Project Report

Umesh Arampath

May 14, 2022

Genetic Algorithm implementation on benchmark TSP problems and analysis of the results of three papers.

Umesh Arampath

May 14, 2022

1 Abstract

Traveling salesman problem(TSP) is a well known combinatorial optimization problem which has studied intensively in various fields such as mathematics, computer science, engineering, bioinformatics and operational research and etc. Problem could be described as below. There is a list of cities with given locations and we need to find the cheapest tour in terms of some cost metric such as distance,time and money is minimized to visit all of the cities, each exactly once and then return to the city of origin.
The most obvious solution would be the brute force method, where we consider all the different possibilities, calculate the cost metric for each, and choose the one that is the minimum.This approach requires a lot of computational time—the runtime of this brute force algorithm would be O(n!). Therefore over the years heuristic search methods have been introduced in order to tackle these TSP problems. In this study Genetic algorithm(GA) approach will be investigated and several results put forward by three published papers[8][2][1] will be analysed with a set of new benchmark TSP datasets.

[2] compares different crossover methods on two benchmark datasets and claim that the OX(order crossover) method works better than the others in terms of faster convergence and better approximation and further states that this is only valid for their considered dataset. I carried out the study for several benchmark TSPLIB[4] datasets and verify the results.

[8] Proposes a new crossover method named OX2 and claims that it performs better than the OX crossover method. I use the same datasets and show that with the tournament selection method combined with the OX crossover method, much better results and faster convergence could be achieved than with the [8] proposed new crossover method.

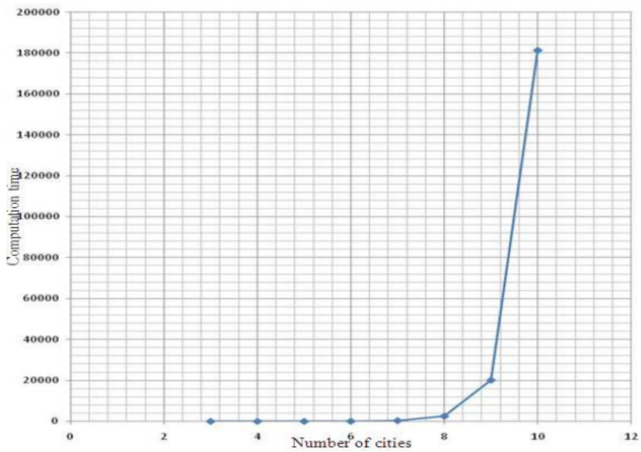
[1] compares several mutation methods and conclude that the Reverse sequence mutation(RSM) method is better than the other mutation methods. In their study they do not consider the Center inverse mutation(CIM) method, and I implement the comparison study to show that CIM method is infact works better than the RSM mutation method.

2 Introduction

The travelling salesman problem (TSP) is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science. A quick calculation shows that the complexity is O(n!) which n is the number of cities.(Figure 1)[1].

Figure 1: Combinatorial explosion of TSP

Number of cities	Number of possibilities	Computation time
5	12	12 µs
10	181440	0,18 ms
15	43 billions	12 hours
20	60 E+15	1928 years
25	310 E+21	9,8 billions of years



There have been really successful methods to solve TSP problem and here I will be focusing on the Genetic algorithm methods. A simple Genetic Algorithm could be defined in the following way

- Create the initial population of chromosomes.
- Evaluate the fitness of each chromosome and implement a selection method.
- Randomly select two parents to create offspring using a crossover operator.
- Apply mutation operators to mutate the resultants.
- Replace the population by implementing the above steps.
- Terminate if the number of generations meet some upper bound.

In the Genetic algorithm implementation, the selection criteria, crossover and mutation are major operators. And there are various implementations of each of these operators.

