GENETIC ALGORITHM
Unit-5

Genetic Algorithm: Genetic Algo is a search base optimization techniques based on the principles of Genetic & Natural selection.

* It is used to find optimal or near optimal solution to difficult problems which otherwise would take a life time to solve.

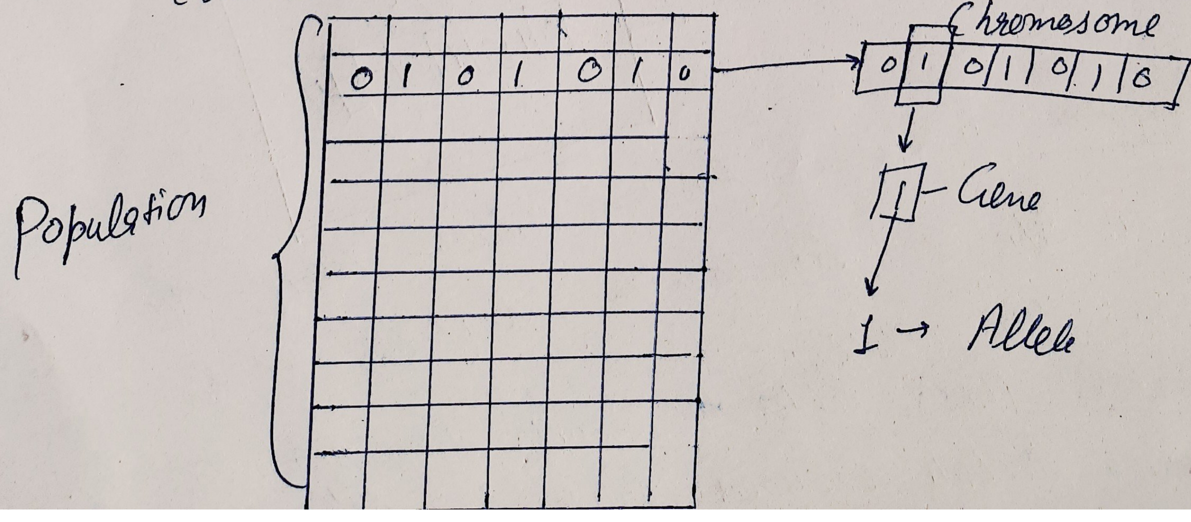
BASIC TERMINOLOGY:

POPULATION: It is a subset of all the possible encoding solution to the given problem.

CHROMOSOMES: It is one such solution to the given problem

GENE: It is one element position of a Chromosome

Allele: It is the value a gene take for a particular chromosomes



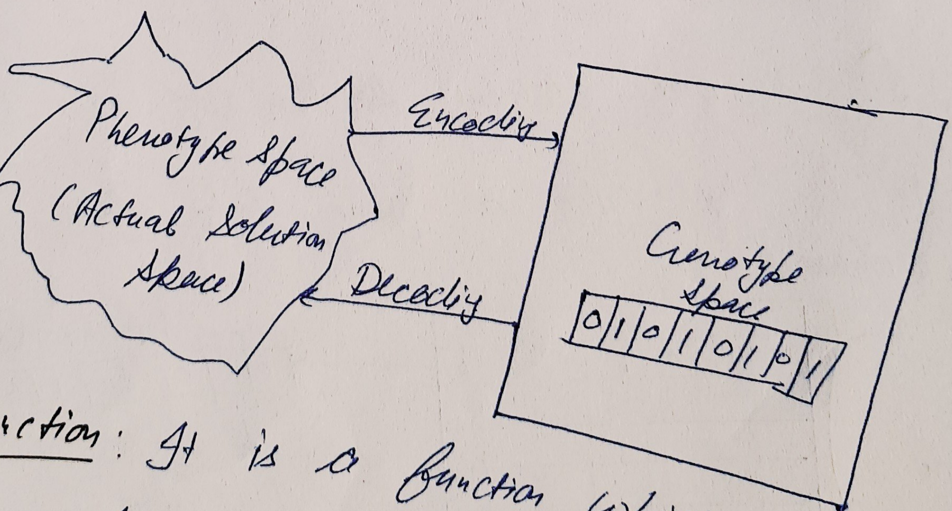
Genotype: Genotype is a population in the Computation space. In the Computation space, the solution are represented in a way which can be easily understood & manipulated using a computing sys.

