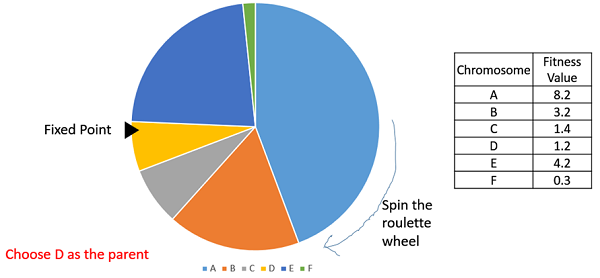
Q3

Understand the different types of selection, cross-over, mutation operation that are employed by Genetic algorithms. Illustrate the working of each over a population of chromosomes (Refer David Goldberg or any other online sources). Explore a minimum of at least 4 to 5 operators under each type.

1. Selection Operators

Roulette Wheel Selection

In a roulette wheel selection, the circular wheel is divided as described before. A fixed point is chosen on the wheel circumference as shown and the wheel is rotated. The region of the wheel which comes in front of the fixed point is chosen as the parent. For the second parent, the same process is repeated.



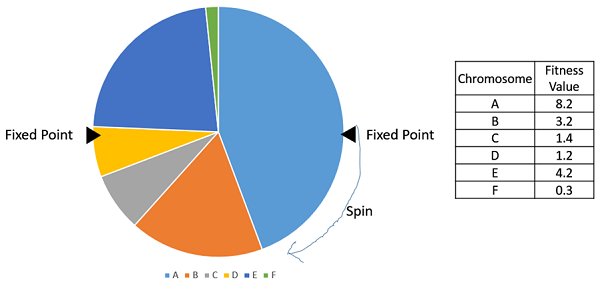
It is clear that a fitter individual has a greater pie on the wheel and therefore a greater chance of landing in front of the fixed point when the wheel is rotated. Therefore, the probability of choosing an individual depends directly on its fitness.

Implementation wise, we use the following steps −

* Calculate S = the sum of a fitnesses.
* Generate a random number between 0 and S.
* Starting from the top of the population, keep adding the finesses to the partial sum P, till P<S.
* The individual for which P exceeds S is the chosen individual.

Stochastic Universal Sampling (SUS)

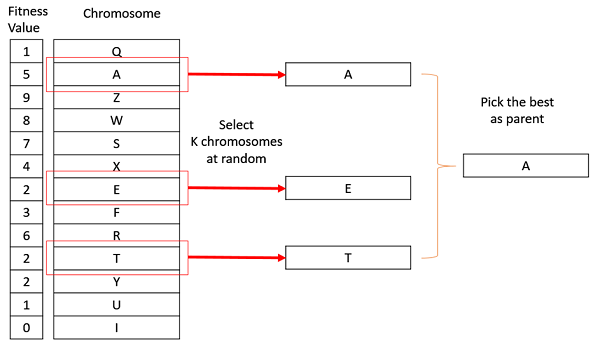
Stochastic Universal Sampling is quite similar to Roulette wheel selection, however instead of having just one fixed point, we have multiple fixed points as shown in the following image. Therefore, all the parents are chosen in just one spin of the wheel. Also, such a setup encourages the highly fit individuals to be chosen at least once.



It is to be noted that fitness proportionate selection methods don’t work for cases where the fitness can take a negative value.

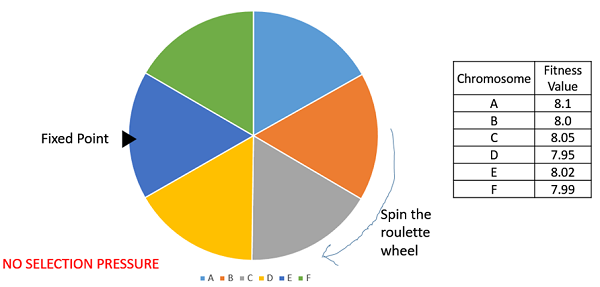
Tournament Selection

In K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent. Tournament Selection is also extremely popular in literature as it can even work with negative fitness values.



Rank Selection

Rank Selection also works with negative fitness values and is mostly used when the individuals in the population have very close fitness values (this happens usually at the end of the run). This leads to each individual having an almost equal share of the pie (like in case of fitness proportionate selection) as shown in the following image and hence each individual no matter how fit relative to each other has an approximately same probability of getting selected as a parent. This in turn leads to a loss in the selection pressure towards fitter individuals, making the GA to make poor parent selections in such situations.



In this, we remove the concept of a fitness value while selecting a parent. However, every individual in the population is ranked according to their fitness. The selection of the parents depends on the rank of each individual and not the fitness. The higher ranked individuals are preferred more than the lower ranked ones.

|  |  |  |
| --- | --- | --- |
| **Chromosome** | **Fitness Value** | **Rank** |
| A | 8.1 | 1 |
| B | 8.0 | 4 |
| C | 8.05 | 2 |
| D | 7.95 | 6 |
| E | 8.02 | 3 |
| F | 7.99 | 5 |

Random Selection

In this strategy we randomly select parents from the existing population. There is no selection pressure towards fitter individuals and therefore this strategy is usually avoided.

1. Crossover Operators

One Point Crossover

In this one-point crossover, a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.



Multi Point Crossover

Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.



Uniform Crossover

In a uniform crossover, we don’t divide the chromosome into segments, rather we treat each gene separately. In this, we essentially flip a coin for each chromosome to decide whether or not it’ll be included in the off-spring. We can also bias the coin to one parent, to have more genetic material in the child from that parent.



Whole Arithmetic Recombination

This is commonly used for integer representations and works by taking the weighted average of the two parents by using the following formulas,

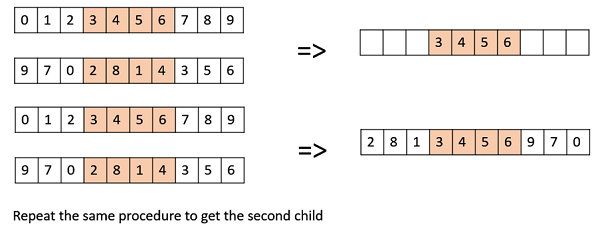
Obviously, if , then both the children will be identical as shown in the following image.



Davis’s Order Crossover (OX1)

OX1 is used for permutation-based crossovers with the intention of transmitting information about relative ordering to the off-springs. It works as follows −

* Create two random crossover points in the parent and copy the segment between them from the first parent to the first offspring.
* Now, starting from the second crossover point in the second parent, copy the remaining unused numbers from the second parent to the first child, wrapping around the list.
* Repeat for the second child with the parent’s role reversed.



1. Mutation Operators

Bit Flip Mutation

In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.

Bit Flip Mutation

Random Resetting

Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation-based encodings.

Swap Mutation

Scramble Mutation

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.

Scramble Mutation

Inversion Mutation

In inversion mutation, we select a subset of genes like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.

Inversion Mutation