Module 3 :-

K- Means algorithm :-

Clustering :-

→ clustering is the classification of objects into different groups or partitioning of a data set into subsets or clusters. Each subset of data share some common trait

Types of clustering are

1) Hierarchical algorithms
2) Partitional clustering.

* K means algorithm comes under partitional clustering type of clustering.

Common distance Measures :-

→ Distance measure will determine how the similarity of two Elements is calculated & will influence the shape of the clusters.

Euclidean distance (2-norm distance) is given by

$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

Scanned with CamScanner

→ K means clustering is an algorithm to cluster n objects based on attributes into K partitions. where $K < n$.

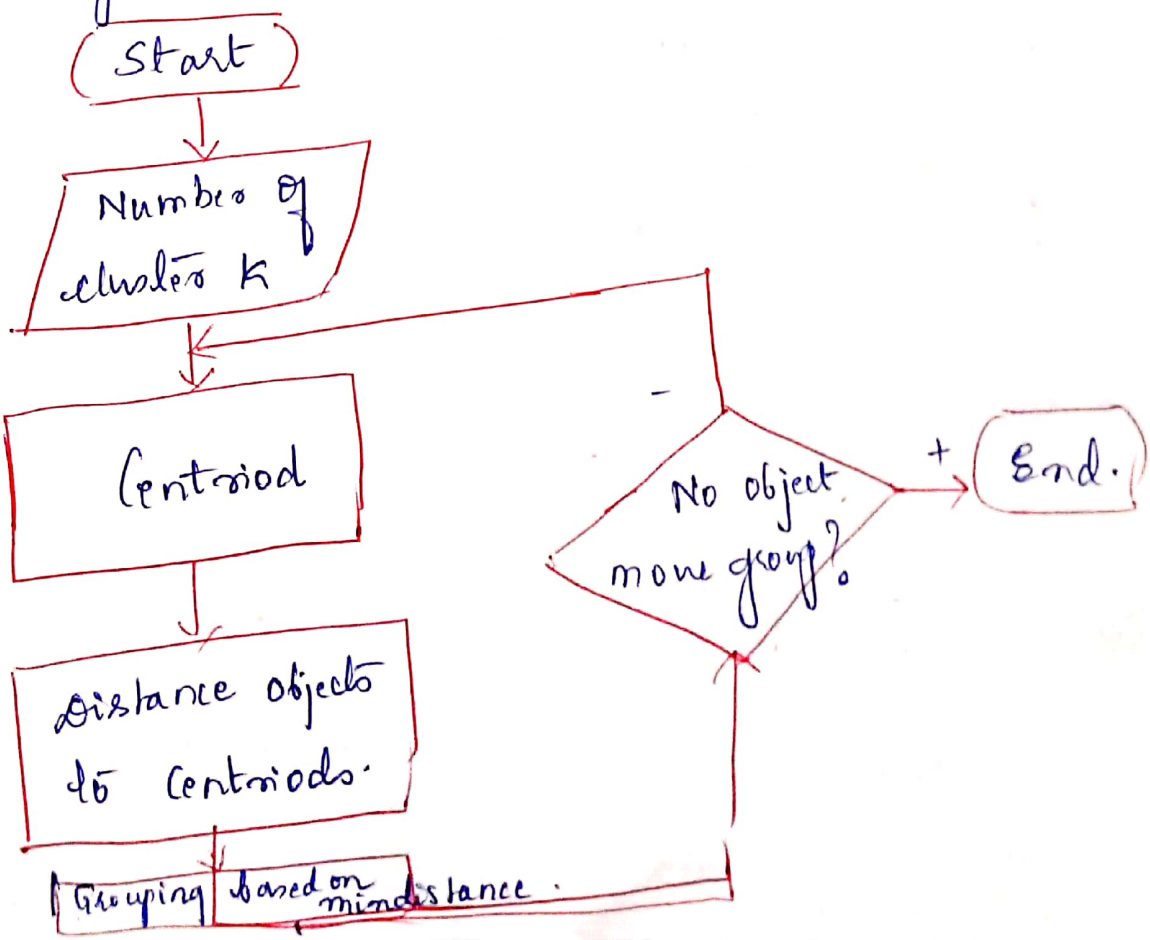
→ It assumes that the object attributes form a vector space.

note :- Supervised algorithms → Target variables are Known

Unsupervised Algorithms → set of features are Known, target or label are not Known

clustering → find clusters of data

How the Algorithm Works :-



Scanned with CamScanner

Step 1 :- For each data point, place it in a cluster whose current centroid to which it is nearest.

Step 2 :- After all data points are assigned, ~~regard~~ update the locations of centroid of the K clusters.

Step 3 :- Reassign all points to their closest centroid

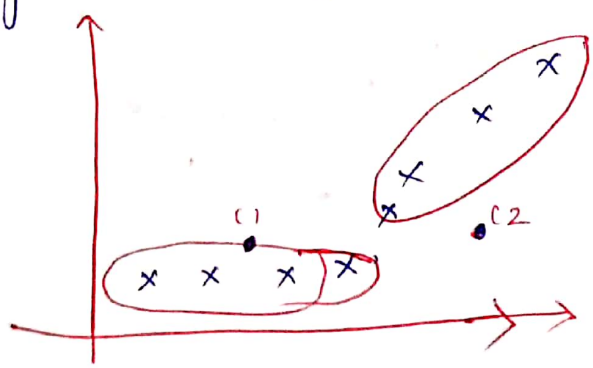
note :- Sometimes. data points changes its clusters centroids.

Step 4 :- Repeat step 2 & Step 3 untill no data points change clusters & untill all converge.

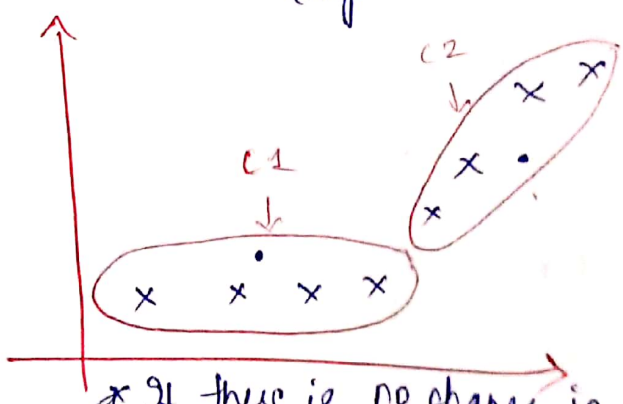
~~Con~~ note :- Convergence - Points do not move between clusters & centroids are stable.

Example :-

step 1



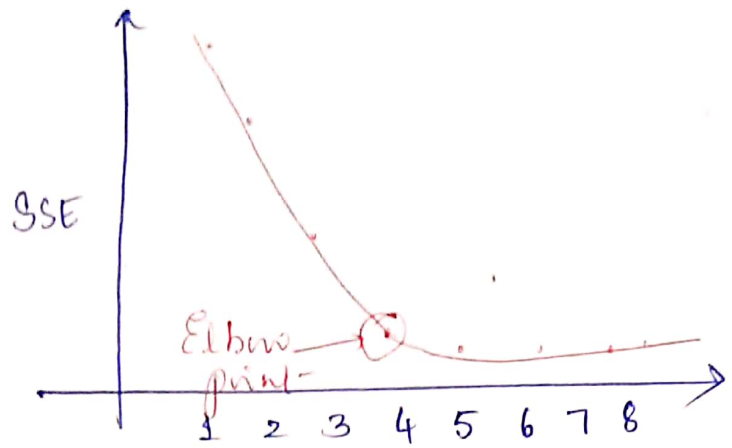
Step 2



* If there is no change in clusters then it

Scanned with CamScanner

→ How to determine correct number of clusters (K)



* Elbow Technique.

So value of $\boxed{K = 4}$ for a set of data points.

note :- SSE [sum of Square Error]

$$SSE = \sum_{i=0}^{n} dist(x_i - c_i)^2$$

c_i → cluster number < c1, c2, c3 ....

$$\therefore SSE = SSE_1 + SSE_2 + \dots + SSE_K$$

SSE_1 → for cluster 1; SSE_2 → cluster 2; SSE_K → for cluster K.

Conclusion :-

→ K - means algorithm is useful for undirected Knowledge discovery & is relatively simple.

Scanned with CamScanner

Fuzzy K-means Algorithm :-

clustering

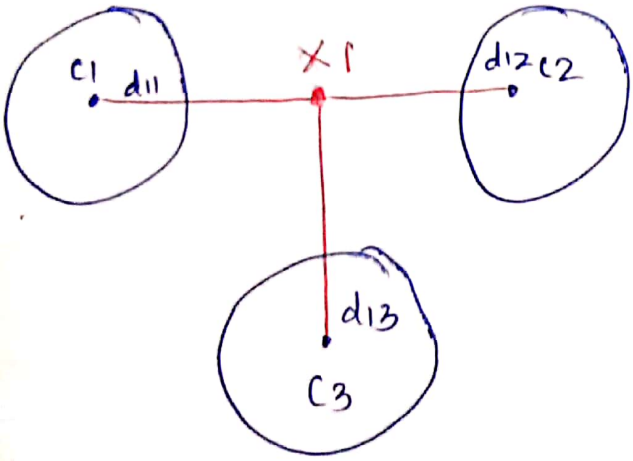
- Hard clustering
- Soft clustering

note :- K-means is a hard-clustering approch.

→ In soft or Fuzzy clustering, instead of putting each data points into seperate clusters, a probability of that point to be in that cluster assigned.

→ In soft clustering or fuzzy clustering, Each data point can belong to multiple clusters along with its probability score or likelihood.

→ Here Fuzzy set theory is used to obtain the optimal values of the Centroid.



→ In this technique the particular vector (x_1) belongs to all the clusters with different membership value.

Scanned with CamScanner

note :- sum of membership value is '1'.

→ Consider vector x_1 belongs to the cluster 1 whose Centroid is c_1 then $(x_1 - c_1)^2$ must be minimized. But if x_1 belongs to cluster 1 with membership value m_1 then $m_1 (x_1 - c_1)^2$ has to be minimized.

→ The fuzzy K-means problem is treated as the optimization technique for obtaining the membership values along with the centroids of the individual clusters such that

$$\sum_{i=1}^{N} \sum_{K=1}^{K} M_{ik}^2 (x_i - C_K)^2 \text{ is minimized}$$

M_{iK} → Membership value of vector x_i belongs to Cluster K

X_i → i^{th} vector

C_K → C_K is K^{th} Centroid.

note :- Here we use converging the membership values to find an optimal solution.

Scanned with CamScanner

Algorithm :-

Step 1 :- Initialize the membership values (M_{ij}) randomly.

Step 2 :- Compute the Centroids of the clusters

$$C_1 = \frac{\left[\text{vector } 1 * (\text{membership values of vector } 1 \text{ belonging to cluster } 1)\right]^2 \text{ i.e., } M_{11}^2 + M_{12}^2 + \ldots}{\text{Sum of the squared values of membership values belonging to the cluster } 1.}$$

III^{r} all Centroids (Eg :- $C_2, C_3 \ldots C_6$) are obtained.

Step 3 :- Update the membership values M using new Centroids.

$$M_{11} = \frac{1}{\left[\frac{(x_1 - C_1)^2}{(x_1 - C_1)^2}\right]^2 + \left[\frac{(x_1 - C_1)^2}{(x_1 - C_2)^2}\right]^2 + \ldots + \left[\frac{(x_1 - C_1)^2}{(x_1 - C_6)^2}\right]^2}$$

vector

Centroid.

