Introduction

Abbreviations and Acronyms	
MaxIt	Maximum number of iterations
nPop	Population size
pc	Crossover percentage
nc	Number of offsprings in crossover population
nm	Number of mutants
beta	Selection pressure (β)
BestSol	Best solution/fitness
CPU Time	Time Taken

Table 1: Abbreviations and Acronyms

Description on the Parameters in Genetic Algorithm.

Tuning of Parameters :

- **Maximum Number of Iterations, *MaxIt***
  - The best solution would be obtained under *MaxIt* iterations, too many iterations are redundant for a longer runtime over the same result, while a low number of iterations will cause the best cost to be remain unexplored.
- **Population Size, *nPop***
  - Population for solutions of travelling path to perform the crossover. A higher value will slow down the genetic algorithm runtime, while a lower value make a few of search space to be explored.
- **Crossover Percentage, *pc***
  - A slight crossover percentage based on the population is appropriate in this model, to obtain some good parts from previous solution. For example, 0% will make the crossover to be the same from previous solution, while a large percentage will change a large proportion of the crossover.
- **Number of Offsprings/ Crossover Populations, *nc***
  - Indicates the number of offspring is being performed in the crossover.
- **Number of Mutants, *nm***
  - Indicates the number of mutants is being performed in the scrambled mutation.

- **Selection Pressure, *beta* (*β*)**
    - Improve the fitness with selection pressure on the selection probability used, while evolving better individuals over time.

Targets :

- **Best solution/fitness, *BestSol***
    - The best solution either minimization or maximization of the particular function.
- **Time taken, *CPU Time***
    - The total time for computer to run and obtain the best solution. (As low as possible)

Methodology

As explained in Introduction, the parameters values will affect the efficacy and the efficiency of the algorithm. Hence, it is significant to obtain an optimal set of parameters values so as to enhance the performance of the algorithm.

Since the generation of the initial populations, the selection of the parents, the crossover method, and the mutation method are subjected to some degree of randomness, the best solution can be obtained regardless of the values set for the parameters as long as the algorithm is run until the best solution is obtained. However, this is tedious. Hence, to evaluate each set of parameters fairly, for each set of parameters, the algorithm is run 10 times. The success rate, *s*, where the algorithm with a particular set of parameters is capable to give the best solution is found by $\frac{n}{10}$, where *n* represents the frequency of the algorithm manages to give the best solutions. To calculate the success rate, multiple testing should be conducted to obtain the known best solution. Nonetheless, this is not a completely fair evaluation as the algorithm is still subjected to randomness.

To obtain the optimal set of parameters for each model, a systematic guide is followed to conduct the trial and error. In the first trial, *maxIT* and n_{pop} are both set as 50, while p_c is set as 2. For β, test its values starting from 2, for model which initial populations have large cost, 2 might be too big that it will result in an error where the calculated probabilities are undefined as well. In the fact that, the python default decimal storing of values, as the value is approaching to zero will be assign to 0, and divided by it will return undefined. Therefore, when the array is completely undefined values and passed into the selection function, it will result in an error as shown in *Figure 2*. Thus, when selecting the first β value, if error in *Figure 2* appears, β will be decreased until the error does not appear anymore. That value is known as β_{max} which the values of beta should not exceed in order to get the algorithm run successfully.

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-15-b963960898f8> in <module>
     12
     13         # SUS Selection
---> 14         p1 = pop[sus_selection(probs)[0]]
     15         p2 = pop[sus_selection(probs)[1]]
     16

<ipython-input-11-3157a28380cc> in sus_selection(Probabilities)
      8     ind1 = np.argwhere(r1 <= c)
      9     ind2 = np.argwhere(r2 <= c)
---> 10     return ind1[0][0], ind2[0][0]

