Ajay Babu Gorantla

Artificial Intelligence

Programming 2

8 Queens Problem using Genetic Algorithms

The objective of this work is to analyze how genetic algorithms work in solving the 8 queens problem.

For the convenience of computation in my program I have used a sequence representation of

Procedure:

Initial Population

First initialized population based on the population parameter. Each element in the population list is an object with attributes sequence, fitness and survival probability.

Selection of Parents

Used a fitness function to determine the fitness of a sequence.

The fitness of a sequence is determined by the number of non-attacking queen pairs a sequence has.

To decide the survival of a sequence through future generations, used a normalized function as described in the assignment

$$S_i = \frac{f_i}{\sum_{j=1}^{PopSize} f_j}$$

Here f is nothing but the fitness values of each sequence as described above.

Randomly generated a threshold value between 0 and 1 and randomly selected two parents among the population whose survival probability is less than the threshold value

Cross Over:

Using the parents selected above, randomly selected a spot to slice the sequences and replace each sequence with the sub part of the other.

Mutation:

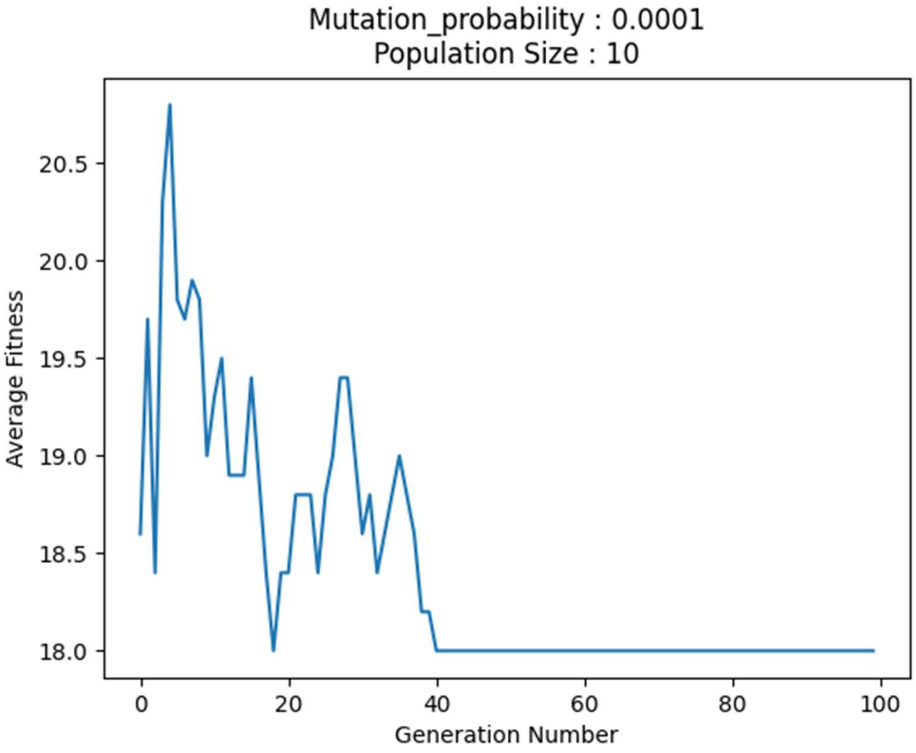
Initialized a hyperparameter mutation probability to decide if a sequence is to be mutated or not. Randomly generated a value between 0 and 1, mutate if the generated value is less than the hyperparameter mutation probability. If a sequence is eligible to be mutated, a randomly selected gene is mutated with a random value.

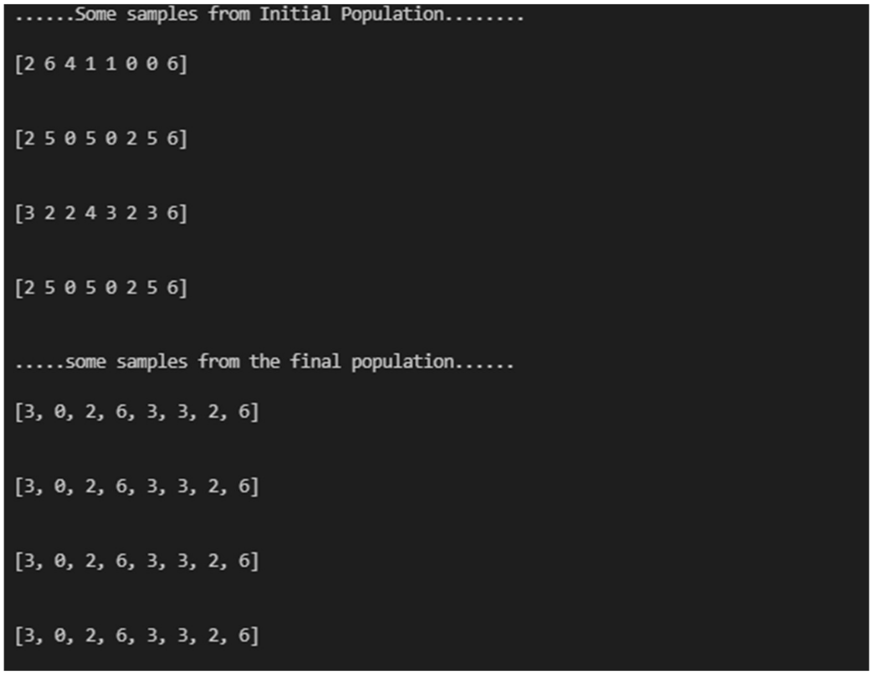
Analysis

The whole assignment comes down to how the genetic algorithm works for different hyperparameters for example the population size, number of generations, mutation probability.