PHENOTYPE: phenotype is the population in the actual real world solution space in which solution are represented in a way they are represented in real world solution

Decoding & Encoding:

Decoding is the process of transforming a solution from Genotype to phenotype.

Encoding: Phenotype to Genotype.



FITNESS function: It is a function which takes the solution as I/p & produce the suitably of the solution as the output.

Genetic operators: These alter the Genetic Composition of the offspring which include Crossover, mutation & selection.

SIMPLE GENETIC ALGORITHM (GA)

GA handles a population of possible solution.

1 Start with a randomly generated population

2 Calculate the fitness of each chromosome in the population

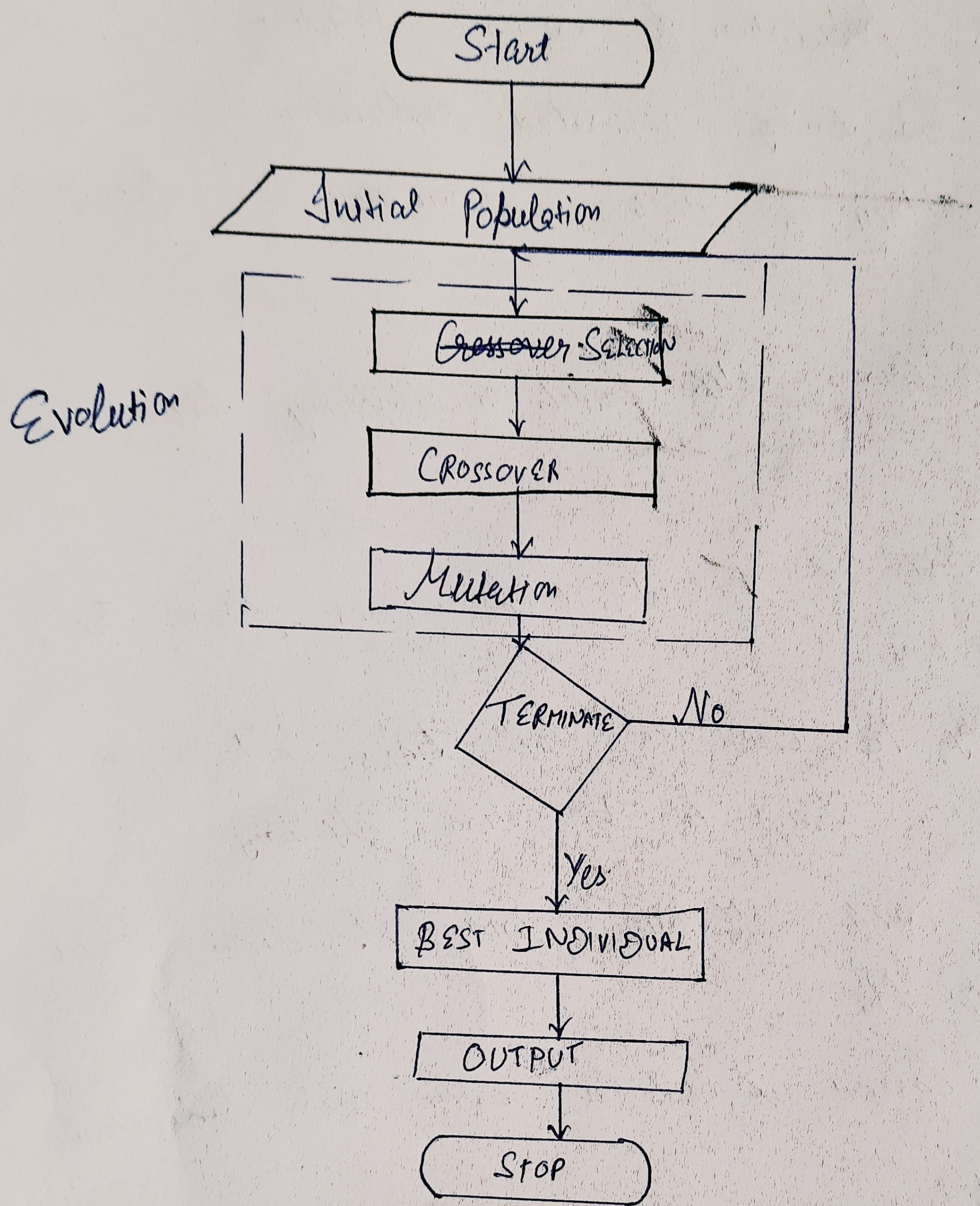
3 Repeat the following step until n offspring have been created

- Select a pair of parent chromosomes from the current population
- With probability P_c crossover the pair at a randomly chosen point to form two offspring
- Mutate the each offspring with at each locus with probability P_m.

