UNIT-4. Genetic Algorithm (G.A).

-> Introduction.
-> Genetic Algorithm.
-> Fitness Value.
-> Cross over.
-> Mutation.
-> Variants in GA.
-> Applications of GA.
-> Evaluationary Programming.
-> Ant colony Algorithm.
-> Particle swarm Algorithm.
-> Bees optimization Algorithm.

* Introduction:- Genetic Algorithms are developed by John Hollad in the year 1965. The first research article published in the year 1975. The GA's are inspired by Darwinian's theory of Survival of Fittest. It belongs to family of Evolutionary algorithms.

Evolutionary algorithms are one which make use of biological & physical behaviour of humans, creatures, animals, etc.. Biological behaviours such as the behaviour of the biological neuron is used to develop artificial neuron, the genetics of human beings is used to develop genetic algorithms, etc....

Physical behaviour movement of particle swarm is used to develop particle swarm algorithm, the movement of bees is used to develop Bees optimization algorithm, etc...

* Genetic algorithm is one which makes use of genetics and evaluations to find solution to complex problems. We can say genetic algorithm is a

optimization algorithm used to find optimized or best solutions out of the available solutions.

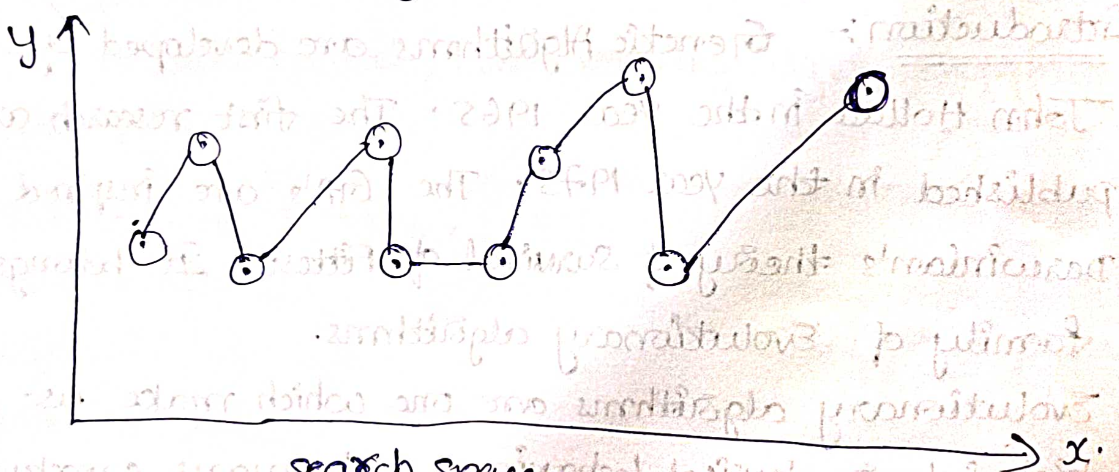
It make use of biological behaviours like genetics and Evaluations.

→ Genetics:- A branch of biology that deals with the study of genes, genes variation and heridittary.

→ Evaluation:- Evaluation is the process by which the genes of the population varies over the generations.

* Search Space:-

The search space is the space of all possible solutions. A typical search space is as shown in the below figure, and x and y axis are purely depends on the problem considered



search space.

- Each solution will be marked with fitness value. Looking for best solution is nothing but looking for maximal or minimal. The problem is that search space is complicated, contains a large number of solutions and doesnot know where to look for solution or where to start from and this is where genetic algorithm is useful.

* Genetic algorithms take huge and potential search spaces and navigating them looking for optimal solutions which we might not find in a lifetime.

* Genetic Algorithm :- The basic step in Genetic Algorithm is to find the objective or fitness function. Then it needs to identify all possible solutions which is called as "Creation of population". Once the population is created, it needs to calculate fitness value of each solution. Once the fitness value is calculated, the genetic operators selection, crossover, Mutation will be applied on the population until the condition is satisfied.

The steps involved in genetic algorithm are as given below :-

Step-1 :- Creation of population.

Step-2 :- Evaluate fitness value of each solution in the population.