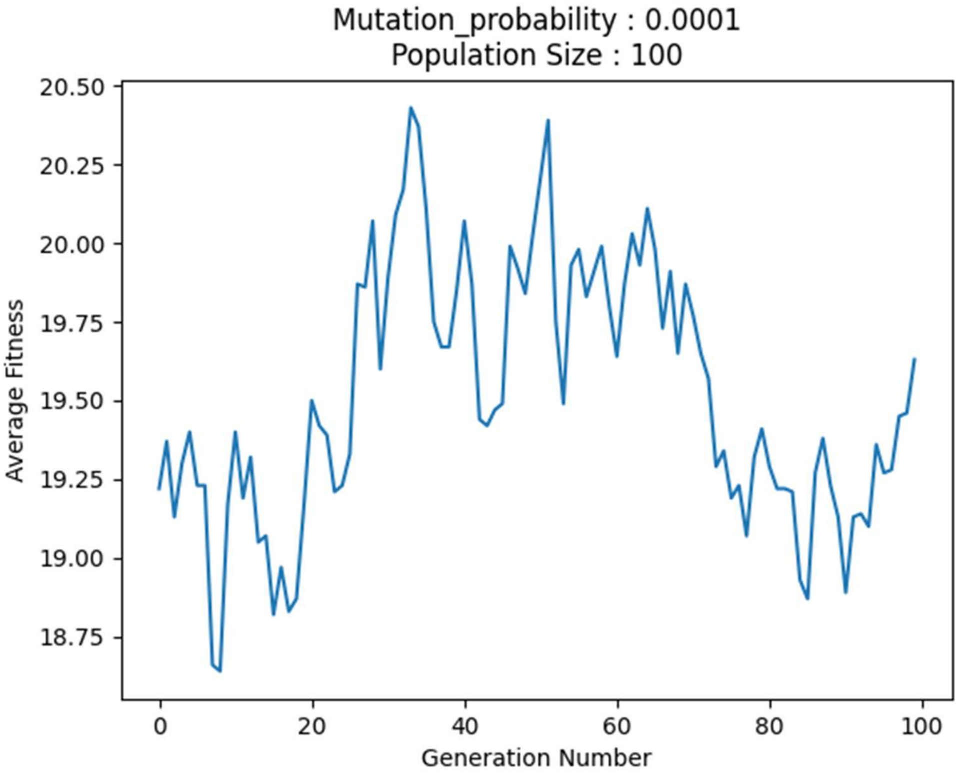
I have tried for different values of the above-mentioned hyper parameters and plotted graphs accordingly.

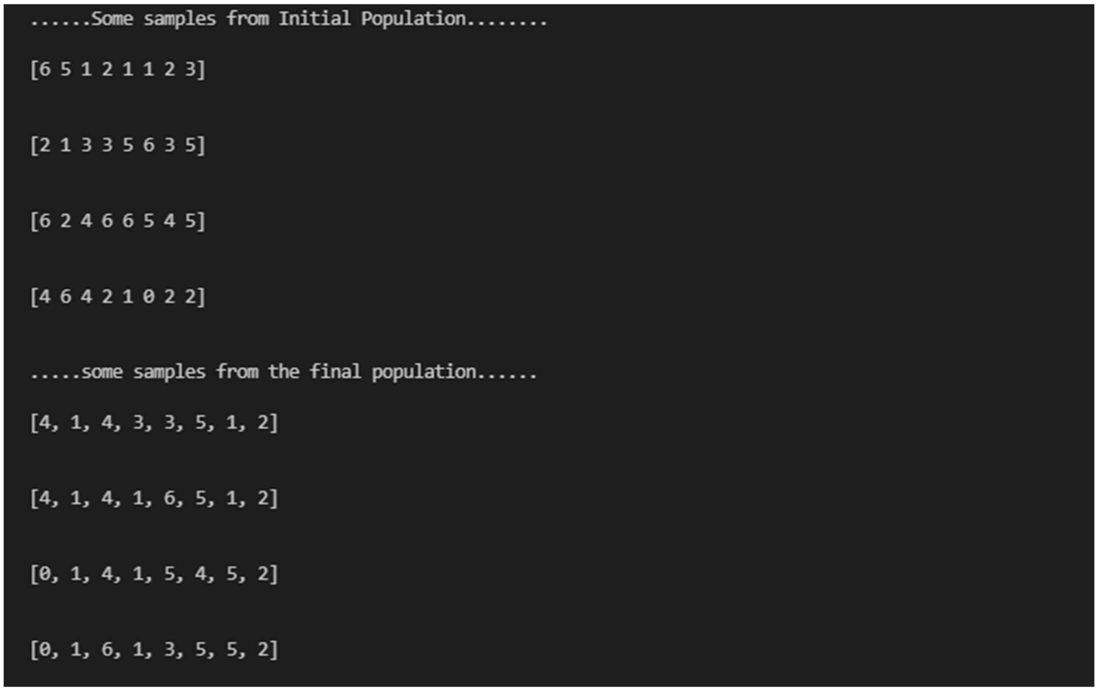
Below are the graphs for each experiment



