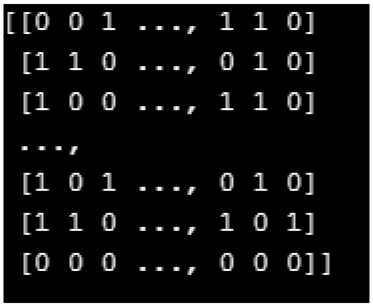
Assignment 1 – Genetic Algorithm Project Report

Implementation with the given parameters

Given parameters are as below

Population size: 100 Crossover rate: 70% Mutation rate: 0.1% Maximum number of generations: 1,000 Selection method: Tournament (K=2)

With the given sample code in Python, I have generated 100 overlay networks with 300 links (active-1 and passive-0) in each, by using the random and numpy libraries. Below is the sample of generated 100 networks.



```
[[0 0 1 ..., 1 1 0]
 [1 1 0 ..., 0 1 0]
 [1 0 0 ..., 1 1 0]
 ...,
 [1 0 1 ..., 0 1 0]
 [1 1 0 ..., 1 0 1]
 [0 0 0 ..., 0 0 0]]
```

Next, imported the given ‘network.txt’ file using the pre-defined ‘loadNetwork()’ function. Then, calculated the fitness of all overlay networks using the given ‘evaluateFitness()’ function to return best and average_fitness values.

Have chosen the parents according to their best fitness values using the ‘selection_Ofparents()’ function, two at a time. Tournament selection with k=2 is used for the selection. Called the ‘crossover()’ function over the two overlay networks that are selected in the previous selection operation. A mutation() function is developed that randomly converts bits from ‘0’ to ‘1’ and vice-versa. The new set of networks are replaced with their parent networks and the fitness evaluation is calculated. This process is repeated for given number of iterations and best and average values of fitness are computed and plotted on the graph.

Experiments by varying the given parameters

1. Low, Medium and High population size with all parameters remaining same

popSize = 10, 100, 300 chromLength = 300, iteration_max = 100, crossover_rate = 0.7, mutation_rate = 0.001

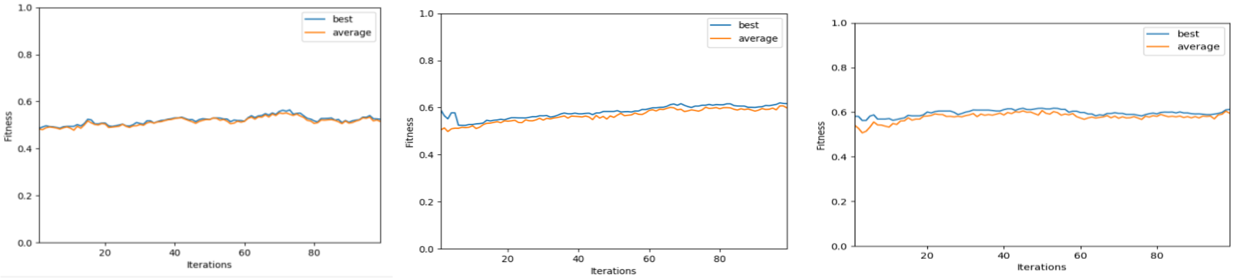
2. Low, Medium and High iterations – 100, 500, 1000

3. Low, medium and High crossover_rate = 0.2, 0.7, 0.9

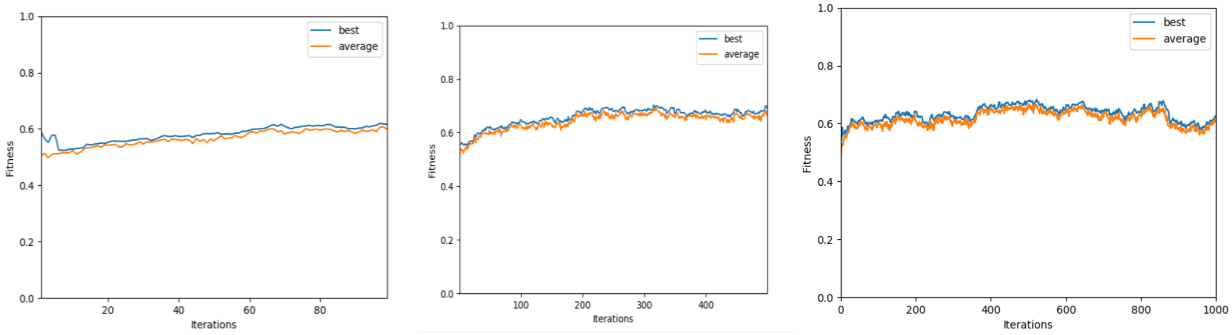
4. Low, medium and High mutation_rate = 0.0001, 0.001, 0.1