4 Replace the current population with new population

5 Go to setp-2

GENERAL GENETIC ALGORITHM & it's FLOW Chart



Step1: Create a random Initial state.

Step2: Evaluate fitness

Step3: Reproduce: (Children Mutate)

Step4: Next Generation

Genetic Representation

One of the Most Important decision to make while Implementing a genetic algorithm is deciding the representation. Improper representation can lead to poor performance of the G.A

1 Binary Representation:

This is one of the Simplest & Most widely used representation in GA's.

* For some problem when the solution space consist of Boolean decision Variables- Yes or no The Binary representation is natural.
* For eg 0/1 Knapsack Problem. If there are n item, we can represent a solution by a binary string of n element. Where the x^{th} element tell wheather the x^{th} is picked (1) or not (0)

0	0	1	0	1	1	1	0	0	1

2 Real-Valued Representation:

For some problem, where we want to define the genes using Continous rather than discrete Variables. the real Value representation is the most natural.

* The Precision of these real Valued or floating Point number is However Limited to the Computer.

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

3 Integer Representation:

For discrete valued genes we can not always limit the solution space to binary.

* For example: If we want to encode the four distance - North, South, East & West. We can encode them as {1,2,3,4}

1	2	3	4	3	2	4	1	2	1

4 PERMUTATION REPRESENTATION:

In many problems, the solution is represented by an oscles of element. In such case permutation representation is most used

* A Classic enample: TSP (Travelling salesman problem)
In this the salesman has to take a tour of all the City. Visiting each city enactly once & come back to the starting problem.
The total distance of the tour is minimized.
* The solution of this TSP is naturally an ordering or permutation of all the cities & therefore using a permutation representation make sense.

1	5	9	8	7	4	2	3	6	0	5

ENCODING TECHNIQUES

a) **Binary Encoding**: Represented a gene in the term of bits (0s & 1s)

b) **Real Value Encoding**: Representing a gene in term of Values, symbols or string

c) **Permutation (order) Encoding**: Representing a sequence of element

d) **Tree Encoding**: Represented in the form of tree of objects.

BINARY Encoding

Example: 0/1 Knapsack Problem

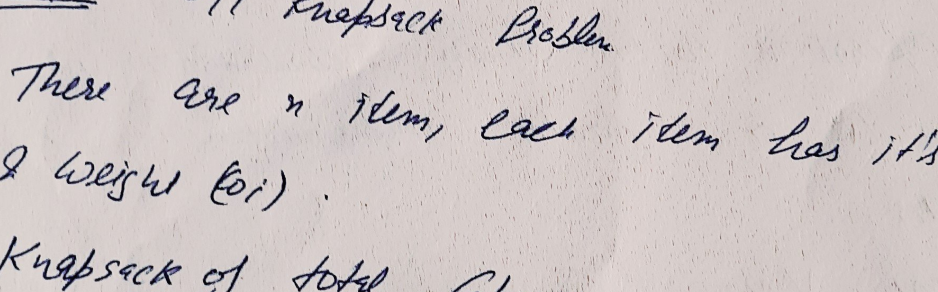
→ There are n item, each item has it's own Cost (C_i) & Weight (ω_i).

→ Knapsack of total Capacity W

This is the optimization problem

Maximize $C_i * \omega_i * x_i$

Subject to $\sum x_i * \omega_i \leq W$



Example-2 Minimize $f(x) = \frac{x^2}{2} + \frac{125}{x}$

When $0 \leq x \leq 15$ and x is any discrete Value

Genotype | x |

Phenotype | 01101 |

Any Value b/w 0 & 15 is possible solution [5 bit] → 1bit for s(n)

Encoding represent all the possible solution

Limitation: * Needs an effort to Convert into binary form.

* Accuracy depend on the binary representation

Advantages: Since Operations with binary representation is faster, It provides a faster implementation of all GA operators

REAL VALUE ENCODING

* The real-coded GA is most suitable for optimization in a Continous search space

- Uses the direct representation of design parameters
- Thus avoid any Intermediate encoding & decoding steps

Genotype | x | y |

Phenotype – | 5.28 | -475.36 |

Real Value Representation

2. REAL VALUE ENCODING

→ The real coded GA is most suitable for optimization in a continuous search space.

