Metaheuristic Optimization

Assignment 1

Name: Mohammed Alom

Student Id: R00144214

Cork Institute of Technology

Part 1: NP-completeness

Q1.

The student number the last digit is four, so I will do the following formula.

F = (-w3 V w4) /\ (w1 V -w2 V w3 V -w4 V -w5 V w6)

= (- w3 V w4 V y1) /\ (- w3 V w4 V – y1) /\ (w1 V – w2 V y2) /\ (w3 V – w4 V – y2)

/\ (- w5 V w6 V y3) /\ (- w5 V w6 V -y3)

F = (- w3 V w4 V y1) /\ (- w3 V w4 V – y1) /\ (w1 V – w2 V y2) /\ (w3 V – w4 V – y2) /\ (- w5 V w6 V y3) /\ (- w5 V w6 V -y3)

(F V T V T) /\ (F V T V F) /\ (T V F V T) /\ (T V F V F) /\ (F V T V T) /\ (F V T V F)

F = (-w3 V w4) /\ (w1 V -w2 V w3 V -w4 V -w5 V w6)

(F V T) /\ (T V F V T V F V F V T)

Q2.

Here, I will do the first and fourth clauses as the first name first letter is M.

(- w3 V w4 V y1) /\ (w3 V – w4 V – y2)

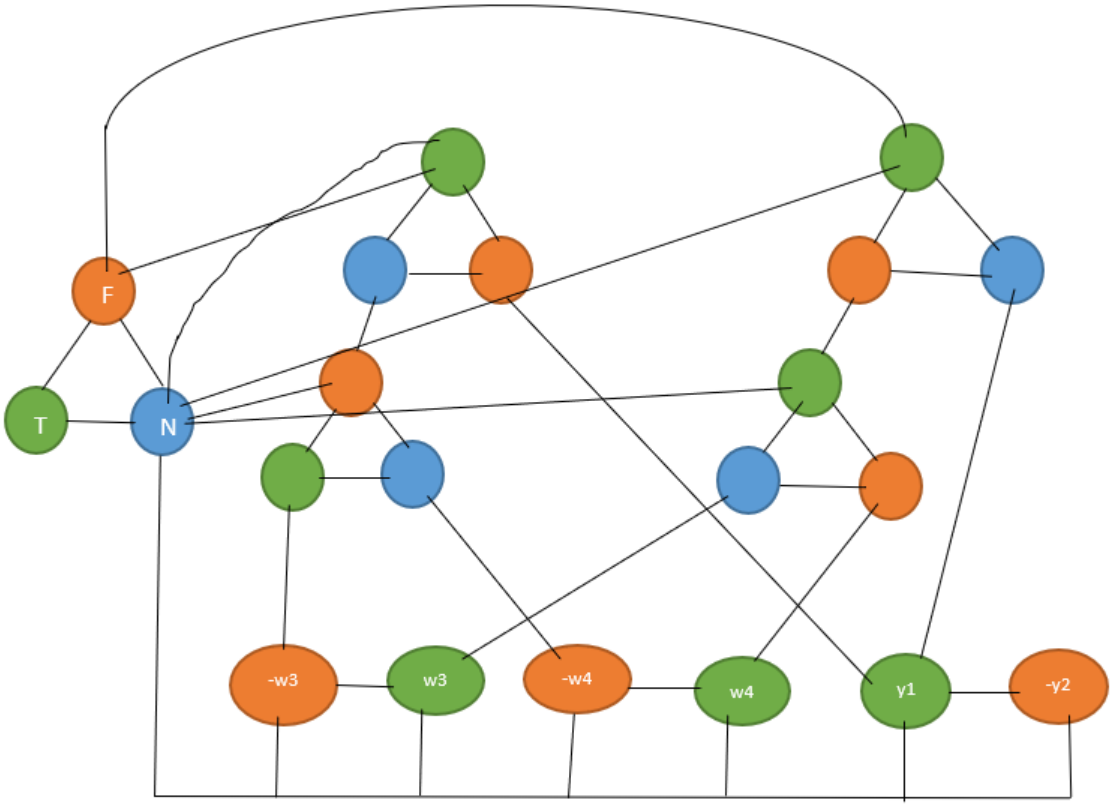

Fig-1 3Col

Here, T = True

F = False

N = Neutral

Each variable has two literal nodes w3 -w3.

