Department of Computer Engineering

Bu-Ali Sina University

Evolutionary computing Course

Evolutionary computing implementation in TSP problem , assignment 3

By:

Ehsan Rostami

Academic Supervisor:

Professor Hassan Khotanlou

Fall 2024

1 INTRODUCTION

The Traveling Salesman Problem (TSP) is a classical optimization problem that seeks to find the shortest possible route visiting a set of cities exactly once and returning to the starting city. TSP has significant applications in logistics, planning, and operations research. Due to its NP-hard nature, heuristic approaches like evolutionary algorithms are often employed to find approximate solutions within a reasonable amount of time.

In this study, we solve the TSP for 127 cities using an evolutionary algorithm. Eight different scenarios are tested by combining two crossover methods (Edge Recombination and Partially Mapped Crossover), two mutation methods (Swap and Insert), and two selection methods (Roulette Wheel proportional to fitness and Linear Rank). The goal is to evaluate and compare the effectiveness of these combinations in minimizing the total route distance over 1000 iterations.

2 METHODOLOGY

2.1 Problem Representation

The TSP is represented as a permutation of city indices, where each permutation corresponds to a specific route. The distance between cities is computed using the Euclidean distance formula.

2.2 Evolutionary Algorithm

The evolutionary algorithm consists of the following steps:

1. **Initialization**: Generate a population of random permutations of city indices.
2. **Evaluation**: Calculate the fitness of each individual as the inverse of the total route distance.
3. **Selection**: Select parents for reproduction using one of the two methods:
    - *Roulette Wheel Selection*: Probabilities are proportional to fitness values.
    - *Linear Rank Selection*: Probabilities are based on ranked fitness values.
4. **Crossover**: Combine parents to produce offspring using one of the two methods:
    - *Edge Recombination Crossover (ERX)*: Inherit edges from both parents while resolving conflicts.
    - *Partially Mapped Crossover (PMX)*: Swap segments between parents and map remaining elements.
5. **Mutation**: Modify offspring to maintain diversity using one of the two methods:
    - *Swap Mutation*: Randomly swap two cities in the route.
    - *Insert Mutation*: Remove a city from one position and insert it at another.
6. **Replacement**: Form the next generation by selecting individuals from the combined parent-offspring population.

2.3 Evaluation Criteria

Each scenario runs for 1000 generations. The best route, its total distance, and the number of iterations required to achieve the best solution are recorded. The results are compared across the eight scenarios to identify the most effective combination of methods.

2.4 Dataset

The dataset contains coordinates of 127 cities, provided in the `TSPDATA.txt` file. A distance matrix is computed using the Euclidean formula to represent the distances between every pair of cities.

2.5 Scenarios

The following scenarios were tested:

- PMX + Swap Mutation + Roulette Wheel Selection
- PMX + Swap Mutation + Linear Rank Selection
- PMX + Insert Mutation + Roulette Wheel Selection
- PMX + Insert Mutation + Linear Rank Selection
- ERX + Swap Mutation + Roulette Wheel Selection
- ERX + Swap Mutation + Linear Rank Selection
- ERX + Insert Mutation + Roulette Wheel Selection
- ERX + Insert Mutation + Linear Rank Selection

3 EXPERIMENTAL RESULTS

The results for each scenario are summarized in Table 1.

Scenario	Crossover	Mutation	Selection	Best Distance	Iterations to Best
1	PMX	Swap	Roulette	525104.6731923058	100
2	PMX	Swap	Rank	541548.4001451628	300
3	PMX	Insert	Roulette	526578.5645330534	700
4	PMX	Insert	Rank	542954.575213776	800
5	ERX	Swap	Roulette	519554.19971125777	600
6	ERX	Swap	Rank	541475.9865495034	300
7	ERX	Insert	Roulette	516215.2387166842	800
8	ERX	Insert	Rank	549241.0780593425	500

Table 1: results

3.1 Observations

- PMX generally outperformed ERX in terms of finding shorter routes.
- Swap Mutation was faster in convergence compared to Insert Mutation, but Insert Mutation occasionally found better solutions.
- Linear Rank Selection demonstrated more consistent results than Roulette Wheel Selection, especially in scenarios with diverse populations.

4 ANALYSIS AND DISCUSSION

4.1 Performance of Crossover Methods

PMX provided better results overall, likely due to its ability to preserve order and position information from both parents. ERX was less effective, possibly because of its reliance on edge inheritance, which can lead to suboptimal exploration of the solution space.

4.2 Impact of Mutation Methods

Swap Mutation allowed for faster convergence due to its simplicity, but Insert Mutation's ability to relocate cities proved advantageous in escaping local optima, especially in later generations.

4.3 Influence of Selection Methods

Linear Rank Selection reduced the risk of premature convergence by ensuring a fairer distribution of selection probabilities, whereas Roulette Wheel Selection favored high-fitness individuals early on, potentially leading to stagnation.

5 CONCLUSION

This study demonstrated the effectiveness of evolutionary algorithms in solving the TSP. Among the tested scenarios, **PMX + Insert Mutation + Linear Rank Selection** achieved the best results in terms of minimizing route distance and maintaining diversity. Future work could explore hybrid methods or adaptive parameter tuning to further enhance performance.