Artificial Intelligence

Assignment:

The fitness function used for evaluating the algorithms was different than the one mentioned in the assignment.

In my fitness function, for a particular edge, if the colours of the corresponding vertices are different, the fitness value gets incremented. So, the range of the fitness value is from [0-Number of edges].

The Intuition behind using this was instead of considering a vertex and its corresponding edges as a whole, the vertices corresponding to every edge are included in the fitness function. As This function had a greater range for improvement , even the smallest improvement in parents will be considered while calculating its weight and correspondingly, it has a better chance to get passed on and thus, higher chance of retention of improvements during the transition to the subsequent generations.

A. These are the plots after implementing the Genetic Algorithm which is given in the text book.

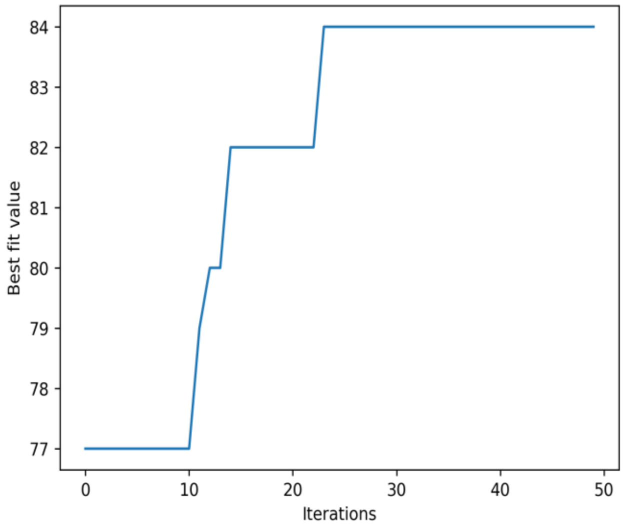
   Parameters:

   - Size of the Population = 100
   - Crossing point = randomly generated number in range 1-50
   - Mutation rate probability = 0.04

The plots after implementing the genetic algorithm for different number of edges were:

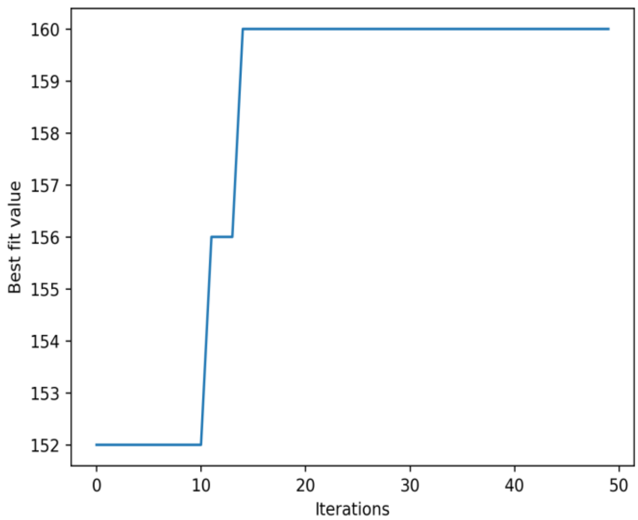
1. Edges = 100

Plot:



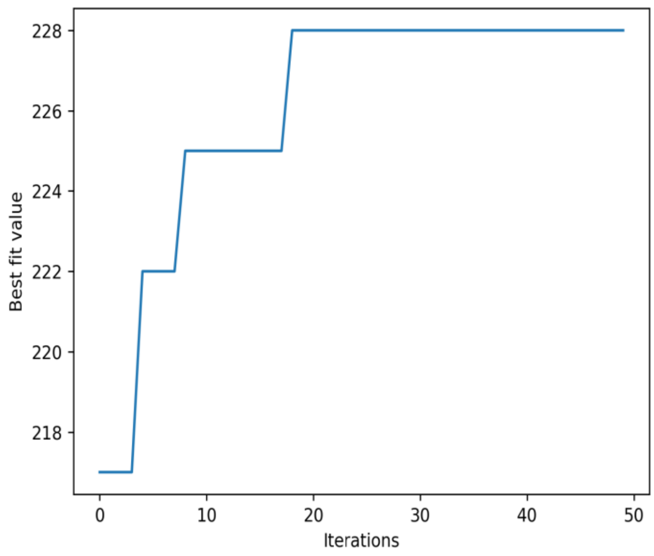
2. Edges = 200

Plot :