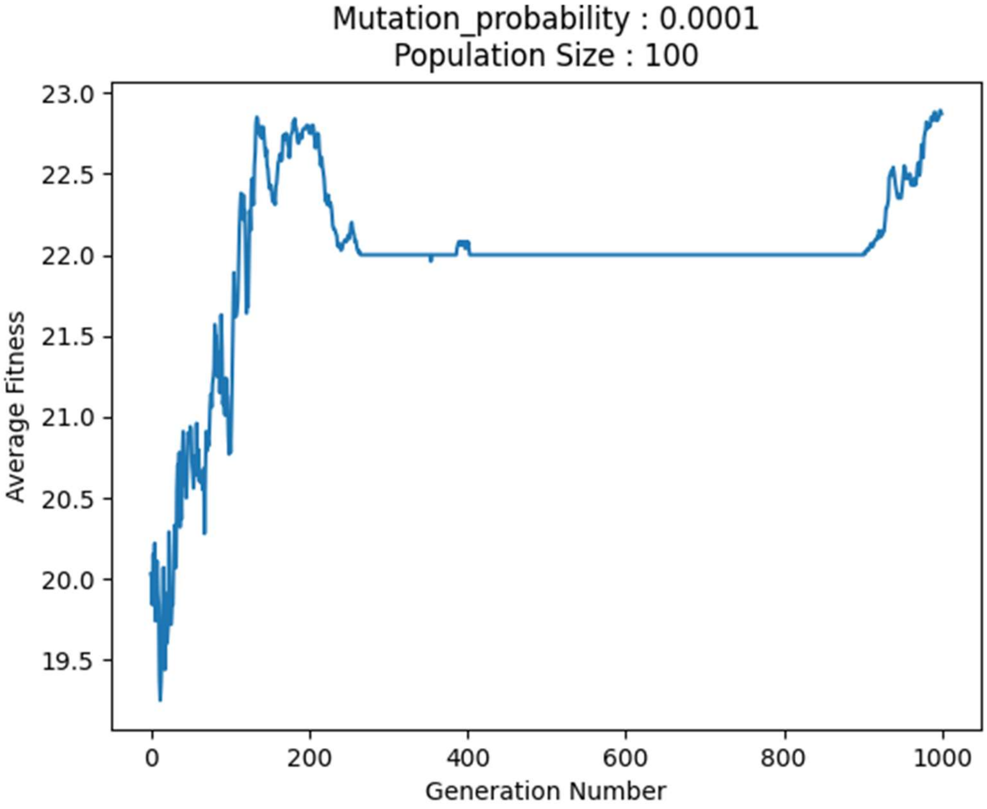


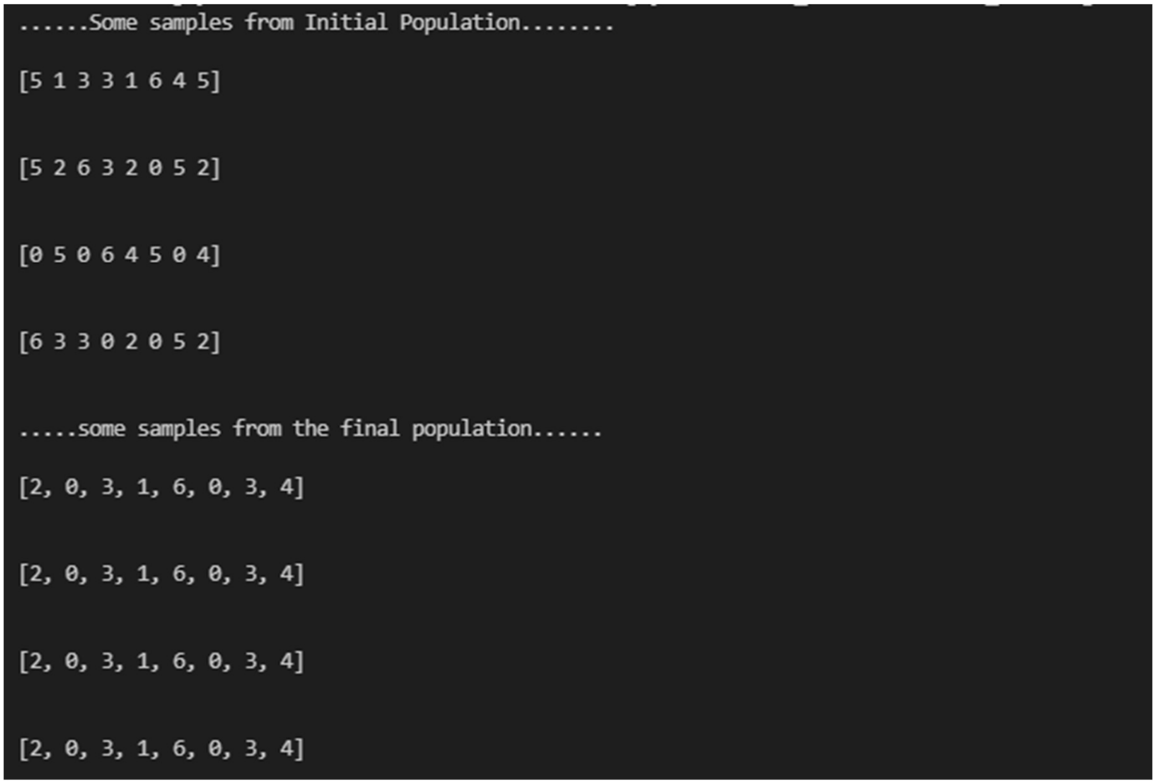
```
......Some samples from Initial Population........

[2 6 4 1 1 0 0 6]


[2 5 0 5 0 2 5 6]


[3 2 2 4 3 2 3 6]


[2 5 0 5 0 2 5 6]


.....some samples from the final population......

[3, 0, 2, 6, 3, 3, 2, 6]


[3, 0, 2, 6, 3, 3, 2, 6]


[3, 0, 2, 6, 3, 3, 2, 6]


[3, 0, 2, 6, 3, 3, 2, 6]
```