- Uses the direct representation of the design parameters.
- Thus avoid intermediate coding & encoding steps.

Example Maximize $f(x,y) = x^3 - x^2y + xy^2 + y^3$

Subject to $x + y \leq 10$

$-1 < x \leq 10$

$-10 \leq y \leq 10$

Genotype

x	y

01101	11001

3 PERMUTATION / order Encoding:

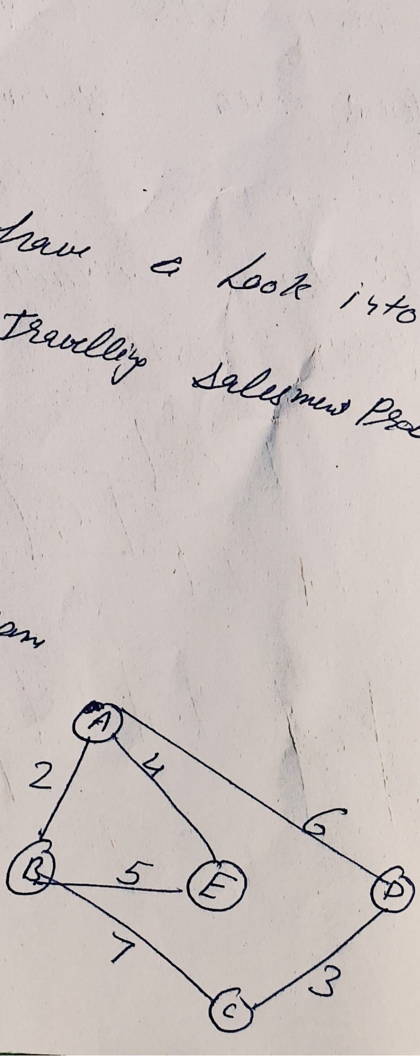
Let's have a look into the following instance of the (TSP) travelling salesman problem TSP

→ Visit all the cities

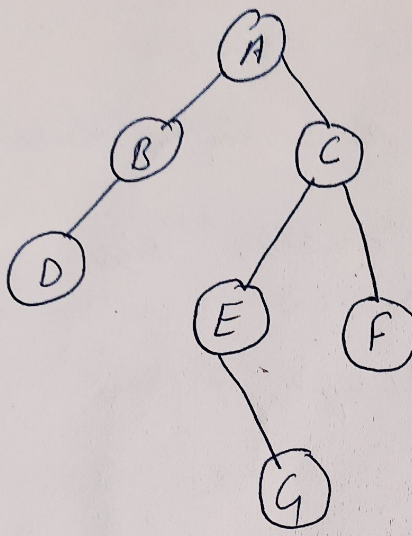
→ One city visit once only

→ Starting & encoding city is the same

d	A	B	C	D	E
A	0	2	∞	6	4
B	2	0	7	∞	5
C	∞	7	0	3	1
D	6	∞	3	0	∞
E	4	5	1	∞	0



4 **TREE ENCODING:** In this encoding scheme, a solution is encoded in the form of a binary tree



