Unit 3
Search algorithms in AI

ZEAL INSTITUTES
www.zealeducation.com

★ Algorithms for uninformed & informed search :-

1) What is an informed search in AI?

→ Algorithms have information on the goal state which help in more efficient searching.
Informed search in AI is a type of search algorithm that use additional information to guide the search process, allowing for more efficient problem solving compared to uninformed search algo.

This information can be in the form of heuristics, estimates of cost or other relevant data to prioritize which state to expand & explore.
eg:- A* search, Best-first search & Greedy search & Graph search.

2) What is Uninformed search in AI?

→ Algorithm have no additional information on the goal node other than the one provided in the problem def^n

The plans to reach the goal state from the start state differ only by the order & length of action. Uninformed search in AI refer to a type of search algorithm that does not used addition information to guide the search process

Instead, these algo explore the search space in a systematic but blind manner without considering the cost of reaching the goal.
eg:
BFS - Breadth first search
DFS - Depth First search
Depth - limited search.

	Parameter	Informed search	Uninformed search
1.	Known as	Heuristic search	Blind search
2.	Using knowledge	It is uses knowledge for searching process	It does not use
3.	Performance	It find a sol^n more quickly	slow
4.	completion	It may or may not be complete	It is always complete
5.	Cost factor	low	high
6.	Time	It consume less time because of quick searching	moderate time.
7.	Direction	There is a direction given about the sol^n	No suggestion is given regarding the sol^n in it
8.	Implementation	It is less lengthy while implemented	more lengthy
9.	Efficiency	more	less
10.	computational requirement	are lessened	higher
11.	size of search problems	Having wide scope in terms of handling large search problem	solving a massive search task is challenging.

	Informed	Uninformed
Ex of Algo.	• Greedy search • A* search • AO* search • Hill climbing algo.	• Depth First search DFS • Breadth Firstsearch BFS • Branch & Bound.

★ What is heuristics in AI?

→ Heuristics is a method of problem-solving where the goal is to come up with a workable solution in a feasible amount of time.

A problem specific approach that employs a practical method that often provides sufficient accuracy for the immediate goals.

(brute force method)

8 puzzle problem with Heuristic.

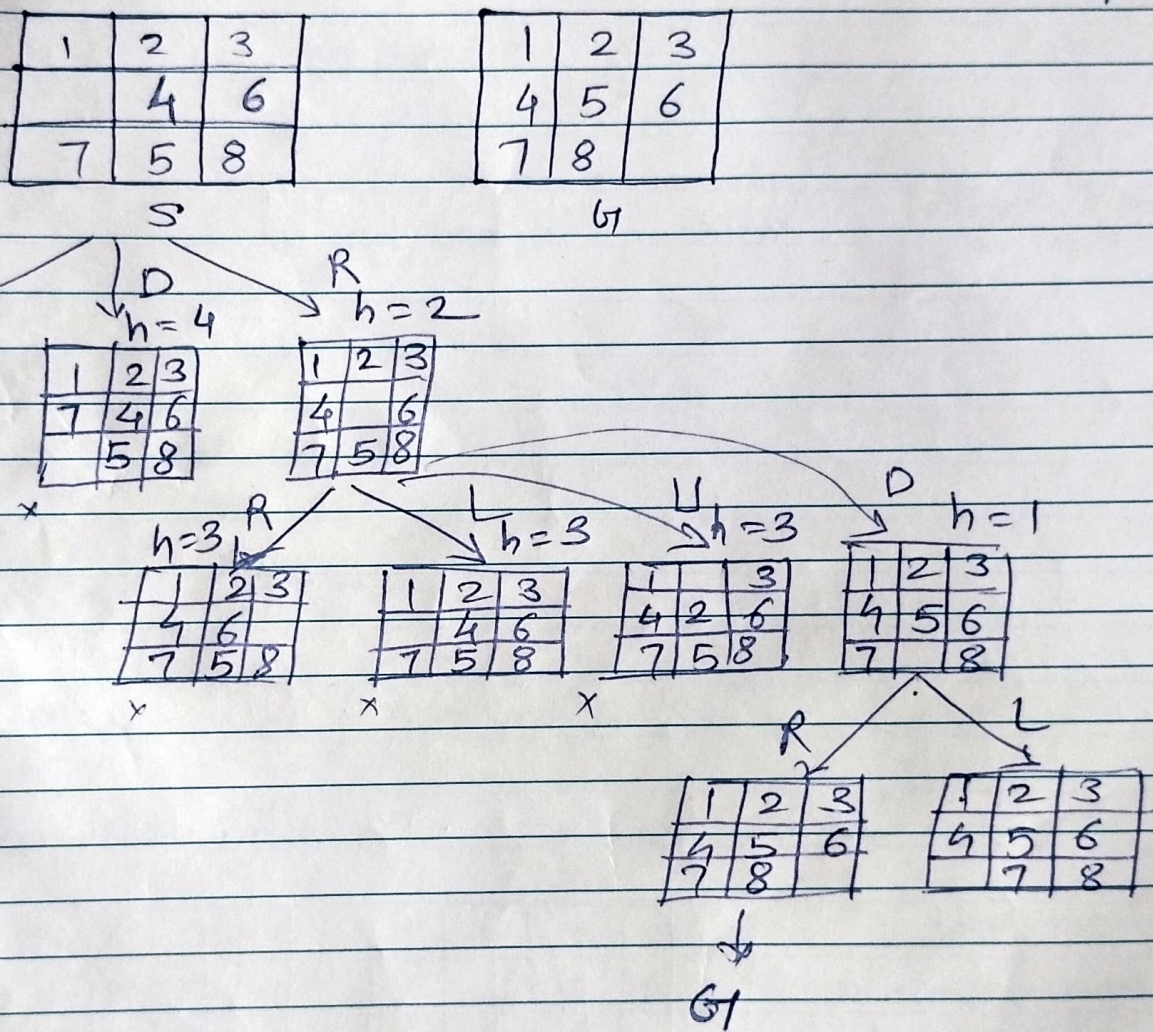
↑ informed search technique.

↑ Best for search

Calculate. First step. No's of misplaced type. tiles.

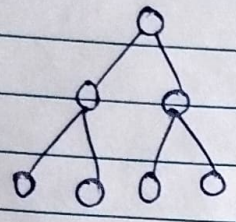
Heuristic value. h = 3

d = 3 depth



NP → P.

* Heuristic in AI (Rule of thumb) [What, Why, How]
↑ assumption
It is technique designed to solve a problem quickly



8 puzzle problem. $O(b^d)$ Uniformed case.