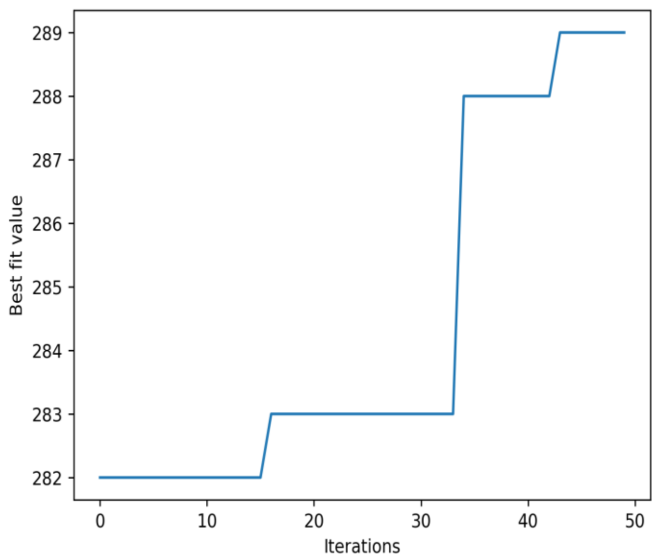
3. Edges = 300

Plot :



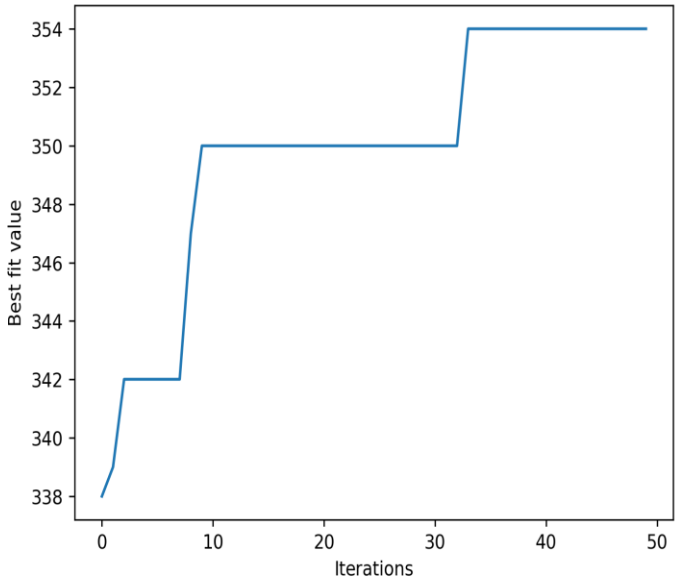
4. Edges = 400

Plot :



5. Edges = 500

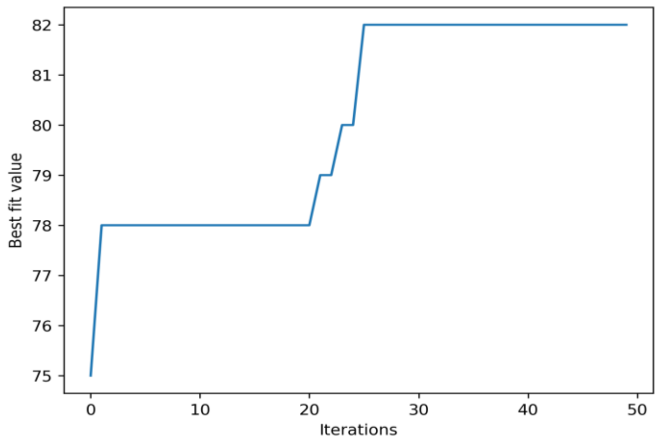
Plot :



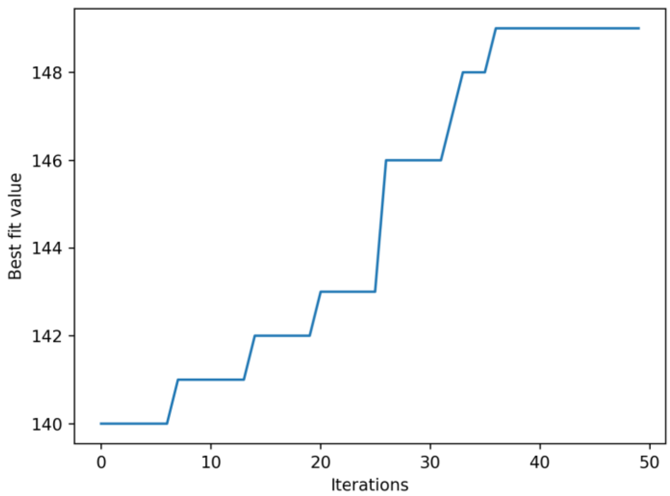
B. The population size of 100 was very big. Due to the large size, the variability increases and as a result, the more fitter states are not passed on effectively to form fitter children .
   The crossover point was randomly chosen in A so in approximately 50% of the cases, the fitter parent does not get passed on more than the other parent if the crossover point is less than 25.
   Mutation rate is also reduced to minimize random changes in the child.
   Changes in parameters:
      Population size = 10
      Crossover point = Chosen randomly in range 20-40.
      Mutation rate = 0.02.