Selection criteria

- Tournament selection - Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover.This ideal for large datasets since the fitness of all individuals will not be calculated.[9]
- Roulette wheel selection - Roulette selection is a stochastic selection method, where the probability for selection of an individual is proportional to its fitness.The larger the fitness of an individual is, the more likely is its selection.
- Elite selection - Elitist selection is a selection strategy where a limited number of individuals with the best fitness values are chosen to pass to the next generation, avoiding the crossover and mutation operators.

Remark

Here in my study I considered the above selection criteria. And based on the published work on this, it is stated that the tournament selection method works better than the other two methods when it comes to relatively large tsps. And in my study, it was clearly seen that the tournament selection method works better than the other two methods and also even implementing Roulette wheel selection for majority of the problems was not feasible because of the high running time.

Crossover Operators

In genetic algorithms and evolutionary computation, crossover, also called recombination, is a genetic operator used to combine the genetic information of two parents to generate new offspring.[3] There are a number of different crossover methods proposed over the year. But here in my study I will be focusing on two crossover methods based on the two papers that I'm referring to.[2][8]

Remark

Here in my study I considered two crossover operators, namely, partially mapped crossover (PMX) and order crossover operator (OX) which are described in the section 3.

Mutation Method

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to the biological random mutation in the evolutionary process. Similar to selection and crossover, Mutation could also be done in several ways and various methods have been proposed over the years. Namely,

- Twors Mutation - Twors mutation is a mutation method that also can be referred to as swap. Two genes are randomly chosen, and their positions are swapped.
- Reverse Sequence Mutation (RSM) - The RSM method chooses two random points in the chromosome and selects the section of genes in between. The gene sequence inside the selected section is then reversed.
- Partial Shuffle Mutation (PSM) - The PSM method iterates through each gene in a chromosome. Each gene uses the mutation probability to determine if the gene should be swapped with another. If the gene is determined to be swapped, the gene will swapped with another randomly chosen gene.

As it is established in this paper [2], it had been shown that the Center Inverse Mutation (CIM) works better compared to the other methods. Therefore in my study, I will be focusing on implementing the Center Inverse Mutation (CIM) in the genetic algorithm. And in the results analysis of the paper [1], I will be comparing the Reverse Sequence Mutation (RSM) with the performance of the Center Inverse Mutation (CIM) method. In the paper[1], they test different mutation methods and come to the conclusion that the Reverse Sequence Mutation (RSM) method works better than the rest of the methods. In that analysis they do not consider the Center Inverse Mutation (CIM) method. Therefore in the analysis part 3, It will be shown that CIM works better than the RSM method.

Center Inverse Mutation (CIM) [2]

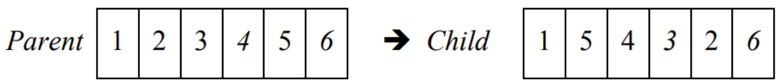
The CIM method chooses one random point, which divides a chromosome into two sections. The two sections are flipped.

Figure 2: Example of CIM method for mutation

Reverse Sequence Mutation (RSM) [1]

In the reverse sequence mutation operator, we take a sequence S limited by two positions i and j randomly chosen, such that $i < j$. The gene order in this sequence will be reversed by the same way as what has been covered in the CIM operation.

Figure 3: Example of RSM method for mutation



3 Problem and Methodology

Approach is to solve a set of benchmark TSP problem datasets[4] using the GA with a set of different combinations of crossover and mutation methods and see their performance and come to some conclusions on what methods are more suited for solving these type of TSP problems.

And also several selection methods will be evaluated in order to assess their performance and compare the results with a published paper.[2]

For crossover, Following crossover methods are considered.

- Partially mapped crossover (PMX)
- Order crossover (OX)

Brief description of the crossover methods are stated below.