15 puzzle. $\binom{13}{10}$

24 puzzle $\binom{24}{10}$

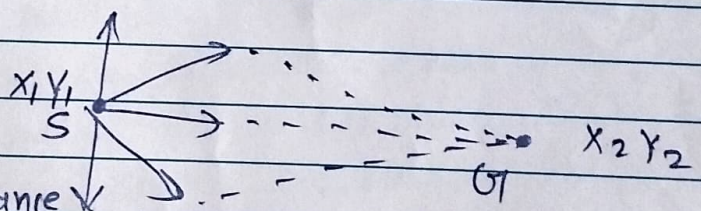
$\binom{20}{3}$ searchspace possible.

Time complexity exponential
Non polynomial problem.

Definitely optimum soln

Time reduce.

How to calculate:-



Straight line distance
Euclidian distance. $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

Manhatten distance. $0+1+1+2+0+2+2+0$

1	3	2
6	5	4
	8	7

1	2	3
4	5	6
7	8	

less time & cost
minimum path.

Calculate heuristic value.

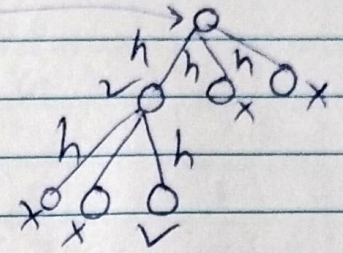
It always give guarantee good soln.

But not give optimal soln.

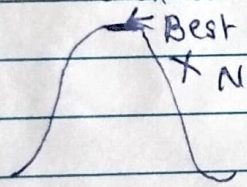
When we want solution quickely.

★ Simple Hill climbing Algorithm :-
(local search algo , greedy approach , No backtrack)
local domain knowledge ↑ Best move tracking.

1. Evaluate the initial state.
✓ New operators find new node.
Three branch based on heuristic value. , beam width is 1
based on that cost.



✓ Best of in 3 state. BW is 1
Explore only this state, other state are not save in the memory
This concept are Hill climbing algorithm.
✓ Best state → Explore of this state next level.
✗ ✓ If the state is better then the current state then it is new current state.
✓ keep going → better state ⇒ If better state available then No backtrack.



✓ Problem :-

Problem :-

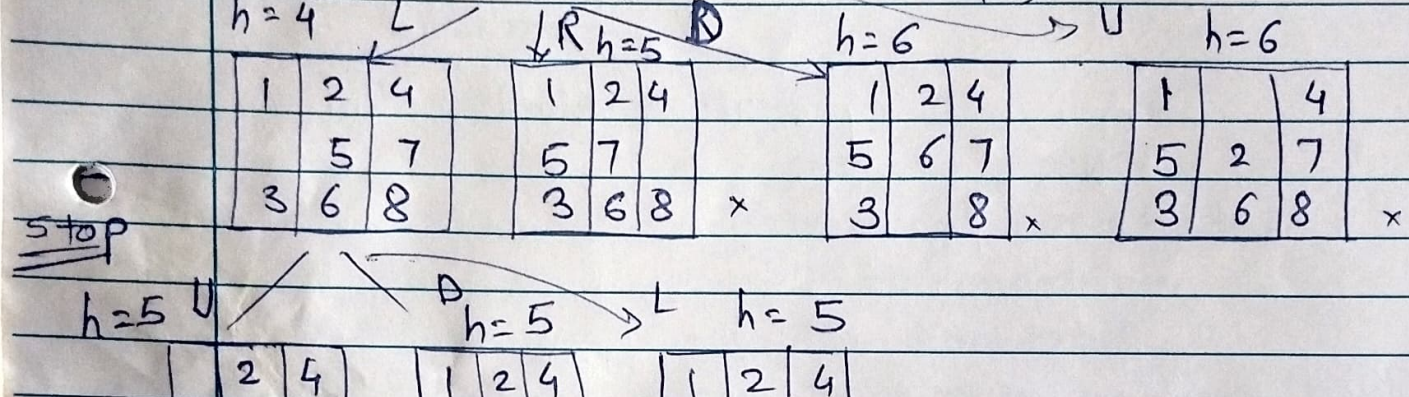
1	2	4
5		7
3	6	8

(S)

1	4	7
2	5	8
3	6	

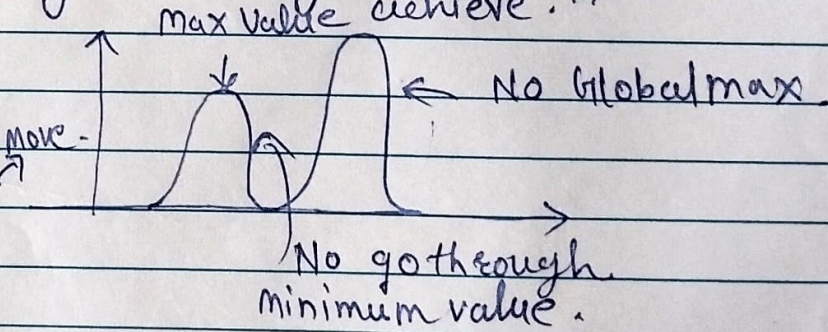
(G)

heuristic value.



Problem in Hill~~ing~~ climbing.

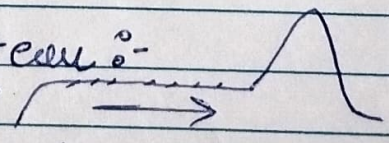
1) Local Maximum



good heuristic value
local based

less heuristic — less cost value decide.

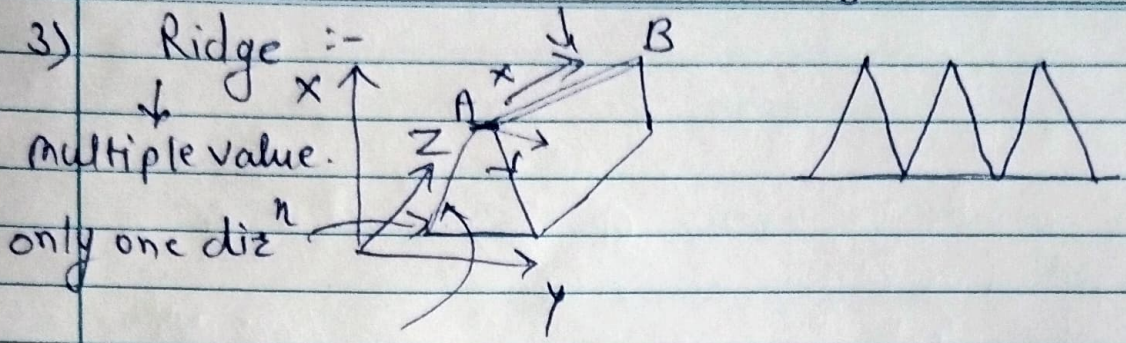
2) Flat maximum / Plateau :-



Same heuristic Value → No answer given
→ then stop.

