

Genetic Algorithms - Mutation

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.

0	0	1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

=>

0	0	1	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Exchange Mutation

Suppose we have a chromosome:

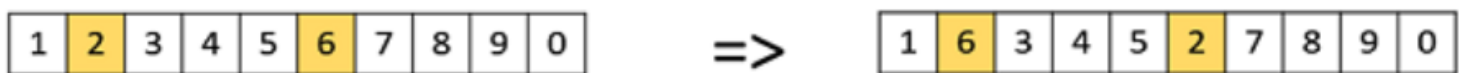
5 3 2 1 7 4 0 6

We simply choose two genes at random (i.e.; 3 and 4) and swap them:

5 **4** 2 1 7 **3** 0 6

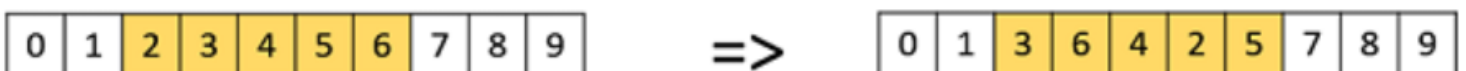
Swap Mutation

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



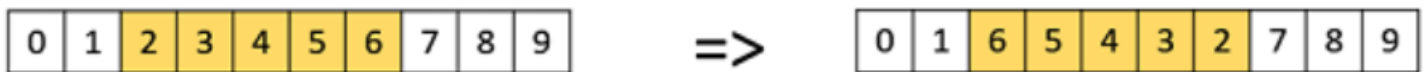
Scramble Mutation

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.



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.



Displacement Mutation

Select two random points (i.e.; positions 4 and 6), grab the genes between them as a group, then reinsert the group at a random position displaced from the original.

0 1 2 **3 4 5** 6 7

becomes

0 **3 4 5** 1 2 6 7

Insertion Mutation

This is a very effective mutation and is almost the same as Displacement Mutation, except here only one gene (i.e.; 2) is selected to be displaced and inserted back into the chromosome. In tests, this mutation operator has been shown to be consistently better than any of the alternatives mentioned here.

0 1 **2** 3 4 5 6 7

Take the 2 out of the sequence,

0 1 3 4 5 6 7

and reinsert the 2 at a randomly chosen position:

0 1 3 4 5 **2** 6 7

Displaced Inversion Mutation

Select two random points (i.e.; positions 5 through 7), reverse the gene order between the two points, and then displace them somewhere along the length of the original chromosome (i.e.; position 2). This is similar to performing Inversion Mutation and then Displacement Mutation using the same start and end points.

0 1 2 3 **4 5 6** 7

becomes

0 **6 5 4** 1 2 3 7