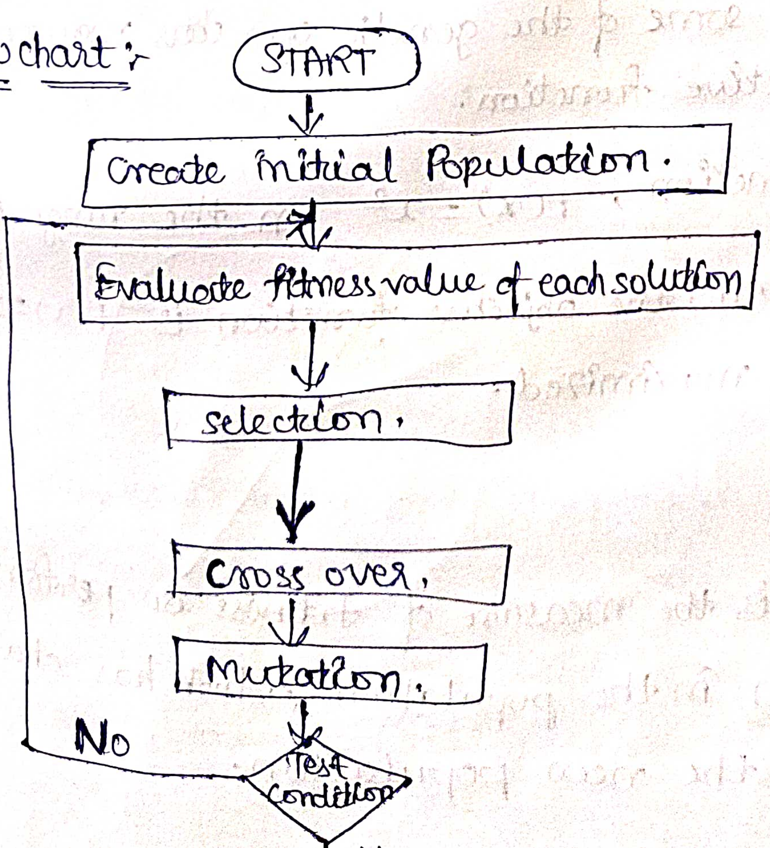
Step-3 :- Apply selection operator.

Step-4 :- Apply crossover operator.

Step-5 :- Apply Mutation operator.

Step-6 :- Test condition : If the condition is satisfied, the loop will be terminated otherwise the steps from 2 to 5 will be repeated until condition is satisfied.

→ Flow chart :-



* Fitness Function and Fitness value :-

- The fitness function is obtained from objective function given in the problem.
- The optimization algorithms are used to maximize (or) minimise the given fitness function.
- Genetic Algorithm is also an optimization algorithm but used for maximization problems only.
- If we wants to apply GA for minimization problems, it should be converted into maximization problem and then we have to apply Genetic Algorithm. It can be mathematically represented as :-

$$F(x) = f(x) \text{ for maximization problems.}$$

$$F(x) = \frac{1}{f(x)} \text{ for minimization problem, If } f(x) \neq 0.$$

$$F(x) = \frac{1}{1+f(x)} \text{ for minimization problem, If } f(x) = 0.$$

* To convert the minimization problem into maximization problem, usually before fitness function, negative sign will be considered but some of the genetic operators may not accept negative objective function.

- Eg:- Maximize a function, $f(x) = x^2$ in the range 0 to 31.

In the above problem, the objective function is $f(x) = x^2$ which has to be maximized.

→ Fitness Value :-

Fitness value is the measure of fitness or performance of a given solution in the population, which has chances to reproduce in the new population.

Let us calculate fitness value of the given problem by considering number of solutions N = 4. Take $x = 13, 8, 24$ and 19 randomly. The fitness values of given fitness function can be calculated as follows:

S.NO	x-value	String (chromosome / Parents).	Fitness value. $F(x) = x^2$.
1.	13.	01101.	169.
2.	8.	01000.	64.
3.	24.	11000.	576.
4.	19.	10011.	361.

★ Encoding :- Encoding is a technique to find strings or chromosome from a given solution. There is so many encoding techniques are there but due to simplicity, Binary encoding will be used.

★ In the above population, the solution $x = 24$ has got highest fitness value which has more chances to reproduce in the new population.

★ Genetic operators :-

The following are the three genetic operators applied on the population (or) mating pool iteratively to get the best or optimum solution among the available solutions.

1.) Reproduction / selection operator.
2.) Crossover operator.
3.) Mutation operator.

1.) Reproduction (or) Selection Operator:-

- Reproduction / selection operator is the first operator applied on the population.
- It is used to select the best solutions from the population based on fitness values.
- As per the Darwinian's Theory of survival of fittest, parent strings are selected from the population to cross over and to produce child (or) offspring.
- The different reproduction (or) selection operators are:-

1.) Roulette-wheel selection.
2.) Boltzmann selection.
3.) Tournament selection.
4.) Rank ordering.
5.) Steady state selection.

The Roulette wheel selection is the simple and basic method used as selection operator.