No direction change.

3) Ridge :-



Best move.

Hill climbing Beta is 1
one best move.
Remaining forget other.
Stop one point
No backtrack.

✓ These all are local maximum problem in Hill climbing
✓ This is Hill climbing.
✓ How to evalute these problem. I will study.

Exponential time complexity } Uninformed.
Exponential space complexity }

ZEAL INSTITUTES
www.zealeducation.com

★ Best first search :- (Informed, Heuristic)

Algorithm :-

let 'OPEN' be priority queue containing initial state.
↓
based on heuristic value.

Loop.

If open is empty return failure.

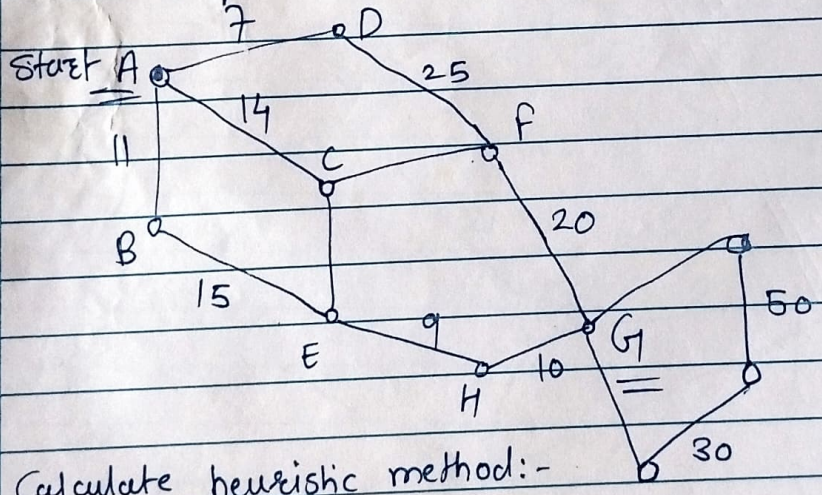
Node ← Remove - First OPEN

If node is goal

then return the path from initial to node.

else generate all successors of node & put the newly generated node into open according to their f value. ← (Heuristic value)

End loop.



Straight line dist

A → G1 = 40
B → G1 = 32
C → G1 = 25
D → G1 = 35
E → G1 = 19
F → G1 = 17
H → G1 = 10
G1 → G1 = 0

Calculate heuristic method :-

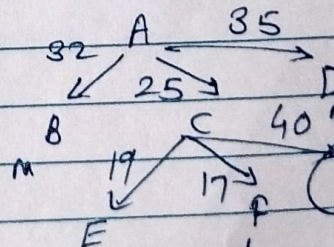
Use straight line distance.

Use Euclidian distance.

$$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

queue ~~A~~, ~~C~~, B, D

F, E



Final path A → C → F → G1

good time complexity

Good soln but not give optimum soln.

Work fast Greedy method.

G1(0) guarantee.

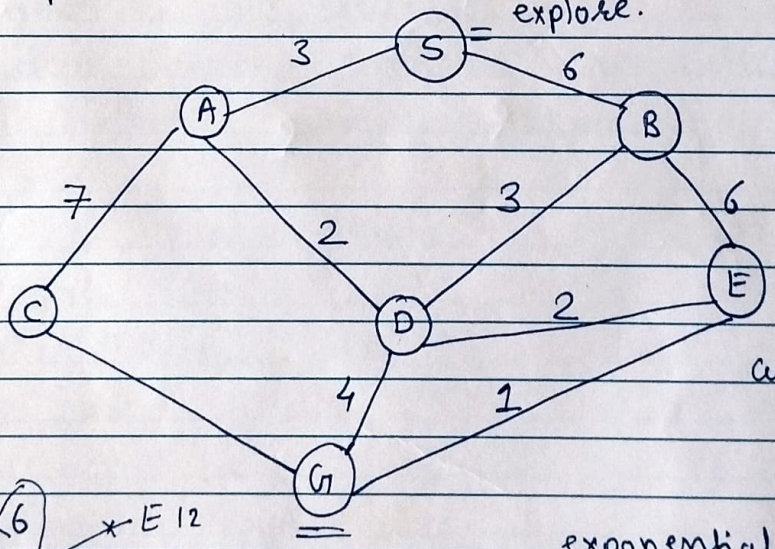
minimum heuristic value.

★ Branch & Bound algorithm :- Widely used find optimal sol^n

↑ Finding subsol^n ↑ Problem solving.

↑ stop ↑ ignore.

- ✓ Branch is the process of generating subproblem.
- ✓ Bounding refers to ignoring partial sol^n that cannot be better than current best sol^n.
- ✓ It is a search procedure to find the optimal sol^n.
- ✓ It eliminates those parts of a search space which does not contain better solution
- ✓ In this method we basically extend the cheapest partial path.



current sol^n = ~~9~~ = 8

exponential time is more.

$S \xrightarrow{3} A \xrightarrow{2} D \xrightarrow{2} E \xrightarrow{1} G = 8$.

Parent Node

Leaf Node.

multiple option

Discard

Bound.

Best sol^n

optimal sol^n.

★ Metaheuristics :-

- General purpose optimization technique.
- Unlike heuristic, it is not limited to a particular problem.
- Iteratively improves a candidate sol^n till an acceptable sol^n is obtained.
- Useful for solving complex problems where traditional method are not suitable.
- Can find global optimal sol^n.

- Mimics natural phenomena : Nature inspired algorithm.
- stochastic optimization, Use random number.
- combination of two extremes.
  (1) random search (extreme exploration)
  (2) hill climbing (extreme exploitation)

★ Groups of metaheuristics :-

1) Evolutionary algo.
2) swarm intelligence algo
3) physics based algo
4) Human based algo
5) other nature inspired algo

★ swarm intelligence algo. :-

(a) Ant colony optimization (ACO)
(b) Frog leaping algo (FLA)
(c) Artificial bee colony (ABC)

★ Metaheuristics :-

- ✓ General purpose optimization technique.
- ✓ Unlike heuristic, it is not limited to a particular problem.
- ✓ Iteratively improves a candidate sol^n till an acceptable sol^n is obtained.
- ✓ Useful for solving complex problems where traditional method are not suitable.
- ✓ Can find global optimal sol^n.