Step 4 :- Compute the sum of squared difference between the previous membership value & current membership value.

Scanned with CamScanner

$$\left[m_{11}^{old} - m_{11}^{new}\right]^2 + \left[m_{12}^{old} - M_{12}^{new}\right]^2 + \ldots$$

* If computed value is not less than the threshold then goto step 2.

* If computed value is less than the threshold then stop the iteration.

Self Organizing Maps :-

→ Supervised training techniques, where there is a target output for each input pattern & the network learns to produce the required output.

→ Unsupervised training is where the network learn to form their own classifications of the training data without external help.

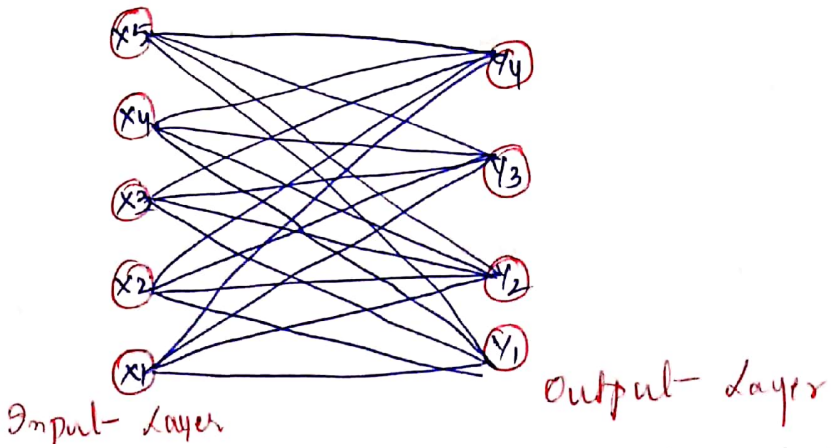
→ One particularly interesting class of unsupervised based on competitive learning in which the o/p neurons compete amongst themselves to be activated & this activated neuron is called a winner-takes-all neuron or winning neuron.

Scanned with CamScanner

→ Such competition can be implemented by having lateral inhibition connections between the neurons. resulting in neurons organizing themselves. Such a network is called as self Organizing Maps (SOM)

→ The goal of SOM is to transform an incoming signal pattern of arbitrary dimension into 1 or 2 dimensional discrete map, ∴ setting up a SOM.

→ We shall concentrate on a particular kind of SOM known as Kohonen network. This is a feed forward structure with a single computational layer arrange in rows & columns. Each neuron is fully connected to all source nodes in the input layer.



Scanned with CamScanner

→ The aim is to learn a "feature map" from the spacially "continuous Input Space" to a low dimensionality spatially discrete output space. which is formed by arranging the computational neurons into a grid.

Algorithm :-

Step 1 :- Each node weights are Initialized

Step 2 :- Every node in the network is Examined to Calculate which one's weight are most like the Input vector. The winning node is commonly Known as the "Best matching unit"

Step 3 :- Update the weight after finding the winning neuron

Step 4 :- Training will stop if Input & weight vectors are Identical otherwise goto step - 2

$$\boxed{w^{new} = w^{old} + \alpha\eta(\text{I/P} - \text{weight vector})}$$

Scanned with CamScanner

K- means algorithm :-

*) Assume Euclidean space / distance

*) start by picking K, the number of clusters

*) Initialize cluster by picking one point per cluster

Note: for this moment, assume we pick the K points at Random.

step 1: for each point, place it in the cluster whose current centroid it is nearest

step 2: After all points are assigned, update the ~~locatio~~ locations of centroid of the K clusters

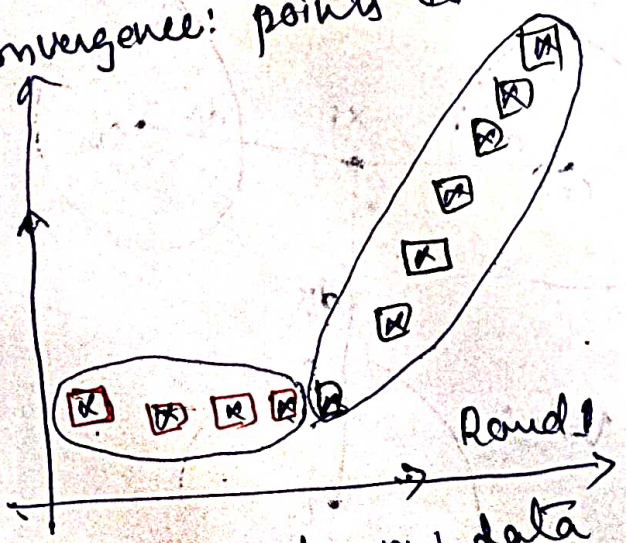
step 3: Reassign all points to their closest centroid

Note: some times moves points b/w clusters

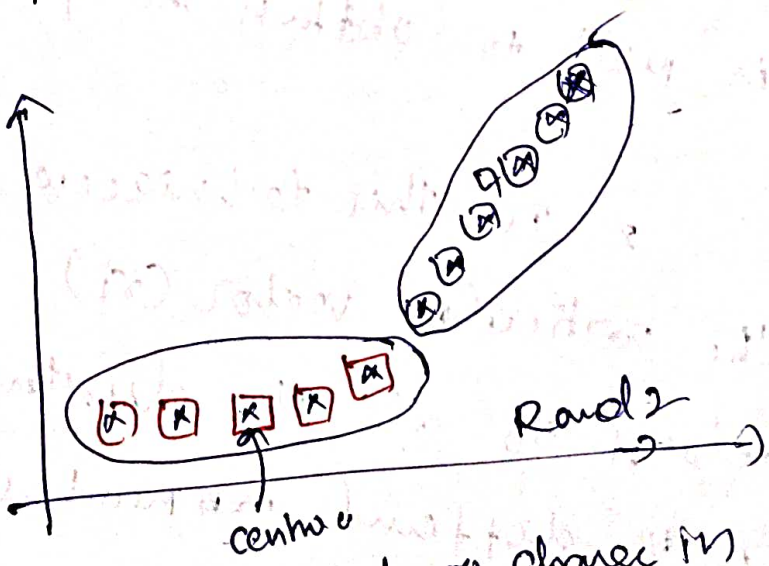
step 4: Repeat step 2 and step 3 ~~until~~ until convergence

convergence: points do not move b/w clusters and centroids stable

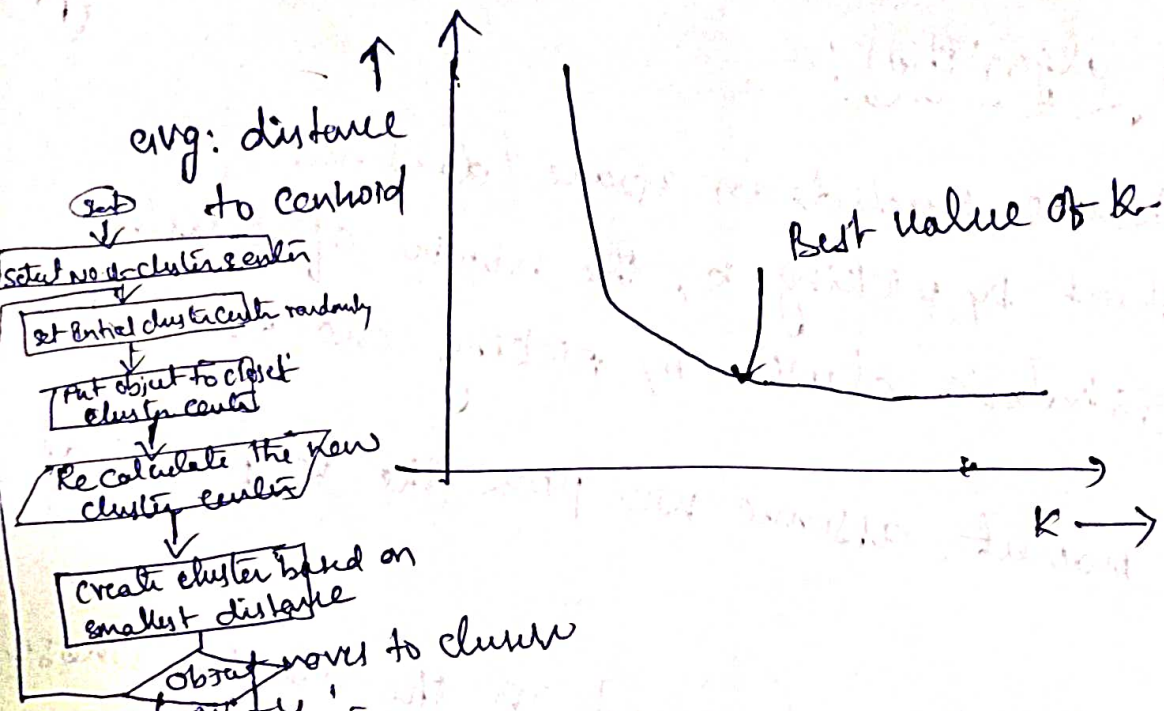
Eg



Because of new data centroids shift



In round - '3' no change in centroid position.



Complexity :-

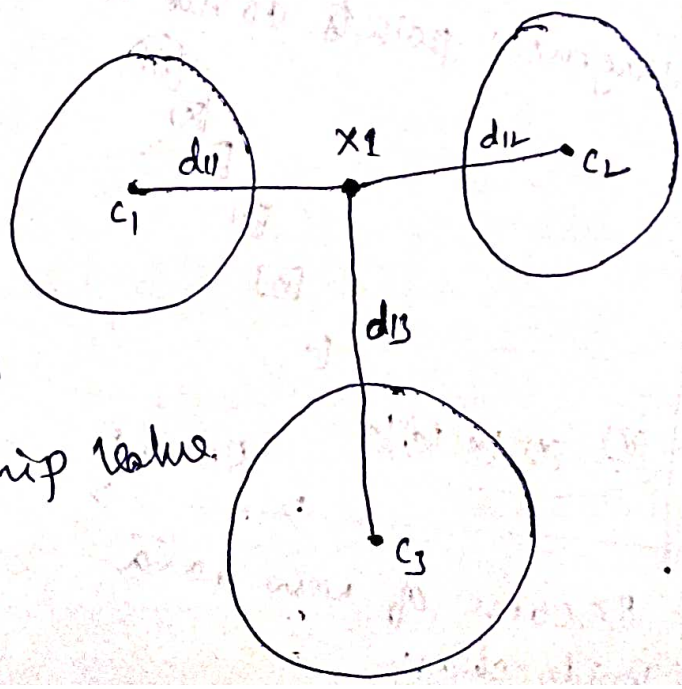
1) In each round, we have to examine each input point exactly once to find closest centroid

2) If data is too large then computation or iterations are increased.

2) Fuzzy K-means algorithm

1. In fuzzy-K means techniques, Fuzzy set theory is used to obtain the optimal values of the centroid

2. In this technique the particular vector (x_1) belongs to all the clusters with different membership value



Note: Sum of membership value is '1'

3. consider the vector x_1 belongs to the cluster 1 whose centroid is c_1, then $(x_1 - c_1)^2$ must be minimized. But if x_1 ~~belo~~ belongs to ~~cluster~~ cluster one with membership value m_1 then $m_1 (x_1 - c_1)^2$ has to be minimized. similarly x_1 belongs to cluster-2 with membership value m_2 then $(x_1 - c_2)^2 . m_2$ has to be minimized.