IndexError: index 0 is out of bounds for axis 0 with size 0
```

Figure 2: Error due to Selection Pressure

At the initial stage, the main objective that is taken into consideration for the assignment of parameters values is to enable the algorithm to get the best solution for at least once out of 10 times. To achieve this, the population size will be increased to increase the diversity of the solutions, while the maximum number of iterations will also be increased to increase the chances for the solution to converge to the best solution. When the algorithm still failed to obtain the best solution with these parameters, the crossover percentage will be increased to create more new solutions.

After the algorithm is able to obtain the best solution for at least once, the next stage will focus on reducing the time taken. This can be accomplished by lowering the *maxIT* and n_{pop}. Stage after this is to increase the success rate of algorithm which can be completed by increasing crossover percentage and decreasing the selection pressure so that the diversity of the solutions will be increased. The final stage is to find the best balance between the success rate and the time taken. Hence, the values for *maxIT*, n_{pop}, and β will be fine-tuned.

To determine the set of parameters with the best balance between the success rate and the time taken, two criteria are applied. The first criterion is to achieve a success rate above 0.5. Since the randomness of the algorithm will always exists, it is impossible to have a set of parameters which will definitely give the best solution every time the algorithm is run, unless the other factor such as time taken is neglected. The second criterion is the least time taken among those who qualified from first criterion.

The First Model – States of West Malaysia

Through multiple times of testing, the best solution found for this model is as shown in *Figure 1.1* which gives the cost of 852.1688232127713.

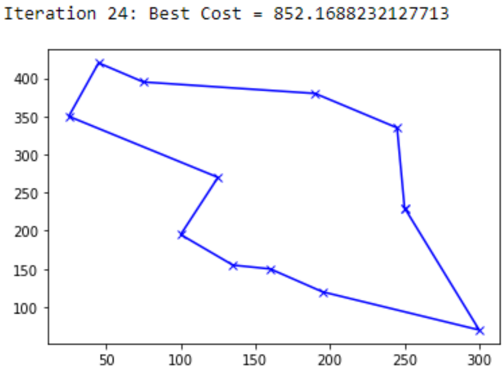

Figure 1.1: Best Solution (Model 1)

15 sets of different parameter values are tested with the algorithm and the result is shown in *Table 1.1*.

No.	**Parameters**				**Best Cost Obtained**	**Time Taken (s)**	**Success Rate**	
	maxIT	n_{pop}	p_c	β			n	$s = \frac{n}{10}$
1	50	50	2	0.4	855.3503859481812	13.43	0	0
2	50	100	2	0.4	855.3503859481812	14.24	0	0
3	100	100	2	0.4	855.3503859481812	28.17	0	0
4	100	100	5	0.4	852.1688232127713	31.67	2	0.2
5	50	100	5	0.4	852.1688232127713	15.95	1	0.1
6	50	100	10	0.4	852.1688232127713	19.6	4	0.4
7	50	100	10	0.04	852.1688232127713	18.45	3	0.3
8	50	100	10	0.012	852.1688232127713	18.58	8	0.8
9	30	100	10	0.012	852.1688232127713	11.07	5	0.5
10	50	80	10	0.012	852.1688232127713	17.5	8	0.8
11	40	80	10	0.012	852.1688232127713	13.51	3	0.3
12	50	50	10	0.012	852.1688232127713	15.84	4	0.4
13	30	100	10	0.02	855.3503859481812	11.2	0	0