A Binary tree

In-order

DABEGCF

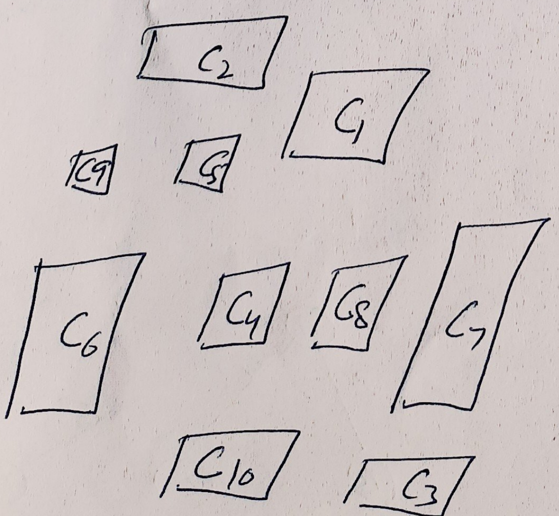
ABDCEGF

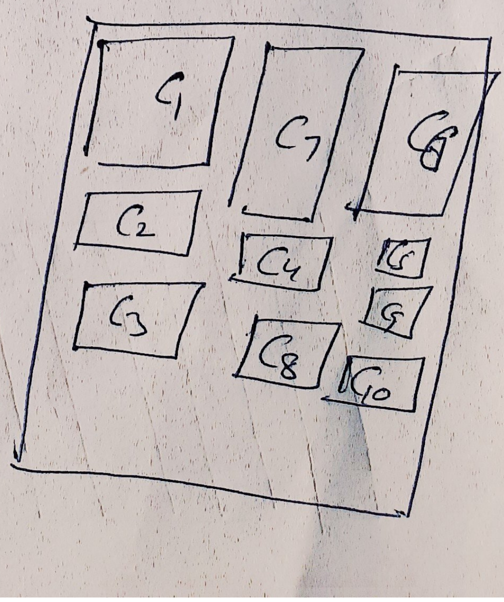
Pre order

DBGEFCA

Post order

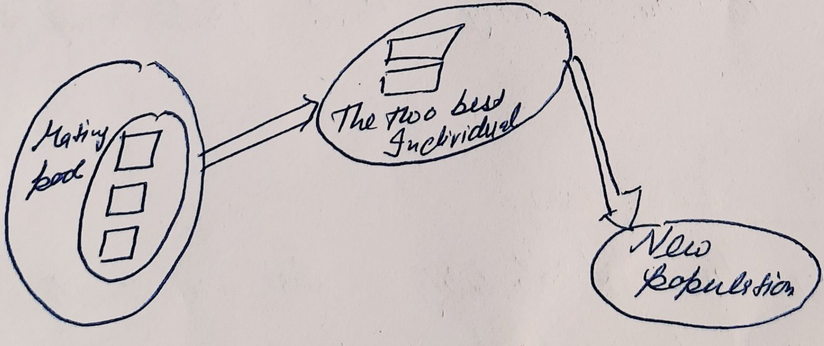
5 **FLOOR PLANNING:** Floor Planning is a standard Problem in VLSI Design. Here given, n circuits of different area requirement, we are to arrange them into a floor of Chip Layout, so that all the circuits are placed in a minimum Layout possible





SELECTION & INITIALIZATION

* Selection is the process of choosing two parents from the population for crossing.
* The purpose of selection is to emphasize fitter individual in the population in hopes that their offspring have higher fitness
* Selection is the process of creating the population for next generation from current generation.
* To Generate new population:
  * Create a mating pool
  * select a pair
  * Reproduce



FITNESS Evaluation: A simple strategy could be take the confidence of the value(s) of the objective function

* Simple, If there is a single objective function.
* But need a different treatment, If there are two or more objective function

Selection Schemes in GA:

1> Canonical Selection (Propostionate-based selection)

In this techniques, fitness is defined for the i-th individual as follow

$$fitness(i) = \frac{f_i}{\bar{F}}$$

→ Where f_i is the evaluation associated with the ith Individual in the population

→ $\bar{F}$ is the Average Evaluation of all Individual in the population size N & is defined as

$$\bar{F} = \sum_{i=1}^{N} \frac{f_i}{N}$$

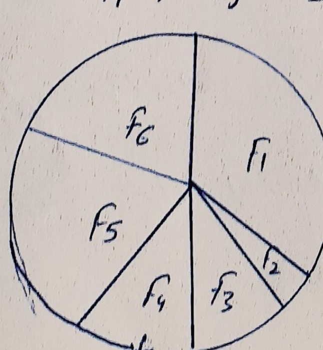
→ In an Iteration, we calculate $\frac{f_i}{\bar{F}}$ for all Individual, the probability of that Individual in the current population are copied & placed in the Mating Pool. Propostionate this fitness.

2 ROULETTE - WHEEL SELECTION:

In this Scheme, the probability for an Individual is being selected in the Mating pool is considered to be prepostionate to its fitness.

$i \longrightarrow f_i$

$j \longrightarrow f_j$ $\quad f_i > f_j \Rightarrow$ (i is preferrable from j)



Wheel Game Rotate & see the pointer

Roulette - Wheel Selection Algo

* The top surface area of the wheel is divided into N Part in proportion to the fitness values $f_1, f_2, f_3 - f_N$.