The fuzzy k-means problem is treated as the optimization technique for obtaing the membership values along with the centroids of the individual clusters such that

$$\boxed{\sum_{i=1}^{N} \sum_{k=1}^{K} M_{ik}^{2} . (x_i - c_k)^2 \text{ is minimized}}$$

$M_{ik} \rightarrow$ membership value of vector x_i belongs to the cluster K.

$x_i \rightarrow$ i^{th} vector

$c_k \rightarrow$ c_k is k^{th} centroid.

~~No~~ Note: Here we uses converging the membership values.

Fuzzy K-means Algorithm :

Step 1 : Initialize the membership values (M_{ij}) randomly

Step 2 : Compute the centroids of the clusters (Eg : 6 clusters)

Hint $C_1 = \frac{X_i M}{}$

Hint

$C_1 = [$ vector 1 * (membership value of vector 1 belongs to cluster 1)^v i.e M_{11}^v + vector 2 * (membership value of vector 2 belongs cluster 1)^v i.e M_{12}^v + ... vector 100 * (membership value of vector 100 belongs to cluster 1)^v i.e M_{1100}^v] / sum of the squared values of the membership values belongs to the cluster - 1.

Thy all centroids (Eg: $C_2, C_3 \ldots C_6$) are obtained

Step 3 : Update the membership values M using new (current) values of centroids.

$$M_{11} = \frac{1}{\left[\frac{(x_1-C_1)^v}{(x_1-C_1)^v}\right]^v + \left[\frac{(x_1-C_1)^v}{(x_1-C_2)^v}\right]^v + \cdots + \left[\frac{(x_1-C_1)^v}{(x_1-C_6)^v}\right]}$$

Similarly other membership values are computed.

$$M_{45} = \frac{1}{\left[\frac{(x_4-C_5)^v}{(x_4-C_1)^v}\right]^v + \left[\frac{(x_4-C_5)^v}{(x_4-C_2)^v}\right]^v + \cdots + \left[\frac{(x_4-C_5)^v}{(x_4-C_6)^v}\right]}$$

vector centroid

step 4 : compute the sum of squared difference between the previous membership value and the current membership value.

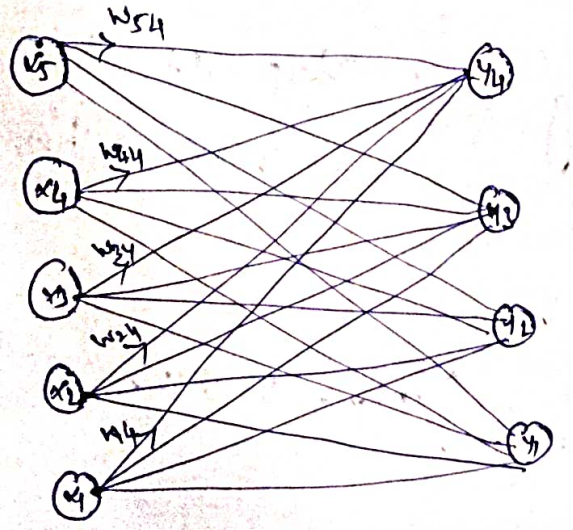
If computed value is not less than the threshold then go to step-2.

If computed value is less than the threshold then stop the iteration.

Eg : $(M_{11}^{old} - M_{11}^{new})^2 + (M_{12}^{old} - M_{12}^{new})^2 + \cdots$

KSOM (Kohonen self-organizing Maps)

This has a feed-forward structure with a single computational layer of neurons. ~~arranged~~ Each neuron is fully connected to all the source units in the i/p layers.



cluster unit

Input units

The aim is to learn a "feature map" from the spacially "continuous input space", in which ~~out~~ our I/p vectors live, to the low dimentional spatially discrete output space which is formed by arranging the computational neurons into a grid.

KSOM (Algorithm)

step 1: Each node's weights are initialized (all I/p & o/p are normalized).

step 2: Every node in the network is examined to calculate which ones' weights are most like the I/p vector. The winning node is commonly known as the "Best matching unit".

step 3: Update the weight after find the winner neuron (more importance to the neuron which is closer to winner, less importance to the ~~the~~ neuron which is far from winner)

step 4: Training will stop if I/p & weight vectors are identical. otherwise go to step-2.

$$\boxed{w^{New} = w^{old} + \alpha\eta\,(\text{I/p} - \text{weight vector})}$$

Particle SWARM Algorithm :-

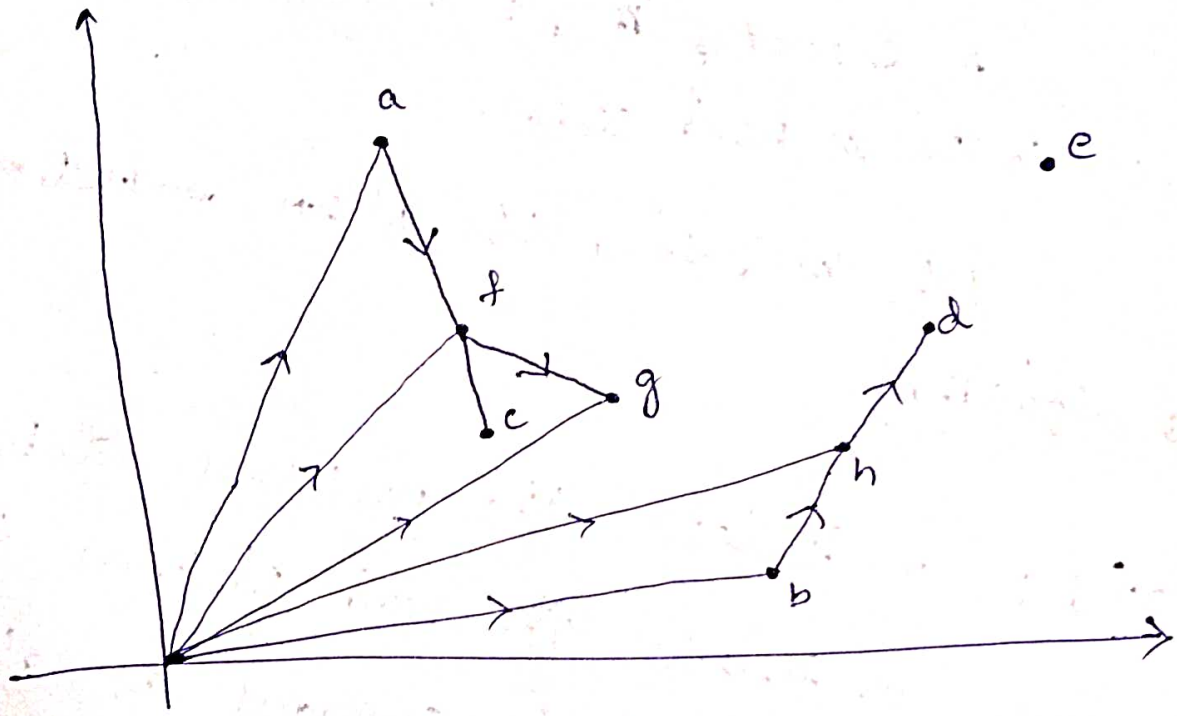


fig: vector Representation of PSO Algorithm.

1) Let the position of the swarms be at 'a' and 'b'. Both are trying to reach 'e'.

2) Let 'a' decided to move 'c' and 'b' is decided to move 'd'.

3) The distance b/w position 'c' and 'e' is greater than the distance b/w d and 'e'.

4) So, 'd' is a common position decided by the a and b

5) The position 'c' is the Individual decision taken by 'a', The position 'd' is the " " " by 'b'.. the position 'd' is common position to take by both a & b.

Now. let us consider three swarms (A, B, C) are ~~trying~~ trying to reach the particular destination point. 'D'

A decides 'A', B decides B', C decides C' "

~~which is the~~ as next position.

~~Let the distance b/w. A' etc.~~ ~~what~~

PSO algorithm

step 1 : Initialize the positions a, b, c, d, e

step 2 : Initialize the next possitions decided by The Individual swarms as a', b', c', d', e'

step 3 : Global decision regarding the next position is computed as follows.

compute f(a', b, c, d, e), f(a, b', c, d, e), f(a, b, c', d, e) f(a, b, c, d', e) and f(a, b, c, d, e'). find the minimum among the computed values.

If f(a', b, c, d, e) is minimum among all, the global position decided regarding the next position is a'

If f(a, b', c, d, e) is minimum the global position is b' and so on.

Let

Let the selected global position is represented as "global"

step 4 : Net value for 'a' is computed as the linear combination of 'a', (a'-a), (global - a). i.e

weight of individual decision

weight of random decision

next a = a + C1 * RAND * (a'-a) + C2 * RAND * (global - a)

next b = b + C1 * RAND * (b'-b) + C2 * RAND * (global - b)

⋮

next e = e + C1 * RAND * (e'-e) + C2 * RAND * (global-e)

step 5 : change the current value for a, b, c, d, e as next a, next b, next c, next d, next e.

step 6 : If f(next a, b, c, d, e) is less than f(a, b, c, d, e) then update the value for "a" as "next a" otherwise 'a' is not change (do this for b', c', d', e' also.)

step 7 : Repeat the steps 3 to 6 for much iterations to reach the final decision.

The values C1 & C2 are decided based on the weightage given to individual decision and global decision respectively.

Let $\Delta a(t)$ is the change in the value for updating the value for 'a' in 't' th iteration. then next a at (t+1) th iteration can be computed using the following formula.

This is considered as the velocity for updating the position of the swarm in every iteration