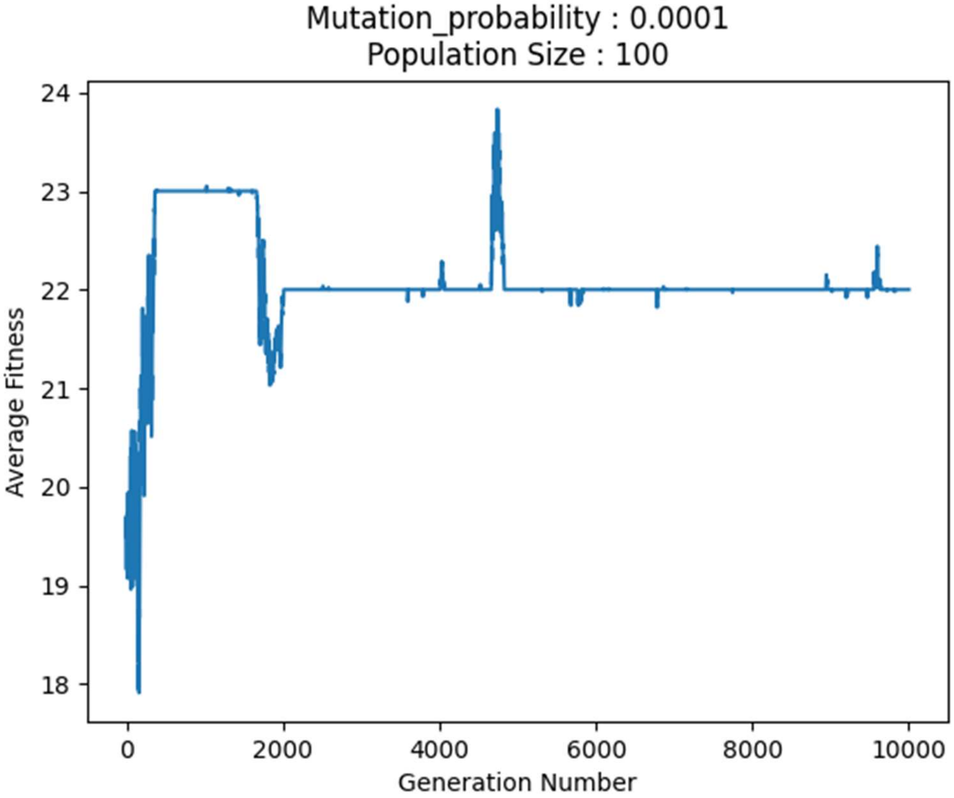


```
......Some samples from Initial Population........

[6 5 1 2 1 1 2 3]


[2 1 3 3 5 6 3 5]


[6 2 4 6 6 5 4 5]


[4 6 4 2 1 0 2 2]


.....some samples from the final population......

[4, 1, 4, 3, 3, 5, 1, 2]


[4, 1, 4, 1, 6, 5, 1, 2]


[0, 1, 4, 1, 5, 4, 5, 2]


[0, 1, 6, 1, 3, 5, 5, 2]
```