In this Roulette wheel selection, the best solutions are selected from the population with a probability proportional to the fitness value. The probability of each solution can be calculated using the following formula:

$$\boxed{P_i = \frac{F_i}{\sum_{j=1}^{n} F_j}}$$

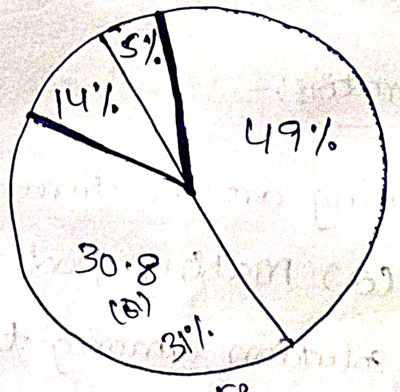


Fig-1.

Problem:- Maximise the $f(x) = x^2$

SNO	x value	String	$f(x) = x^2$	Pi
1	13	01101	$13^2 = 169$	0·14
2	8	01000	64	0·05
3	24	11000	576	0·49
4	19	10011	361	0·308 (or) 0·31

To implement the roulette wheel selection, a roulette wheel is considered with a circumference.

The roulette wheel is divided into n-number of spuns, each part is represented with the Probability of survival in the next new population. For the above problem, roulette wheel is designed as above Fig-1.

From the above roulette-wheel the probability of survival of a solution in the next new population is selected based on area occupied in the roulette-wheel.

*(2.) Crossover operator :-

After the selection / reproduction process is over, the population is enriched with better individuals.

The second operator applied on the population is cross over operator. The crossover operator is applied with a hope that better individual strings will be formed. The following are the different crossover operators:

(i.) Single point crossover.
(ii.) Two point crossover.
(iii.) Multipoint crossover.
(iv.) Uniform crossover.
(v.) Matrix crossover.

(i.) Single Point Crossover :- In a single point / single site crossover, a cross site is randomly selected along the length of the mated strings and bits next to the cross-site are exchanged as shown below.

Eg:- → cross-site

1001 1|100 → Parent-1

1101 1|011 → parent-2

Before swapping

10011011 → child-1

11011100 → child-2

After swapping.

- If an appropriate cross site is chosen, better children can be obtained by combining parents, since the knowledge of appropriate site is not known, it is randomly selected. Anyway, because of the crossing of parent strings, better children will be produced.

(ii.) Two point crossover :-

In a two point crossover, two random crosssites are selected, the bits between the two crosssites will be swapped to produce the child strings as shown in the below figure.