- ✓ Mimics natural phenomena : Nature inspired algorithm.
- ✓ stochastic optimization, Use random number.
- ✓ combination of two extremes.
  - (1) random search (extreme exploration)
  - (2) hill climbing (extreme exploitation)

★ Groups of metaheuristics :-

1) Evolutionary algo
2) swarm intelligence algo
3) physics based algo
4) Human based algo
5) other nature inspired algo

★ swarm intelligence algo :-

- (a) Ant colony optimization (ACO)
- (b) Frog leaping algo (FLA)
- (c) Artificial bee colony (ABC)

Metaheuristic :-

- In computer science & mathematical optimization, a metaheuristic is higher level procedure or.
- heuristic designed to find generate, tune or select a heuristic that may provide a sufficiently good solution to an optimization problem / machine learning problem.
- compared to optimization algo & iterative method, metaheuristics do not guarantee that a globally optimal sol^n can be found on some class of problem.
- Many metaheuristics implement some form of stochastic optimization. so that the sol^n found is dependent on the set of random variables generated.
- In combinatorial optimization by searching over a large set of feasible sol^n
- metaheuristic can often find good sol^n with less computational effort than optimization algo., iterative method or simple heuristic
- Metaheuristics is experimental in nature, describing empirical result based on computer experiment with algo.
- But some formal theoritical results are also available. often on convergence & the possibility of finding the global optimum.

* Physics based heuristics :-
(a) Simulated annealing.
(b) Gravitational local search (GLSA)
(c) charged system search

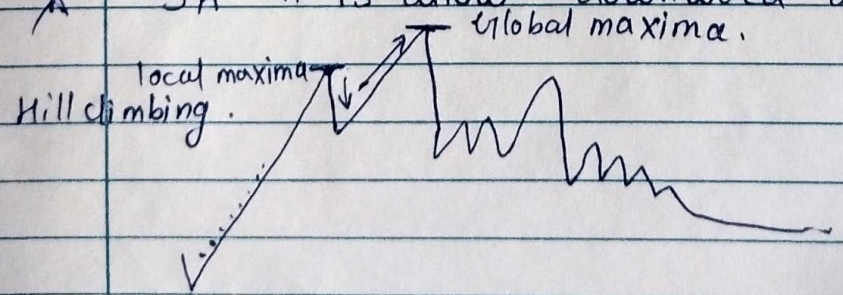
* Human based heuristics :-
(a) Tabu search
(b) Harmony search
(c) Group search optimizer (GSO)

{Variation Hill climbing method.}

Simulated annealing :- (SA) {checks all neighbors}

- ✓ It is a probabilistic technique for approximating the global optimum of a given f^n.
- ✓ Specifically, it is a metaheuristic to approximate global optimization in a large space for an optimization problem.
- ✓ eg :- ① traveling salesman problem.
  ② Boolean satisfiability problem.
  ③ Protein structure prediction
  ④ Job-shop scheduling.
- ✓ SA may be preferable to exact algo such as gradient descent or branch & bound.

- ✓ The name of the algo comes from annealing in metallurgy, a technique involving heating & controlled cooling of a material to alter it's physical property.
- ✓ Both are attributes of the material that depends on their thermodymic free energy.
- ✓ Heating & cooling the material affect both the temp. & the thermodynamic free energy or Gibbs energy.
- ✓ The term objective f^n → minimum value achieve. is used in place of heuristic f^n.
- ★ Annealing is a process in metallurgy where metals are slowly cooled to make them reach a state of low energy where they are very strong.
- ★ SA it is allow downward direction.



The probability that the metal will jump to a high energy level is given by

$$P = e^{-\Delta E / KT}$$

Where K = Boltzmann's constant

ΔE = change in value of the objective fn.

When K = 1

$$P' = e^{-\Delta E / T}.$$

Algorithm :-

★

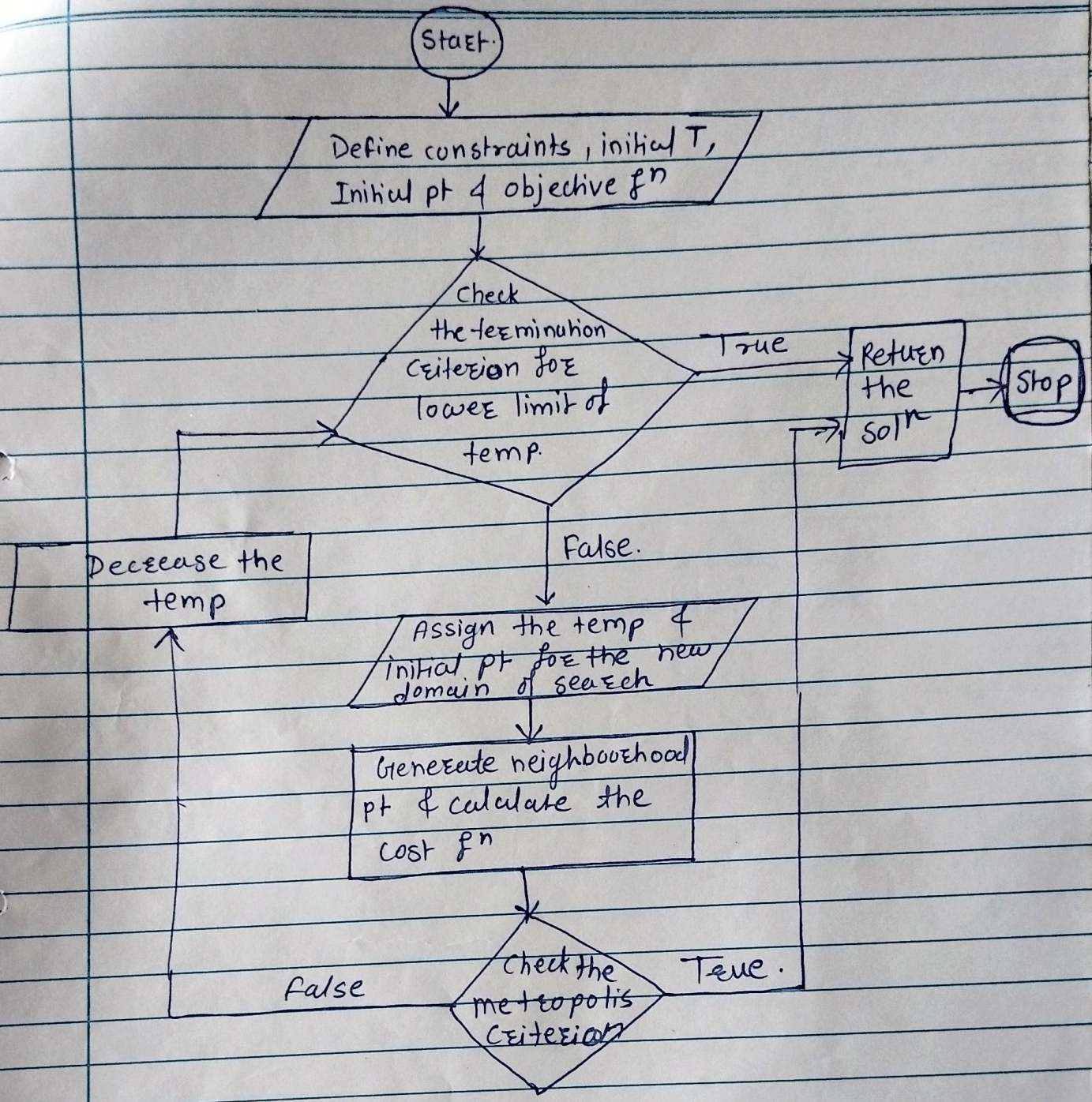
Advantages.