Each clause is connected with the blue color, which is neutral.

Furthermore, for each clause added six more clause nodes. All clauses are satisfied, so the above two clauses are also satisfied.

Conclude proof that 3COL Є P -> 3SAT Є P.

Since each clause has a true value and the complete formula also evaluates to true, a correct solution has been satisfied.

Part 2: Genetic Algorithms

For an optimum solution, the genetic algorithm is widely used in different applications. Here genetic algorithm is used in the traveling salesman problem to find out the optimal solution. The core concept of the genetic algorithm is Darwin's theory, which is survival in the fittest.

The basic structure of the genetic algorithm:

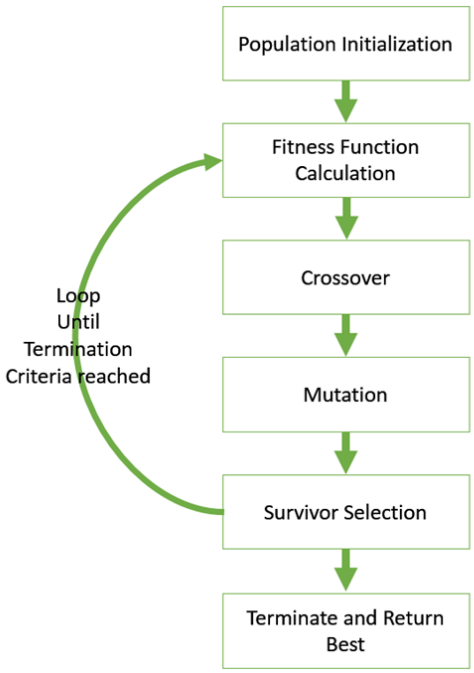

Fig -1 – Basic Structure of the GA [1]

Initial Population (Initial Solution)

The first step of the GA is to initialize the population. It can be performed in several ways. Here are the two approaches.

1. Random approach.
2. Heuristic approach.

Selection

In the selection steps, parents are selected for mating to generate the offspring, and that offspring used to create a new generation. In this GA TSP problem here, Binary Tournament Selection has been used for all six configurations.

Crossover

To perform the crossover operation at first will select two individuals: the parents to get the two new individuals, which is the child.

For this experiment, here two crossover methods used.

1. Uniform order based crossover
2. Order 1 crossover

Uniform order based crossover

here at first selecting the smallest one from both parents and then create the binary template. The binary slot slicing part of the array then finds the empty sloat and adds to the binary sloat. Once the adding operation finished, then finding the genes' zero position in the binary template then sorting the genes and again replacing the genes inside the binary template.

Order 1 crossover

here at first, select randomly part of the parent one. Then drop down those values to the child one, and the remaining value placed according to the parent2.

Mutation

A mutation is to ensure diversity in the chromosome. It happens when a random number is generated to decide whether a mutation in the chromosome is required. If the random number is greater than the current mutation probability, then no mutation is performed; otherwise, the chromosome is mutated.

Here, two approaches have is followed.

1. Inversion Mutation.
2. Scramble Mutation.

Inversion Mutation

Inversion mutation is similar to scramble mutation, but instead of shuffling the subset's portion, it reverses the entire subset.

Scramble Mutation

A subset of the chromosome is selected in the scramble mutation stage, and then their values are shuffled randomly.

