CS 451 Report – Assignment 2

Barış Özbaş - S014669

Definition

In this assignment, I implemented Genetic Algorithm (GA) to solve one of the most well-known Computer Science problem, which is Traveling Salesman Problem (TSP).

Traveling salesman problem tries to find the solution of question "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?" It is an NP-hard problem in combinatorial optimization, important in operations research and theoretical computer science.

Approach

While solving the Traveling Salesman Problem (TSP) there are a bunch of algorithms that can be used such as following, given with their pseudo codes.

Naive Solution:

1) Consider city 1 as the starting and ending point.

2) Generate all (n-1)! Permutations of cities.

3) Calculate cost of every permutation and keep track of minimum cost permutation.

4) Return the permutation with minimum cost.

Dynamic Programming:

If size of S is 2, then S must be {1, i},

C(S, i) = dist(1, i)

Else if size of S is greater than 2.

C(S, i) = min { C(S-{i}, j) + dis(j, i)} where j belongs to S, j != i and j != 1.

Genetic Algorithm:

1) Creating initial population.

2) Calculating fitness.

3) Selecting the best genes.

4) Crossing over.

5) Mutating to introduce variations.

Genetic Algorithm with Modified Cycle Crossover Operator:

Step 1. Create an initial population of P chromosomes.

Step 2. Evaluate the fitness of each chromosome.

Step 3. Choose P/2 parents from the current population via proportional selection.

Step 4. Randomly select two parents to create offspring using crossover operator.

Step 5. Apply mutation operators for minor changes in the results.

Step 6. Repeat Steps 4 and 5 until all parents are selected and mated.

Step 7. Replace old population of chromosomes with new one.

Step 8. Evaluate the fitness of each chromosome in the new population.

Step 9. Terminate if the number of generations meets some upper bound; otherwise go to Step 3.

Implementation and Results

For the assignment, I implemented Genetic Algorithm (GA) to solve Traveling Salesman Problem (TSP) in Python. I edited the RouteManager.py (for a correction) and GeneticAlgorithmSolver.py.

For testing purposes, main_test.py has written (it is just to get bulk results and testing purposes, main.py still exists and usable).

The results and improvement % have shown below for beginning route cost = 1808, number of cities = 20 and population size = 50.

MR/TS	4	6	8	10	12
0.1	1460	1295	1191	953	925
0.2	1841	1552	1700	1582	1688
0.3	1864	1698	1947	1844	1841
0.4	1959	1825	1868	1751	1936

MR / TS	4	6	8	10	12
0.1	19.25%	28.37%	34.13%	47.29%	48.84%
0.2	-1.83%	14.16%	5.97%	12.50%	6.64%
0.3	-3.10%	6.08%	-7.69%	-1.99%	-1.83%
0.4	-8.35%	-0.94%	-3.32%	3.15%	-7.08%

Conclusion

Finally, the results are compared for different Mutation Rate (MR) and Tournament Size (TS). Best improvement that is achieved is 48.64% with MR = 0.1 and TS = 12. It can be seen that the internal variables can affect the improvement ratio of the algorithm crucially. Further work would be implementing different algorithms to solve the Traveling Salesman Problem (TSP) which have shown in Approach section.

Reference:

1. Abid Hussain, Yousaf Shad Muhammad, M. Nauman Sajid, Ijaz Hussain, Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator, 2017
2. Travelling Salesman Problem – View: https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/