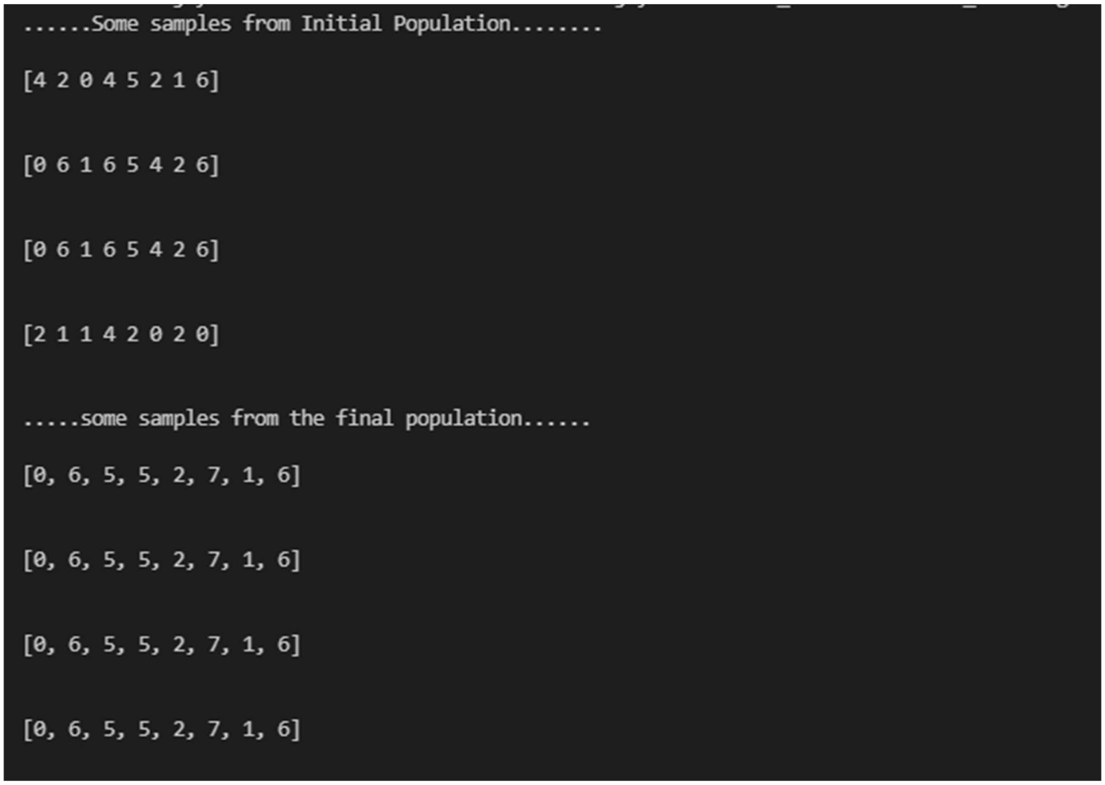




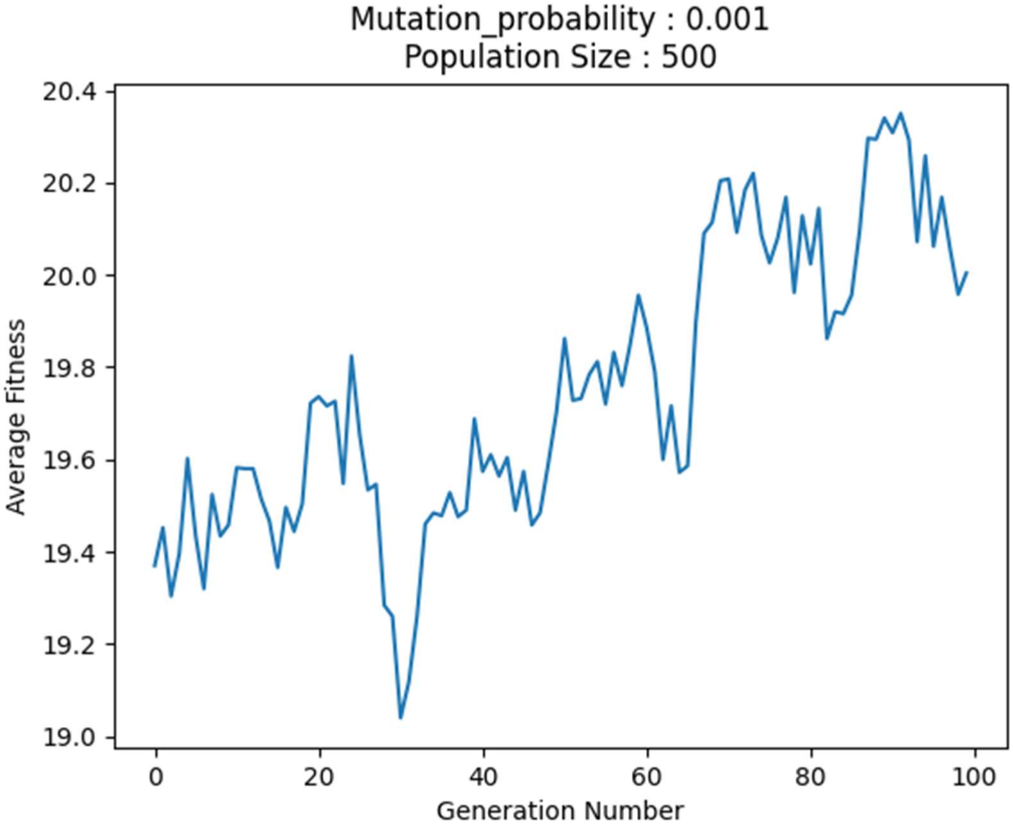
```
......Some samples from Initial Population........

[5 1 3 3 1 6 4 5]


[5 2 6 3 2 0 5 2]


[0 5 0 6 4 5 0 4]


[6 3 3 0 2 0 5 2]


.....some samples from the final population......

[2, 0, 3, 1, 6, 0, 3, 4]


[2, 0, 3, 1, 6, 0, 3, 4]


[2, 0, 3, 1, 6, 0, 3, 4]


[2, 0, 3, 1, 6, 0, 3, 4]
```