* The wheel is rotated in a particular direction & a fixed pointer is used to indicate the ~~fixed~~ Winning area when its stop rotation

* A particular sub- area representing a GA solution is selected to be winner probabilistically & the probability that the ith area will be declared as

$$P_i = \frac{f_i}{\sum_{i=1}^{N} f_i}$$

* In other ~~words~~ words, The Individual having higher fitness Value is likely to be selected More

* The Wheel is rotated for N_P times (where $N_P = P\% N$ for some P) & each time only one area is identified by the pointer to be winner

ALGORITHM:

Input: A population of size N with their fitness values

O/P: A mating pool of size N_p

Steps:

1. Compute $P_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \forall i = 1, 2, -- N$

2) Calculate the Cumulative Probability for each Individual starting from the top of the List

$$P_i = \sum_{j=1}^{i} P_j \quad \text{for all } j = 1, 2 -- N$$

3) Generate a random no. say r b/w 0 & 1

4) Select the jth Individual such that $P_{j-1} < r \leq P_j$

5) Repeat step 3 & 4 to select N_p Individual

6) END

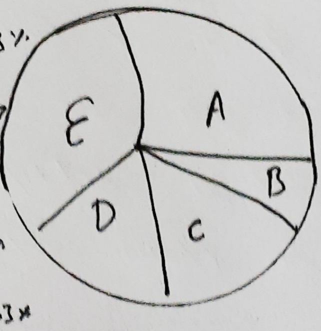
3) Random Selection: This technique randomly selects a parent from the population.

4) Rank Selection:

1) Select a pair of Individual at random. Generate a random no R b/w 0 & 1, If $R < r$ use the first Individual as a parent. If $R \geq r$ then use the second Individual as the parent. This is repeated to select the second parent

2 Select the two Individual at random, The Individual with the highest evaluation become a parent.

	Fitness	Rank	
A	5	5	33%
B	2	4	26.67%
C	0.5	1	6.67%
D	1.5	3	20%
E	1	2	13.33%
		15	



$A \rightarrow \frac{5}{15} \times 100 = 33\%$

1

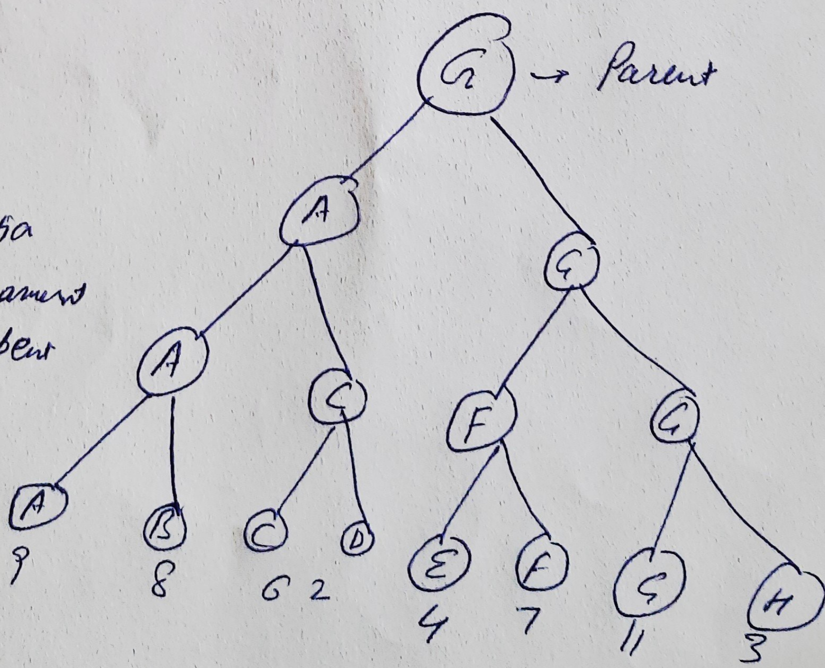
5 TOURNAMENT SELECTION:

The tournament selection strategy provide selective Pressure by holding a tournament Competition among N_p individual.

→ The best Individual from the tournament is the one with the highest fitness who is the winner of N_p. WINNER Are then Interested into the Mating Pool.