PMX crossover method

The PMX crossover method is used for crossover in permutation problems. This was proposed by Goldberg and Lingle.[7] In this method, two parents are chosen using a selection method in the algorithm and then two random cut points are defined on parents to build the offspring. The portion between the cut points, one parent's numbers are mapped onto the other and the remaining information is exchanged. And it is to be noted that PMX could be defined with a single cut point or with arbitrary number of cut points as well. But the basic idea of one portion being copied exactly the same for the offspring and the rest being exchanged remains the same. Consider the following example below with two cut points defined.

$$P_1 = (8\ 7\ 1|\ 3\ 6\ 5|\ 4\ 2)$$

$$P_2 = (4\ 6\ 2|\ 1\ 5\ 8|\ 3\ 7)$$

Two parents of a 8 node problem are mentioned above. Two randomly cut points between the 3rd and the 4th bits and the other between the 6th and the 7th bits are defined.
Portions between the cut points are mapped onto each other to create the new offsprings.

$$O_1 = (x\ x\ x|\ 1\ 5\ 8|\ x\ x)$$

$$O_2 = (x\ x\ x|\ 3\ 6\ 5|\ x\ x)$$

Then we would fill further bits for those which have no conflict as follows.

$$O_1 = (x\ 7\ x|\ 1\ 5\ 8|\ 4\ 2)$$

$$O_2 = (4\ x\ 2|\ 3\ 6\ 5|\ x\ 7)$$

The first x in the first offspring is 8 which comes from first parent but 8 is already in this offspring. The we see that $8 \leftrightarrow 5$. But 5 again exists in the first offspring. Then we check $5 \leftrightarrow 6$ and 6 is not in the first offspring so first x is denoted to be 6. Similarly rest of the bits are exchanged. This gives us the following two offsprings from the two parents.

$$O_1 = (6\ 7\ 3|\ 1\ 5\ 8|\ 4\ 2)$$

$$O_2 = (4\ 8\ 2|\ 3\ 6\ 5|\ 1\ 7)$$

OX crossover method

OX method was proposed by Davis.[5] and it is also a crossover method for permutation problems. This method is based on randomly selecting a section of genes within the parents, for example, the 4 middle genes. Child 1 will then directly inherit these 4 middle genes from parent 1 (into the same position in the child), while child 2 will inherit from parent 2. The remaining genes are then filled with values from the other parent. Again, since chromosomes represent permutations, it is important that there are no duplicate genes (values) in the child. Therefore, creating the child starts with looking at the index of the first non-assigned gene in the other parent. If this gene value does not exist in the child, it is copied into the child. If the value already exists in the child, the procedure continues to check the next gene of the other parent. The process can be illustrated with the example in Figure 4.[2]

Figure 4: Example of order 1 crossover (OX)

Parent 1	1	2	3	4	5	6	7	8	9
Parent 2	9	~~7~~	~~6~~	1	3	~~5~~	8	2	~~4~~
Selected section copied to child 1									
Child 1				4	5	6	7		
Starts with first gene of parent 2, which is 9. Since 9 does not exist in child 1 yet, it can be copied.									
Child 1	9			4	5	6	7		
The next gene in parent 2 is 7, this already exist in child 1. The next gene is checked, which is 6. It also exists in child 1. The next gene is 1, this is not in child 1 yet and therefore inherited in the next free position.									
Child 1	9	1		4	5	6	7		
The next gene is 3, which can be inherited, and so on. Finally, you get the child below.									
Child 1	9	1	3	4	5	6	7	8	2

4 Approach

4.1 Analysis 1 - related to the paper [2]

- Use the tournament selection method and compare the results with OX and PMX crossover methods on several symmetric TSP datasets from the **TSPLIB** library.[4]
- Population size, crossover probability, Mutation probability were set at 200,0.5 and 0.8 respectively.
- Implement GA with OX and PMX crossover methods on the TSPLIB dataset[4] and compare it with the results obtained in the paper [2].

4.2 Analysis 2 - related to the paper [8]

- Then critique of the results obtained in the paper[8] and show that the same performance can be achieved just by changing the selection method and without using the proposed new crossover (OX2) method.
- Tournament selection method was used and it was shown that without using the proposed new crossover(OX2) method, we can obtain optimal results with the OX crossover method.