Eg:- crosssite-1 → crosssite-2

1 1 0 | 0 1 0 | 1 1 → Parent-1

0 1 0 | 1 1 1 | 0 0 → Parent-2

Before crossover

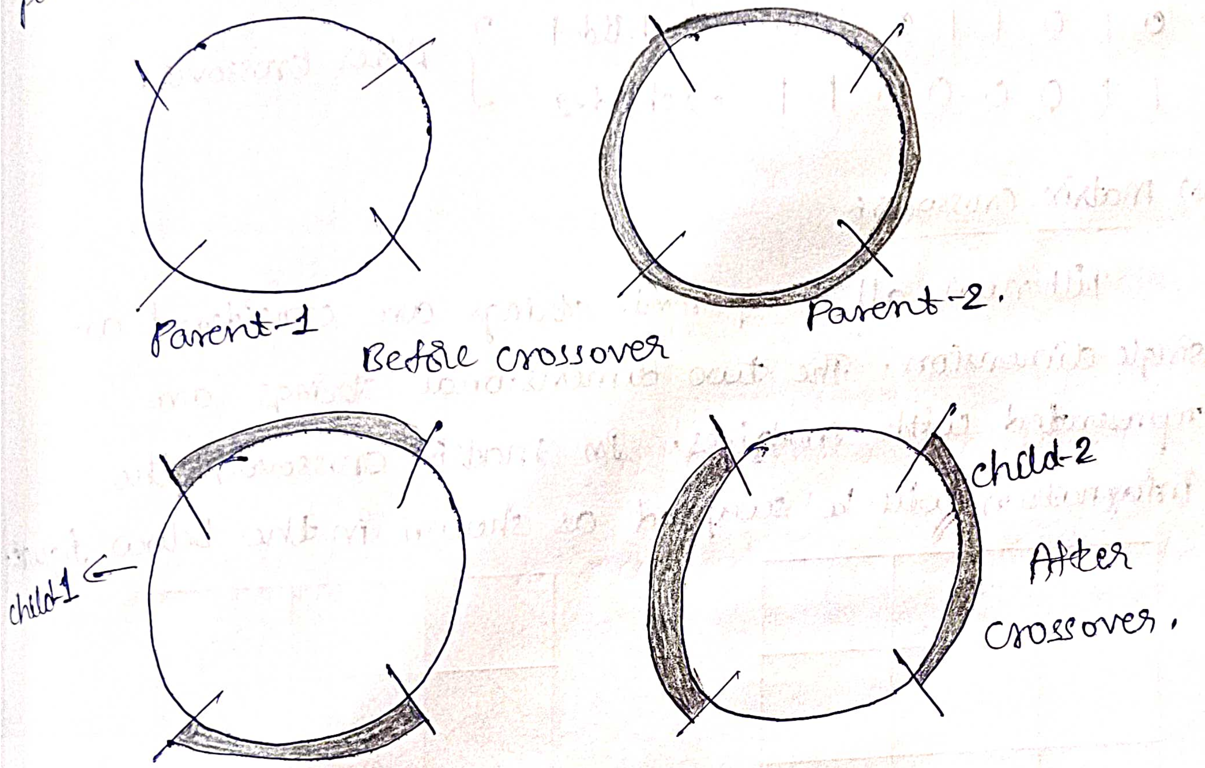
1 1 0 1 1 1 1 1 → child-1

0 1 0 0 1 0 0 0 → child-2

After crossover.

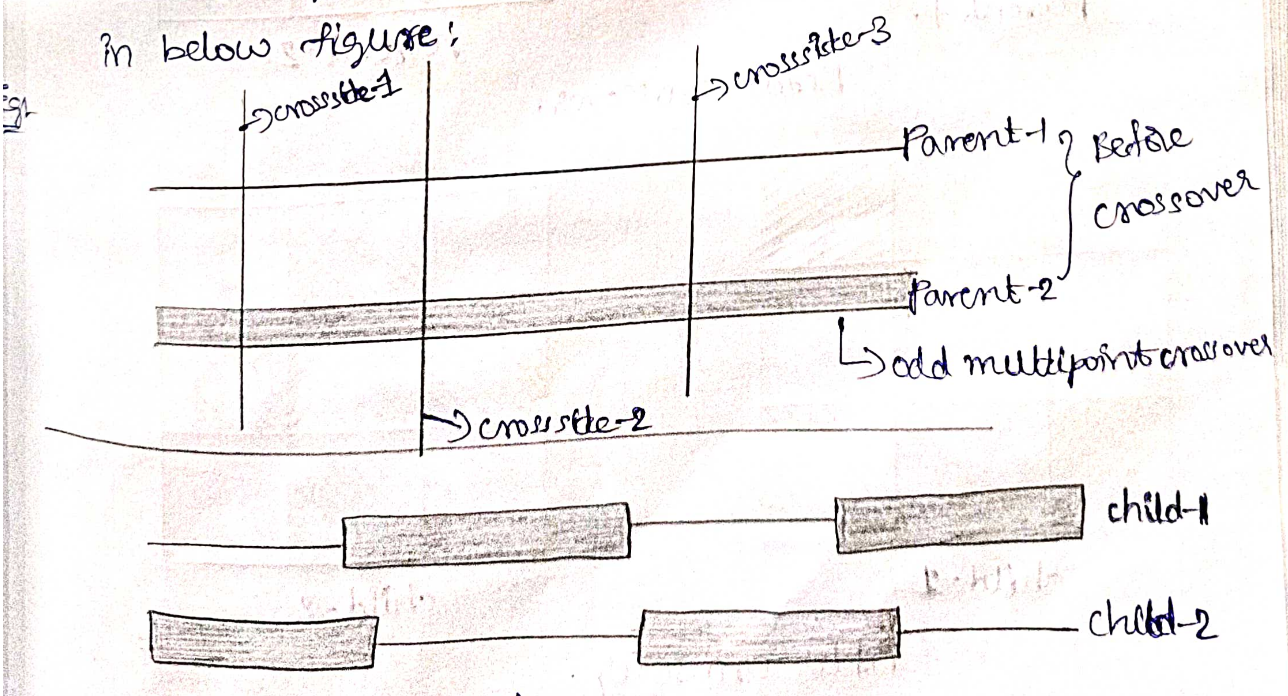
(iii) Multi point crossover :- In a multi point crossover, there is two cases, one is even number of cross sites and the second is odd number of cross sites.

In case of even number of cross sites, the string is treated as a Ring with no beginning & ending. The cross sites are selected around the circle uniformly at random. Now the information between alternate pairs of sites is interchanged as shown in below figure.



