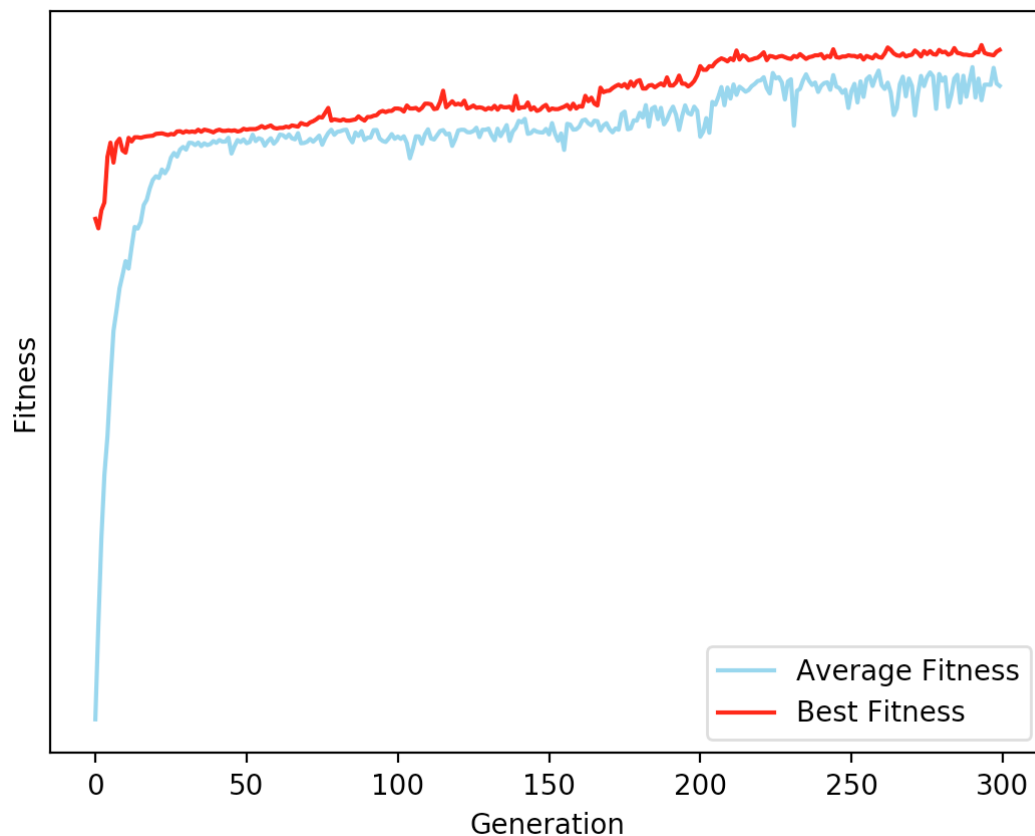Base: runGA(100, 1.0, 0.005,**"GAoutput.txt"**) Generation:300