The plots after implementing the modified genetic algorithm for different number of edges were:

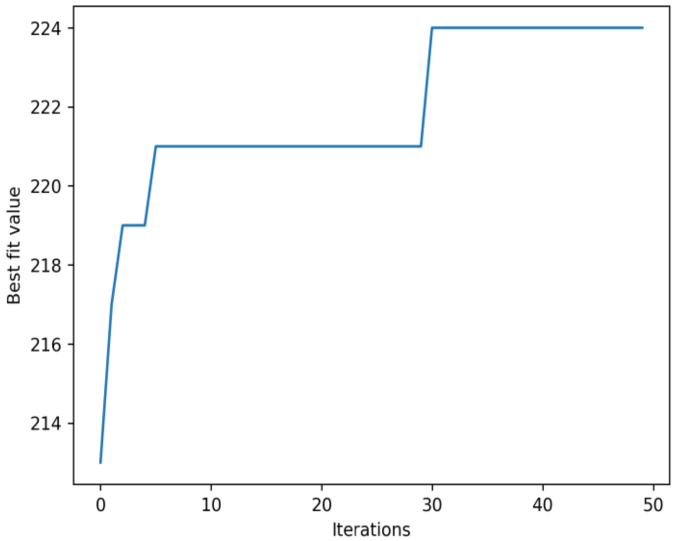
1. Edges = 100



2. Edges = 200

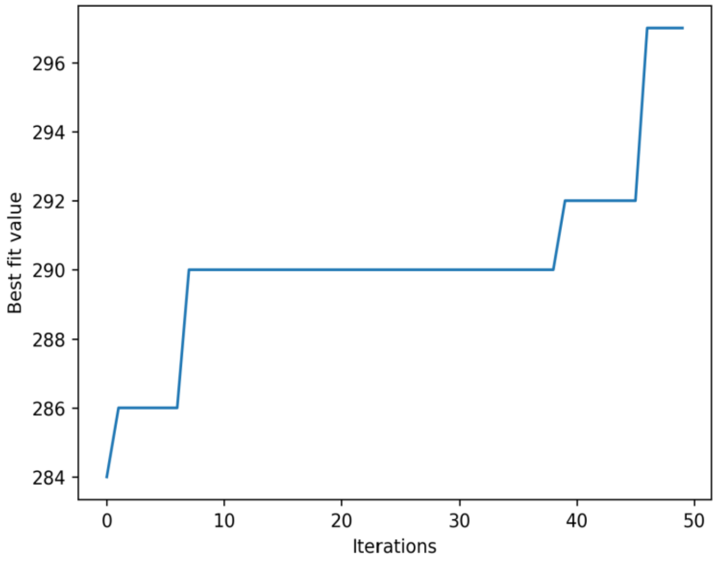


3. Edges = 300

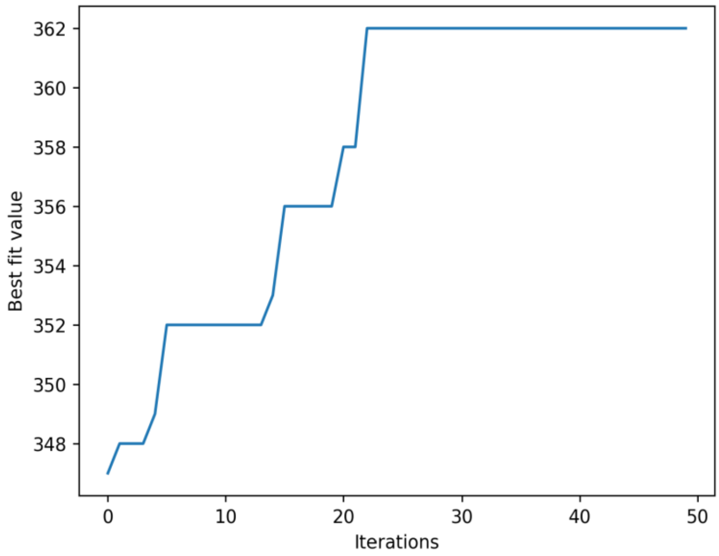


Best fit value
Iterations
224
222
220
218
216
214
0
10
20
30
40
50

4. Edges = 400



5. Edges = 500



Effects of the Modifications on the Algorithm:

- In most cases, The best fit value was increased.
- Generally, The best state is found before the 40th iteration.
- This algorithm does get stuck on some local maxima. So ,it is not very effective in finding the global best fit state.