- The Example of odd multi point crossover is as shown in below figure:

eg:



After crossover

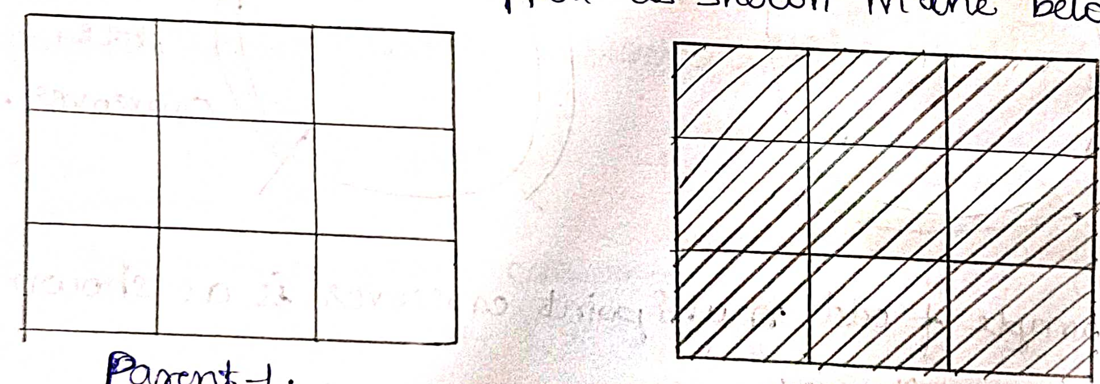
(iv) Uniform crossover:- An extreme of multi point crossover is the uniform crossover operator. In an uniform crossover operator, each bit from either parent strings can be selected randomly and swapped.

Eg:-

1100 1011	→ Parent-1	}	Before crossover.
0101 0010	→ Parent-2		
01011010	→ child-1	}	After crossover.
11000011	→ child-2		

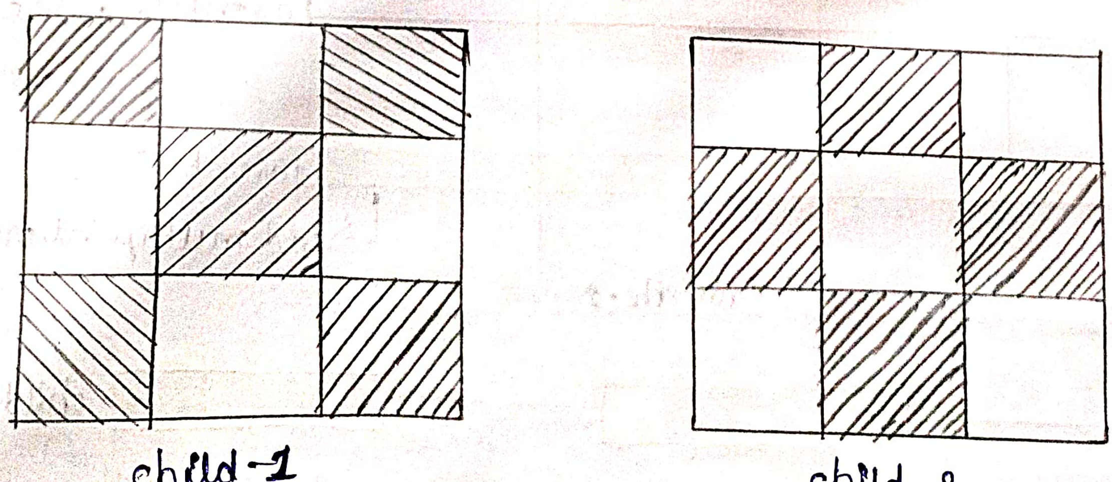
(v) Matrix Crossover:-

Till now, all the parent strings are considered are single dimension. The two dimensional strings are represented with matrices. In matrix crossover, the information will be swapped as shown in the below figure.



Parent-1. Parent-2.

Before crossover.



child-1 child-2

After crossover.

(3.) Mutation Operator :-

- After cross over operator, Mutation operator will be applied on the population.
- It is the third operator applied on the population.
- Mutation is the only operator which can flip either 0 to 1 (or) 1 to 0.

Eg-1:- 0 1 0 1 → 5

2^3 2^2 2^1 2^0 → before flipping

0 4 0 1.

$f(x) = x^2$.

$f(5) = 25$.

1 1 0 1 → 13.

→ $f(13) = 169$.

Using mutation operator, any bit can be randomly selected if that bit is '0' that can be flipped to '1'; if that bit is '1' that can be flipped to '0'.

After flipping 1101, x value = 13 and the fitness value will be 169. Like this mutation operator improves the fitness value of the string such that there is more possibility for that string to survive in next population.