GA-TSP Result Analysis with Different Instances

Instance: inst-0.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
1	**100**	**500**	**Random**	**Order 1 crossover**	**Inversion**	**0.05**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**290**	**25304020.0219857**	**21626682.733027004**
2	**408**	**24442130.21153252**	**21360181.590954565**
3	**129**	**24123435.39584713**	**21431578.7331915**
4	**415**	**24814806.97362244**	**22030247.65952938**
5	***126***	***23815340.052803557***	***21862895.310869213***

Test running no 5 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 5
('BEST INITIAL SOLUATION: ', 23815340.052803557)

TSP FILE NAME : ./data/inst-0.tsp
POPULATION SIZE : 100
MUTATION RATE : 0.05
TOTAL ITERATION : 500
CONFIG NUMBER : 1
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.RANDOM
CROSSOVER TYPE : Crossover.ORDER_ONE_CROSS_OVER
MUTATION TYPE: Mutation.INVERSION_MUTATION
('ITERATION : ', 0, 'BEST : ', 23788904.637023266)
('ITERATION : ', 0, 'BEST : ', 23782752.398495827)
('ITERATION : ', 0, 'BEST : ', 23666703.167431)
('ITERATION : ', 0, 'BEST : ', 23658195.368414853)
('ITERATION : ', 0, 'BEST : ', 23639843.89386298)
('ITERATION : ', 0, 'BEST : ', 23638949.141991246)
('ITERATION : ', 0, 'BEST : ', 23592305.002344914)
('ITERATION : ', 0, 'BEST : ', 23590776.478350405)
('ITERATION : ', 0, 'BEST : ', 23395996.72080404)
('ITERATION : ', 0, 'BEST : ', 23394543.866193555)
('ITERATION : ', 0, 'BEST : ', 23360953.424663316)
('ITERATION : ', 3, 'BEST : ', 23342869.373221654)
('ITERATION : ', 4, 'BEST : ', 23177321.196384408)
('ITERATION : ', 4, 'BEST : ', 23086765.136947528)
('ITERATION : ', 4, 'BEST : ', 23065546.55116892)
('ITERATION : ', 23, 'BEST : ', 22898044.681372806)
('ITERATION : ', 30, 'BEST : ', 22813375.980144884)
('ITERATION : ', 30, 'BEST : ', 22771833.64090916)
('ITERATION : ', 30, 'BEST : ', 22739657.286056265)
('ITERATION : ', 31, 'BEST : ', 22710224.65644197)
('ITERATION : ', 31, 'BEST : ', 22620884.327842016)
('ITERATION : ', 31, 'BEST : ', 22587139.748079095)
('ITERATION : ', 31, 'BEST : ', 22561886.23427253)
('ITERATION : ', 31, 'BEST : ', 22513403.577436507)
('ITERATION : ', 31, 'BEST : ', 22304522.389363278)
('ITERATION : ', 31, 'BEST : ', 22280713.3630084)
('ITERATION : ', 126, 'BEST : ', 21862895.310869213)
('BEST SOLUTION: ', 21862895.310869213)
```

Instance: inst-0.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
2	**100**	**500**	**Random**	**Uniform Crossover**	**Scramble**	**0.05**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**180**	**25304020.0219857**	**21105624.632543188**
2	**485**	**25884110.48743124**	**21373619.78395664**
3	**31**	**24804760.32484403**	**21675903.994911443**
4	**130**	**25119270.493506966**	**21696962.79390975**
5	***108***	***25406640.933535133***	***21703373.942602187***

Test running no 5 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 5
('BEST INITIAL SOLUATION: ', 25406640.933535133)

TSP FILE NAME : ./data/inst-0.tsp
POPULATION SIZE : 100
MUTATION RATE : 0.05
TOTAL ITERATION : 500
CONFIG NUMBER : 2
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.RANDOM
CROSSOVER TYPE : Crossover.UNIFORM_CROSS_OVER
MUTATION TYPE: Mutation.SCRAMBLE_MUTATION
('ITERATION : ', 0, 'BEST : ', 24418735.821290564)
('ITERATION : ', 0, 'BEST : ', 24324631.347821575)
('ITERATION : ', 0, 'BEST : ', 23917402.336600427)
('ITERATION : ', 0, 'BEST : ', 23911345.94710057)
('ITERATION : ', 0, 'BEST : ', 22883653.949369185)
('ITERATION : ', 2, 'BEST : ', 22537087.177447785)
('ITERATION : ', 2, 'BEST : ', 22356487.18460083)
('ITERATION : ', 2, 'BEST : ', 21993497.108069282)
('ITERATION : ', 9, 'BEST : ', 21835822.57612775)
('ITERATION : ', 108, 'BEST : ', 21703373.942602187)
('BEST SOLUTION: ', 21703373.942602187)
```

