## → "Genetic-Algorithms"

↳ Genetic-algorithms belong to a class of evolutionary algorithms

↳ They are adaptive, heuristic search algorithms.

↳ These algorithms are adaptive to the number and type of parameters that we are giving, or the type of parameters,

↳ These algorithms adapt themselves with respect to the change in environment.

↳ They are based on Genes & natural selection. ↳ (based on evolution)
(Genotype & Phenotype)

↳ Genetic algorithms are used for creating high-quality solution for optimization problem.

↳ Individual :⇒ In terms of genetic-algorithms, "individual" is considered as a possible solution for a given problem.

↳ Population :⇒ Groups of individuals is called a population

## → "Operators of Genetic-Algorithm"

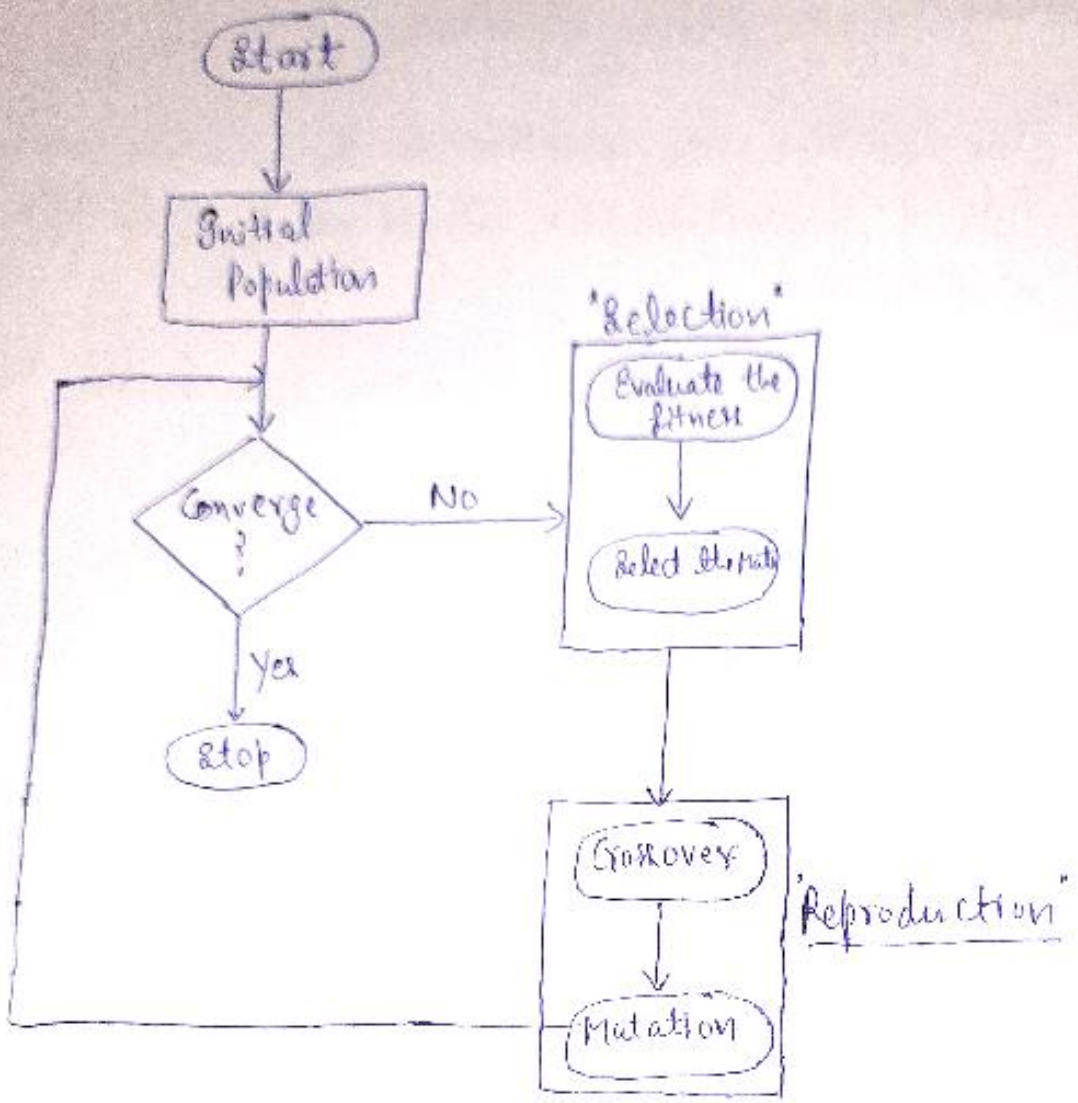
1) "Selection" :⇒ To select the fittest individual.