- ✓ It is easy to code for complex power.
- ✓ It's gives good soln
- ✓ statistically guruntee finding optimal soln

Disadvantages.

- ✓ slow process
- ✓ can't tail whether an optimal soln is found.
  - ↳ some other method is also require.

	Simulated annealing	Hill climbing.
1)	Annealing schedule maintained	No.
2)	Moves to worst states may be accept.	No.
3)	Best state found so ~~far~~ is also maintained	No.


Start
Define constraints, initial T, Initial pt & objective fn
Check the termination criterion for lower limit of temp.
True
Return the soln
Stop
False.
Decrease the temp
Assign the temp & initial pt for the new domain of search
Generate neighbourhood pt & calculate the cost fn
Check the metropolis criterion
True.
False

★ Tabu Search :-

- ✓ Tabu search is based on introducing flexible memory structure in conjuction with strategic restrication of aspiration levels as a means for exploiting search spaces
- ✓ It is a meta heuristic that guides a local heuristic search procedure to explore the soln space beyond local optimum by use of a Tabu list.
- ✓ Used to solve combinatorial optimization problems.
- ✓ A dynamic neighborhood search method.
- ✓ It uses a flexible memory to restrict the next solution choice to some subset of neighborhood of current solution.

★ 3 main strategies

- ✓ Forbidding strategy → controls what enters the tabu list
- ✓ Freeing strategy → control what exits from the tabu list.
- ✓ Short term strategy → Manages interplay between the forbidding strategy & freeing strategy to select trial solutions.

* Basic ingredients of Tabu search:-

- Primary way to exploit memory in tabu search is classify a subset of the moves in a neighborhood as forbidden.
- A neighborhood is constructed
- To identify adjacent solution that can be reached from current solution

- The classification depends on search history & on frequency that certain move or solution component called attributes, have participated in generating past soln.
- A tabu list records forbidden moves, which are referred to as tabu moves.
- Aspiration criterion provides exception to Tabu restriction. When a tabu move has a sufficiently attractive evaluation where it would result in a solution better than any visited so far, then its tabu classification may be overridden.

* Basic Tabu search algorithm :-

Step1: Choose an initial solution i in s.
Set $i^* = i$ & $k=0$

Step2: Set $k=k+1$ & generate a subset V^* of solution in $N(i, k)$ such that either one of the Tabu conditions is violated or at least one of the aspiration conditions holds.

Step3: Choose a best j in V^* & set $i=j$

Step4: If $f(i) < f(i^*)$ then set $i^* = i$

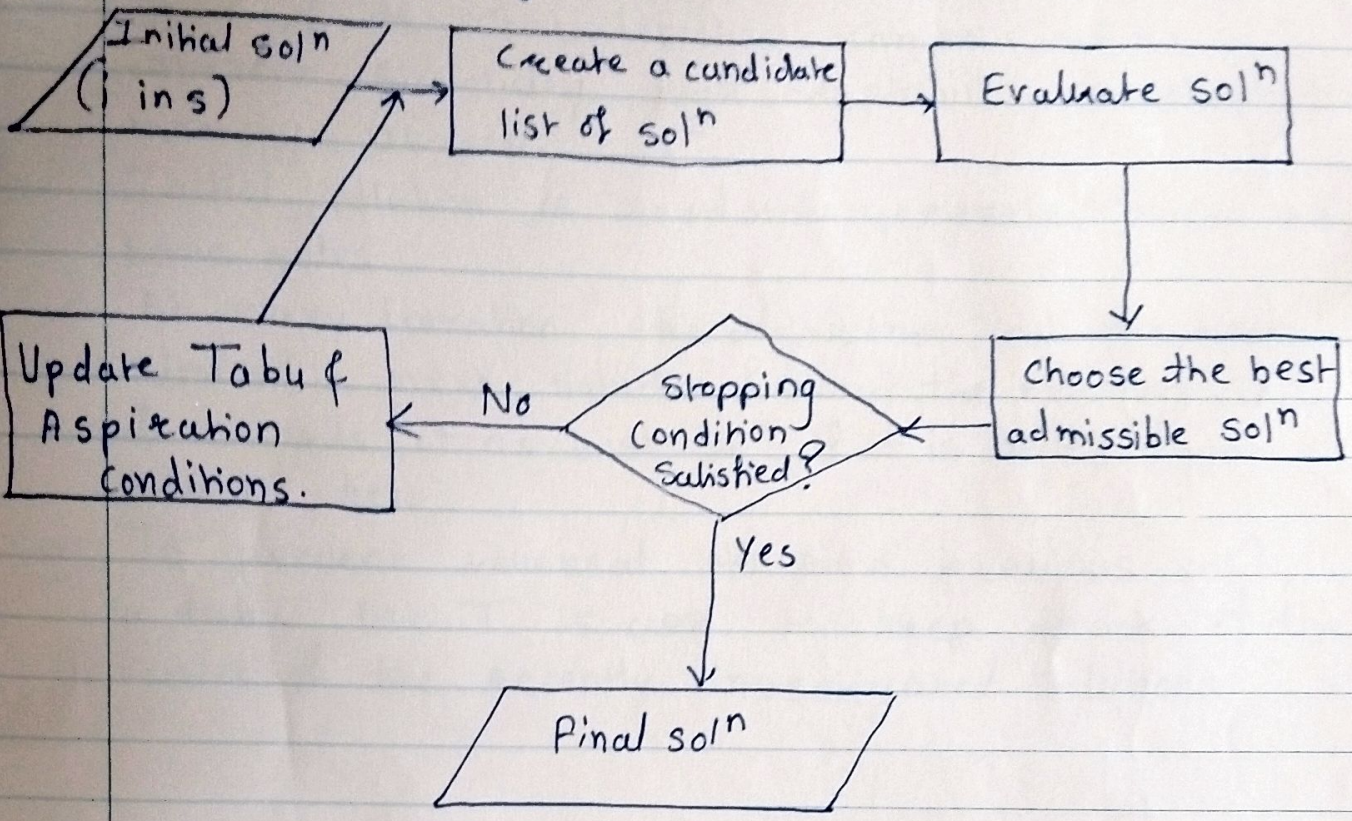
Step5: Update Tabu & aspiration conditions.