C. In majority of the cases, the modified algorithm performed better than the initial Algorithm. In this iteration, the range of the crossover point will further be increased to a randomly chosen point in range 30-40 to further increase the percentage of the more fitter parents in the next generation.

Reducing the population size from 10 to 6 because, in case of a higher population size, the number of times the more fitter parents gets picked for reproduction is less as the probability is distributed across the entire population. Thus, reducing the population size might ensure that more fitter parents will get passed on more.

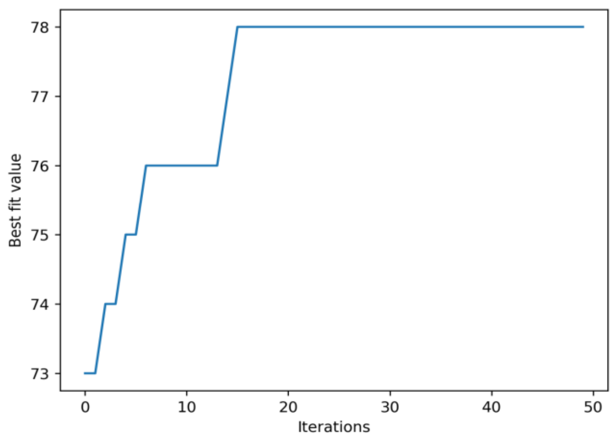
Changes in parameters:

Crossover point = Chosen randomly in range 30-40.

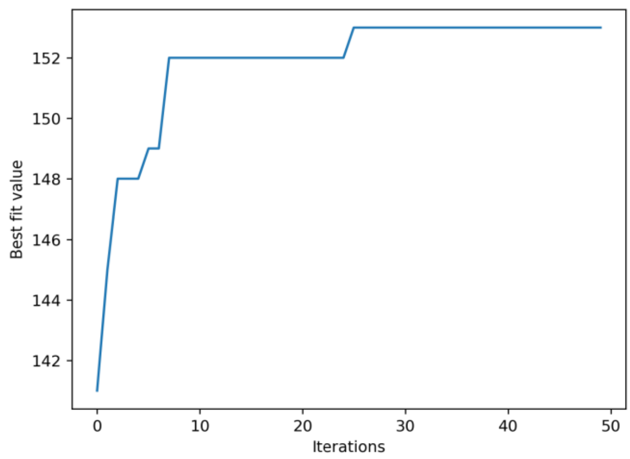
Population size = 6.

The plots after implementing the modified genetic algorithm for different number of edges were:

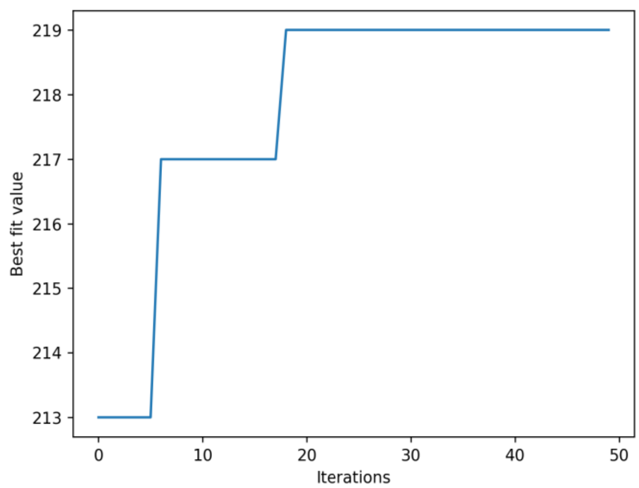
1. Edges = 100



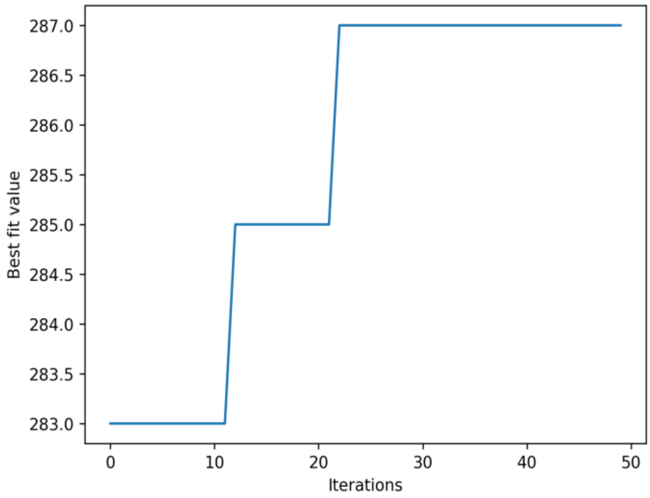
2. Edges = 200



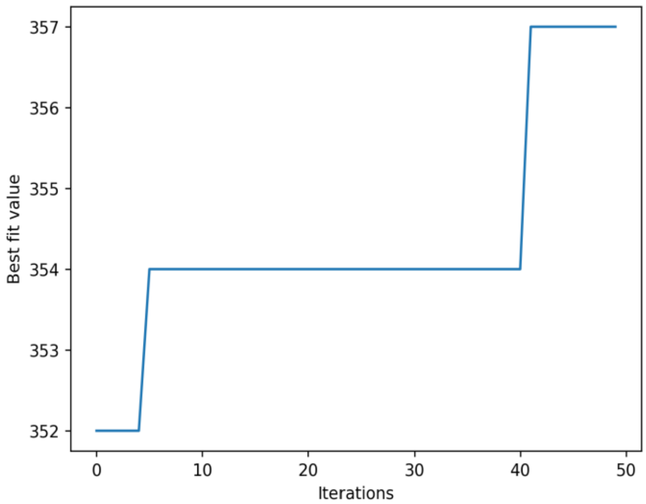
3. Edges = 300



4. Edges = 400



5. Edges = 500



Effects of the Modifications on the Algorithm:

- In most cases, The best fit value was less than the best fit value in the previous algorithm.
- In majority of cases, this algorithm got stuck very early and thus there was minimal improvement in the best fit value.

D. The population size of 6 was clearly very low to find the best fit value through the genetic algorithm because in most cases ,after initialization there was no change in the best fit value. Therefore, the population size is increased to 15.

In this Iteration, Culling and Elitism is implemented.

The top three fittest states are present in the next generation and the three most unfit states are not included in reproduction.

Changes in parameters:

Crossover point = Chosen randomly in range 10-40.

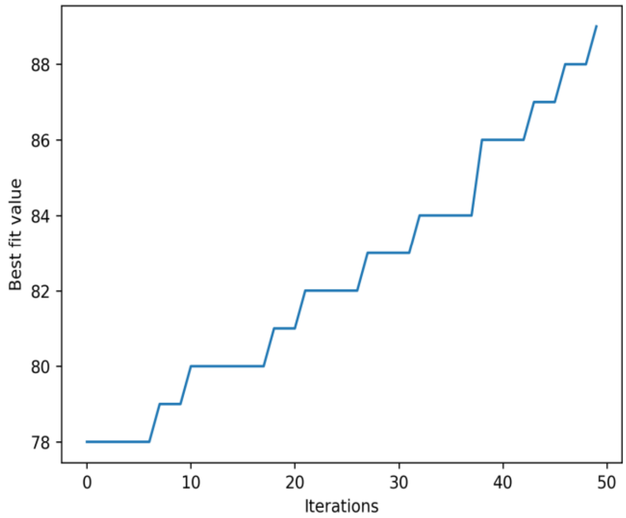
Population size = 15.

Elitism.

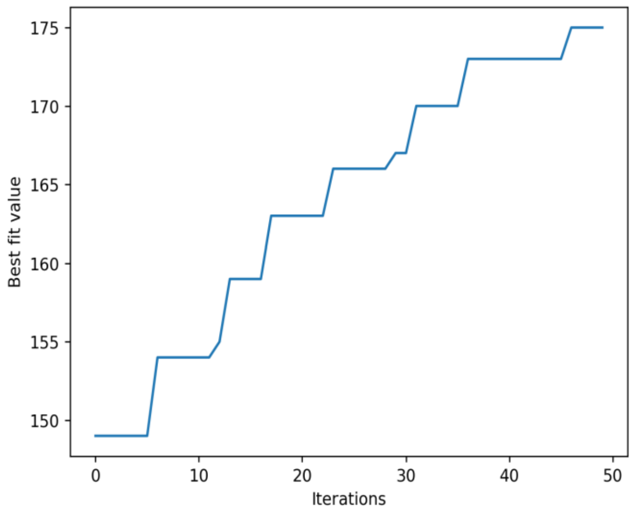
Culling.