2) "Crossover" :⇒ To generate better offsprings (solutions).

3) "mutation" :⇒ To alter some of the genes.

4) "Encoding" :⇒ To apply encoding schemes on initial population.

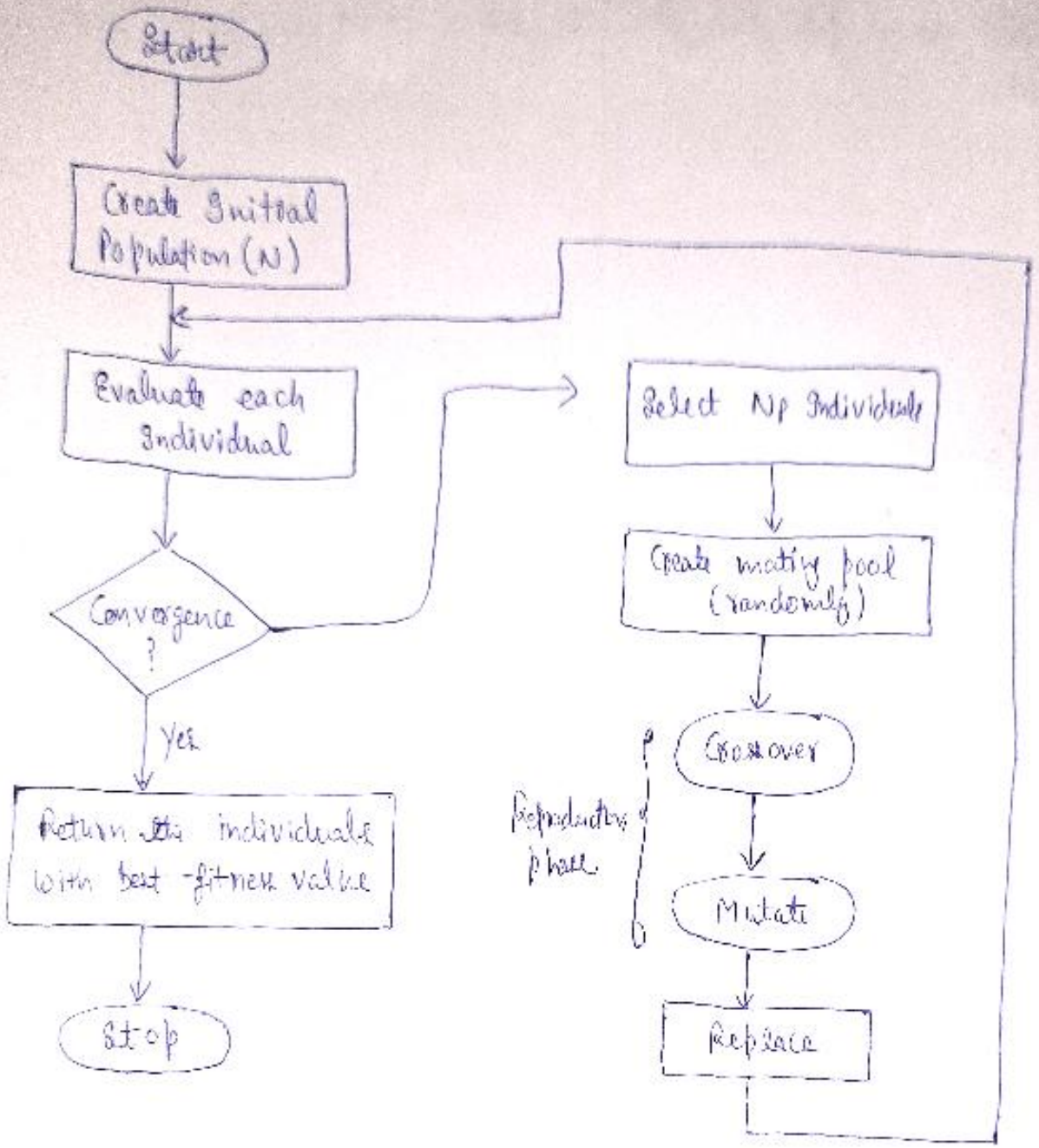
Genetic Algorithm is an iterative process =>