4.3 Analysis 3 - related to the paper [1]

- In this paper several mutation methods were analysed and it was shown that the Reverse Sequence mutation method works better than the other mutation methods.
- They do not consider the center inverse mutation (CIM) in their analysis.
- CIM method is implemented and it is shown that in fact CIM method works better with the same set of benchmark problems.

5 Methods and Results with comparing OX and PMX crossover methods

5.1 Analysis 1 - related to the paper [2]

This test was done with the tournament selection method and to make a comparison of the performance of the OX and PMX crossover methods on some symmetric TSP problems that we can find on the **TSPLIB** datasets.[4] And these problems have been solved for their optimal value over the years using different methods and their optimal value results are used to make the comparisons.[4]

Results

TSP Dataset	no.of Cities	Optimal Value	Best Cost OX	Best Cost PMX
Berlin52	52	7542	7544.365	19382
Dantzig42	42	699	699	1762
fri26	26	937	937	1469
gr17	17	2085	2085	2243
gr21	21	2707	2707	3807
gr24	24	1272	1272	1930
gr48	48	5046	5063	13343
hk48	48	11461	11461	30584
gr120	120	6942	7512	-

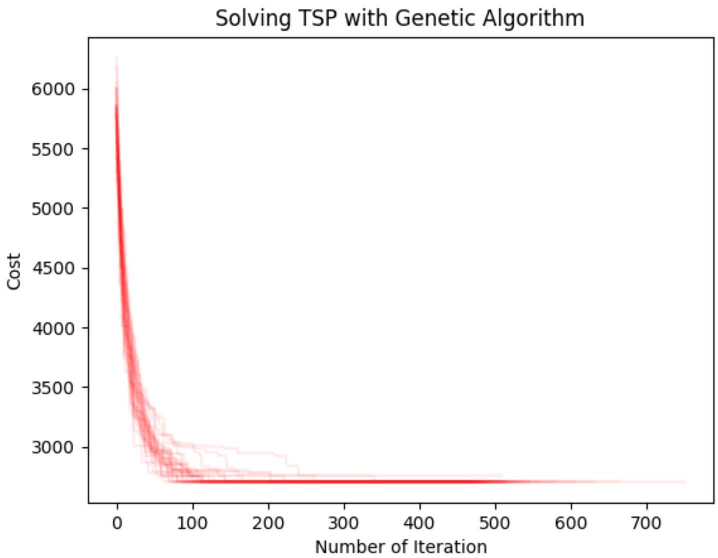
Table 1: Results with OX and PMX

Observations

Here with these results we can see that given the selection method is Tournament, Order crossover (OX) method works much better than the PMX crossover method. This further confirms the conclusion in this published paper [2]. In the paper it states that **" GA with OX crossover was the best choice however it is important to note that this claim is only valid for the two datasets and the choice of GA settings described."**

In the paper[2], they are using the GA settings as , Population size, crossover probability and the mutation ratio as 300,0.7 and 0.05. And they are implementing this method on just two benchmark

Figure 5: Cost evolution for the gr21 dataset with order 1 crossover (OX)



datasets , Western Sahara dataset and the Djibouti dataset.

In my experiment a totally different GA settings were used as mentioned in the section 4 and it was tested in a number of different Benchmark datasets (TSPLIB) and it confirms the results of the paper that GA with OX works better than the other crossover methods.

And also if we consider the size of the TSP problems considered, I have considered 8 TSP symmetric problems where the no of cities are of maximum 120. In all these cases, GA with the OX crossover operator is able to find the exact optimal solution for 5 problems and the other 3 approximate solutions which are very close to the optimal value.

For the problem with 120 cities, the best solution is 7512 and this was implemented using the **HPC uiowa with 30 runs and this is achieved at an avg.time for a run of 180 seconds.**

5.2 Analysis 2 - related to the paper [8]

Critique of the results obtained in the paper[8]

In this paper, They propose a new crossover operator based on OX. Which was named OX2. Then PMX,OX and OX2 were implemented on several TSP datasets and the results were compared. They conclude that their proposed OX2 crossover method performs much better than the other two methods.