$$next\,a(t+1) = a(t) + \Delta a(t+1)$$

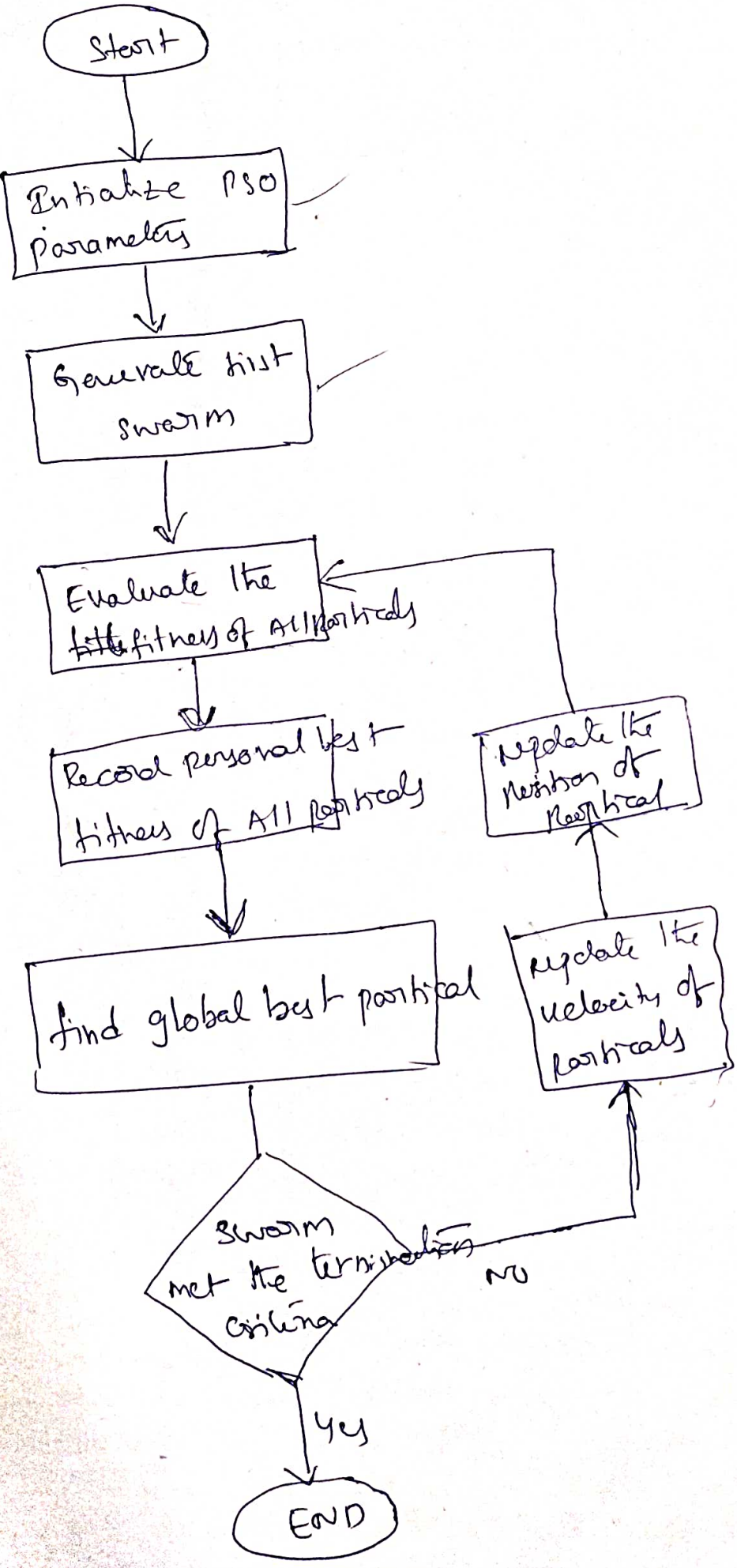
where

$$\Delta a(t+1) = c1 * rand * (a' - a) + c2 * rand * (Global\,a) + w(t) * \Delta a(t)$$

w(t) is the weight at t^{th} iteration. The value for 'w' is adjusted at every iteration as given bellow, where 'iter' is total no. of iterations used.

$$\boxed{w(t+1) = \frac{w(t) - t * w(t)}{iter}}$$

PSO Algorithm flow chart



Ant colony optimization

Ant colony optimization technique exploit the natural behavior of ants in finding the shortest path to reach destination from the source

Algorithm

consider the problem of finding the optimum order in which the numbers from 1 to 8 are arranged so that the cost of the order is maximized. cost of the particular order is computed using the two ~~matrix~~ matrices A and B as described below

a_1							
a_2	a_3						
a_4	a_5	a_6					
a_7	a_8	a_9	a_{10}				
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}			
a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}		
a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	
a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}

Matrix A

b_1							
b_2	b_3						
b_4	b_5	b_6					
b_7	b_8	b_9	b_{10}				
b_{11}	b_{12}	b_{13}	b_{14}	b_{15}			
b_{16}	b_{17}	b_{18}	b_{19}	b_{20}	b_{21}		
b_{22}	b_{23}	b_{24}	b_{25}	b_{26}	b_{27}	b_{28}	
b_{29}	b_{30}	b_{31}	b_{32}	b_{33}	b_{34}	b_{35}	b_{36}

Matrix B

The cost of the order [3 2 1 5 7 2 4 6 8] is computed as the sum of the products of values obtained from the positions (3,1) or (1,3), (2,2), (3,1), (4,5) or (5,4), (5,7) or (7,5), (6,2), (7,4), (8,6) of matrix A & B

$(a_6 \times b_6) + (a_3 \times b_3) + (a_6 \times b_{36}) + (a_{14} \times b_{14}) + (a_{26} \times b_{26}) + (a_{27} \times b_{27}) + (a_{24} \times b_{25}) + (a_{36} \times b_{36})$

The problem of finding the optimal order is achieved using Ant cololony optimization technique as described below.

Step 1: Generate the five sets of orders randomly which are treated as the intial values selected by the 5 ants respectively.

The following are the orders selected by the 5 ants along with the corresponding cost as given bellow

ANT	ORDERS	COST
ANT 1	7 1 3 4 6 2 5 8	C1
ANT 2	4 2 8 5 7 1 3 6	C2
ANT 3	5 1 3 2 4 7 6 8	C3
ANT 4	3 7 4 1 8 6 5 2	C4
ANT 5	8 1 2 4 3 5 7 6	C5

Step 2: Pheromone is the value assigned for ith number for the jth position of the sequence selected by all the ants. Initially all the values of matrix are completely filled up with ones. The values of pheromone matrix are updated in every iteration.

pheromone matrix (n+1) = pheromone matrix (n) + updating pheromone matrix.

	1	2	3	4	5	6	7	8
1	0	$\frac{1}{c_1}+\frac{1}{c_3}+\frac{1}{c_5}$	0	$\frac{1}{c_4}$	0	$\frac{1}{c_2}$	0	0
2	0	$\frac{1}{c_2}$	$\frac{1}{c_5}$	$\frac{1}{c_3}$	0	$1/c_1$	0	$\frac{1}{c_4}$
3	$\frac{1}{c_4}$	0	$\frac{1}{c_1}+\frac{1}{c_3}$	0	$\frac{1}{c_5}$	0	$\frac{1}{c_2}$	0
4	$\frac{1}{c_2}$	0	$\frac{1}{c_4}$	$\frac{1}{c_5}+\frac{1}{c_1}$	$\frac{1}{c_3}$	0	0	0
5	$\frac{1}{c_3}$	0	0	$1/c_2$	0	$\frac{1}{c_5}$	$\frac{1}{c_4}+\frac{1}{c_1}$	0
6	0	0	0	0	$\frac{1}{c_1}$	$\frac{1}{c_4}$	$\frac{1}{c_3}$	$\frac{1}{c_2}+\frac{1}{c_5}$
7	$\frac{1}{c_1}$	$\frac{1}{c_4}$	0	0	$1/c_2$	$\frac{1}{c_3}$	$\frac{1}{c_5}$	0
8	$1/c_5$	0	$\frac{1}{c_2}$	0	$\frac{1}{c_4}$	0	0	$\frac{1}{c_1}+\frac{1}{c_3}$

Step 3: Next set of orders selected by the ants is as described bellow.

Eg: The random positions 7 8 2 3 6 4 1 5 (say)

Let the order decided by the ant 1 be represented as [u1, u2 u3 u4 u5 u6 u7 u8]

solut

* The value for the u7 in the order as mentioned above is selected first as the first number of the ~~ran~~ randomly generated position sequence is 7.

* To select the value for $u7$, 7th column of the pheromone matrix is used as below,

$$\text{column } 7 = \left[0 \quad 0 \quad \frac{1}{c_2} \quad 0 \quad \frac{1}{c_1}+\frac{1}{c_4} \quad \frac{1}{c_3} \quad \frac{1}{c_5} \quad 0\right]$$

* Now calculate the normalized value of column 7

This is also called as the probability vector used for selecting the value for $u7$ in the order

$$S = \left[\frac{1}{c_2}+\frac{1}{c_1}+\frac{1}{c_4}+\frac{1}{c_3}+1/c_5\right] = \underline{\underline{\text{value}}}$$

$$\text{Normalized vector } 7 = \frac{\text{column } 7}{S} = \left[\frac{0}{S} \quad \frac{0}{S} \quad \frac{1/c_2}{S} \quad \frac{0}{S} \quad \frac{c_1c_4+\frac{1}{c_1}}{S} \quad \frac{1/c_3}{S} \quad \frac{1/c_5}{S} \quad \frac{0}{S}\right)$$

$$= [P1 \quad P2 \quad P3 \quad P4 \quad P5 \quad P6 \quad P7 \quad P8]$$

* find the lowest value position, for Example If P_5 is the lowest value the 5 is assigned to $u7$.

<u>Note:</u> position no. '5' is not used again.

* The next value in position sequence is 8, so column '8' is selected from pheromone matrix, and select the normalized value

$$\text{column } 8 = \left[0 \quad 1/c_4 \quad 0 \quad 0 \quad \frac{1}{c_2}+\frac{1}{c_5} \quad 0 \quad \frac{1}{c_1}+\frac{1}{c_3}\right]$$

$$S = \left[\frac{1}{c_4} + \left(\frac{1}{c_2} + \frac{1}{c_7}\right) + \left(\frac{1}{c_1} + \frac{1}{c_3}\right) \right]$$

normalized vector $8 = \frac{\text{column } 8}{S} = (a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7)$

find the lowest value in normalized vector '8' and replaced with v8. Examp If a_3 is lowest then number '3' is assigned to v8.

* This process is repeated to obtain the order selected by the ~~ant~~ Ant 2, Ant 3, Ant 4, Ant 5.

* This is called one Iteration.

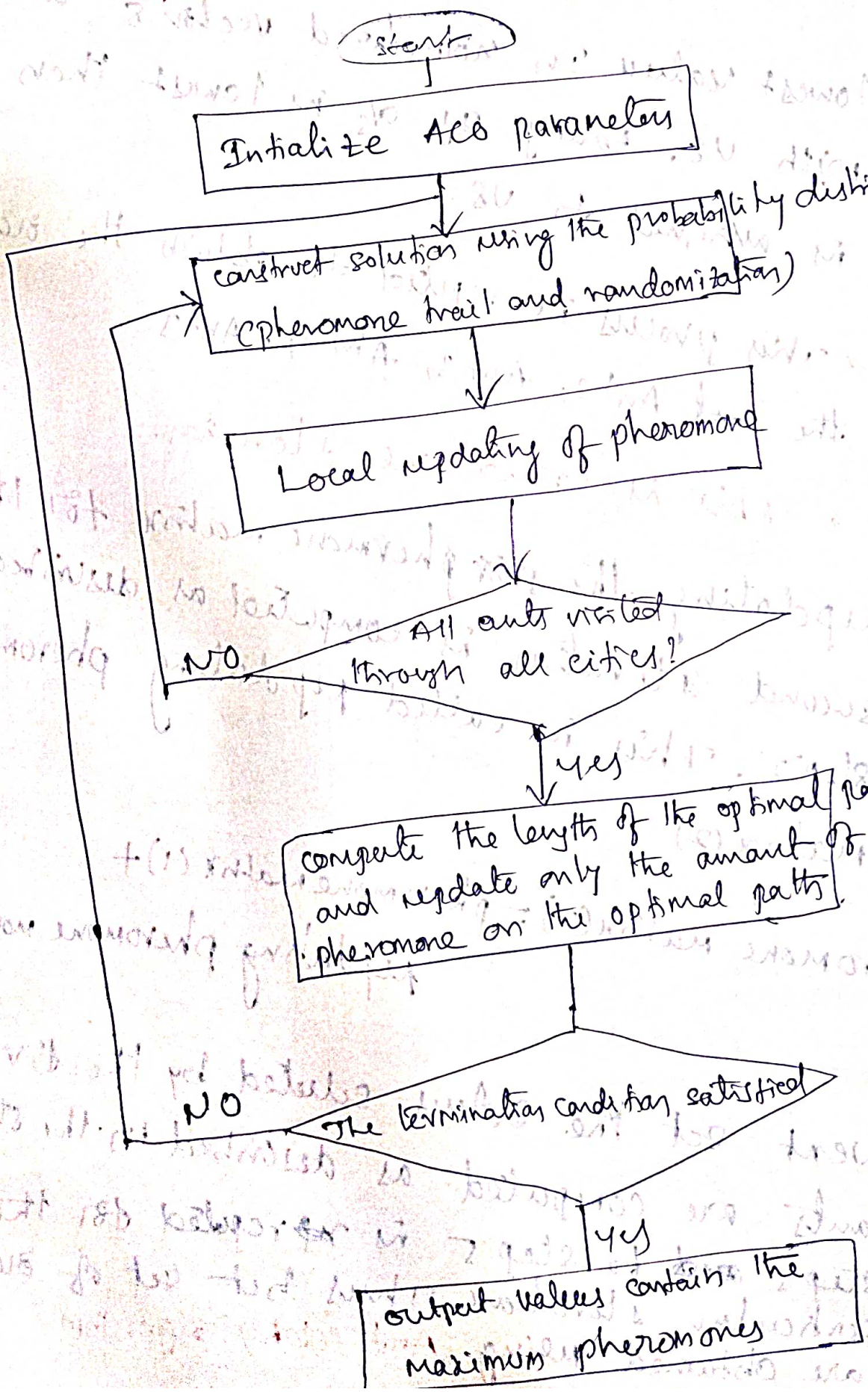
step 4 : Updating the ~~pre~~ pheromone matrix for the second Iteration is computed as described in step '2'. This is called updating pheromone matrix (2)

$$\therefore \boxed{\text{pheromone matrix}(2) = \text{pheromone matrix}(1) + \text{updating pheromone matrix}(2)}$$

step 5 : Next set the orders selected by the five ants are computed as described in the step '3'

step 6 : step 3 ~~and~~ to step 5 is ~~re~~ repeated for the particular Iterations. Thus best set of orders are obtained using Ant colony algorithm

Note: Best among the best set of orders selected by the five ~~ants~~ ants in the last iterations is selected as the gobel best order.