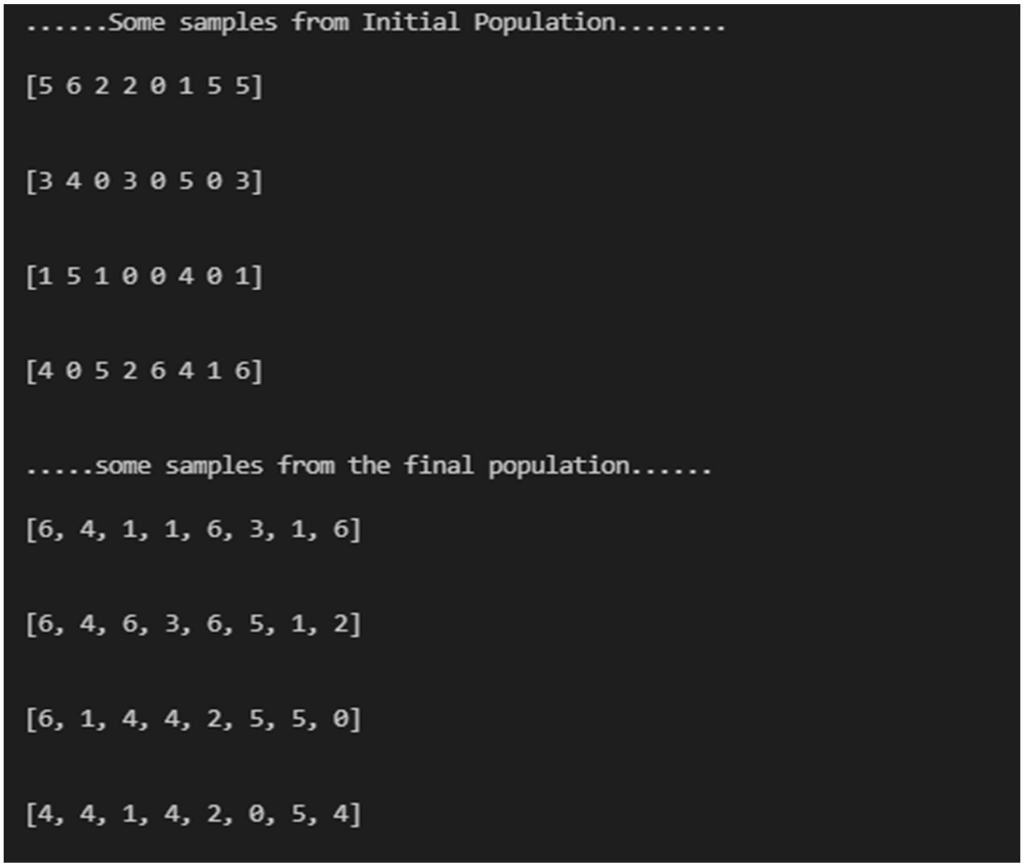


Mutation_probability : 0.0001
Population Size : 100

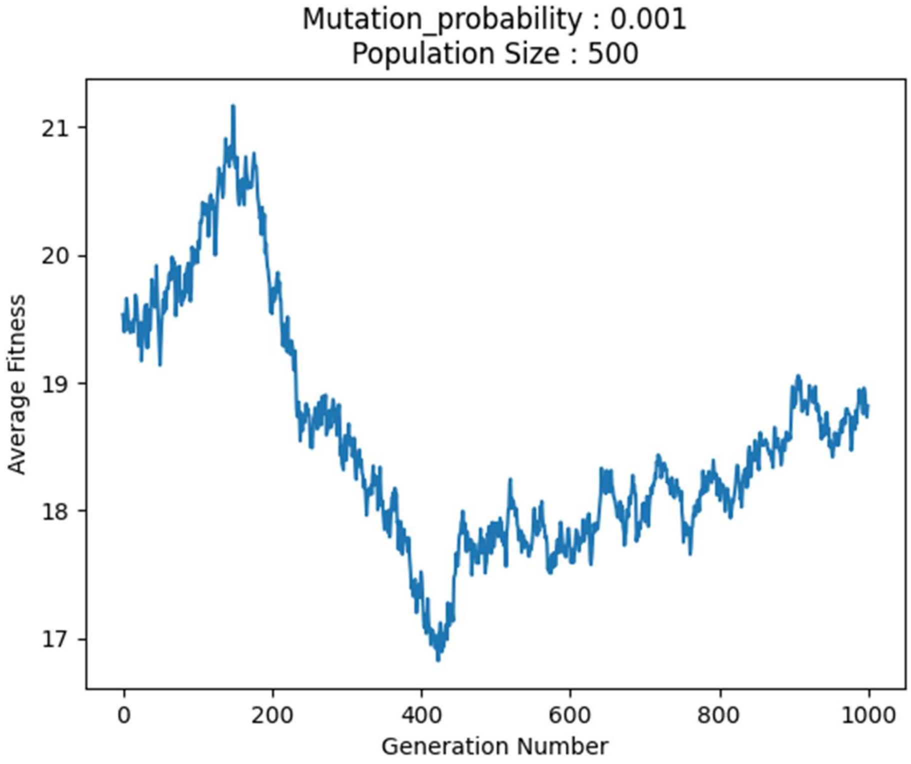


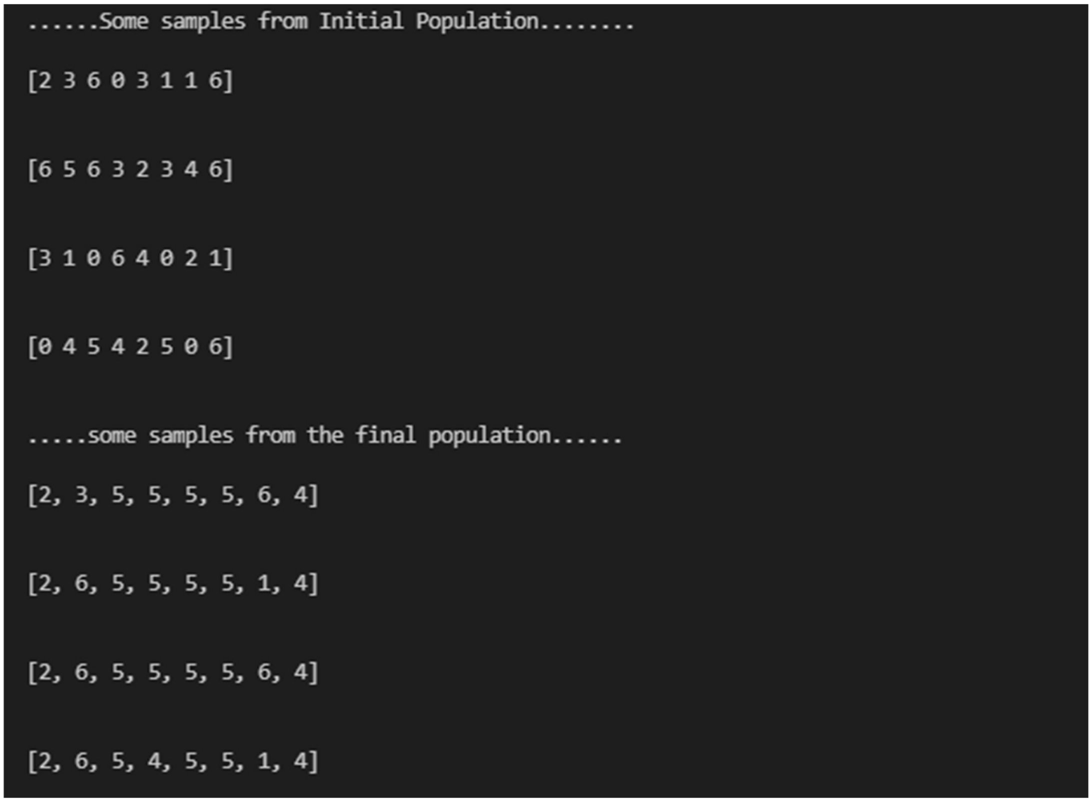
```
......Some samples from Initial Population........

[4 2 0 4 5 2 1 6]


[0 6 1 6 5 4 2 6]


[0 6 1 6 5 4 2 6]


[2 1 1 4 2 0 2 0]


.....some samples from the final population......

[0, 6, 5, 5, 2, 7, 1, 6]


[0, 6, 5, 5, 2, 7, 1, 6]


[0, 6, 5, 5, 2, 7, 1, 6]


[0, 6, 5, 5, 2, 7, 1, 6]
```