It is an iterative procedure to find out the optimal solution.

P.T.O

→ 'Simple Genetic Algorithm Flowchart' is




→ Difference between Genetic Algorithm & Traditional Algorithm.

| "Genetic - Algorithm" | "Traditional - Algorithm" |
|---|---|
| (i) Genetics & Natural selection to solve optimization problem | (i) Step by step procedure to solve a given problem. |
| (ii) More advanced. | (ii) Not as advanced. |
| (iii) It is used in fields such as ML, AI | (iii) Used in fields such as programming, mathematics |

PTO

(iv) Probabilistic Rules

(v) Search on a population of points.

(iv) Fully deterministic Rules.

(v) Search on a single point.

---

→ Convergence test / Termination Condition :→

1) Manual checking.
2) Solution found that satisfy objective criteria.
3) Fixed number of generation.
4) Budget limit reached (Resources exhausted).

---

→ Encoding :→

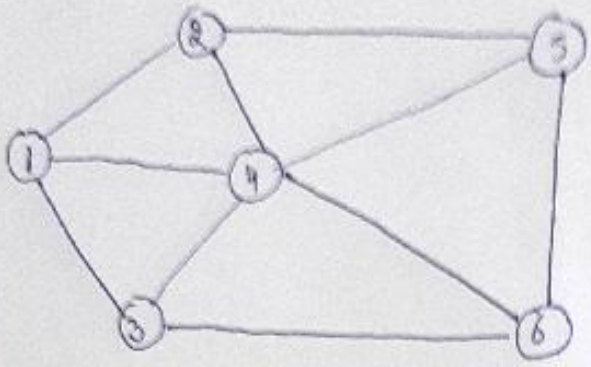
The first step in genetic algorithms is to create an initial population.

To create an initial population, we can use these "encoding operations".

Binary encoding is one such example of encoding.

→ "Fitness evaluation :- Fitness function or objective function"

Eg 1:→ TSP




| | | sum |
|---|---|---|
| Path1: | 1 2 5 6 4 3 1 | (18) |
| Path2: | 1 2 5 4 6 3 1 | (20) |
| Path3: | 1 2 4 5 6 3 1 | (12) |
| Path4: | 1 2 5 6 3 4 1 | (17) |

Eg.2:→ 0,1 Knapsack

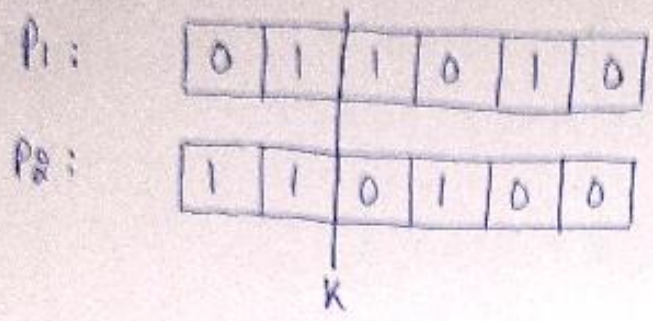
(A) - 5 (100)
(B) - 10 (150)
(C) - 15 (200)
M = 25

| | A B C | W | P | |
|---|---|---|---|---|
| (A) - | 1 0 0 | 5 | 100 | ✓ |
| (AB) - | 1 1 0 | 15 | 250 | ✓ |
| (AC) - | 1 0 1 | 20 | 300 | ✓ |
| (BC) - | 0 1 1 | 25 | 350 | ✓ |
| (ABC) - | 1 1 1 | 30 | 450 | X |