...al Foraging

1. Bacterial tries to move towards the region where more nutrients are available (i.e) moving ~~to~~ towards Increasing concentration of nutrients

$$J(\theta) \downarrow\downarrow \rightarrow \text{objective function}$$

θ = the position of the Bacteria

$J(\theta)$ = is the nutrient profile

$J(\theta) < 0 \longrightarrow$ presence of Nutrients

$J(\theta) = 0 \longrightarrow$ Neutral

$J(\theta) > 0 \longrightarrow$ noxious substance

1. Chemotaxis : Tendency to move ~~to~~ towards distant sources of nutrients

a) Tumbling [change in directions]

$$\theta^{new} = \theta^{\hat{}} + C\,\phi$$

$\phi \rightarrow$ must walk in random directions

$$\phi = \frac{\Delta}{\sqrt{\Delta^T \Delta}}$$

$C \rightarrow$ chemotactic step size

b) Swimming :

movement in same direction as the previous tumbling.

Note: Bacterium will tend to keep moving if it is headed in the direction of an increasingly favorable environment

If at θ^{new}, J^{new} is lower (better) than the cost at θ^{i}, another swimming step is taken and is continued as long as it continues maximum no. of swimming steps N_s.

2. Reproduction

After N_s chemotactic steps, Reproduction steps is taken

a) Least healthy bacteria die

b) "S_r" healthiest bacteria split into two bacteria which are then placed in the same location

Note: Health of the bacteria are measured by objective function J.

Higher cost represents that the bacterium did not get many nutrients (worse as healthy) and hence unlikely to reproduce

3. Elimination - Dispersion

In real world process, bacteria can be dispersed to new locations, for example wind dispersal.

Individual bacterium is stochastically selected for elimination from the population and is replaced by a new Bacterium located at a random new location within the optimization domine, according to the present probability P_{ed}.

4. Social communication

Bacterium that has uncovered good source of nutrients can release a "chemical attract signal" which attracts the other bacteria to converge to its current location. (This help in making all the bacterium to come closer)

Also "chemical Repellent signal" is released to ensure that the bacteria do not get too close to each other.

Let θ^i be the position of the ith bacteria then the objective function J is modified as,

$$J_{sw}^{i} = J^{i} + J_{cc}(\theta^{i}, \theta)$$

$J_{cc}(\theta^i, \theta)$ is choosen such that the distance b/w ith bacteria and all other bacterium are maximized (for Repellent signal) and distance b/w "ith" bacteria and all other bacterium are minimized (for attractant signal).

Note: The width of the attractant and repellent signals are used based on the requirement.

$$J_{cc}(\theta^i, \theta) = -M\left(\sum_{k=1}^{S} e^{-W_a\|\theta^k - \theta^i\|^2} - \sum_{k=1}^{S} e^{-W_r\|\theta^k - \theta^i\|^2}\right) \quad (a)$$

width of the attraction

width of the repellent

Let

P = Dimensions of the search space

S = Total no. of bacteria in the population

N_c = The no. of chemotactic steps

N_s = The swimming lengths

N_{re} = No. of Reproduction steps

N_{ed} = No. of elimination-dispersal events

P_{ed} = Elimination dispersal probability

$C(i)$ = The size of the step taken in the random direction specified by the tumble.

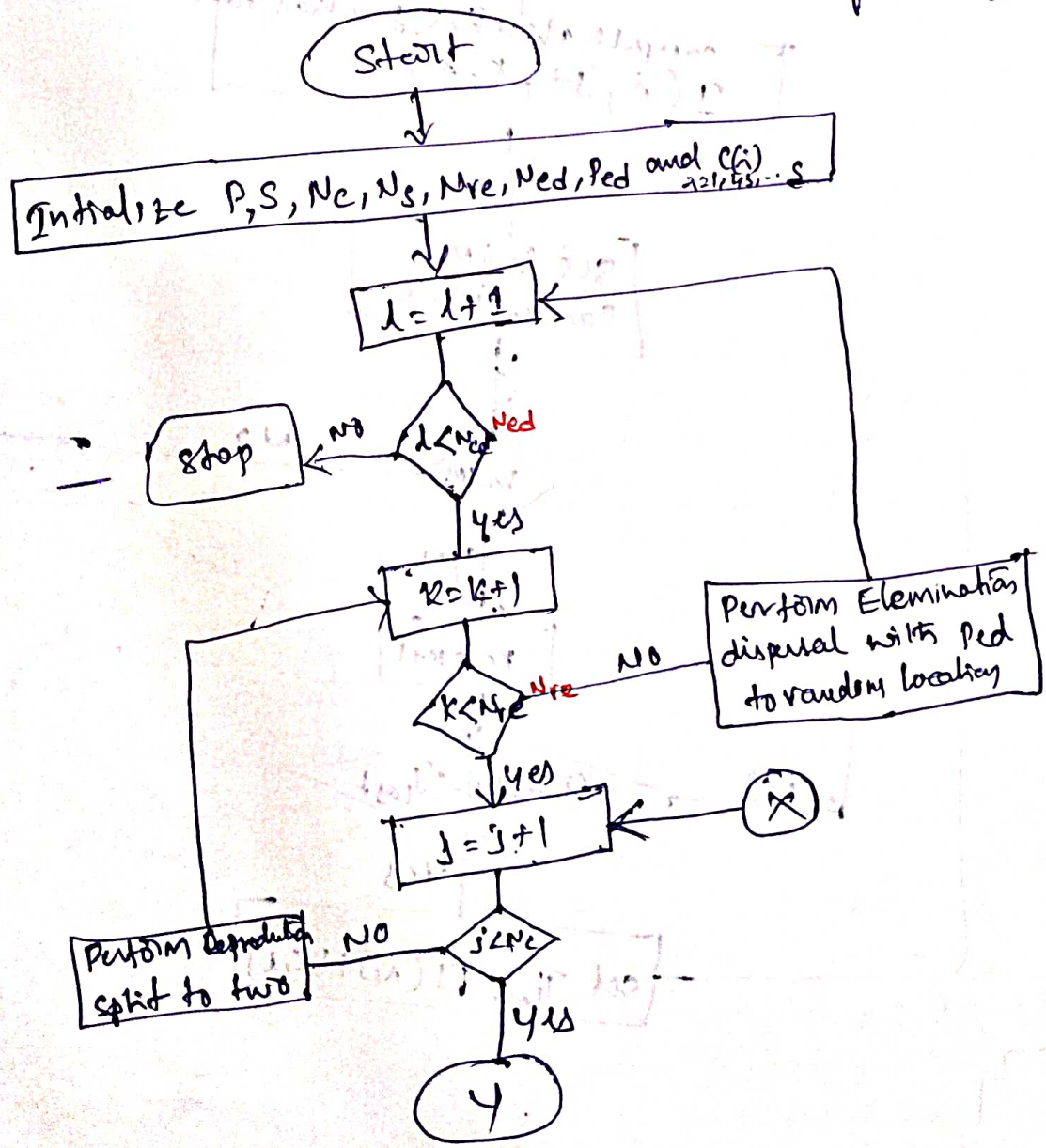
let : $\theta^i(j, k, l)$ = i^{th} bacterium at j^{th} chemotactic, k^{th} reproductive and l^{th} elimination dispersion step.

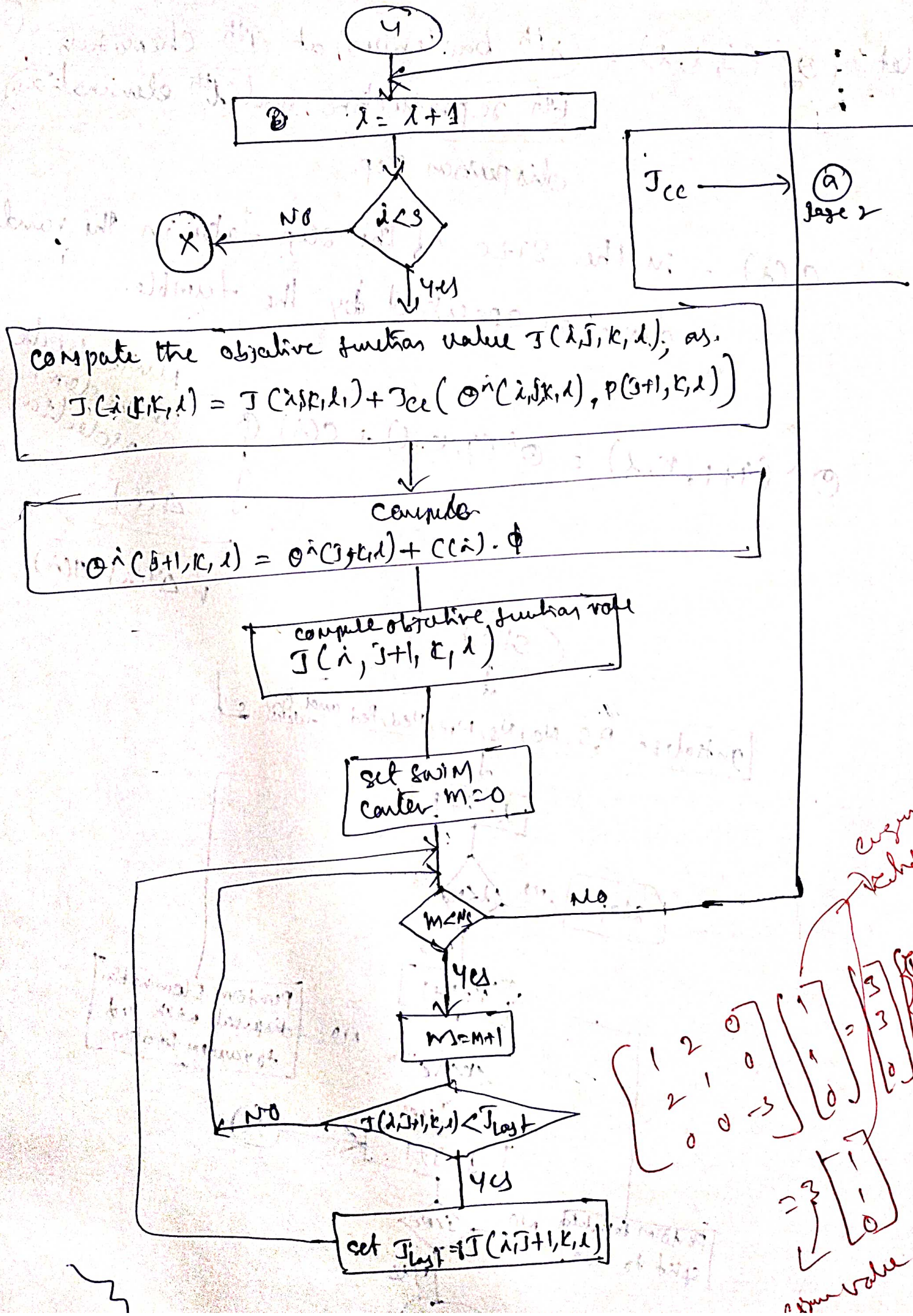
$C(i)$ = is the size of the step taken in the random direction specified by the tumble.