Step6. If a stopping condition is met then stop
Else go to step2.

* Stopping criterion :-

1. $N(i, k+1) = 0$ (No feasible sol^n in the neighborhood of sol^n i)
2. k is largeer than the maximum no's of iteration allowed.
3. The no's of iteration since the last improvement of i^* is largeer than a specified no's.
4. There is evidence that an optimum sol^n has been obtained.

* Tabu search algorithm flow chart :-



* Advantages :-
- ✓ Accepts non-improving soln so as to escape from a local optimum
- ✓ Can be applied to both discrete & continuous soln spaces
- ✓ Can be used for complex problem on scheduling, quadratic assignment & vehicle routing. It obtains competitive solution vis-a-vis other similar approaches.

* Disadvantages:-
- ✓ Too many parameters to be determined
- ✓ Number of iterations could be very large.
- ✓ Global optimum may not be found, depends on parameter settings.

☆ Soln strategy by Tabu search:-

✓ Some of its characteristics can be given as: inherent simplicity, high adaptability, a short term memory process.

✓ Initial solution is randomly generated based on some rules.

✓ At every iteration, the algorithm find the best admissible next solution from the existing soln & records it as best soln if it is better than overall best.

✓ To prevent reversal to some previous soln, a tabu list T is used to keep track of the codes of the recently investigated solutions.

☆ Initial parameters :-

✓ Maximum no's of iterations.

✓ Size of tabu list-

Aspiration criteria

✓ The criteria for overruling the tabu constraints & differentiating the preference of among the neighbors.

Encoding schema.

Distributors → 1 2 3 4 5

Retailers → 1 5 2 4 3

Remarks on Tabu search :-

- ✓ Neighbor search
- ✓ Prevents being trapped in the local minimum with tabu list.
- ✓ Directs the selection of neighboz.
- ✓ cannot quaruntee the optimal solution.
- ✓ Is sequential.

- Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects & of other animals.
- In particular, ants have inspired a no's of method & techniques among which the most studied & the most successful is the general purpose optimization technique. known as ant colony optimization.

What actually it is?

- Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species.
- These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony.

- Pheromone :→ Chemical substance produced & released into the environment by an animal, especially a mammal or an insect affecting the behavior or physiology of others of its species.

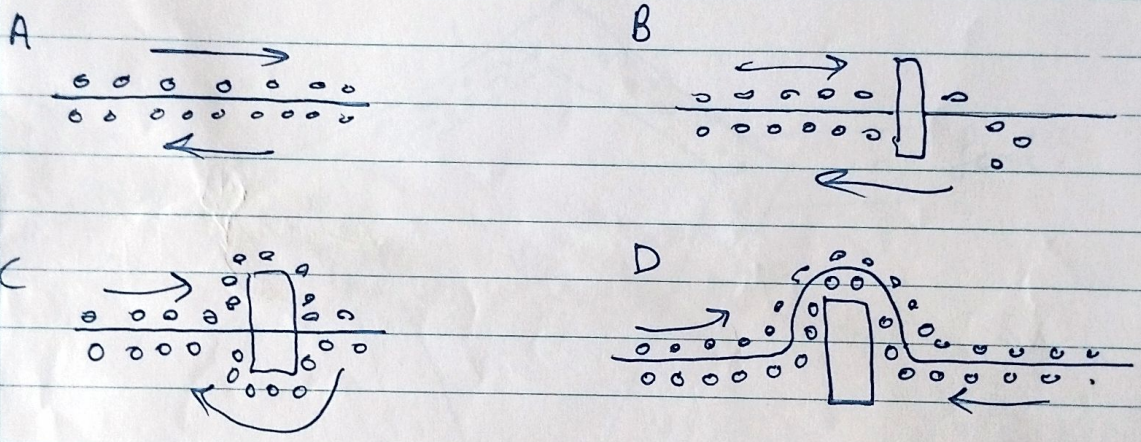
Type of swarm optimization.

Ant colony optimization :-

- Ants search for food
- The shorter the path the greater the pheromone left by an ant.
- The probability of taking a route is directly proportional to the level of pheromone on that route.
- As more ants take the shorter path, the pheromone level increases.

one sort of liquid released find out path for another ant

Path



★ The ACO Algorithm is constructed using 3 procedure :-

1) Construct Ant Solution
2) Update Pheromone
3) Daemon Actions.

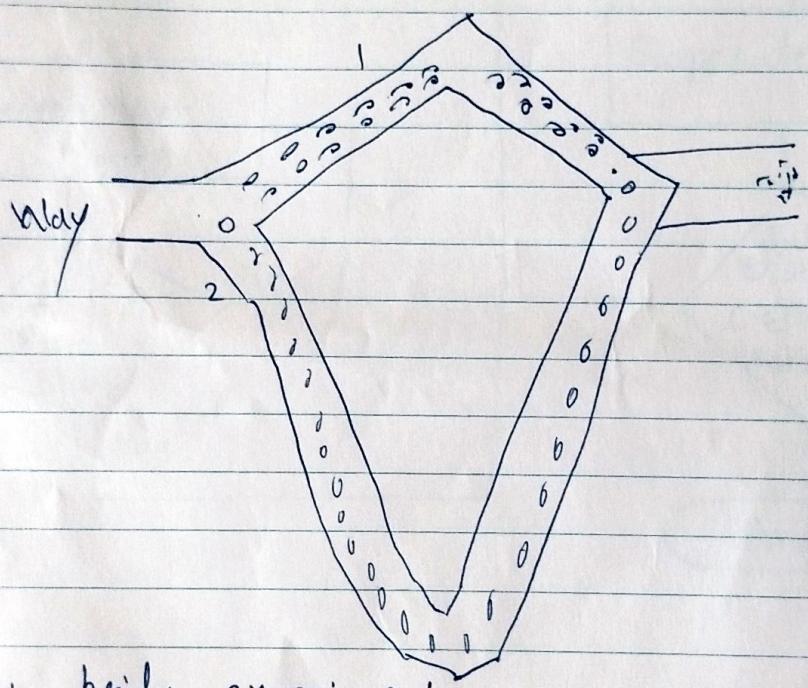
Soln have bee constructed. & before updating pheromone value often some specific value are required called Daemon action.