14	40	80	10	0.02	852.1688232127713	14.32	4	0.4
15	40	80	10	0.015	852.1688232127713	14.05	6	0.6

Table 1.1: Selection of Parameters Values (Model 1)

After 15 trials, the optimal set of parameters is chosen as the highlighted one in *Table 1.1* which gives the result shown in *Figure 1.2*. A graph which depicts the process of the algorithm in finding the best solution is plotted as shown in *Figure* .

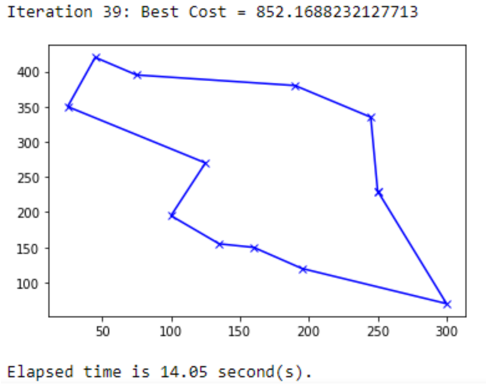

Figure 1.2: Best Solution for Model 1

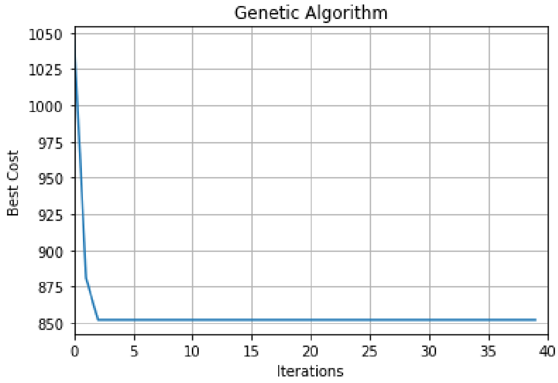

Figure 1.3: Graph of Best Cost Against Iterations (Model 1)

As observed from *Table 1.1*, the time taken is proportional with the maximum number of iterations, *maxIT*, and the number of populations, n_{pop}. However, these two parameters cannot be assigned with values that are too low, else the possibility for the algorithm to get the best solution will decrease. On the other hand, crossover percentage and selection pressure do affect the success rate, which is the possibility for the algorithm to obtain the best solution. This is because both of these parameters influence the diversity of the solutions. In this case, the time taken is 14.05 seconds, while the success rate is approximately 60%, which can be considered favourable for a number of 12 data.

Figure shows that the algorithm is able to obtain the best solution with less than 5 iterations. Solely based on the graph, it could be inferred that the maximum number of iterations can actually be reduced to optimize the algorithm by increasing the computing speed. However, since the algorithm is subjected to randomness, by doing this, the possibility for the algorithm to obtain the best solution will probably decrease. Therefore, it is preferable to give some allowance on assigning the value for *maxIT*.

The Second Model – Highlights of Malaysia

In second model, we have the travelling across the highlights in Malaysia. A large distance between two city coordinates implies higher cost of travelling. To find the shortest path, genetic algorithm is designed to minimize the cost or achieve the best solution to travel among all the highlights.

By obtaining the best solution after running multiple times in *Figure 2.1*, and explain the weakness of extrapolation of the parameters (upper and lower limit).

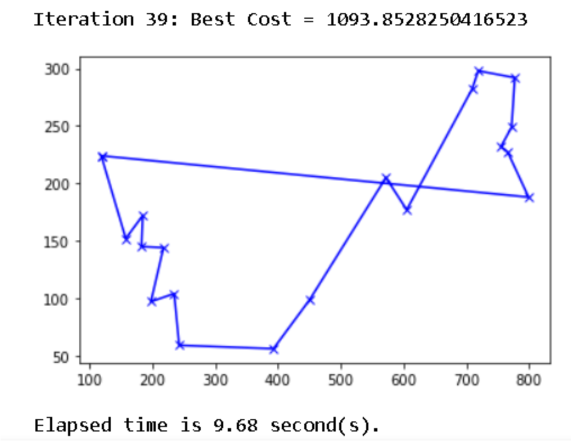

Figure 3.1: Best Solution for Model 2

MaxIt	nPop	pc	Beta (β)	BestSol	CPU Time	Remark
40	30	5	0.0005	1093.8528	9.68	-
10	30	5	0.0005	1802.4946	4.37	Not Best Cost
200	30	5	0.0005	1093.8528	28.08	Long Runtime
50	**10**	5	0.0005	1683.2103	11.02	Not Best Cost
50	**100**	5	0.0005	1093.8528	15.53	Long Runtime
50	30	**0.1**	0.0005	3205.9780	12.07	Not Best Cost
50	30	**10**	0.0005	1360.5834	12.80	Not Best Cost
50	30	5	**0.05**	1289.9737	11.29	Not Best Cost
50	30	5	**0.000005**	1171.4702	11.27	Not Best Cost