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\,\phi$$

ϕ → Unit walk in random direction

$$\phi = \frac{\Delta(i)}{\sqrt{\Delta^T(i) \cdot \Delta(i)}}$$





$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix}$$

eigen value

$$= 3 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

eigen value

Q.1. Use k-means clustering to divide data into 2 clusters

x_1	1	2	2	3	4	5
x_2	1	1	3	2	3	5

Step 1 :- choose 2 random cluster centres.

Say $v_1 = (2,1)$ $v_2 = (2,3)$

Data point	Distance from v_1	Distance from v_2	Ans
a_1 (1, 1)	1	2.24	
a_2 (2, 1)	0	2	
a_3 (2, 3)	2	0	
a_4 (3, 2)	1.41	1.41	
a_5 (4, 3)	2.83	2	
a_6 (5, 5)	5	3.61	

(x_1, x_2) (y_1, y_2)

$(1,1)$ $(2,1)$

$$\sqrt{(1-2)^2 + (1-1)^2} = \sqrt{1+0}$$

$$= 1$$

Scanned with CamScanner

Step 3 :- Cluster 1 of V1

: ~~{1, 2, 3}~~ $\{a_1, a_2, a_4,\}$

cluster 2 of V2 : $\{a_3, a_5, a_6\}$

Step 4 :- Recalculate cluster centers.

$$V1 = \frac{1}{3}\left[(1,1) + (2,1) + (3,2)\right]$$

$$= \frac{1}{3}\left[6, 4\right]$$

$$= (2, 1.33)$$

$$V_2 = \frac{1}{3}\left[a_3 + a_5 + a_6\right]$$

$$= \frac{1}{3}\left[(2,3)(4,3)(5,5)\right]$$

$$= \frac{1}{3}\left[(11, 11)\right]$$

$$= (3.67, 3.67).$$

Step 5 :- Repeat from S2 until we get same cluster centers as same cluster elements.

Distance table :-

Data point	Distance $v_1(2, 1.33)$	$v_2(3.67, 3.67)$	Assign cen
a_1	1.05	3.78	v_1
a_2	0.33	3.15	v_1
a_3	1.67	1.8	v_1
a_4	1.204	1.8	v_1
a_5	2.605	0.75	v_1
a_6	4.44	1.88	v_2

cluster 1 of v_1 : $\{a_1, a_2, a_3, a_4\}$

v_2 : $\{a_5, a_6\}$

Recalculating cluster centers

$$v_1 = \frac{1}{4}[a_1 + a_2 + a_3 + a_4]$$

$$= \frac{1}{4}[(1,1) + (2,1) + (2,3) + (3,2)]$$

$$= \frac{1}{4}(8, 7) = (2, 1.75)$$

Scanned with CamScanner

$$v_2 = \frac{1}{2}[a5 + a6]$$

$$= \frac{1}{2}[(4,3) + (5,5)]$$

$$= \frac{1}{2}(9,8) = 4.5, 4$$

* centroid not same.

Again Distance table.

	(2, 1.75)	(4.5, 4)	
	V1	V2	Assigned cluster
a1	1.25	4.61	V1
a2	0.75	3.9	V1
a3	1.25	2.69	V1
a4	1.03	2.5	V1
a5	2.36	1.12	V2
a6	4.42	1.12	V2

cluster 1 of $V_1 = \{a_1, a_2, a_3, a_4\}$

of $V_2 = \{a_5, a_6\}$

* cluster elements are same as in previous iteration.

Scanned with CamScanner

cluster 1 : { (1, 1), (2, 1), (2, 3)
(3, 2) }

cluster 2 : { (4, 3), (5, 5) }.

Scanned with CamScanner

$$[m_{11}^{old} - m_{11}^{new}]^2 + [m_{12}^{old} - M_{12}^{new}]^2 + \dots$$

* If computed value is not less than the threshold then goto step 2.

* If computed value is less than the threshold then stop the iteration.

Self Organizing Maps :-

→ Supervised training techniques, where there is a target output for each input pattern & the network learns to produce the required output.

→ Unsupervised training is where the network learn to form their own classifications of the training data without external help.

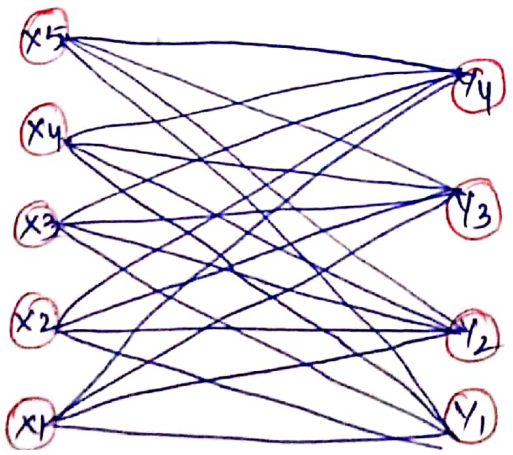
→ One particularly interesting class of unsupervised based on competitive learning in which the o/p neurons compete amongst themselves to be activated & this activated neuron is called a winner-takes.

Scanned with CamScanner

→ Such competition can be implemented by having lateral inhibition connections between the neurons resulting in neurons organizing themselves. Such a network is called as self Organizing Maps (SOM)

→ The goal of SOM is to transform an incoming signal pattern of arbitrary dimension into 1 or 2 dimensional discrete map, ∴ setting up a SOM.

→ We shall concentrate on a particular kind of SOM known as Kohonen network. This is a feed forward structure with a single computational layer arrange in rows & columns. Each neuro is fully connected to all source nodes in the input layer.



Scanned with CamScanner

→ The aim is to learn a "feature map" from the spacially "continuous Input space" to a low dimensionality spatially discrete output space. which is formed by arranging the computational neurons into a grid.

Algorithm :-

Step 1 :- Each node weights are Initialized

Step 2 :- Every node in the network is Examined to Calculate which one's weight are most like the Input vector. The winning node is commonly Known as the "Best matching unit"

Step 3 :- Update the weight after finding the winning neuron

Step 4 :- Training will stop if Input & weight vectors are Identical otherwise goto step - 2

$$\boxed{w^{new} = w^{old} + \alpha\eta(\text{I/P} - \text{weight vector})}$$

Scanned with CamScanner

Genetic Algorithm :-

→ The Genetic Algorithm was developed by John holland & is a model or abstraction of biological Evolution based on charle's Darwin theory of natural selection.

→ charle's Darwin theory of natural selection Explains survival of fittest overtime, change within species leads to replacement of old species by new species as less successful species becomes Extinct.

→ Genetic Algorithms are a class of probabilistic optimization algorithms Inspired by biological Evolution process. These algorithms use "natural selection" & "genetic Inheritance".

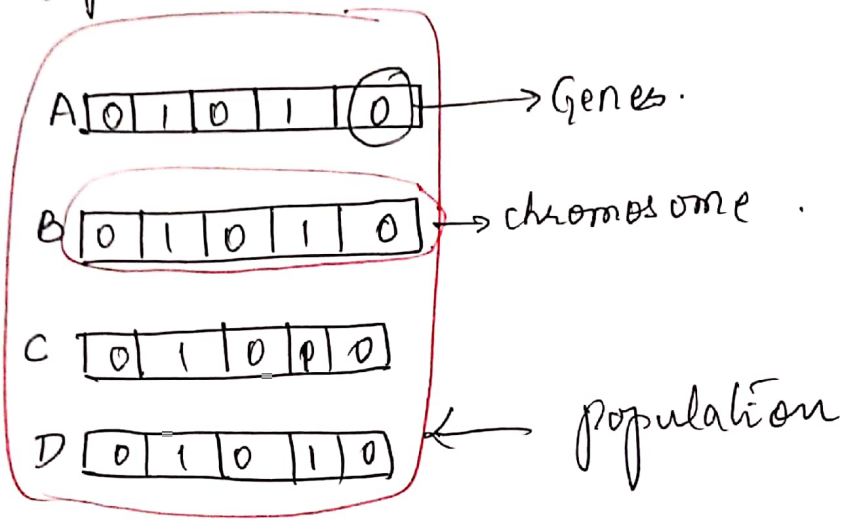
→ Phases of Genetic Algorithm :-

GE Begins with a Large population & gradually reduces it using some heuristics until only the best is left.

Scanned with CamScanner

1) Initial population :-

→ Process starts with a set of individuals known as a population.



note :- Genes → chromosomes are divided into several parts called genes.

chromosome - All genetic information of a species is stored in chromosomes.