Results

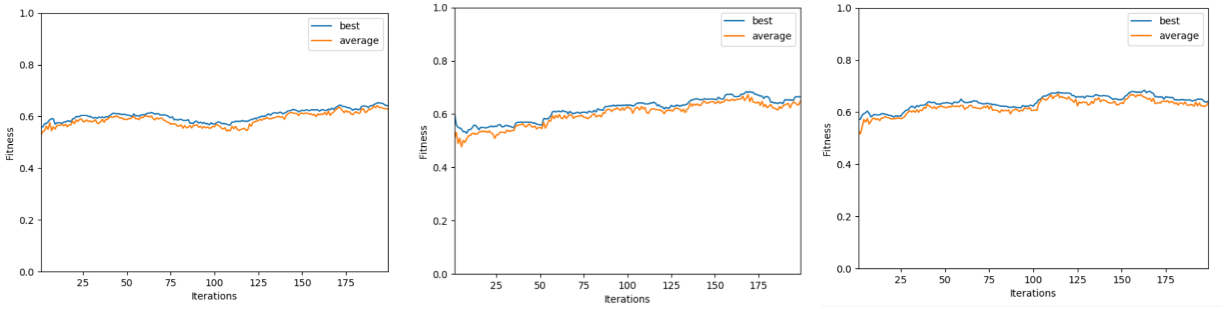
1. popSize = 10, 100, 300



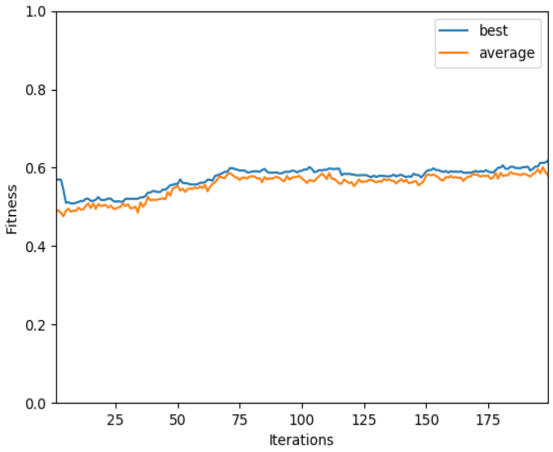
2. Iterations = 100, 500, 1000

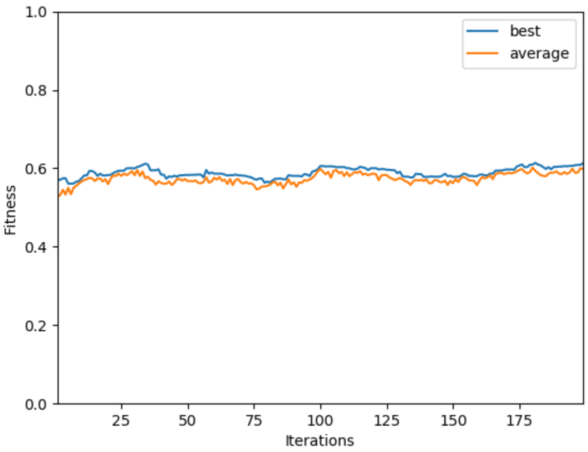


3. Cross_over rate = 0.2,0.7,0.9



4. Mutation rate = 0.01, 0.2





Conclusion:

Increase in population size require more time for execution. Small population is not adequate to get the results of Genetic Algorithm. Large population took more time for execution with little improvement in the results. Hence, according to the requirement of the project we must select proper population size. Similarly, to population size, with the number of iterations the time taken is more with an advantage of better results. With less crossover rate the results are not as expected as the children produced have increased their fitness only by a little value. More crossover rate gave better results. Conversely, less mutation rates gave better results.