Travelling salesman problem

Bicu Daniel(A1), Bogos(cas. Iftinca) Corina Elena(E1)

January 7, 2020

1 Introduction

The traveling salesman problem (TSP) is one of the most famous benchmarks, significant and very hard combinatorial optimization problem(NP-hard). It is the fundamental problem in the fields of computer science, engineering, operations research, discrete mathematics and graph theory. The TSPs are real world problems related to transportation, logistics, etc. In transportation, the school buses routes are established to find a cheapest link through every stop. Another real life application can be found in transportation in logistics where the aim is to find the cheapest route to deliver goods to customers.

TSP can be described as the minimization of the total distance traveled by touring all cities exactly once and return to depot city. Initially there are n different cities from which the salesman may choose. In the first city the salesman may choose one of different n-1 cities. In the second city, the salesman may choose one of the different n-2 cities. As a result, the salesman have the right to choose through n! different tour. These is the reason to say that TSP is NP-hard problem.

2 Methods

We have tried to find the optimal solution for the TSP problem using a genetic algorithm and an heuristic one.

2.1 Genetic algorithm

Solution representation For the genetic algorithm the solution is represented was by integer numbers. Each number represents a vertex from the graph.

Mutation The function mutation will choose with the probability of 1% to select a gene from the chromosome. Through mutation process will be selected a random number from the (0,1) interval , and if the generated number is smaller than 0.01 that gene will be selected. After the mutation process, in case there is an odd number of genes, the last one selected will be removed.

Selection For selection we used the roulette wheel. Fitness function is represented by the total cost of the path for a chromosome. Due to each chromosome's fitness, the best chromosome will be chosen for the next generation.

Population We generated the values random for the 100 chromosome such that each value is different from the other ones.

2.2 Heuristic algorithm

For the heuristic algorithm we choose a random vertex as a start point. Then at each step we choose random another vertex which was not visited until we visit all the vertices.

3 Experiment

The heuristic algorithm was implemented with a random search method. In order to get the statistical results, which can be seen at section (4), the algorithm was itterated 10000 times.
The genetic algorithm was ran 30 times for 400 generations.

4 Results

Operation	**Heuristic(1run)**	**Heuristic (10000runs)**	**Genetic Alg.**
Best known solution	-	-	**291**
Minim	487	387	317
Maxim	707	696	496
Average	565.33	416.322	326.364
σ	53.797	58.8893	23.368
Execution time	0.000405	1.0012	1.49216

The results for p01.tsp (15 cities)

Operation	**Heuristic(1run)**	**Heuristic (10000runs)**	**Genetic Alg.**
Best known solution	-	-	**2084**
Minim	3022	2322	2108
Maxim	5262	3310	3051
Average	4024	2884	2595.64
σ	415.4089	423.8893	167.2763
Execution time	0.03631	1.0032	1.50398

The results for gr17.tsp

Operation	**Heuristic(1run)**	**Heuristic (10000runs)**	**Genetic Alg.**
Best known solution	-	-	**937**
Minim	2143	1788	1343
Maxim	2937	3005	2038
Average	2593.5	2592.921	1820.7
σ	195.32	216.355	74.8511
Execution time	0.001053	0.0872	2.25497

The results for fri26.tsp

Operation	**Heuristic(1run)**	**Heuristic (10000runs)**	**Genetic Alg.**
Best known solution	-	-	**699**
Minim	2502	2129	1120
Maxim	3441	3680	2317
Average	2996.8	3552.868	2195.64
σ	247.2938	242.8893	68.2763
Execution time	0.001627	0.064532	3.3473

The results for dantzig42.tsp

Operation	**Heuristic(1run)**	**Heuristic (10000runs)**	**Genetic Alg.**
Best known solution	-	-	**33523**
Minim	134255	112611	79890
Maxim	177630	192306	127681
Average	152488.1	156850.20	113233.64
σ	9872.9536	12696.8893	8241.9763
Execution time	0.004465	1.3232	5.5786

The results for att48.tsp

4.1 Interpretation

The order of the tables isn't random, the tables are ordonated by the number of the cities of instances, therefore this fact makes the differences between algorithms used in experiment to be smoothly to understand.

Then:

1.An important fact to point out about the tables is that: the distances between cities are very important and more important than the number of cities when we talk about the minimum value obtained (eg.: for the instance **fri26**, even though it has 26 cities, the minimum obtained is **1343** and for the instance **gr17** we obtained **2108**). This happened because **the cost will influence the result of permutations**, therefore the cost is more important than the number of cities of an instance.

2. The parameters which influence the obtained values are represented by the number of executions and by the distances between the cities. **The greater the number of execution is, the better average result will be obtained** because for example with the heuristic algorithm on 10000 runs, even though the execution time is greater, the minimum obtained is definitely better due to the fact that a greater random node(city) could be selected. Also an important thing is that the minimum obtained with the heuristic algorithm on 10000 runs is lower and lower than the minimum obtained by just one run of the heuristic when the number of cities is bigger(eg. on instance dantazig42.tsp it is obtained the value **2502** on 1 run and the value: **2129** on 10000 runs.)

5 Conclusion

In this report we presented a genetic algorithm and an heuristic one for solving the TSP problem. After the experimental part and after comparing the results, we came to the conclusion that using a genetic algorithm we obtain better results than using an heuristic one. The last one obtains results which can compete with the genetic algorithm after 10000 iterations. On the other hand, due to of the exponential way of choosing a tour , solving TSP with mathematical methods becomes almost impossible when the number of cities increases to much, so the best solution remains the genetic algorithm .

6 References

https://www.hindawi.com/journals/cin/2017/7430125/ https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35
https://www.hindawi.com/journals/cin/2017/7430125/
https://pediaa.com/what-is-the-difference-between-genetic-algorithm-and-traditional-algorithm/
https://www.researchgate.net/publication/259745241
https://www.tandfonline.com/doi/full/10.1080/25765299.2019.1615172