Instance: inst-13.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
2	**200**	**500**	**Random**	**Uniform Crossover**	**Scramble**	**0.1**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**81**	**125245935.10856143**	**106420680.55181614**
2	**176**	**130698931.09522116**	**106692711.46668994**
3	**495**	**1327738800.50098448**	**106664678.13651152**
4	**425**	**129392631.14663124**	**107048690.00453603**
5	***478***	***122896784.8652386***	***105838944.01366356***

Test running no 5 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 5
BEST INITIAL SOLUATION:  122896784.8652386

TSP FILE NAME : ./TSPdata/inst-13.tsp
POPULATION SIZE : 200
MUTATION RATE : 0.1
TOTAL ITERATION : 500
CONFIG NUMBER : 2
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.RANDOM
CROSSOVER TYPE : Crossover.UNIFORM_CROSS_OVER
MUTATION TYPE: Mutation.SCRAMBLE_MUTATION
ITERATION :  0 BEST :  113906797.86774424
ITERATION :  0 BEST :  111290912.30315404
ITERATION :  7 BEST :  110308037.14395969
ITERATION :  35 BEST :  109632924.79430711
ITERATION :  46 BEST :  109444355.02785248
ITERATION :  57 BEST :  108475414.73558486
ITERATION :  116 BEST :  108144138.39431387
ITERATION :  164 BEST :  107826370.61560386
ITERATION :  164 BEST :  107609621.19466144
ITERATION :  166 BEST :  107525079.77398619
ITERATION :  379 BEST :  106215102.48370579
ITERATION :  478 BEST :  105838944.01366356
BEST SOLUTION:  105838944.01366356
```

Instance: inst-5.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
3	**200**	**500**	**Random**	**Order 1 crossover**	**Scramble**	**0.1**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**130**	**461239164.77997386**	**423309423.340693**
2	**154**	**466914688.5550637**	**422790159.4573943**
3	**130**	**459342821.3316884**	**419904421.0534524**
4	**216**	**45064615.92350155**	**418428287.57919115**
5	**379**	**454952606.1129233**	**421951177.68356293**

Test running no 1 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 1
('BEST INITIAL SOLUATION: ', 461239164.77997386)

TSP FILE NAME : ./data/inst-5.tsp
POPULATION SIZE : 200
MUTATION RATE : 0.1
TOTAL ITERATION : 500
CONFIG NUMBER : 3
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.RANDOM
CROSSOVER TYPE : Crossover.ORDER_ONE_CROSS_OVER
MUTATION TYPE: Mutation.SCRAMBLE_MUTATION
('ITERATION : ', 0, 'BEST : ', 460711845.4336542)
('ITERATION : ', 0, 'BEST : ', 458276664.1166581)
('ITERATION : ', 0, 'BEST : ', 456333259.8619329)
('ITERATION : ', 0, 'BEST : ', 455448821.882391)
('ITERATION : ', 0, 'BEST : ', 446212725.22168916)
('ITERATION : ', 0, 'BEST : ', 445741487.63652045)
('ITERATION : ', 0, 'BEST : ', 442581407.31480867)
('ITERATION : ', 0, 'BEST : ', 441624652.3697981)
('ITERATION : ', 0, 'BEST : ', 438729137.66378695)
('ITERATION : ', 0, 'BEST : ', 435461026.93248373)
('ITERATION : ', 0, 'BEST : ', 428857157.197311)
('ITERATION : ', 40, 'BEST : ', 428041951.89915353)
('ITERATION : ', 46, 'BEST : ', 427571966.01922804)
('ITERATION : ', 97, 'BEST : ', 425658998.1551874)
('ITERATION : ', 130, 'BEST : ', 423309423.340693)

('BEST SOLUTION: ', 423309423.340693)
```