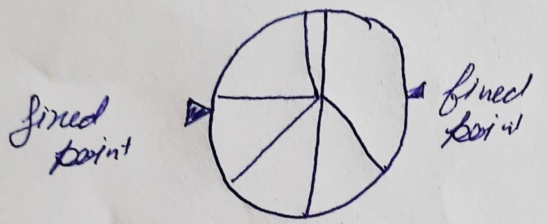
$N \leq N_p$

for eg N - Population
$N_p \rightarrow$ Tournament participent



6 Stochastic Universal Sampling (SUS)

It is quite similar to Roulette wheel selection, However instead of having just one fixed point, we have multiple fixed point, Therefore all the parents are chosen in just one spin of the wheel. such a setup encourages the highly fit individual to be chosen at least once



Chromosome	fitness value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Crossover (Recombination)

* Crossover is the process of two parent solutions is producing from them a child After Crossover, the population is enriched with better Individual
* Crossover is a recombination operator that proceeds in three steps
1) The reproduction operator select at random a pair of two Individual string for the mating
2) A Cross site is selected at random along the string Length
3) finally, The position Value are swapped b/w the two string following the Cross site

⇒ Types of Crossover:

1) **Single-Point Cross-over:** A random crossover point is selected & the traits of its two parent are swapped to get new off-springs

Parent 1	1 0 1 1 0 | 0 1 0
Parent 2	1 0 1 0 1 | 1 1 1

Child	1 0 1 1 0 | 1 1 1
Child	1 0 1 0 1 | 0 1 0

2) **Two-point Crossover:** Two crossover point are chosen & the Content between them points are exchanged b/w two mated parents

Parent 1	1 1 | 0 1 1 | 0 1 0
Parent 2	0 1 | 1 0 1 | 1 0 0

Child	1 1 | 1 0 1 | 0 1 0
Child	0 1 | 0 1 1 | 1 0 0

3 Multi-point Crossover (N-Point Crossover):

Two way cross-overs

1) Even no. of cross-sites and the other Odd numbers of Cross sites

* In case of even no of cross sites, the cross-sites are selected randomly around a Circle & Info is exchanged.

Parent1	0	1	1	0	1	0
Parent2	1	1	0	1	0	1

Remain same

Others are Change

Child1	1	1	0	0	1	0
Child2	0	1	1	1	0	1

4 Uniform Crossover:

Where There is a 1 in the Crossover Mask The Gene is Copied from the first parent, where there is a 0 in the Mask The Gene is Copied from the second parent.

Parent1	1	0	1	1	0	0	1	1
Parent2	0	0	0	1	1	0	1	0
Mask	1	1	0	1	0	1	1	0
Child1	1	0	0	1	1	0	1	0
Child2	0	0	1	1	0	0	1	1

5 Three parent Crossover: In this Crossover technique, Three parent are randomly chosen. Each bit of the first parent is Compared with the bit of the second parent. If both are same, the bit is taken for the offspring, otherwise the bit from the Third parent is taken for the offspring.

Parent 1	1 1 0 1 0 0 0 1
Parent 2	0 1 1 0 1 0 0 1
Parent 3	0 1 1 0 1 1 0 0
Child	0 1 1 0 1 0 0 1

6 Shuffle Crossover: A single crossover point is selected but before the variable are exchanged, they are randomly shuffled in both parent. After

Parent 1	1 1 0 0 0	1 1 0
Parent 2	1 0 0 1 1	0 1 1

Parent 1	0 0 1 0 1	1 0 1
Parent 2	0 1 0 1 1	1 0 1

Child 1	0 0 1 0 1	1 0 1
Child 2	0 1 0 1 1	1 0 1

INTRODUCTION TO MUTATION

* Mutation may be defined as a small random tweak in the Chromosome to get a new solution.
* It used to maintain & Introduce diversity in the genetic population & is usually applied with a low probability - Pm. If the probability is very high. The GA gets reduce to a random search.

* Mutation is the part of search space which is related to 'Exploration' of the search space

Mutation operators :

1) **Bit-flip operator**: We select one or more random bits & flip them. This is used for binary encoded GA's

| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | ⇒ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

2) **Random Resetting**: Random Resetting is the extention of the bit flip for the Integer representation. In this, a random value from the set of permissible value is assigned to a randomly chosen gene.