The plots after implementing the modified genetic algorithm for different number of edges were:

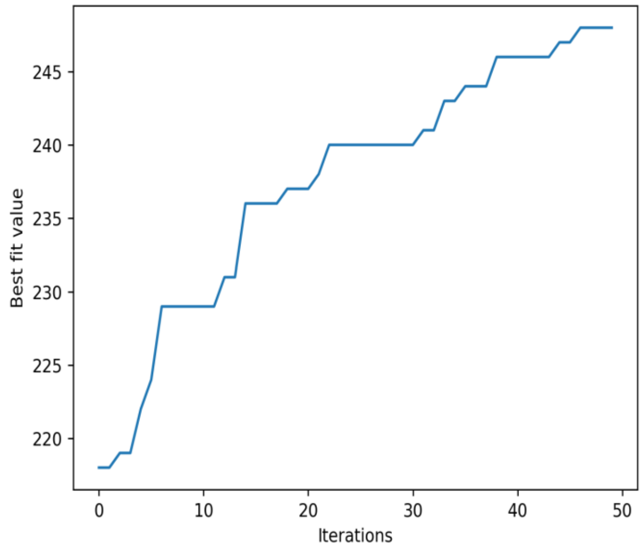
1. Edges = 100



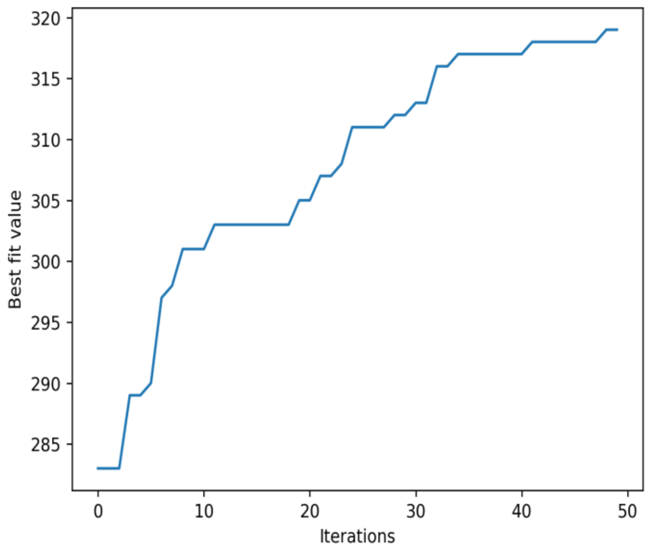
2. Edges = 200



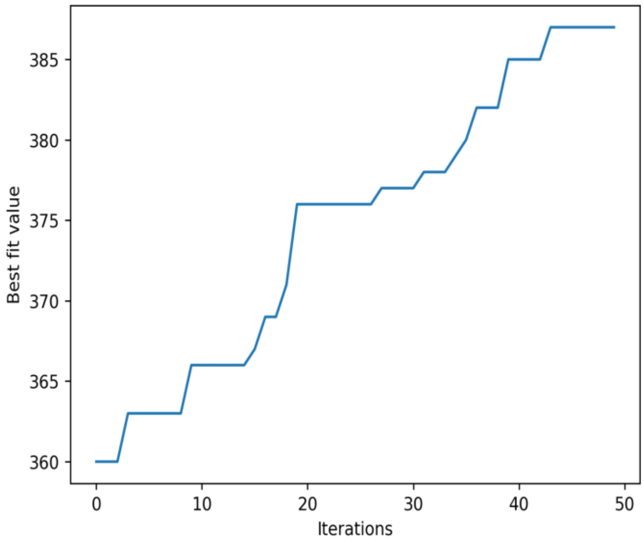
3. Edges = 300



4. Edges = 400



5. Edges = 500



Effects of the Modifications on the Algorithm:

- In Every case, the best fit value was greater than the best fit value in all the previous algorithms.
- In majority of cases, this algorithm does not get stuck throughout and thus there was significant improvement in the best fit value.
- This algorithm can be used to find the global best state.

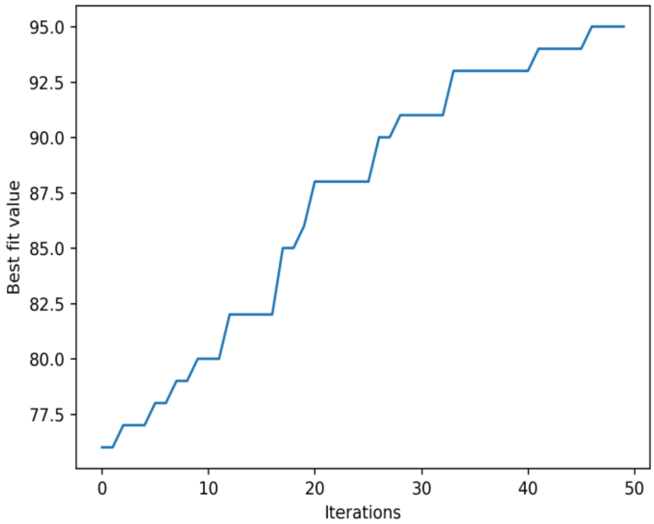
E. The last iteration of the modified algorithm gave the most promising results. So the culling and elitism parameters will kept same. The next generation child was made of parent1 till the crossover point and then the child consisted of parent2.In this iteration, the parents will not be passed on as it is, rather the color of each vertex will come from either parent1 or parent2 with a probability of 0.5 from parent1 and 0.5 from parent2.