Instance: inst-0.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
4	**300**	**500**	**Random**	**Uniform Crossover**	**Inversion**	**0.2**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**333**	**25181303.048181646**	**21132951.39947786**
2		**25709324.38339636**	**21456880.924304206**
3	**144**	**24483377.81511987**	**21553693.561969236**
4	***400***	***25393777.033949822***	***21810115.395905446***
5	**148**	**24769918.503492016**	**21592381.33643805**

Test running no 4 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 4
BEST INITIAL SOLUATION:  25393777.033949822

TSP FILE NAME : ./TSPdata/inst-0.tsp
POPULATION SIZE : 300
MUTATION RATE : 0.2
TOTAL ITERATION : 500
CONFIG NUMBER : 4
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.RANDOM
CROSSOVER TYPE : Crossover.UNIFORM_CROSS_OVER
MUTATION TYPE: Mutation.INVERSION_MUTATION
ITERATION :  0 BEST :  22545840.0908598
ITERATION :  0 BEST :  22384833.084288754
ITERATION :  0 BEST :  22323746.84422943
ITERATION :  2 BEST :  22289738.047337346
ITERATION :  18 BEST :  22201627.728134256
ITERATION :  23 BEST :  22180378.04323297
ITERATION :  23 BEST :  22114851.757623773
ITERATION :  23 BEST :  22045086.604381554
ITERATION :  23 BEST :  22022841.001338575
ITERATION :  23 BEST :  21972955.245586634
ITERATION :  195 BEST :  21965954.069542836
ITERATION :  195 BEST :  21885709.21937863
ITERATION :  322 BEST :  21849944.790513515
ITERATION :  400 BEST :  21810115.395905446
BEST SOLUTION:  21810115.395905446
```

Instance: inst-0.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
5	**100**	**500**	**Heuristic**	**Order 1 crossover**	**Scramble**	**0.2**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**340**	**25304020.0219857**	**211555997.040908664**
2	***111***	***23552129.632993057***	***21732680.001899328***
3	**231**	**25503040.480132144**	**21643469.94527176**
4	**222**	**24188668.001459766**	**21406536.61891494**
5	**181**	**24100490.524118714**	**21748872.725402698**

Test running no 5 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 2
('BEST INITIAL SOLUATION: ', 23552129.632993057)

TSP FILE NAME : ./data/inst-0.tsp
POPULATION SIZE : 100
MUTATION RATE : 0.2
TOTAL ITERATION : 500
CONFIG NUMBER : 5
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.HEURISTIC
CROSSOVER TYPE : Crossover.ORDER_ONE_CROSS_OVER
MUTATION TYPE: Mutation.SCRAMBLE_MUTATION
('ITERATION : ', 0, 'BEST : ', 23258820.817935184)
('ITERATION : ', 0, 'BEST : ', 22676403.82958652)
('ITERATION : ', 9, 'BEST : ', 22641973.65278264)
('ITERATION : ', 9, 'BEST : ', 22495179.16118465)
('ITERATION : ', 16, 'BEST : ', 22351398.506194446)
('ITERATION : ', 23, 'BEST : ', 21785870.641966518)
('ITERATION : ', 111, 'BEST : ', 21732680.001899328)
('BEST SOLUTION: ', 21732680.001899328)
```

Instance: inst-13.tsp

Config. No	Population size	Total Iteration	Initial Solution	Crossover	Mutation	Mutation Rate	Selection
6	**300**	**500**	**Heuristic**	**Uniform crossover**	**Inversion**	**0.3**	**B. Tournament**

Performance of the Algorithm

Test Run No	Iteration Took	Initial Solution	Best Solution
1	**138**	**130713495.85379668**	**108214493.43096422**
2	***404***	***123390120.00117944***	***109055767.40520926***
3	**447**	**129148870.89532204**	**108509592.677678**
4	**11**	**128727222.7510627**	**106779232.83597541**
5	**333**	**123163483.96626519**	**109227033.64974199**

Test running no 4 has a screenshot below.