In the above mentioned paper, they have considered 3 TSP datasets as I have done in this study. They are namely, **dantzig42, fri26 and gr21**. Mentioned below are the results that they have obtained for those problems.

TSP Dataset	no.of Cities	Optimal Value	Best Cost OX	Best Cost PMX	Best Cost OX2
Dantzig42	42	699	1301	1425	802
fri26	26	937	1158	1133	1128
gr21	21	2707	3208	3127	3145

Table 2: Results in the paper[8] with OX,PMX and their proposed new OX2 crossover methods

Here we can see that for these three problems, they do not obtain the optimal value with any of the crossover methods. But in my study, I could obtain the optimal solution with the OX crossover operator combined with the tournament selection method. And this behavior could be seen for the rest of the datasets that they have considered.

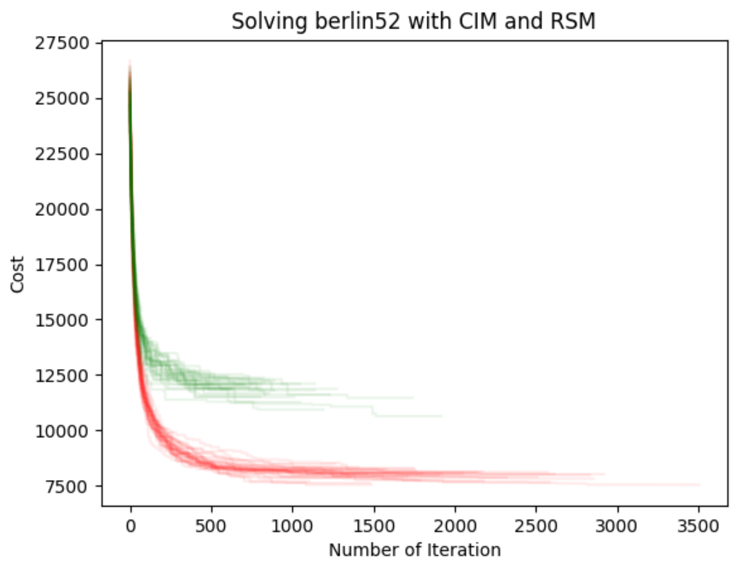
In this paper proportional selection method was considered. In the proportional selection method an individual can become a parent with a probability which is proportional to its fitness.

Therefore it can be concluded that the selection method plays a huge role in the GA algorithm and just by changing the selection criteria we could significantly increase the accuracy of the results without tweaking the crossover methods.

5.3 Analysis 3 - related to the paper [1]

- In this paper several mutation methods were compared with a TSPLIB [4] dataset, namely **berlin52** and it was shown that the **RSM** method works better than the other methods.
- Therefore I have implemented the **CIM** mutation method and it was compared with the **RSM** method with the same dataset and all the other benchmark datasets in [4] that I previously considered in the Analysis 1.
- It is shown that the **CIM** mutation method is in fact more stable and efficient than the **RSM** method when it comes to all these benchmark datasets.

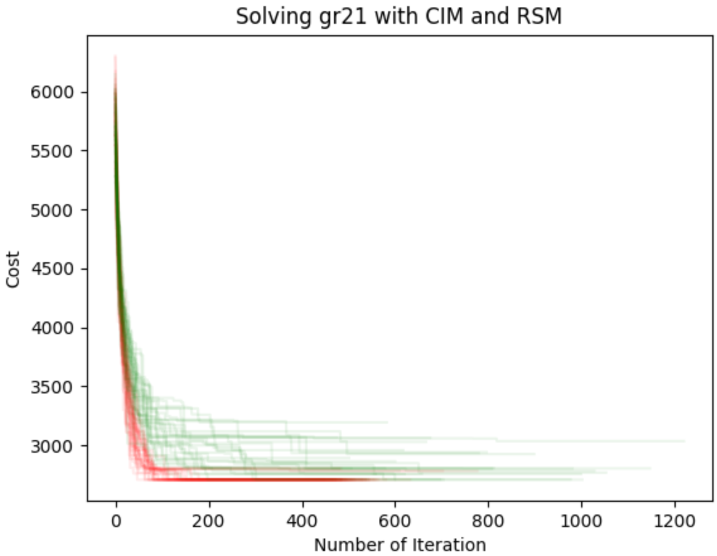
Figure 6: Cost evolution for the berlin52 dataset with RSM and CIM mutation methods