* Variants of GA and Applications of GA :-

- Variants of GA :-

- The fundamental operators used in Genetic Algorithm is selection / Reproduction, crossover and Mutation.
- Based on the type of logic used in the Genetic operators, there are following Variants of GA:

1.) Real coded GA.
2.) Binary coded GA.
3.) Sawtooth GA.
4.) Micro GA.
5.) Improved GA.
6.) Differential Evaluation GA.

Applications of GA:- The following are the different applications of GA:

1.) Optimization :- GA's are mostly used for optimization problems where we have to maximise & minimise an objective function under constraints.

2.) Neural Networks:- The GA is used to find the optimum weights in neural networks. It is mostly used in Feedback (or) Recurrent Artificial Neural Networks.

3.) Image Processing:- GA's are used for different digital image processing tasks.

4.) DNA Analysis:- GA's are used to identify the structure of DNA efficiently.

5.) Travelling salesman Problem:- In this, GA's are used to find the solution with shortest path.

6.) Scheduling:- In scheduling, GA's are used to plan the actions & activities in a most optimum way.

7.) Machine learning:- GA's are mainly used in GAML (Genetic based machine learning). In machine learning, the data to be feeded to the machine is optimised before training.

8.) Parameters design of Aircraft:-
GA's are also used to design parameters of Aircraft by varying of the parameters automatically based on the requirements.

9.) Robotics:- GA's are used to create learning Robots which will behave as a human.

10.) Academic (or) Finance:- GA's are used to characterise various academic models

* Evolutionary Programming or Algorithm:-

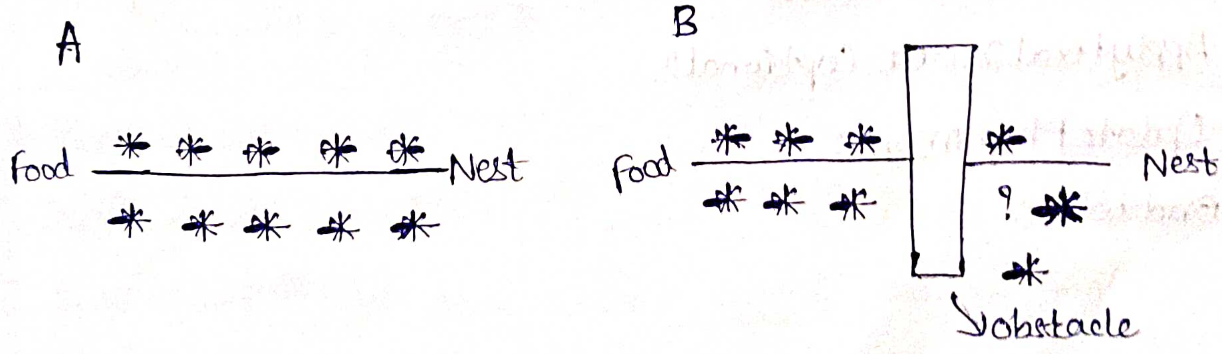
The evolutionary Programming inspired by biological and physical behaviours.

Biological behaviours are nothing but biological neurons of the humans, genes of humans from which Artificial neural networks and the Genetic Algorithms were developed.

Physical behaviours are nothing but movement of ants, movement of swarm and the movement of the bees. Based on which Ants Colony Algorithm, Particle Swarm Algorithm and Artificial Bee Algorithm developed.

* Ant Colony Algorithm - (ACO):-

The ant colony optimization algorithm (ACO) is a probabilistic technique for addressing computational problems that can be reduced to finding good pathways through graphs in computer science and operations research. The concept of ant colony optimization (ACO) is based on the efficient and effective way that ant colonies find food. At the heart, this behavior is the ants' use of chemical pheromone trails for indirect communication with one another, which allows them to locate efficient routes between their nest and food sources.