Table 2.1: Selection of Parameters Values (Model 2)

Maximum of iterations at 40 is optimal, while lower iterations can cause best solution not found, and higher iterations lead to longer time to run. Population size is set at 30, as lower size has fewer permutation sequence, and higher size took longer time to achieve the best solution. Percentage of crossover of 5 is used, which gives us 150 number of offsprings for crossover population as well as the number of mutants. Lastly, 0.0005 of selection pressure is the best for solving this model, as different values would take more iterations to find out the best solution.

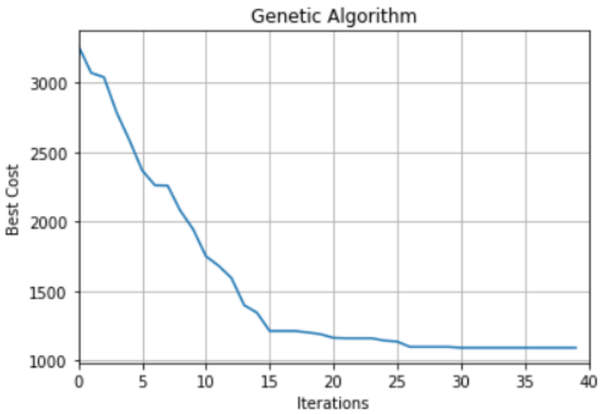

Figure 2.2 Genetic Algorithm performance (Model 2)

In *Figure 2.2,* it shows the best cost optimization over iterations of genetic algorithm. Clearly, a large minimization of cost is obtained quickly within the first 15th iterations. The following iterations attempt to minimize the fairly similar cost in between the solutions within 15nd to 25th iterations. The minimum travelling cost is obtained around 26th iteration.The remaining iterations, however, are just to ensure the minimization is completed.

The Third Model – All Island of Malaysia

Through multiple times of testing, the best solution found for this model is as shown in *Figure* which gives the cost of 2950.4147175309167.

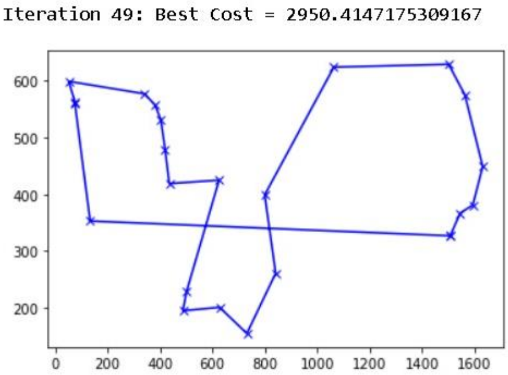

Figure 3.1: Best Solution (Model 3)

10 sets of different parameter values are tested with the algorithm and the result is shown in *Table 3.1*.

No.	Parameters				Best Cost Obtained	Time Taken (s)	Success Rate	
	maxIT	n_{pop}	p_c	β			n	$s = \frac{n}{10}$
1	50	50	2	0.06	4161.810702131713	15.06	0	0
2	50	150	2	0.06	3723.32936393196	14.85	0	0
3	100	150	2	0.06	3667.1746843624037	30.7	0	0
4	100	150	5	0.06	2950.4147175309167	38.57	2	0.2
5	50	150	5	0.06	3044.9545127973797	18.47	0	0
6	50	150	5	0.02	2950.4147175309167	18.65	1	0.1
7	50	150	5	0.006	2950.4147175309167	18.73	1	0.1
8	50	150	5	0.002	2950.4147175309167	18.53	1	0.1
9	50	150	5	0.0006	2950.4147175309167	18.5	4	0.4
10	**50**	**150**	**5**	**0.0002**	**2950.4147175309167**	**18.33**	7	0.7

Table 3.1: Selection of Parameters Values (Model 3)

After 10 trials, the optimal set of parameters is chosen as the highlighted one in Table 3.1 which gives the result shown in *Figure 3.*. A graph which depicts the process of the algorithm in finding the best solution is plotted as shown in *Figure 3.2*.

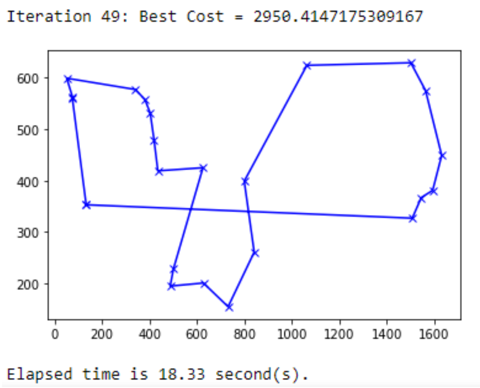

Figure 3.2:Best Solution for Model 3

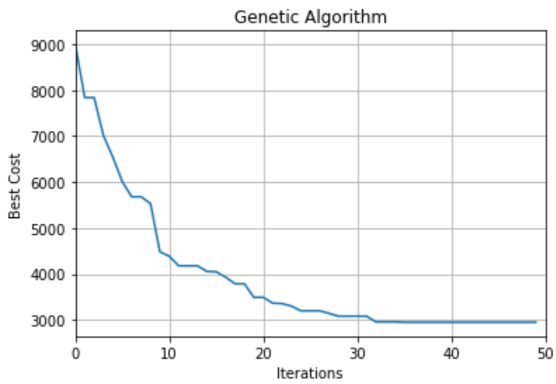

Figure 3.3: Graph of Best Cost Against Iterations (Model 3)