In the Figure 6 and 7, green color evolutions corresponds to RSM and the red color evolutions corresponds to CIM. And here we can see that the convergence to the optimal value is much more efficient and much better solutions are obtained with the CIM mutation method compared to the RSM method.

This could be seen for every dataset that I considered in the Analysis 1. Therefore it can be concluded that CIM method works better than RSM in this TSPLIB[4] dataset and I surmise that it is the same with other benchmark datasets as well.

Figure 7: Cost evolution for the gr21 dataset with RSM and CIM mutation methods



6 Conclusion and summary

- Here in my study I studied the GA implementation on several benchmark datasets from the TSPLIB [4] library.
- This study was done in relation to 3 published papers [1][2][8].
- [2] implements the GA algorithm with various with various crossover methods and come to the conclusion that GA with OX crossover method works better than the rest with their considered datasets and go on to further state that their claim is only valid for their considered datasets of Western Sahara and Djibouti dataset.
- I implemented the study on the TSPLIB dataset and have shown that we could verify that claim on this benchmark dataset as well.
- Then in relation to the Analysis 2 on the paper [8], they propose a new crossover method named OX2 and go on to claim that it performs much better and obtains better results with the given datasets.
- But I implemented the same study with the tournament selection operator and with the OX crossover operator and I could obtain better and much faster convergence than using a newly proposed crossover operator as it was suggested in the [8] paper.
- In relation to the analysis 3, Then compare different crossover methods as it was mentioned and go on to conclude that the RSM method works much better in terms of faster convergence and better approximation to the optimal value.

- They do not consider the CIM method in their study. And I implement the comparison with the CIM and RSM method with the benchmark TSPLIB datasets and go on to show that CIM converges much fasters and achieves a better approximation for every dataset instance.
- For larger datasets **hpc** could be used to get results and further evaluations on hybrid GA algorithms could be done on larger datasets.

7 Brief Documentation

In this documentation section, Brief descriptions on how I implemented the code and dealt with the relevant datasets are given. And a few of the terminal outcomes are given for reference.

- Benchmark datasets in TSPLIB [4] has different formats. In the problems that I considered, edge weights are given in two ways. **Explicitly** or using 2- coordinates which is referred to as **EUC_2D**.
- Then ones in the **Explicit** category are given in either as a **Full_MATRIX** or a **LOWER_DIAG_ROW** matrix.
- Therefore when creating the adjacency matrix with edge distances/cost three different ways of implementation were needed.
- This implementation of adjacency matrix using the **EUC_2D** and the **LOWER_DIAG_ROW** matrix were given in the **main.py** file, under **# calculate adjacency matrix** and **#Lower_triangular_row Adjacency Matrix** respectively.

Calculation of cost

- Cost calculation is done in **tsp.py** file.
- Two separate implementations were needed for the above two scenarios.
- They are implemented by the **get_cost** and **get_cost_LT** functions respectively.