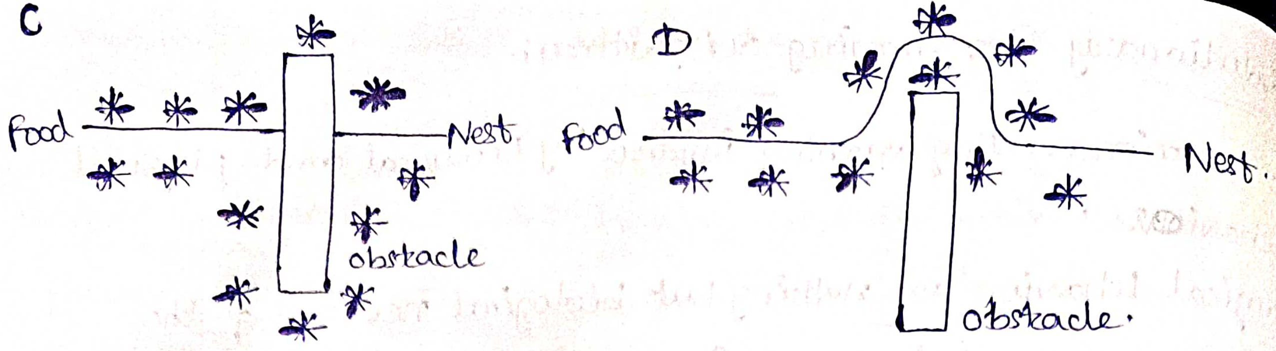


Fig 1.

A: Ants in a pheromone trail between nest and food,

B: an obstacle interrupts the trail.

C: Ants find two paths to go around the obstacle.

→ 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.

Three important steps in the algorithm is :-

(i.) Construct Ant Solutions.

(ii.) Apply Local Search.

(iii.) Update Pheromones.

→ Algorithm-1: The Ant Colony optimization Metaheuristic.

Set parameters, initialize pheromone trails
while termination condition not met do
    Construct Ant Solutions
    Apply Local Search (optional)
    Update Pheromones
End while

* Partical Swarm Optimization (PSO):-

- PSO is stochastic optimization technique proposed by Kennedy and Eberhart (1995).
- A population based search method with position of particle is representing solution and swarm of particles as searching agent.
- PSO is a robust evolutionary optimization technique based on the movement and intelligence of swarms.
- PSO find the minimum value for the function.
- The idea is similar to bird flocks searching for food.

→ Bird = a particle, Food = a solution

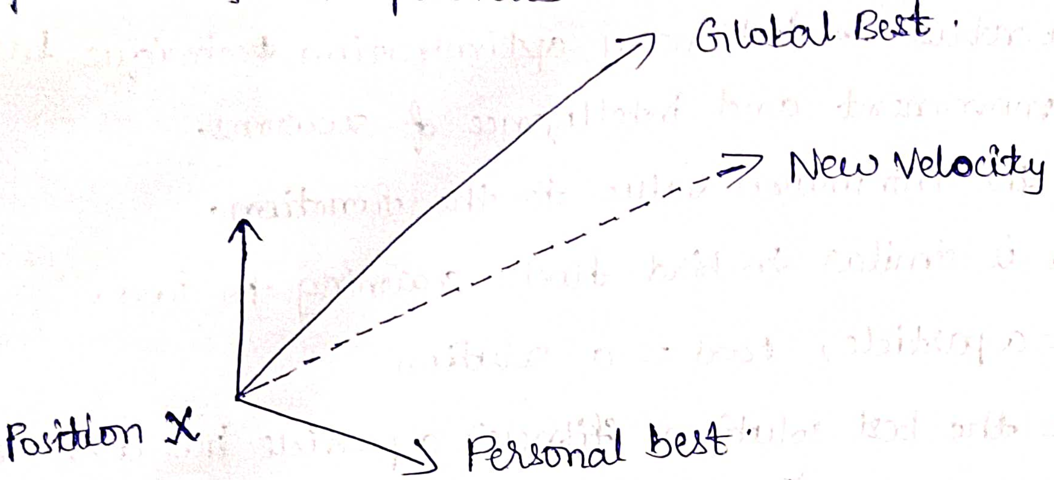
→ pbest = the best solution (fitness) a particle has achieved so far

→ gbest = the global best solution of all particles within the swarm.

- pbest: the best solution achieved so far by that particles.
- gbest: the best value obtained so far by any particle in the neighborhood of that particle.

- The basic concept of PSO lies in accelerating each particle toward its pbest and the gbest locations, with a random weighted acceleration at each time.
- PSO uses a number of agents, i.e., particles, that constitute a swarm flying in the search space looking for the best solution.
- Each particle is treated as a point (candidate solution) in a N-dimensional space which adjusts its "flying" according to its own flying experience as well as the flying experience of other particles.

- PSO uses a number of agents, i.e., particles, that constitute a swarm flying in the search space looking for the best solution.
- Each particle is treated as a point (candidate solution) in a N-dimensional space which adjusts its "flying" according to its own flying experience as well as the flying experience of other particles.



→ The PSO algorithm pseudocode as following:-

Input:- Randomly initialized position and velocity of Particle $X_i(0)$ and $V_i(0)$.

Output:- Position of the approximate global minimum X^*.