3 **Swap Mutation**: We select two position on the chromosomes at random, & Interchange it's value. This is common in permutation band encoding

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | ⇒ | 1 | 6 | 3 | 4 | 5 | 2 | 7 | 8 | 9 | 0 |

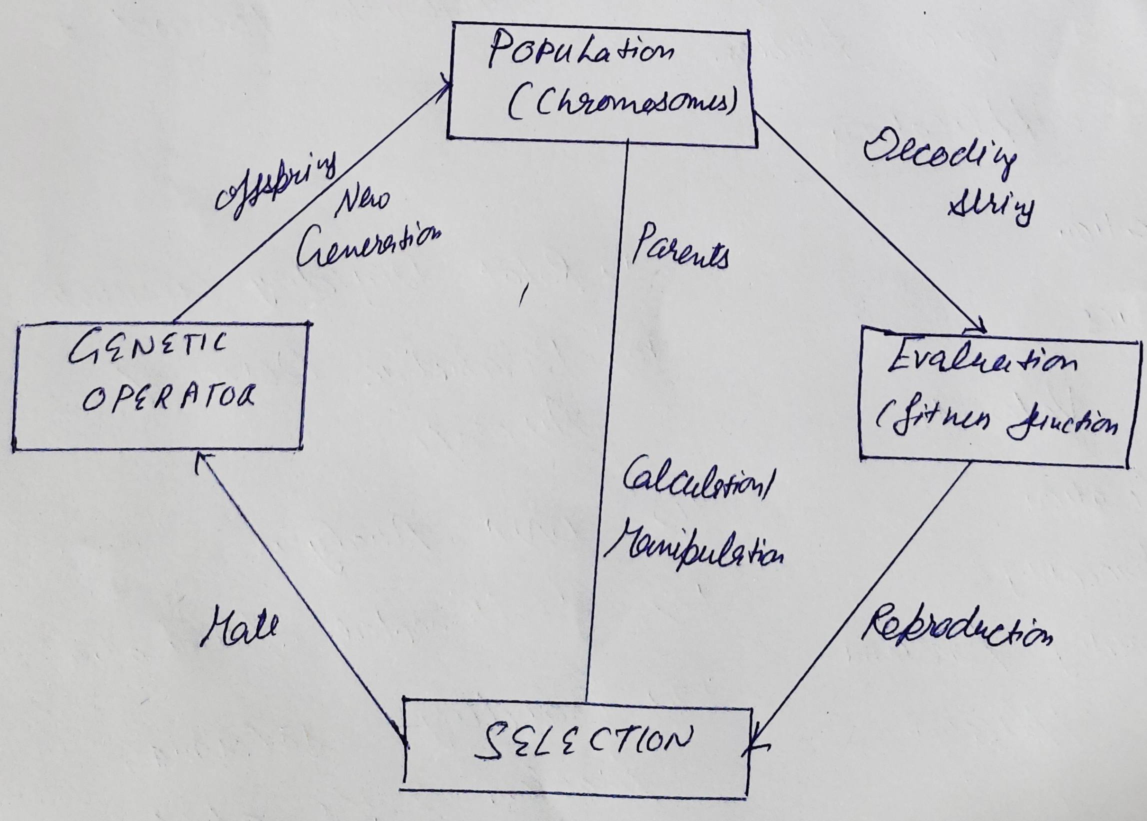
4) **Scramble Mutation:** It is also popular with permutation representation. In this from the entire Chromosomes, a subset of gene is chosen & their value are scrambled / shuffled randomly

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | → | 0 | 1 | 3 | 6 | 4 | 2 | 5 | 7 | 8 | 9 |

5) **Inversion Mutation:** In Inversion mutation, we select a subset of gene like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | → | 0 | 1 | 6 | 5 | 4 | 3 | 2 | 7 | 8 | 9 |

GENERATIONAL CYCLE



Component of Generational Cycle in GA:

1) **Population** (Chromosomes): A population is the Collection of Individual. It Consist of a numbers of Individual being tested, The Phenotype parameter defining the Individual & some Info about the Search Space

2) **Evaluation**: A fitness function is a popular type of objective function that Quantifies the optimality of a solution in a GA so that particular Chromosome may be ranked against all other Chromosomes

3) **SELECTION**: During each successive generation, a proportion of Existing population is selected to breed a new generation, Individual solutions are selected through a fitness based process

4) **Genetic operator**: A Genetic operator is an operator used in genetic algorithm to guide the algorithm toward a solution to a given problem

Applications

1) Optimization
2) Economics
3) Neural Networks
4) Parallelization
5) Image processing
6) Vehicle routing problems
7) Scheduling application
8) Machine Learning
9) Robot Trajectory Generation
10) Parametric Design of Aircraft
11) DNA Analysis
12) Multimodal optimization
13) Travelling salesman problem & it's application