Problem

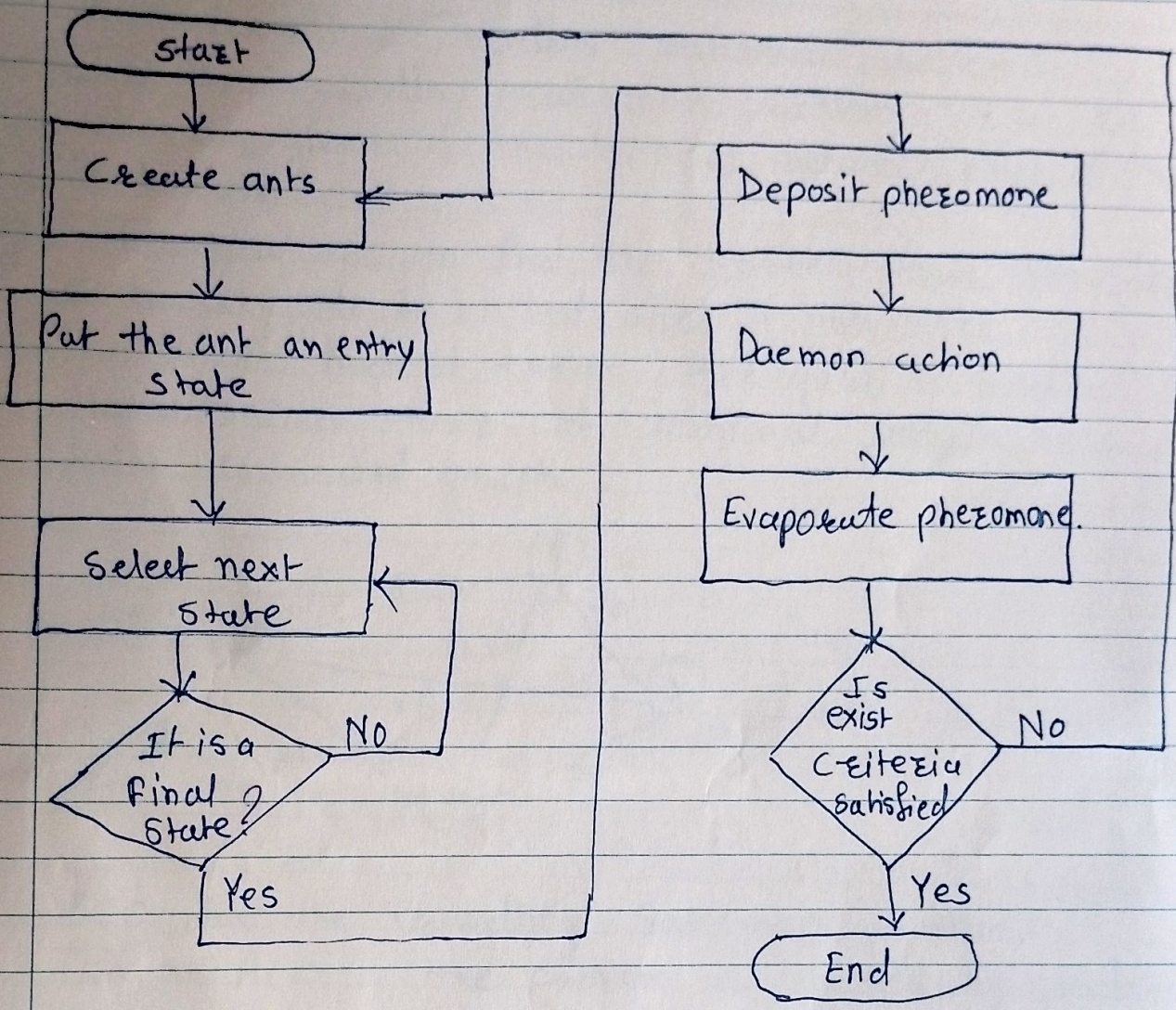
not single ant → But Group of ant.

Concept :-

- ✓ Ants navigate from nest to food source Ants are blind.
- ✓ Shortest path is discovered via pheromone trails.
- ✓ Each ant moves at random
- ✓ pheromone is deposited on path
- ✓ More pheromone on path increases probability of path being followed.



Double bridge experiment.

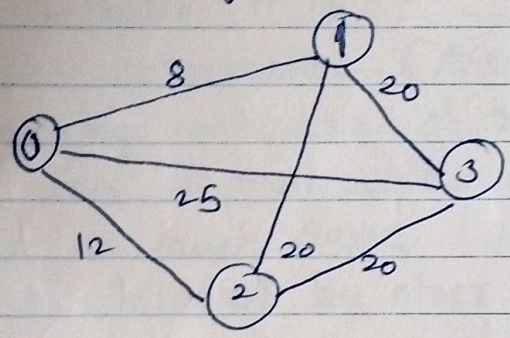

Start
Create ants
Put the ant an entry state
Select next state
It is a final state?
No
Yes
Deposit phezomone
Daemon action
Evaporate phezomone.
Is exist criteria satisfied
No
Yes
End

ACO for the Traveling salesman problem:-

In the traveling salesman problem, a set of cities is given & the distance between each of them is known.

The goal is to find the shortest toue that allows each city to be visited once & only once.

In more formal terms, the goal to find a Hamiltonian toue of minimal length on a fully connected graph.



ACO for the Traveling Salesman problem.

In ant ACO, the problem is tackled by simulating a number of artificial ants moving on a graph that encodes the problem itself:

each vertex represents a city & each edge represent a connection between two cities.

A variable called pheromone is associated with each edge & can be read & modified by ants.

ACO: Updating Pheromone.

- After constructing a tour in n time steps.
- each ant k deposits an amount of pheromone Q/L_k on the edges it traversed.
- Which is inversely proportional to the cost of the tour L_k is found. Q is an appropriate constant.
- Total pheromone deposited on edge ij is $\Delta \tau_{ij}(t, t+n)$
- The total pheromone on edge ij is updated as

$$\tau_{ij}(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t, t+n)$$

Where ρ is the rate of evaporation of pheromone.

★ The ACO algorithm (more detail)

TSP - ACO( )

1. best Tour ← nil

repeat

Randomly place M ants on N Cities.

for each ant a → construct tour

for n ← 1 to N

ant a selects an edge from the distribution P_n^a

Update best Tour

for each ant a → Update pheromone

for each edge (u, v) in the ants' tour

deposit pheromone $\propto \frac{1}{\text{tour - length on edge } (u,v)}$

Until some termination Criteria

return best Tour.