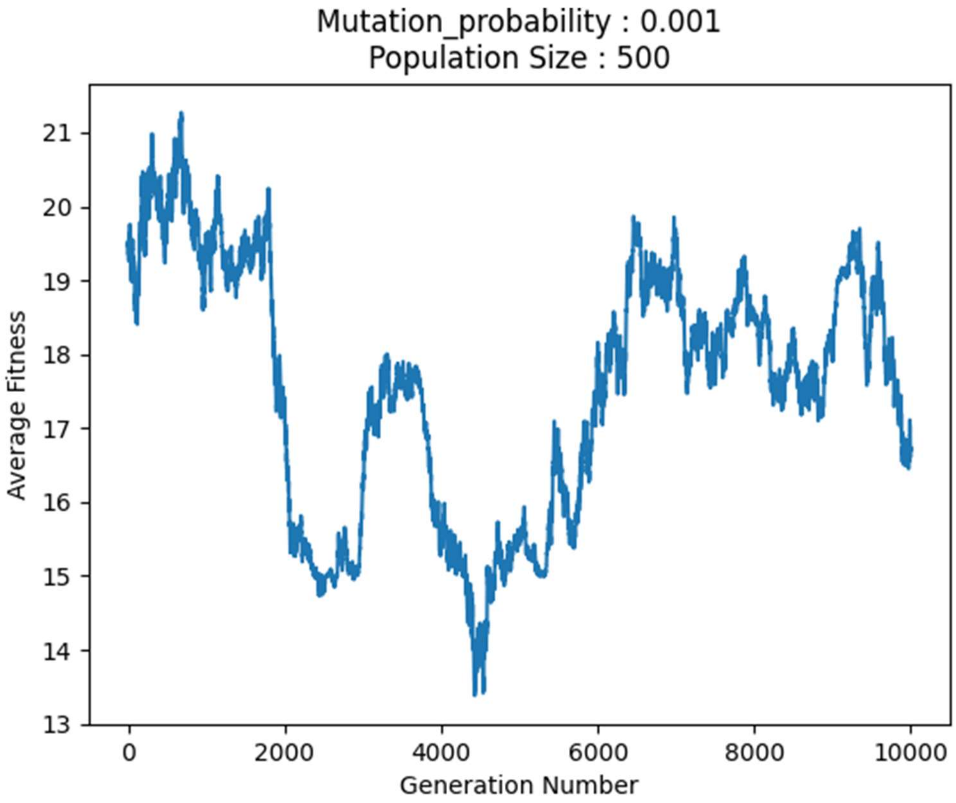


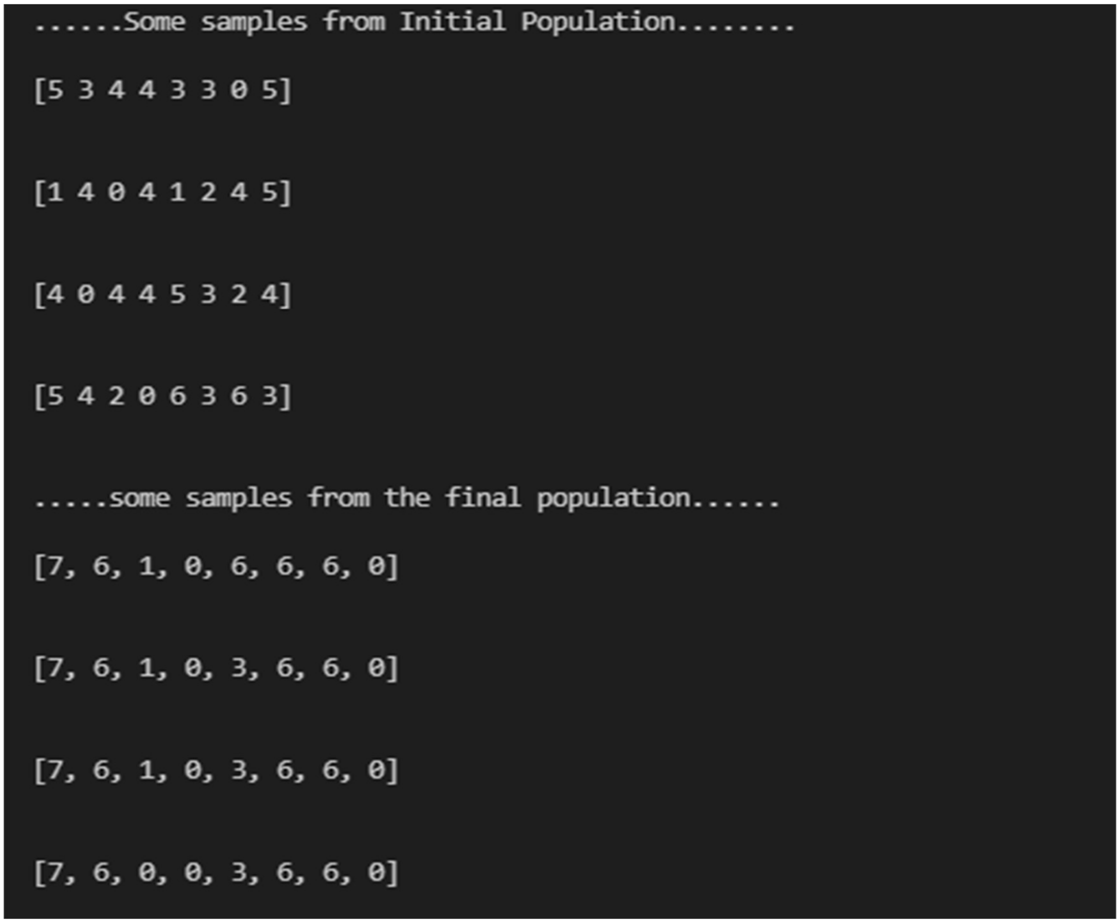
```
......Some samples from Initial Population........

[5 6 2 2 0 1 5 5]


[3 4 0 3 0 5 0 3]


[1 5 1 0 0 4 0 1]


[4 0 5 2 6 4 1 6]


.....some samples from the final population......

[6, 4, 1, 1, 6, 3, 1, 6]


[6, 4, 6, 3, 6, 5, 1, 2]


[6, 1, 4, 4, 2, 5, 5, 0]


[4, 4, 1, 4, 2, 0, 5, 4]
```





```
......Some samples from Initial Population........

[2 3 6 0 3 1 1 6]


[6 5 6 3 2 3 4 6]


[3 1 0 6 4 0 2 1]


[0 4 5 4 2 5 0 6]


.....some samples from the final population......

[2, 3, 5, 5, 5, 5, 6, 4]


[2, 6, 5, 5, 5, 5, 1, 4]


[2, 6, 5, 5, 5, 5, 6, 4]


[2, 6, 5, 4, 5, 5, 1, 4]
```





```
......Some samples from Initial Population........

[5 3 4 4 3 3 0 5]


[1 4 0 4 1 2 4 5]


[4 0 4 4 5 3 2 4]


[5 4 2 0 6 3 6 3]


.....some samples from the final population......

[7, 6, 1, 0, 6, 6, 6, 0]


[7, 6, 1, 0, 3, 6, 6, 0]


[7, 6, 1, 0, 3, 6, 6, 0]


[7, 6, 0, 0, 3, 6, 6, 0]
```

Conclusions

One thing that I observed is that the more the population size the more the variance in the children in the final population. Of course, this highly depends on the number of generations we are willing to go through. A low population size is forming a convergence.

The change in mutation percentage is also making a considerable difference in the children in the final population. i.e., the high the mutation probability the higher the chances of mutation and consequently higher the non-identical children in the final population.