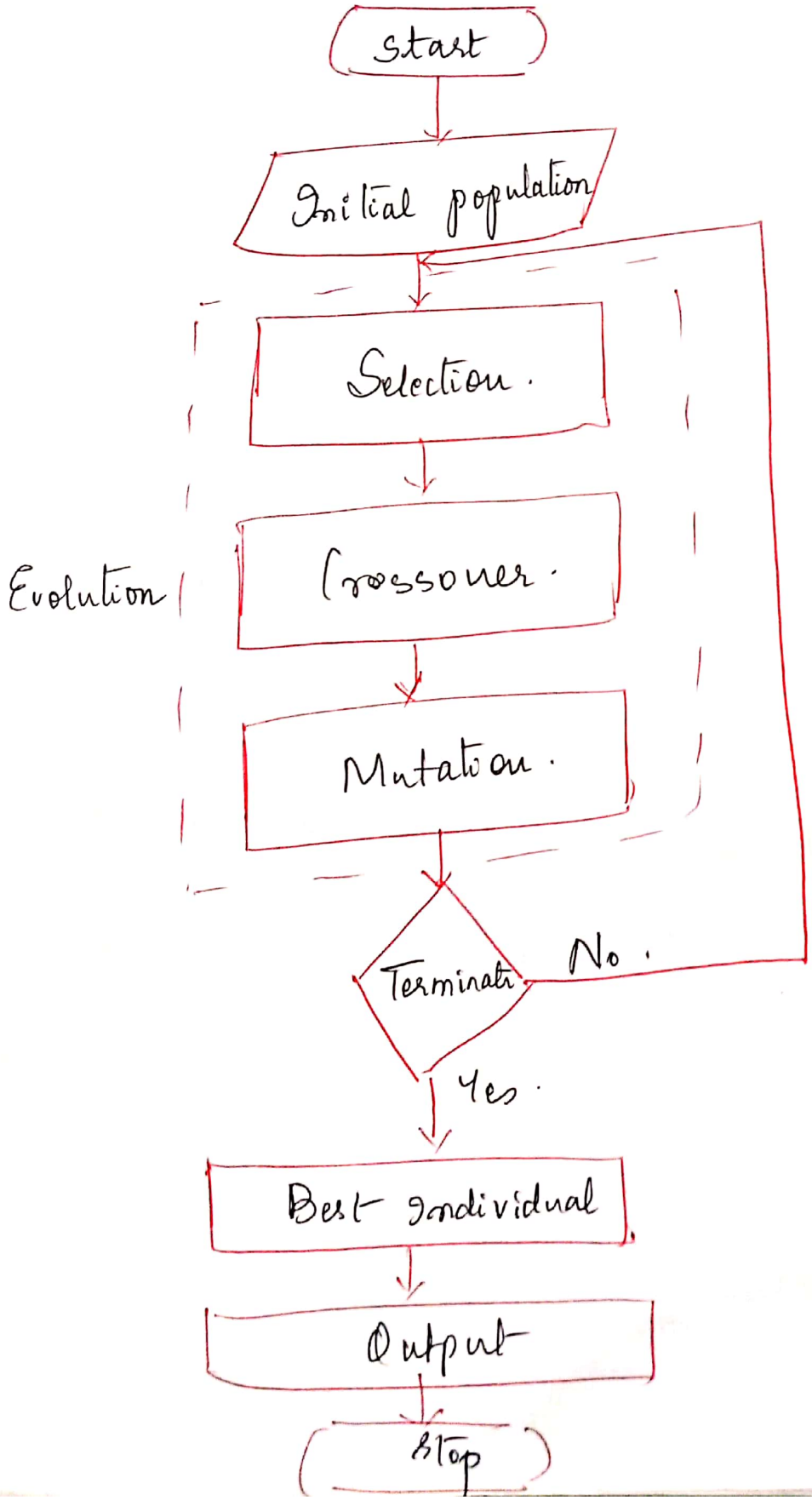
Natural Evolution	Genetic Algorithm
1) Gene	Feature
2) chromosome	string.
3) Genotype	coded string
4) Phenotype	Decoded structure

Scanned with CamScanner

Applications of Genetic Algorithm :-

1) Traveling & shipment routing.

2) Robotics.



Scanned with CamScanner

2) Fitness Function :-

→ Determines how fit an Individual member of the population is.

→ A fitness score is associated with Each Individual & higher the fitness score, higher the probability that Individual will be selected for reproduction to next generation.

3) Selection :-

→ This phase main goal is to find the region where getting the best solution is more.

4) Crossover :-

→ Also called as recombination, where genetic merging takes place to produce new offspring.

5) Convergence.

→ When there is no Improvement in solution quality.

Scanned with CamScanner

Bacterial Foraging Optimization Algorithm (BFO):-

→ Foraging means searching for food & Exploiting food resources. It affects an animal's fitness because it plays an important role in an animal's ability to survive & reproduce.

→ Natural selection tends to Eliminate those poor "foraging strategies" & favor the reproduction of those animals that have successful foraging strategies

→ Such Evolutionary principles have led scientists in the field of "foraging theory" to hypotherize.

→ Here the algorithm is based a bacteria E.coli. ∴ this optimization is named as Bacterial Foraging optimization.

→ "E.coli" [Escherichia coli] is a type of bacteria that normally lives in our Intestines of healthy people & Animals.

Scanned with CamScanner

It helps digest the food we eat.

→ During foraging a real bacteria does two basic Operations.

1) Swim

2) Tumble.

Algorithm :-

1) CHEMOTAXIS

→ a) When a bacterium meets a favorable Environment (rich in nutrients)

i) It will continuosly swim in the same direction

tumble → run → run.

b) when it meets an unfavorable Environment

i) It will tumble, change the direction of swim.

tumble → run → tumble.

2) SWARMING

→ E.coli bacterium has a specific sensing actuation & decision making mechanism.

→ On each move it releases attracts to signal

Scanned with CamScanner

other bacteria towards it.

3) Reproduction :-

→ After calculating fitness value of each bacteria reproduction allows the bacteria to survive & reproduce.

4) Elimination & dispersal

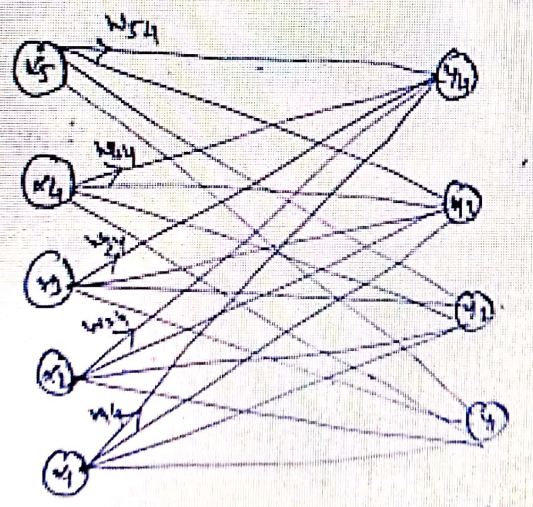
→ Unfit bacterias are Eliminated

Scanned with CamScanner

eg: (M,, ...

KSOM (Kohonen self-organizing Maps)

This has a feed-forward structure with a single computational layer of neurons. ~~connected~~ Each neuron is fully connected to all the source units in the i/p layers.



cluster unit

Input units

step 4 : compute the sum of squared difference between the previous membership value and the current membership value.

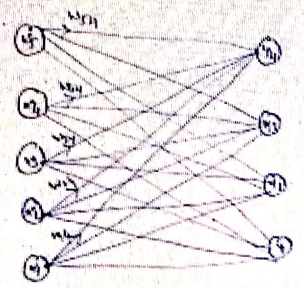
If computed value is not less than the threshold then go to step-2.

If computed value is less than the threshold then stop the iteration.

Eg: $(m_{11}^{old} - m_{11}^{new})^2 + (m_{12}^{old} - m_{12}^{new})^2 + \ldots$

KSOM (Kohonen self-organizing Maps)

This has a feed-forward structure with a single computational layer of neurons. Each neuron is fully connected to all the source units in the I/p layers.



Input units cluster units

The aim is to learn a "feature map" from the spatially "continuous input space", in which our I/p vectors live, to the low dimensional spatially "discrete output space" which is formed by arranging the computational neurons into a grid.

KSOM (Algorithm)

Step 1 : Each node's weights are initialized (all I/p & o/p are normalized).

Step 2 : Every node in the network is examined to calculate which ones' weights are most like the I/p vector. The winning node is commonly known as the "Best matching unit".

Step 3 : Update the weight after find the winner neuron (more importance to the neuron which is closer to winner, less importance to the neuron which is far from winner)

Step 4 : Training will stop if I/p & weight vectors are identical. otherwise go to step-2.

$$w^{new} = w^{old} + \alpha\eta\,(\text{I/p} - \text{weight vector})$$

Particle Swarm Algorithm :-

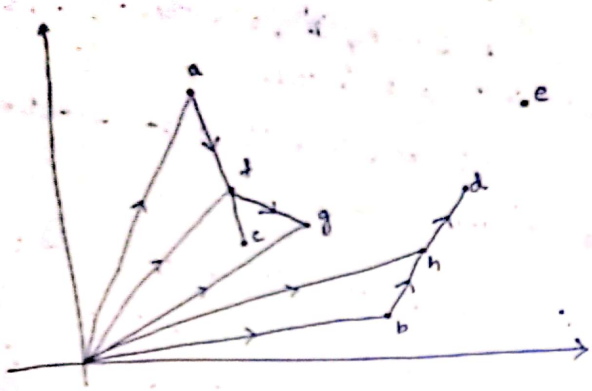


fig: vector representation of pso Algorithm

1) Let the position of the swarms be at 'a' and 'b'. Both are trying to reach 'e'.

2) Let 'a' decided to move 'c' and 'b' is decided to move 'd'.

3) The distance b/w position 'c' and 'e' is greater than the distance b/w 'd' and 'e'.

4) So, 'd' is a common position decided by the a and b

5) The position 'c' is the individual decision taken by 'a'. The position 'd' is the " " " by 'b'. The position 'd' is common position to take by both a & b

Now. let us consider three swarms (a, b, c) are trying to reach the particular destination point 'D'.

A decides 'a', B decides b', c decides c' as next position.

PSO algorithm

step 1 : Initialize the positions a, b, c, d, e

step 2 : Initialize the next positions decided by the individual swarms as a', b', c', d', e'

step 3 : Global decision regarding the next position is computed as follows.

compute $f(a', b, c, d, e)$, $f(a, b', c, d, e)$, $f(a, b, c', d, e)$, $f(a, b, c, d', e)$ and $f(a, b, c, d, e')$. find the minimum among the computed values.

If $f(a', b, c, d, e)$ is minimum among all, the global position decided regarding the next position is 'a'.

If $f(a, b', c, d, e)$ is minimum the global position is 'b'. and so on.

Scanned with CamScanner

Let the selected global position is represented as "global".

step 4 : Net value for 'a' is computed as the linear combination of 'a', (a'-a), (global - a). i.e

weight of individual decision weight of global decision

next a = a + C1 * RAND * (a'-a) + C2 * RAND * (global - a)

next b = b + C1 * RAND * (b'-b) + C2 * RAND * (global - b)

next e = e + C1 * RAND * (e'-e) + C2 * RAND * (global - e)

step 5 : change the current value for a, b, c, d, e as next a, next b, next c, next d, next e.

step 6 : If f(next a, b, c, d, e) is less than f(a', b', c', d', e') then update the value for 'a' as "next a" otherwise 'a' is not change (do this for b', c', d', e' also)

step 7 : Repeat the steps 3 to 6 for much iteration to reach the final decision.

The values C1 & C2 are decided based on the weightage given to individual decision and global decision respectively.

Let Δa(t) is the change in the value for updating the value for 'a' in 't^th' iteration. Then next a at (t+1)^th iteration can be computed using the following formula.

This is considered as the velocity for updating the position of the swarm in every iteration

next a(t+1) = a(t) + Δa(t+1)

where

Δa(t+1) = C1 * rand * (a'-a) + C2 * rand * (Global - a) + w(t) * Δa(t)

w(t) is the weight at t^th iteration. The value for 'w' is adjusted at every iteration as given bellow, where 'Itr' is total no. of iteration used.

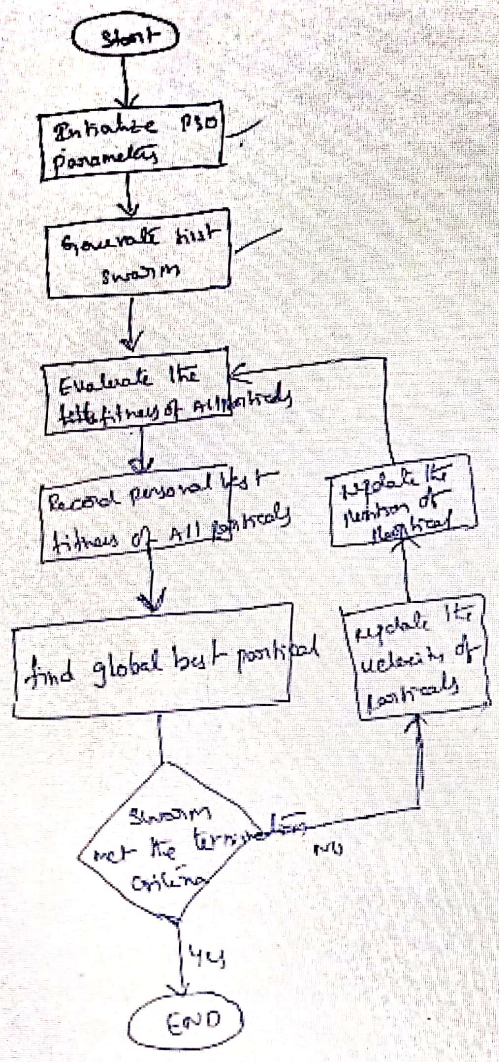
$$w(t+1) = \frac{w(t) - t * w(t)}{Itr}$$

PSO algorithm flow chart

Ant colony optimization

Scanned with CamScanner

PSO algorithm flow chart



Ant colony optimization

Ant colony optimization technique exploit the natural behavior of ants, in finding the shortest path to reach destination from the source.