Result Screenshot

```
NUMBER OF RUN: 4
('BEST INITIAL SOLUATION: ', 128727222.7510627)

TSP FILE NAME : ./data/inst-13.tsp
POPULATION SIZE : 300
MUTATION RATE : 0.3
TOTAL ITERATION : 500
CONFIG NUMBER : 6
SELECTION TYPE : Selection.BINARY_TOURNAMENT_SELECTION
INITIAL SOLUTION TYPEe : Initial_Solution.HEURISTIC
CROSSOVER TYPE : Crossover.UNIFORM_CROSS_OVER
MUTATION TYPE: Mutation.INVERSION_MUTATION
('ITERATION : ', 0, 'BEST : ', 127497356.16824265)
('ITERATION : ', 0, 'BEST : ', 127311187.94748695)
('ITERATION : ', 0, 'BEST : ', 127309351.44088607)
('ITERATION : ', 0, 'BEST : ', 126357639.81556927)
('ITERATION : ', 0, 'BEST : ', 125407209.5547845)
('ITERATION : ', 0, 'BEST : ', 125339642.87427898)
('ITERATION : ', 0, 'BEST : ', 124419517.76575553)
('ITERATION : ', 0, 'BEST : ', 124417382.33683476)
('ITERATION : ', 0, 'BEST : ', 123614127.75113891)
('ITERATION : ', 0, 'BEST : ', 123468858.09086975)
('ITERATION : ', 0, 'BEST : ', 123405256.1800957)
('ITERATION : ', 0, 'BEST : ', 122499004.91796538)
('ITERATION : ', 0, 'BEST : ', 121159549.8317724)
('ITERATION : ', 0, 'BEST : ', 120404727.9061477)
('ITERATION : ', 0, 'BEST : ', 120221635.64226124)
('ITERATION : ', 0, 'BEST : ', 120137198.3964822)
('ITERATION : ', 0, 'BEST : ', 119764234.18393773)
('ITERATION : ', 0, 'BEST : ', 119322355.7714265)
('ITERATION : ', 0, 'BEST : ', 118813552.92119238)
('ITERATION : ', 0, 'BEST : ', 118631529.1569625)
('ITERATION : ', 0, 'BEST : ', 117579029.23881577)
('ITERATION : ', 0, 'BEST : ', 117102782.84639569)
('ITERATION : ', 0, 'BEST : ', 117030104.18923883)
('ITERATION : ', 0, 'BEST : ', 116013268.26063196)
('ITERATION : ', 0, 'BEST : ', 115807022.95874524)
('ITERATION : ', 1, 'BEST : ', 115795173.32567601)
('ITERATION : ', 1, 'BEST : ', 115778499.03288956)
('ITERATION : ', 1, 'BEST : ', 115722744.5002727)
('ITERATION : ', 9, 'BEST : ', 115646129.54231468)
('ITERATION : ', 9, 'BEST : ', 115632176.02152485)
('ITERATION : ', 11, 'BEST : ', 115006131.04104832)
('ITERATION : ', 11, 'BEST : ', 114779946.87188065)
('ITERATION : ', 11, 'BEST : ', 106945777.70283893)
('ITERATION : ', 11, 'BEST : ', 106846350.49074972)
('ITERATION : ', 11, 'BEST : ', 106779232.83597541)
('BEST SOLUTION: ', 106779232.83597541)
```

Result Analysis and Performance Analysis of the GA

From the above tables and result screenshots, it can be seen that the output is different because of the different configuration, which leads to the different optimal best solution. The test was performed with an initial solution by the random and heuristic method.

The first two configurations based on population size 100, mutation rate 0.05, and crossover one of each and selection was a binary tournament selection. The rest of the four was tested with a different configuration setting.

Out of six configurations, configuration no five performed well, followed by configuration 1 and 2. Configuration 3 and 4 perform decently compared to Configuration 1 and 2. On the other hand, configuration five and six, which use the heuristic approach and based on the different settings, cannot perform well compared to the rest of the configuration.

To get the real picture of the algorithm with a much broader scope, it needs to go through different experimenting to get the overall performance to vary a few of the input like population size, mutation rate, and the number of iterations.

Reference:

1. https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm
2. Lectures Notes are used throughout the code and report.