Treveling Salesman Problem

Report made by **Vitalii Nefedov 91376 SD-1**

Part 1: Algorithm Description

The Genetic Algorithm (GA) for the Traveling Salesman Problem (TSP) is an approach used to find a solution to the TSP, where the goal is to determine the shortest possible route that visits a set of cities and returns to the starting city.

* **Initial Population Determination**: the initial population is formed by creating multiple individuals equal to the specified population size by shuffling the order in which the cities are visited.
* **Tournament Selection Method**: Selects individuals (potential solutions) from the current population to become parents for producing offspring in the next generation.
* **Ordered Crossover Technique**: Creates offspring from selected parent solutions by inheriting subsets of the parent chromosomes while maintaining the relative order of the cities.
* **Inversion mutation**: Randomly selects a subset of genes within an individual's chromosome and reverses the order of those genes
* **Solution Representation**: Solutions are represented as permutations of city indices.

Approach Taken:

* **Greedy Algorithm**: Used for comparison purposes, iteratively selects the nearest unvisited city.
* **Genetic Algorithm**: Employed to evolve solutions iteratively based on fitness through selection, crossover, and mutation.

Part 2: Parameter Testing and Analysis

Parameters that were tested: **Size of initial population, Mutation probability and Tournament Size.**

**Test 1**

* **Size of initial population**: 100
* **Mutation probability**: 1%
* **Tournament Size**: 10
* **Score: 8491.705293554089**

**A screen shot of a graph

Description automatically generated**

**Test 2**

* **Size of initial population**: 150
* **Mutation probability**: 5%
* **Tournament Size**: 20
* **Score: 7848.415967213308**

**A screen shot of a graph

Description automatically generated**

**Test 3**

* **Size of initial population**: 200
* **Mutation probability**: 10%
* **Tournament Size**: 20
* **Score: 7935.292434340646**

**A screen shot of a graph

Description automatically generated**

Part 3: Comparative Analysis

**Average scores** of Mean Distance, Standard Deviation, Variance and Best Distance for each of the algotithms.

A table with numbers and a few black text

Description automatically generated with medium confidence

Part 4: Solution Extension and Visualization

The best solution using my Genetic Algorithm for **berlin52.tsp** was found at **996th** generation with score **7848.415967213308 A screen shot of a graph

Description automatically generated**

The best solution using my Genetic Algorithm for **berlin11\_modified.tsp** was found at **10th** generation with score **4038.4379127952357** A screenshot of a computer

Description automatically generated

The best solution using my Genetic Algorithm for **kroA100.tsp** was found at **1000th** generation with score **25390.30619955761**

A screen shot of a computer

Description automatically generated

Part 5: Conclusions

In conclusion, the Genetic Algorithm is a strong approach for solving tricky problems like the Traveling Salesman one.

* Picking the right settings (parameters) really impacts how well it works, so adjusting them to match the specific problem is super important.
* In my case **best result** was **7848 with size of initial population** 200, m**utation probability** 10% and **tournament size** 20,while **the optimal solution** had to be **7542.**
* Even though it needs a lot of computing power and a bit of time to compute the result, Genetic Algorithm is great at finding really good solutions (sometimes even perfect), which makes it a top choice for solving Traveling Salesman Problems.