Changes in parameters:

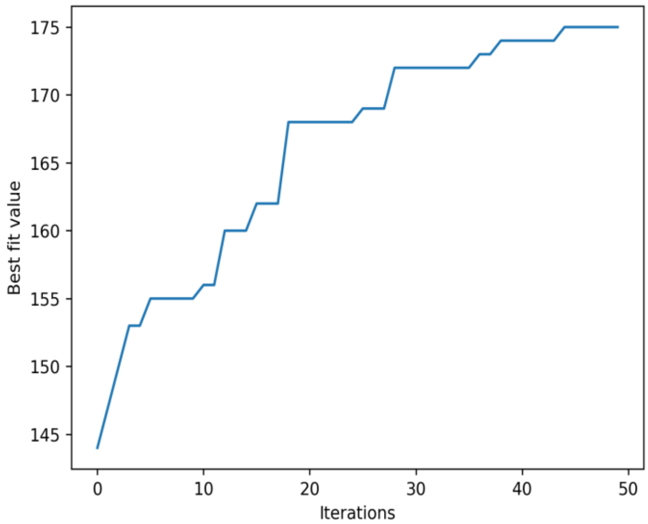
Makeup of the next generation.

The plots after implementing the modified genetic algorithm for different number of edges were:

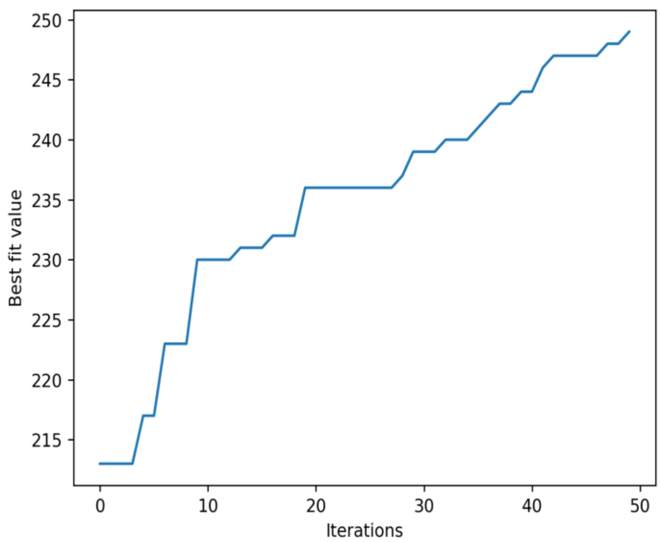
1. Edges = 100



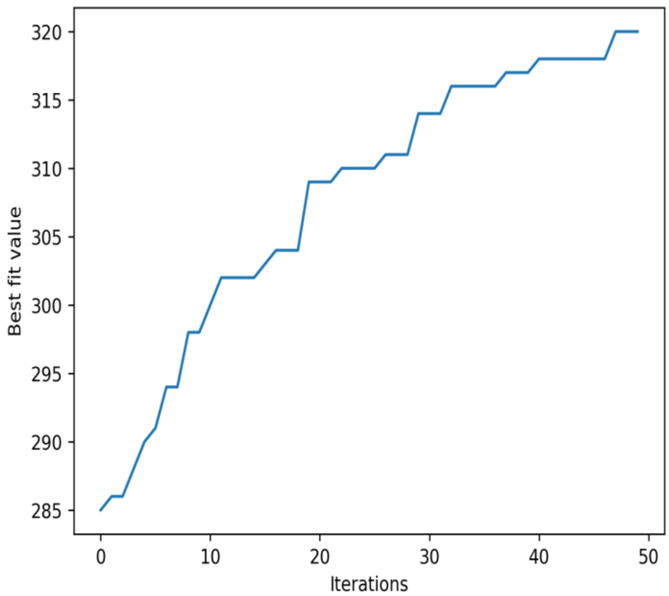
2. Edges = 200



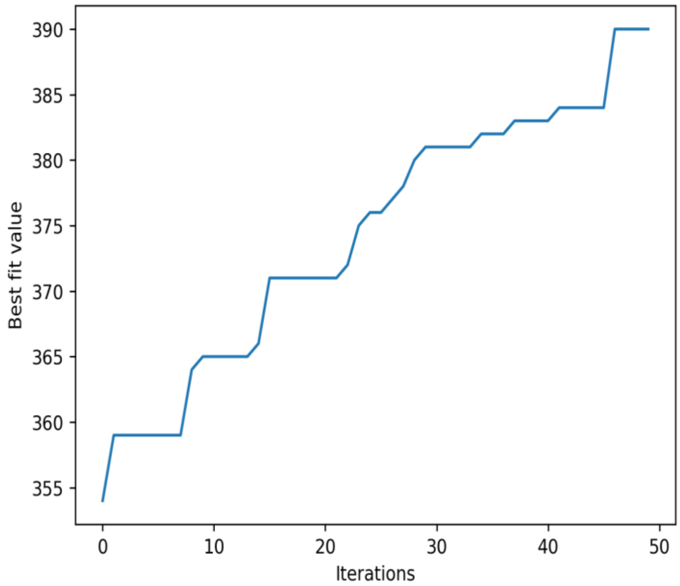
3. Edges = 300.



4. Edges = 400



5. Edges = 500



Effects of the Modifications on the Algorithm:

- In Every case, the best fit value was similar to the best fit value in the previous algorithm.
- Since this algorithm incorporates more randomness in the construction of the child state, this algorithm is less likely to get stuck in a local maxima. Thus this is the improved algorithm which is used for evaluation.

Outputs:

A. For edges = 50,

```
Roll no: 2020A7PS0982G
Number of Edges : 50
Best state :
{0: 'G', 1: 'B', 2: 'B', 3: 'B', 4: 'G', 5: 'R', 6: 'G', 7: 'G', 8: 'R', 9: 'G', 10: 'B', 11: 'B', 1
2: 'B', 13: 'R', 14: 'R', 15: 'R', 16: 'G', 17: 'B', 18: 'R', 19: 'B', 20: 'R', 21: 'G', 22: 'B', 23
: 'G', 24: 'B', 25: 'G', 26: 'B', 27: 'G', 28: 'G', 29: 'B', 30: 'B', 31: 'R', 32: 'B', 33: 'G', 34:
 'R', 35: 'R', 36: 'G', 37: 'B', 38: 'G', 39: 'B', 40: 'B', 41: 'R', 42: 'B', 43: 'R', 44: 'B', 45:
'B', 46: 'B', 47: 'G', 48: 'R', 49: 'G'}
Fitness Value of best state : 50
Time taken : 1.0145995616912842
```

B. For edges = 100,

```
Roll no: 2020A7PS0982G
Number of Edges : 100
Best state :
{0: 'R', 1: 'R', 2: 'B', 3: 'G', 4: 'B', 5: 'R', 6: 'B', 7: 'G', 8: 'R', 9: 'G', 10: 'B', 11: 'B', 1
2: 'G', 13: 'B', 14: 'R', 15: 'R', 16: 'R', 17: 'B', 18: 'R', 19: 'G', 20: 'B', 21: 'G', 22: 'R', 23
: 'G', 24: 'R', 25: 'R', 26: 'G', 27: 'R', 28: 'B', 29: 'G', 30: 'R', 31: 'G', 32: 'G', 33: 'B', 34:
 'R', 35: 'B', 36: 'B', 37: 'R', 38: 'B', 39: 'B', 40: 'B', 41: 'B', 42: 'B', 43: 'R', 44: 'R', 45:
'R', 46: 'G', 47: 'R', 48: 'R', 49: 'G'}
Fitness Value of best state : 50
Time taken : 16.56382727622986
```

C. For edges = 200,

```
Roll no: 2020A7PS0982G
Number of Edges : 200
Best state :
{0: 'R', 1: 'R', 2: 'B', 3: 'R', 4: 'G', 5: 'R', 6: 'G', 7: 'R', 8: 'B', 9: 'G', 10: 'B', 11: 'R', 1
2: 'B', 13: 'G', 14: 'B', 15: 'B', 16: 'R', 17: 'B', 18: 'B', 19: 'R', 20: 'G', 21: 'R', 22: 'B', 23
: 'B', 24: 'B', 25: 'B', 26: 'B', 27: 'R', 28: 'G', 29: 'B', 30: 'B', 31: 'R', 32: 'B', 33: 'B', 34:
 'G', 35: 'G', 36: 'G', 37: 'R', 38: 'B', 39: 'G', 40: 'G', 41: 'G', 42: 'G', 43: 'G', 44: 'R', 45:
'B', 46: 'R', 47: 'R', 48: 'G', 49: 'R'}
Fitness Value of best state : 40
Time taken : 45.019633293151855
```