★ Ant colony optimization for TSP :-

on a city i the K^{th} ant moves to city j with a probability by

$$p^k_{ij}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta}{\sum_{h \in allowed_K(t)} ([\tau_{ih}(t)]^\alpha * [\eta_{ih}]^\beta)} \\ 0 \quad \text{otherwise.} \end{cases}$$

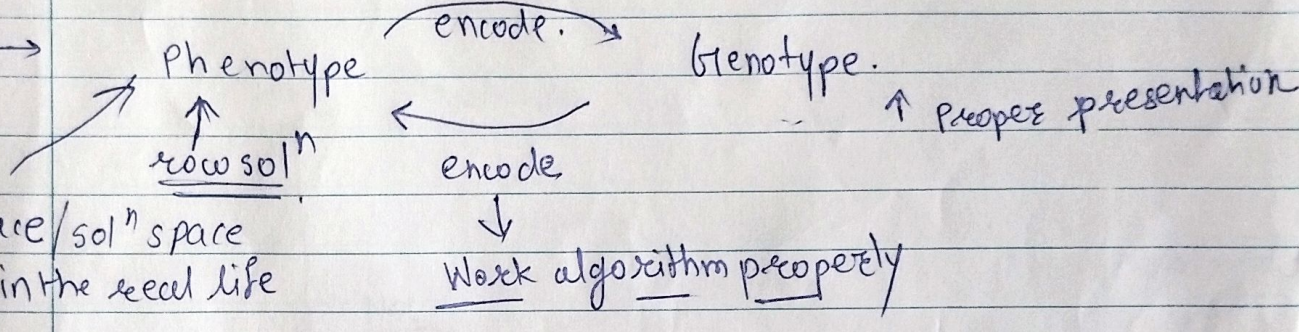
If $j \in allowed_K(t)$ the cities ant_K is allowed to move to.

Where $\tau_{ij}(t)$ is pheromone on edge ij & η_{ij} is called visibility which is inversely proportional to the distance bet^n cities i & j

exponential time complexity

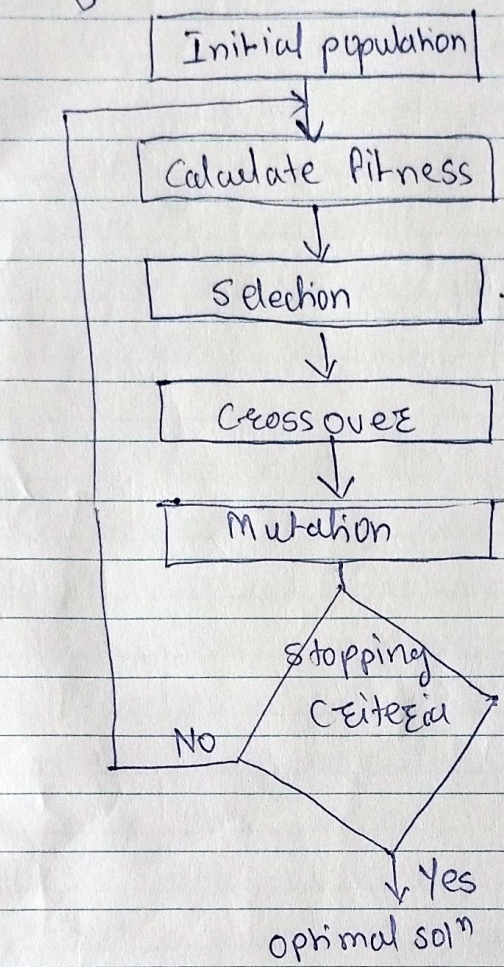
Genetic algorithm :- (John Holland)

→ Abstraction of real Biological Evolution.
→ Solve complex problem (TSP) (NP - hard prob)(0 - 1 knapsack prob)
→ Focus on optimization.
→ Population of possible sol^n for a given problem.
→ From a group of individuals, the best will survive.

→ 

TSP — already soln available.

Diversity = exist pole — one edge gp — No proper soln.
Diverse



Make
Take
fitness value 3

Genetic opn ↑ fitness.

Various method
one point

a b c d e
e f g h i
a b c h i e f g d e

① swap.
② bit-flip.

Best parent → Good fitness. → highest value.

Genetic algorithm:

- The process of sexual reproduction experiments with the genotype by making up different combinations of genes inherited by the two parents.
- The process of competition for survival of the phenotype (the physical entity) in the real world selects the best candidates & their genes propagate.

G.A :- (John Holland 1975)

- A class of methods for optimization problems, more generally known as Evolutionary Algorithm.
- Heuristic stochastic adaptive search algorithm.
- Heuristic stochastic adaptive —//—.
- Implemented on a population of candidates in sol^n space
- Inspired by the process of natural selection.
- A fitness function evaluates each candidate.
- The fittest candidates get to mate & reproduce.

G.A :- Artificial selection

Give a population of candidate sol^n.

Reproduction ← Clone each candidate in proportion to its fitness. Some may get more than one copy, some none.

↓

Crossover ← Randomly mate the resulting population & mix up the genes.

↓

Mutation ← once in a while, in search of a lost gene.

Regain

Genotypes & Phenotypes.

✓ An organism's genotype is the set of genes in its DNA responsible for a particular trait.

✓ An organism's phenotype is the physical expression of these genes.

Crossover have different method:-

1. Single point crossover.
2. Cycle crossover.
3. Partially mapped crossover.

Crossover operators :-

A single point crossover simply cut the two parents at a randomly chosen point & recombines them to form two new soln strings.

$P_1 = X_1 X_2 X_3 X_4 \mid X_5 X_6 X_7 X_8$
$P_2 \quad Y_1 Y_2 Y_3 Y_4 \mid Y_5 Y_6 Y_7 Y_8$

single point crossover.
eg :- SAT.

$C_1 = X_1 X_2 X_3 X_4 \; Y_5 Y_6 Y_7 Y_8$
$C_2 = Y_1 Y_2 Y_3 Y_4 \; X_5 X_6 X_7 X_8$

Any operator the mixes up the genes will do one can have multi point crossovers

Eg :-

$P_1 = X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8$
$P_2 = Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8$

$\Downarrow$

$C_3 = X_1 Y_2 X_3 Y_4 X_5 Y_6 X_7 Y_8$
$= Y_1 X_2 Y_3 X_4 Y_5 X_6 Y_7 X_8$

Genetic Algorithm :-

Genetic Algorithm ( )

1. Initialize a initial population of candidate solution P[1...n]
2. Repeat.
3. Calculate the fitness value of each member in P[1...n]
4. Selected [1...n] ← The new population obtained by picking n members.
5. from P[1...n] with probability proportional fitness
6. Partition selected [1...n] into two halves & randomly mate & cross over members to generate.
7. offspring [1...n]
8. With a low probability mutate some members of offspring [1...n]
9. Replace k weakest members of P[1...n] with the k strongest members of offspring [1...n]
10. Until some termination criteria.
11. return the best member of P[1...n].

GA for TSP :-

✓ Genetic algorithm can be used for the TSP as follows -

✓ Create a population of candidate TSP soln. Let the fitness function be the cost of the tour if it is a minimization problem.

✓ In the path representation the tour is represented by a permutation of the cities, with the assumption that one return from the last city in the permutation to the first.

selection :- Clone each tour in proportion to fitness
The cheapest tours are the fittest.

Crossover :- Randomly pair the resulting population & perform crossover.