Table 3.1 shows a great difference between the time taken for algorithm with *maxIT* of 100 and that with *maxIT* of 50. Decreasing the maximum number of iterations definitely decreases the time taken; however, on the other side, the success rate is also decreased. Therefore, to balance between time taken and success rate, the crossover percentage and the selection pressure is find-tuned so that it can be ensured that most of the time the algorithm is able to get the best solution in reasonably short time. In this case, the time taken is 18.33 seconds, which is acceptable for 23 datas.

As observed from *Figure 3.*, the algorithm only managed to get the best solution after 30 iterations. This justified the statement that decreasing *maxIT* would decrease the possibility for the algorithm to obtain the best solution. To make sure that the algorithm could give the best solution for around 60% of time, instead of setting *maxIT* to 35, which the best solution could be obtained in this case, *maxIT* is assigned with a value of 50 so other cases where the best solution could not be obtained within 35 iterations could be prevented. At the same time, the possibility for the algorithm to obtain the best solution is increased as well.

The Fourth Model – Circle Model (Include 20 cities)

We build the fourth model in circle to find the shortest pathway which is to minimize the cost in genetic algorithm (GA).

MaxIt	nPop	pc	Beta (β)	BestSol	CPU Time	Remark
40	100	6	0.0005	**59.5241**	14.62	BC
40	100	6	0.00006	**59.2541**	15.78	LRT
40	100	6	0.05	76.9037	14.97	-
10	100	6	0.0005	115.9219	**4.48**	-
60	100	6	0.0005	**59.2541**	24.31	LRT
40	20	6	0.0005	65.65824	**12.02**	-
40	200	6	0.0005	**59.2541**	**20.32**	LRT
40	100	10	0.0005	**59.2541**	16.92	LRT
40	100	0.1	0.0005	172.7006	11.59	-

Table 4.1: Selection of Parameters Values (Model 4)

'-': Bad or the Worst Cost
LRT : Long Runtime
BC : Best Cost

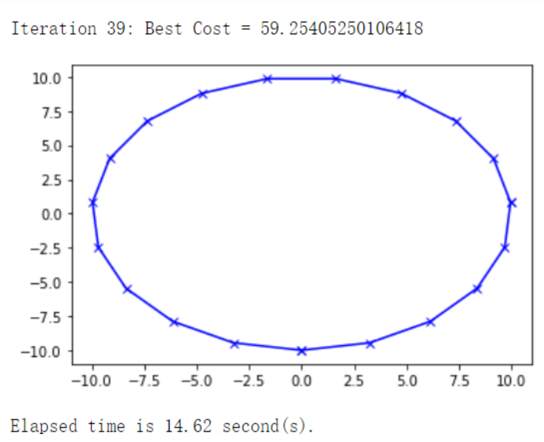

Figure 4.1:Best Solution for Model 4

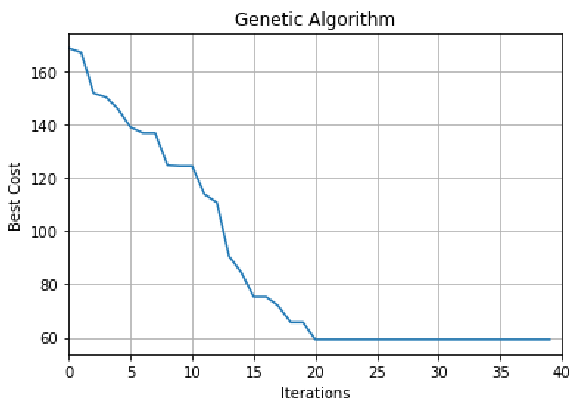

Figure 4.2: Graph of Best Cost Against Iterations (Model 4)

The best result is obtained from *Figure 4.1* above with various of parameter values. In *Figure 4.2*, we can see that 40^{th} in maximum of iteration is suitable iteration while the lower iteration affects the circle cannot be matched, and the higher iteration may lead to the solution in long run process. Setting population size at 100, because the largest population size(200) leads to the solution is not in perfect solution. The crossover percentage is set at 6, and the beta which is the selection pressure should be as small as 0.0005 or even smaller as the largest selection pressure can make the solution become worst.

From the *Figure 4.2* above is showing that the maximum iterations and the best cost in genetic algorithm. In between 5^{th} to 10^{th} iteration which looks similar cost from the result, while, there is a huge drop among 10^{th} and 20^{th} iterations nearly reached to the best result. However, after the 20^{th} iteration is remain constant which is completely become the smallest cost showing by table.

