represents a given decomposition

Page No.

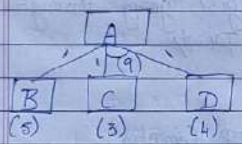
Date __/__/__

→ AO* Algorithm - (AND-OR Graph) (Problem reduction)



- AO* is best algorithm for solving AND-OR Graph
- Problem can be divided into a set of sub-problems and those sub-problems can be solved separetly.

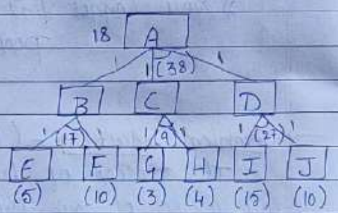
Example:-



OR = $f(n) = g(n) + h(n)$
$1 + 5 = 6$ ✓

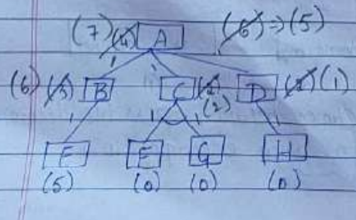
AND = $f(n) = g(n) + h(n)$
$3+1+4+1$
$= 9$

Example 2:-



Here the best path is:-
OR = $f(n) = 5+1+10+1$
$6+11 = 17$
$17+1 \Rightarrow 18$ (best path)

Example 3:-



=> $f(n) = 5$ => (best path)

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

Algorithm:

1. Let G Consists only to the node representing the initial state
2. Traverse the graph following the Current path, accumulating node that has not yet been Expanded (or) Solved
3. Select any of these nodes and Explore it.
   if it has no Successors then Call this Value FUTILITY
   else Calculate f(n) for each of the successors
4. if f(n) = 0, then mark the node as solved
5. Change the value of f(n) for the newly Created node to reflect its successors by backpropagation
6. whenever possible use the most Promising routes, if a node is marked as SOLVED then mark the parent node as SOLVED
7. if the starting node is SOLVED or Value is greater than FUTILITY then stop else repeat from step-2

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

→ Constraint Satisfaction (or) Crypt-Arithmatic Problem

A problem is solved when each variable has a value that satisfies all the constraints on the variable. A problem described this way is called a Constraint Satisfaction Problem (or) CSP

- A Constraint satisfaction problem consists of three components, X, D and C:

X is a set of variables, $\{X_1 \ldots X_n\}$
D is a set of domains, $\{D_1 \ldots D_n\}$ one for each variable
C is a set of constraints that specify allowable combinations of values

Problem:-

```
  SEND
+ MORE
 MONEY
```

(we can assign 0 to 9 values to any variable) & every letter should have unique value

Solution

```
  SEND
+ MORE
 MONEY
```

0 = O
1 = M
2 = Y
3 = -
4 = -
5 = E
6 = N
7 = D
8 = R
9 = S

Step 1:

```
  SEND
+ 1ORE
 1ONEY
```

(S+M will generate a carry, which can be only one)

Step 2 =

```
  9END
+ 1ORE
 1ONEY
```

(only when S=9, it generates a carry, also value for 'O')

Page No.

Scanned with CamScanner

Date __/__/____ Saathi

Step 3 = 9 E N D
+1 0 R E
1 0 N E Y

(so O = zero)

Step 4 = 9 E N D (carry)
+1 0 R E
1 0 N E Y

(E+O should be E only, but we find that it is N, hence this proves that there is a carry in this step)

Expression = E+1 = N
2) N+R(+1) = E+10
$\boxed{R=8}$

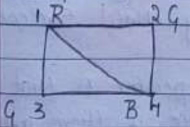
Step 5:- 9 E N D
+1 0 8 E
1 0 N E Y

(D+E = Y, has to generate a carry. D+E has to add up more than 11 as Y can't be 0 (or) 1 => possibility are 7+5 or 7+6 if we take D=7, E=5 hence Y=2)

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

hence this is the possible solution

Example 2: NODE Coloring:



V = {1, 2, 3, 4}
D = {Red, Green, Blue}
C = {adjacent nodes should not have same color}

Sol^n

	1	2	3	4	
Initial	R,G,B	R,G,B	R,G,B	R,G,B	
1=R	R	G,B	G,B	G,B	
2=G	R	G	G,B	B	
3=B	R	G	B		← Conflict
3=G	R	G	G	B	

Page No.

Scanned with CamScanner

Saathi

Date __/__/__

→ Means-Ends analysis

It is a Combination of forward and backward direction

- Means-ends analysis, introduced in 1961 by Allen Newell and Herbert A. Simon

- It works recursively

Methods to solve a problems in MEA

1 First, find the difference between initial state and final state

2 Select the operators which Can be applied for each difference.

3 Apply the operator for each difference

Algorithm

1 Compute CURRENT to GOAL if there are no differences between them the return

2 Otherwise, Select the most important difference and reduce it by doing the following Until Success or failure is Signed

a) Select an as yet Untried operator O that is applicable to the Current difference. if there are no such operators, then Signal failure

b) Attempt to apply O to CURRENT.

Page No.

c) if
(FIRST-PART ← MEA (CURRENT, O-START))
and
(LAST-PART ← MEMO-RESULT, GOAL))
are successful, then signal success and return the result of Concatenating FIRST-PART, O, and LAST-PART