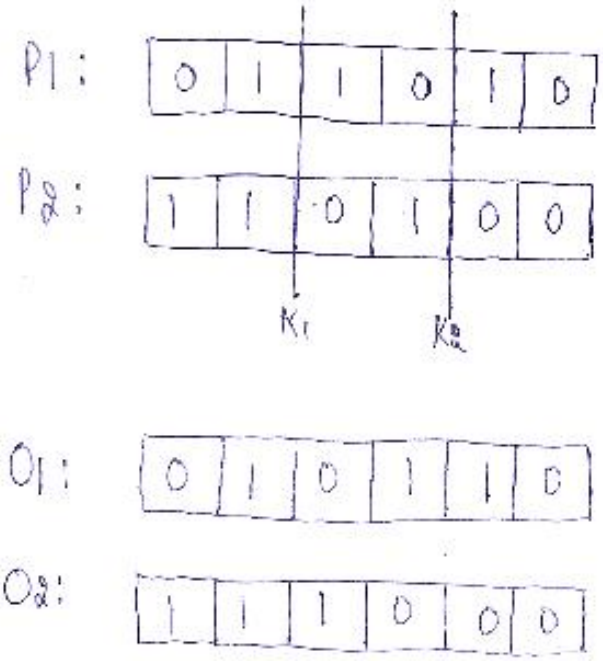
Best possible solution.

→ 'Crossover' :- To generate offspring by combining genetic materials of parents.

1> Single point crossover :-

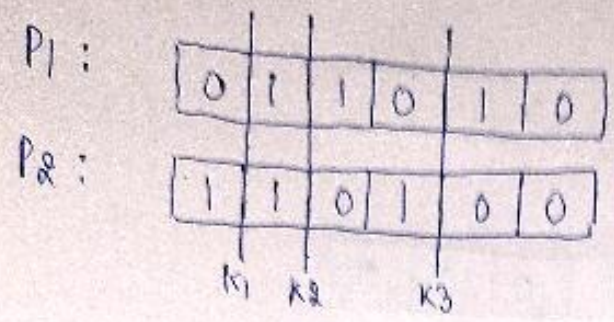


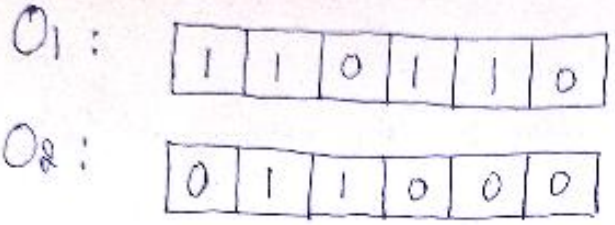

2> Two point crossover (Binary crossover) :-
Keep extremes as it is.




P.T.O

3> Multipoint crossover (swap alternate)




---

→ Mutation in Genetic Algorithms :-

Here we do not use genetic material of parents to create new offsprings but instead we are making alterations in the genetic materials of offsprings only.

First technique to perform mutation is flipping & for doing that, we require something known as "mutation probability"

↙

It tells us whether to flip a particular chromosome bit of an offspring or not.

↓

"modify or not"

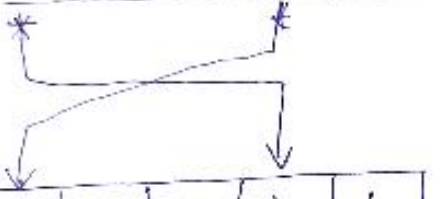
Offspring (O) :- | 0 | 1 | 1 | 0 | 0 | 1 |

MP (MP) :- | 0 | 1 | 0 | 0 | 0 | 1 |

Mutated offspring (MO) :- | 0 | 0 | 1 | 0 | 0 | 0 |

Second Technique is interchanging :- Randomly select two chromosome bits from the offspring & swap them

Offspring :- | 0 | 1 | 1 | 0 | 0 | 1 |



MO :- | 0 | 0 | 1 | 0 | 1 | 1 |