Population: 100 Crossover rates: 1.0 mutation rate: 0.005

Best: Generation 280 : average fitness 40.13 , best fitness: 57.2

Base on the graph, both the average and the best fitness are growing up. The fitness value growing significant false from generation 0 to 50. Then, after generation 50, the growth become slow.
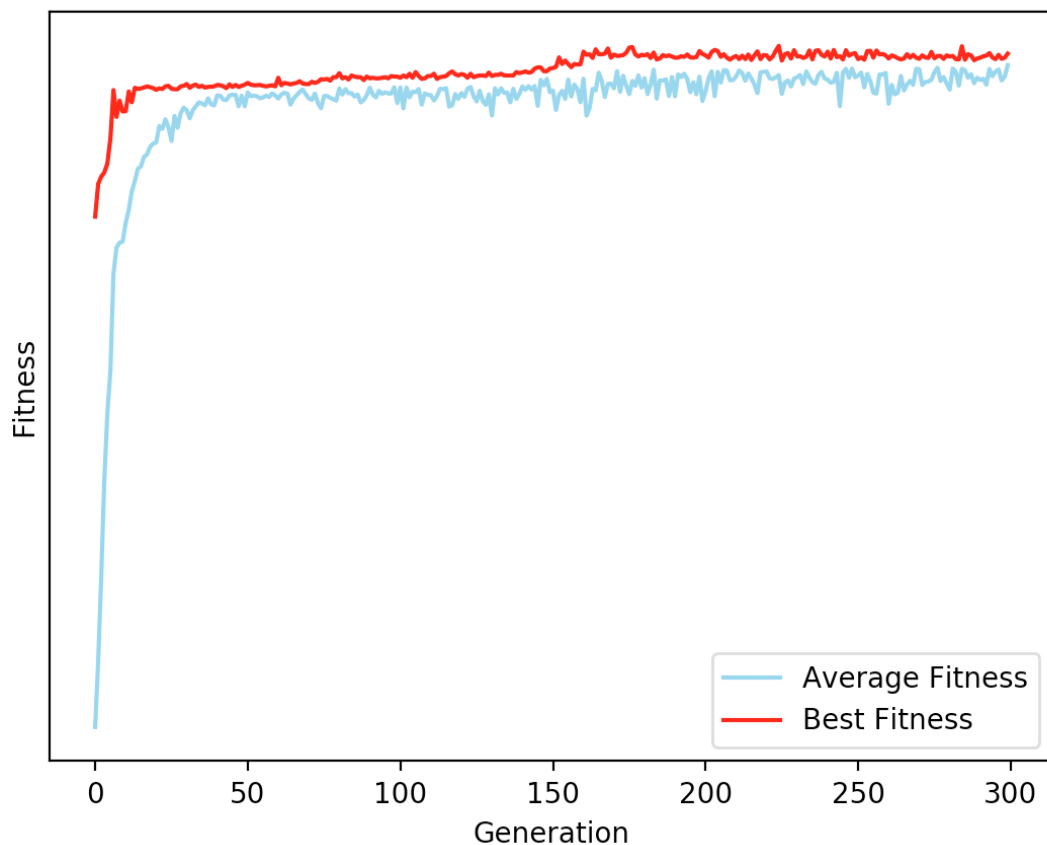
Population:

runGA(50, 1.0, 0.005,**"GAoutput.txt"**) Generation:300

Population: 50 Crossover rates: 1.0 mutation rate: 0.005

Best: Generation: 224 **average fitness**: 11.870399999999995, **best fitness**:32.2

By reduce the population, since it has fewer sample data to do cross over, and as the generation stay the same, the best and average fitness will go down. Vice versa, by increase the population value, it will have better average and best fitness value.
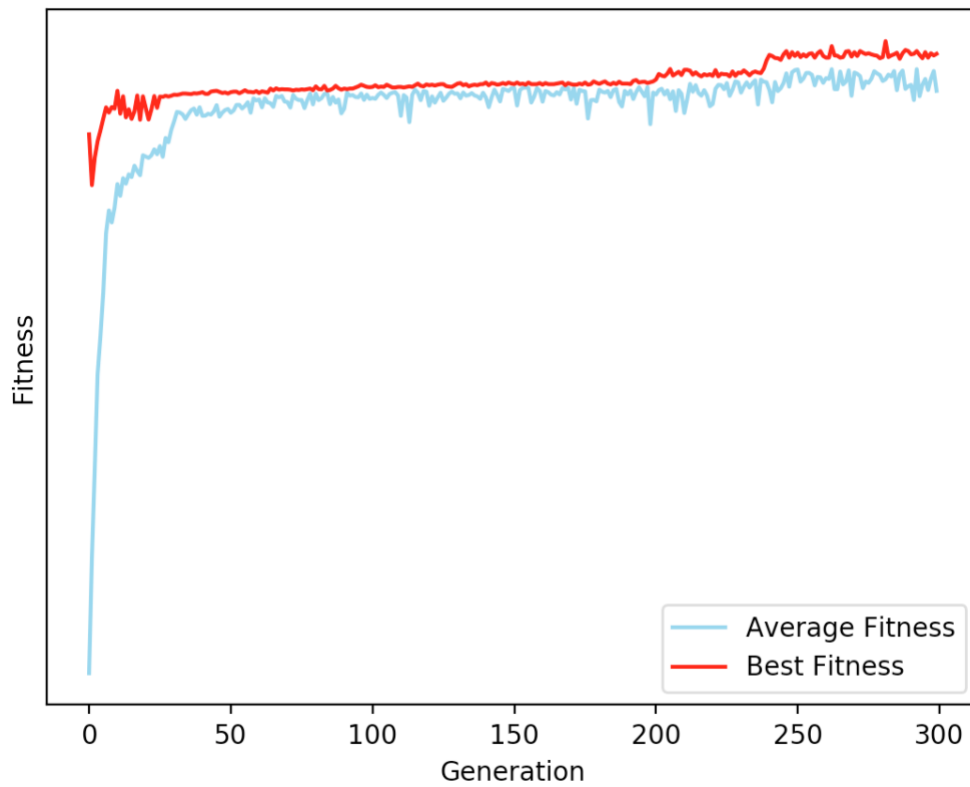
Crossover:

runGA(100, 0.5, 0.005,**"GAoutput.txt"**) Generation:300

Population: 100 Crossover rates: 0.5 mutation rate: 0.005

Best: Generation: 281 **average fitness** : 12.3464 **best fitness:** 40.0

By reduce the crossover rate, since the cross over function randomly generate a number from 0 to 1 and compare to the cross over rate pass into the function. Therefore, by reduce the cross over rate, the function will do less cross over, so it will decrease best and average fitness value.
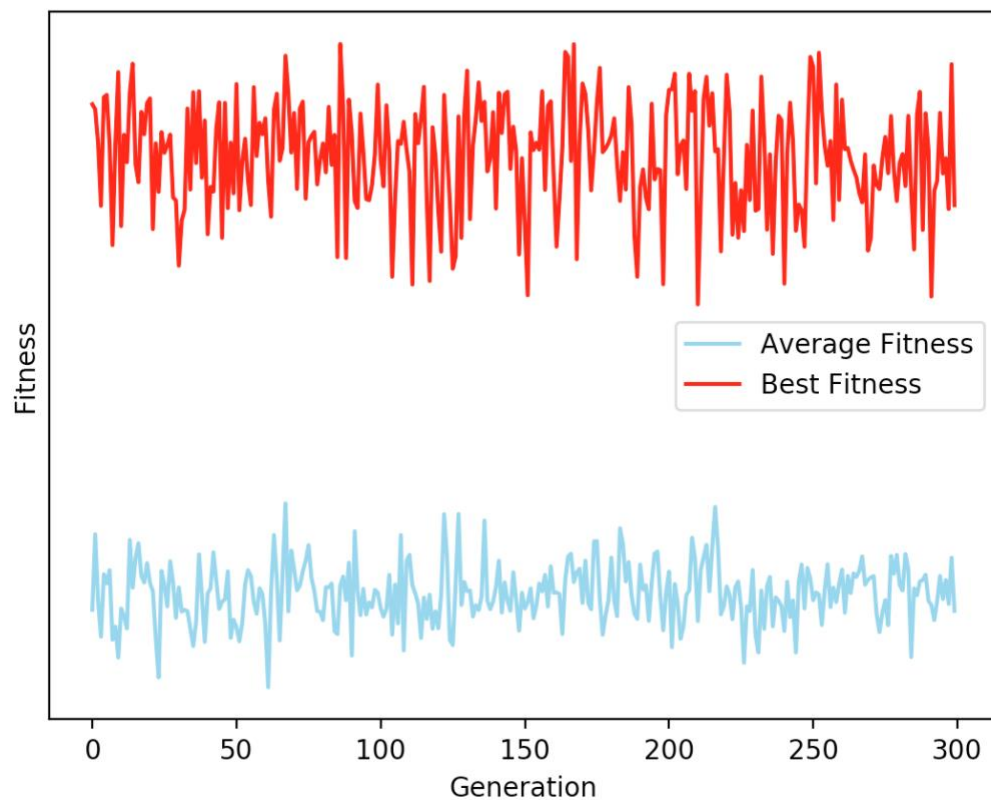
Mutate:

runGA(100, 1.0, 0.9,**"GAoutput.txt"**)

Population: 100 Crossover rates: 1.0 mutation rate: 0.9

Best: Generation: 86 **average fitness** :-445.75679999999994 **best fitness**:2.4

By looking at the graph, the average and best fitness value is not growing as the all the other graph. Because as the mutation rate goes very high, the mutate function will mutate the string very often. Therefore, even though the function finds the optimal string, the mutate function will still mutate it so that it no longer optimal. Thus, by reduce mutation rate to an appropriate low value (like the base case). It will be likely to find the optimal solution.