The Fifth Model – Circle Model (Include 25 cities)

Through multiple times of testing, the best solution found for this model is as shown in *Figure 5.1* which gives the cost of 60.042048421223726.

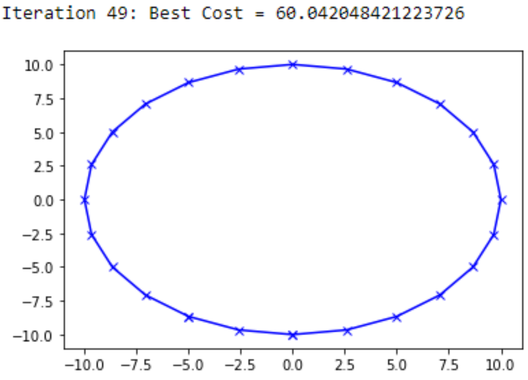

Figure 5.1: Best Solution (Model 5)

10 sets of different parameter values are tested with the algorithm and the result is shown in *Table 5.1*.

No.	**Parameters**				**Best Cost Obtained**	**Time Taken (s)**	**Success Rate**	
	maxIT	n_{pop}	p_c	β			n	$s = \frac{n}{10}$
1	50	50	2	2	105.25700817462224	15.99	0	0
2	50	100	2	2	75.12909574977085	17.18	0	0
3	100	100	2	2	89.73694528844499	34.36	0	0
4	100	200	2	2	60.042048421223726	36.24	1	0.1
5	50	200	2	2	92.05661413610237	20.06	0	0
6	50	200	5	2	69.99738163447209	24.66	0	0
7	50	200	5	0.2	60.042048421223726	23.38	2	0.2
8	50	200	5	0.02	60.042048421223726	22.89	6	0.6
9	40	200	5	0.02	60.042048421223726	18.74	6	0.6
10	40	200	5	0.002	60.042048421223726	18.62	4	0.4

Table 5.1: Selection of Parameters Values (Model 5)

After 10 trials, the optimal set of parameters is chosen as the highlighted one in Table which gives the result shown in *Figure* . A graph which depicts the process of the algorithm in finding the best solution is plotted as shown in *Figure* .

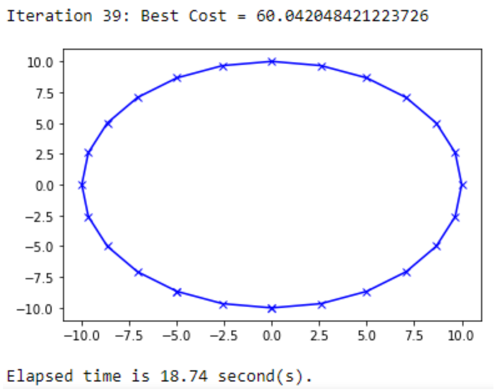

Figure 5.2: Best Solution for Model 5

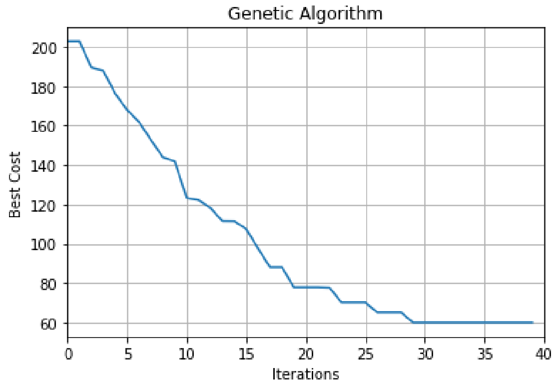

Figure 5.3: Graph of Best Cost Against Iterations (Model 5)

Based on *Table* , it can be noticed that the maximum number of iterations and the number of populations are first increased to obtain the best solution since both of these parameters increase the chance for the algorithm to get the best solution. Once the algorithm is able to obtain the best solution, the parameters p_c and β are varied to increase the possibility for the algorithm to obtain the best solution. At the same time, *maxIT* is slightly decreased to improve the time taken. In this case, the time taken is 18.74 seconds, which is reasonably short to obtain the best solution for a number of 25 datas.

Figure 5.3 shows that the algorithm is able to obtain the best solution within 30 iterations. Nonetheless, the *maxIT* is not set to 30 in consideration of the randomness of the algorithm and to allow flexibility of the algorithm. By setting the *maxIT* to a higher value, 40 in this case, it can be guaranteed that at least half of the times the algorithm is capable to provide the best solution.

Summary

Best Solution for 5 models :

Model	MaxIt	nPop	pc	nc	nm	beta(β)	BestSol	CPU Time
1	40	80	10	800	800	0.015	852.1688	14.05
2	40	30	5	150	150	0.0005	1093.8528	9.68
3	50	150	5	750	750	0.0002	2950.4147	18.33
4	50	100	6	600	600	0.0005	59.5241	14.62
5	40	200	5	1000	1000	0.02	60.0420	18.74

Table 2: Summary of Model

End