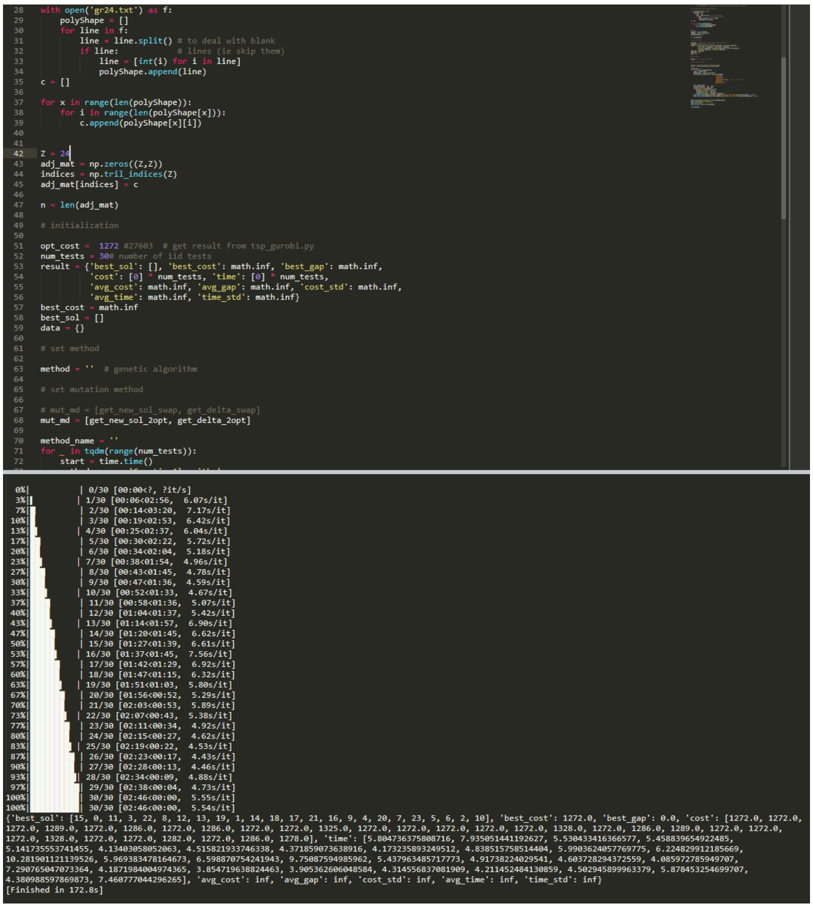
Selection, Crossover and Mutation method implementation

- Tournament selection, Roulette wheel selection and elite selection methods were implemented and some code blocks were taken from already existing implementations.[6]
- Center inverse mutation(CIM) and Reverse inverse mutation (RSM) were implemented and they are implemented by the **mutation** and **RSM** functions respectively.
- Similarly crossover methods are also being implamented in the **ga.py** file.
- Various other methods were also implemented in those three stages and those code blocks also can be seen in the respective files.

Examples of terminal outputs for reference

Figure 8: Terminal output for berlin52 dataset

Figure 9: Terminal output for gr24 dataset



References

[1] Otman Abdoun, Jaafar Abouchabaka, and Chakir Tajani. "Analyzing the performance of mutation operators to solve the travelling salesman problem". In: *arXiv preprint arXiv:1203.3099* (2012).

[2] Robin T Bye et al. "A Comparison of GA Crossover and Mutation Methods for the Traveling Salesman Problem". In: *Innovations in Computational Intelligence and Computer Vision*. Springer, 2021, pp. 529–542.

[3] *Crossover (genetic algorithm)*. `https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)`.

[4] *dataset*. `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/`.

[5] Lawrence Davis et al. "Applying adaptive algorithms to epistatic domains." In: *IJCAI*. Vol. 85. 1985, pp. 162–164.

[6] *GAcode*. `https://github.com/CarlossShi/tsp-meta-heuristic/blob/main/ga.py`.

[7] David E Goldberg, Robert Lingle, et al. "Alleles, loci, and the traveling salesman problem". In: *Proceedings of an international conference on genetic algorithms and their applications.* Vol. 154. Lawrence Erlbaum Hillsdale, NJ. 1985, pp. 154–159.

[8] Abid Hussain et al. "Genetic algorithm for traveling salesman problem with modified cycle crossover operator". In: *Computational intelligence and neuroscience* 2017 (2017).

[9] *TSM.* `https://en.wikipedia.org/wiki/Tournament_selection`.