Algorithm

consider the problem of finding the optimum order in which the numbers from 1 to 8 are arranged so that the cost of the order is maximized. cost of the particle particular order is computed using the two ~~matr~~ matrices A and B as described below

a_1							
a_2	a_3						
a_4	a_5	a_6					
a_7	a_8	a_9	a_{10}				
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}			
a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}		
a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	
a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}

Matrix A

b_1							
b_2	b_3						
b_4	b_5	b_6					
b_7	b_8	b_9	b_{10}				
b_{11}	b_{12}	b_{13}	b_{14}	b_{15}			
b_{16}	b_{17}	b_{18}	b_{19}	b_{20}	b_{21}		
b_{22}	b_{23}	b_{24}	b_{25}	b_{26}	b_{27}	b_{28}	
b_{29}	b_{30}	b_{31}	b_{32}	b_{33}	b_{34}	b_{35}	b_{36}

Matrix B

The cost of the order [3 2 1 5 7 2 4 6 8] is computed as the sum of the products of values obtained from the positions (3,1) or (1,3), (2,2), (3,1), (4,5) or (5,4), (5,7) or (7,5), (6,2), (7,4), (8,6) of matrix A & B

$(a_6 \times b_{21}) + (a_3 \times b_{11}) + (a_6 \times b_{36}) + (a_{14} \times b_{14}) + (a_{21} \times b_{16}) + (a_{15} \times b_{15}) + (a_{28} \times b_{28}) + (a_{36} \times b_{36})$

The problem of finding the optimal sequence using Ant colony optimization heuristic is described below

Step 1: Generate the five sets of orders randomly which are treated as the initial values selected by the 5 ants respectively.

The following are the orders selected by the 5 ants along with the corresponding cost as given below

ANT	ORDERS	COST
ANT 1	4 7 3 5 6 2 1 8	C1
ANT 2	4 2 8 5 7 1 3 6	C2
ANT 3	5 1 3 2 4 7 6 1 8	C3
ANT 4	2 7 4 1 8 6 5 3	C4
ANT 5	8 1 2 4 3 5 7 6	C5

Step 2: Pheromone is the value assigned for ith number for the jth position of the sequence selected by all the ants. Initially all the values of matrix are completely filled up with one. The values of pheromone matrix are updated in every iteration.

The pheromone matrix (n+1) = pheromone matrix(n) + updating pheromone matrix.

	1	2	3	4	5	6	7	8
1	0	$\frac{1}{C_1}+\frac{1}{C_2}+\frac{1}{C_4}$	0	$\frac{1}{C_4}$	0	$\frac{1}{C_2}$	0	0
2	0	$\frac{1}{C_2}$	$\frac{1}{C_5}$	$\frac{1}{C_3}$	0	$1/C_1$	0	$\frac{1}{C_4}$
3	$\frac{1}{C_4}$	0	$\frac{1}{C_1}+\frac{1}{C_3}$	0	$\frac{1}{C_5}$	0	$\frac{1}{C_2}$	0
4	$\frac{1}{C_3}$	0	$\frac{1}{C_4}$	$\frac{1}{C_5}+\frac{1}{C_1}$	$\frac{1}{C_3}$	0	0	0
5	$\frac{1}{C_5}$	0	0	$1/C_2$	0	$\frac{1}{C_3}$	$\frac{1}{C_4}+\frac{1}{C_1}$	0
6	0	0	0	0	$\frac{1}{C_1}$	$\frac{1}{C_4}$	$\frac{1}{C_3}$	$\frac{1}{C_1}+\frac{1}{C_2}$
7	$\frac{1}{C_1}$	$\frac{1}{C_4}$	0	0	$1/C_2$	$\frac{1}{C_3}$	$\frac{1}{C_5}$	0
8	$1/C_1$	0	$\frac{1}{C_2}$	0	$\frac{1}{C_4}$	0	0	$\frac{1}{C_1}+$

Step 3: Next set of orders selected by the ants are as described below

Eg: The random position of 8 2 3 6 4 1 5 (say)

Let the order decided by the ant 1 be sequence

at [x1, x2, x3, x4, x5, x6, x7, x8]

* The value for the x7 in the order as mentioned above is selected that as the first number of the random generated position sequence is 7.

* To select the value for x7, 7th column of

* To select the value for v7, 7th column of the pheromone matrix is used as bellow,

column 7 = $[0 \quad 0 \quad \frac{1}{c_2} \quad 0 \quad \frac{1}{c_4}+\frac{1}{c_4} \quad \frac{1}{c_3} \quad \frac{1}{c_5} \quad 0]$

* now calculate the normalized value of column 7. This is also called as the probability vector used for selecting the value of v7 in the order

$S = [\frac{1}{c_2}+\frac{1}{c_4}+\frac{1}{c_4}+\frac{1}{c_3}+1/c_5]$ = value

Normalized vector 7 = $\frac{\text{column 7}}{S} = [\frac{0}{S} \quad \frac{0}{S} \quad \frac{1/c_2}{S} \quad \frac{0}{S} \quad \frac{(c_4+\frac{1}{c_4})}{S}$

or

$= [p_1 \; p_2 \; p_3 \; p_4 \; p_5 \; p_6 \; p_7 \; p_8] \; \frac{\frac{1}{c_2}}{S} \rightarrow \frac{1/c_5}{S} \quad \frac{0}{S})$

* find the lowest value position, for example

If p_5 is the lowest value the 5 is assigned to v7.

Note: position no. '5' is not used again.

* The next value in position sequence is 8, so column '8' is selected from pheromone matrix, and select the normalized value

column 8 = $[0 \quad 1/c_4 \quad 0 \quad 0 \quad \times \quad \frac{1}{c_2}+\frac{1}{c_3} \quad 0 \quad \frac{1}{c_4}+\frac{1}{c_3}]$

$\therefore S = [\frac{1}{c_4}+(\frac{1}{c_2}+\frac{1}{c_3})+(1/c_4+\frac{1}{c_3})]$

Normalized vector 8 = $\frac{\text{column 8}}{S} = (a_1 \; a_2 \; a_3 \; a_4 \; a_5 \; a_6 \; a_7)$

find the lowest value in normalized vector '8' and replaced with v8. Among 2 of a_3 is lowest then number '3' is assigned to v8.

* This process is repeated to obtain the order selected by the ~~ant~~ Ant 2, Ant 3, Ant 4, Ant 5.

* This is called one Iteration.

step 4: updating the ~~pre~~ pheromone matrix for the second iteration is computed as described in step 2. This is called updating pheromone matrix (2)

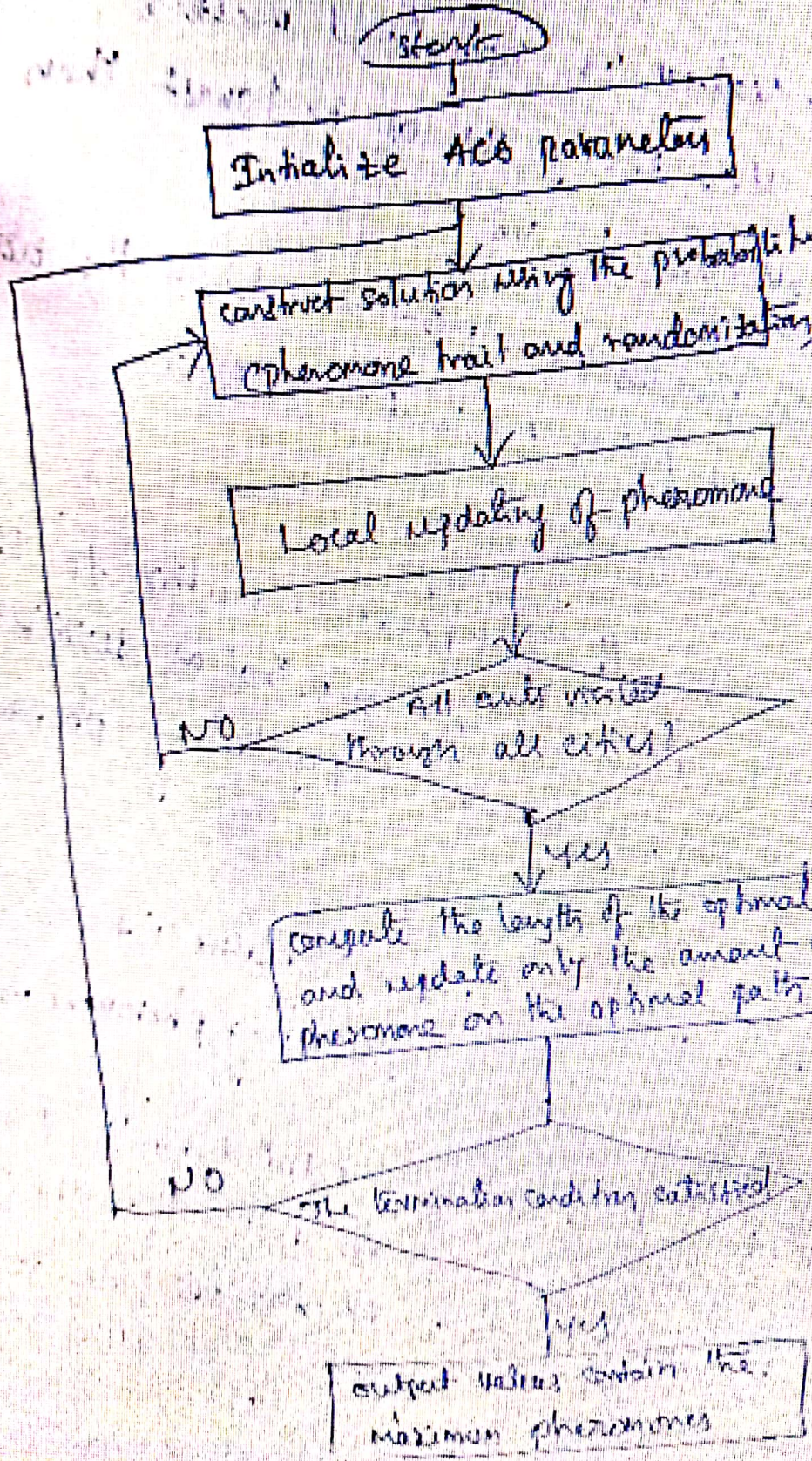
$\therefore$ pheromone matrix(2) = pheromone matrix (1) + updating pheromone matrix

step 5: Next set the orders selected by the five ants are computed as described in the step 3

step 6: step 3 ~~and~~ to step 5 is ~~re~~ repeated for the particular iterations. Thus best set of orders are obtained using Ant colony algorithm

Scanned with CamScanner

Note: Best among the best set of orders selected by the five ants in the last iteration is selected as the global best order.



Scanned with CamScanner