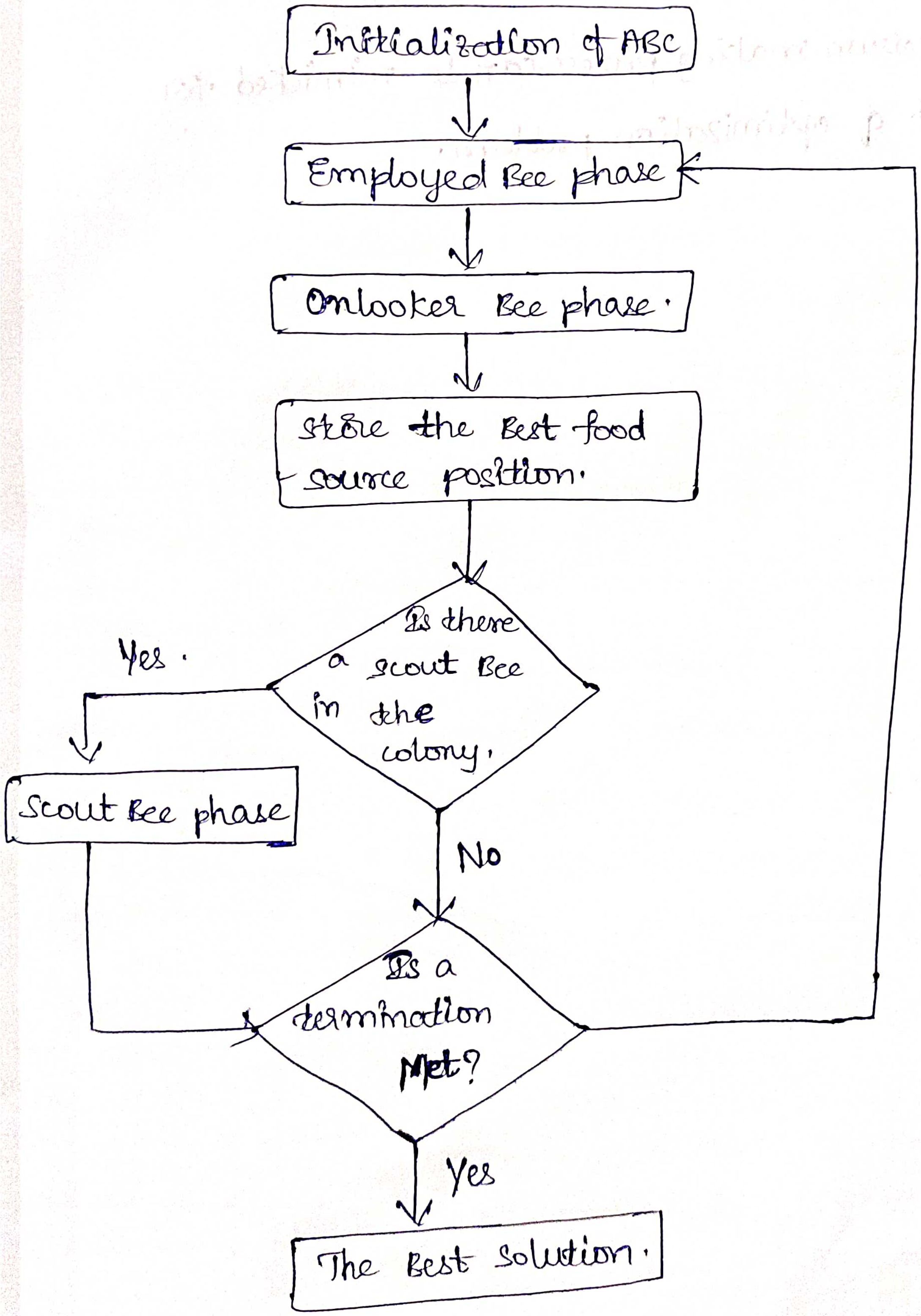
```
while terminating condition is not reached do
    for i = 1 to number of particles do
        Calculate the fitness function f
        Update personal best and global best of each particle
        Update velocity of the particle
        Update the position of the particle
    end for
end while
```

* Artificial Bee Colony Algorithm (ABC):-

Bee uses waggle dance to communicate. Waggle dance performed by Scout bees to inform other foraging bees about net site of food resource.

Scout Bee : The navigator

Foraging Bee: The collector of food form.



Honeybee search for the best nest site between many sites with taking care of both speed and accuracy.

This is analogues to finding the optimal solution to an optimization process. The group decision making process used by bees for searching out the best food resources among various solutions is a robust example of swarm based decision method.

This group decision making process can be mimicked for finding out of optimization problem.