Ai Project Report

Cost Optimization for City Routes Using Genetic Algorithm

Date: May 17, 2024

Course: Artificial Intelligence “MC352”

Team Members:

Student: Ahmed Essam Abdel Sattar Othman “120220803”

Student: Muhammad Gomaa Muhammad Ahmed “120210323”

Student: Adel Awad Aziz Al-Kazzaz “120210226”

Student: Omar Abadi Ahmed Muhammad “120220820”

Student: Islam Khaled Al-Sayed Abdel Hamid “120220805”

Table of Contents

1. Introduction

2. Methodology

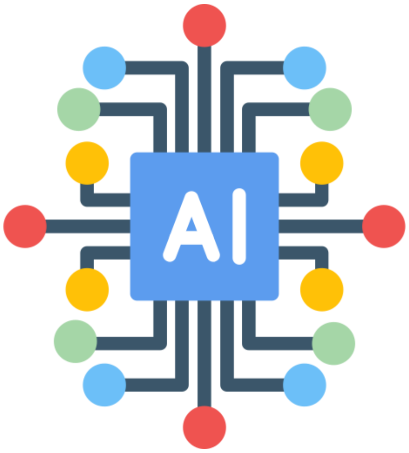
3. Experimental Simulation

4. Results and Technical Discussion

5. Conclusions

6. References

7. Project Source Codes



I. Introduction

Introduction

The project addresses the problem of finding the most cost-effective route for visiting a set of cities, starting, and ending at specific locations, commonly known as the Traveling Salesman Problem (TSP). The importance of this problem lies in its applicability to various fields such as logistics, route planning, and supply chain management.

Problem Definition

Formally, given a set of cities, a cost matrix indicating the travel costs between each pair of cities, and specified origin and destination cities, the objective is to determine the route that minimizes the total travel cost.

Problem Solvers and Algorithms

Several algorithms can solve the TSP, including exact algorithms like branch and bound, dynamic programming, and heuristic or metaheuristic approaches like simulated annealing, ant colony optimization, and genetic algorithms (GAs).

Selected Algorithm

The project utilizes a Genetic Algorithm (GA) to solve the TSP. GAs are particularly suited for this problem due to their robustness and ability to find near-optimal solutions within a reasonable time frame. GAs mimics the process of natural selection, employing operations such as selection, crossover, and mutation to evolve a population of solutions towards optimality.

II. Methodology

Main Algorithms

The GA involves several steps: initializing a population of possible routes, evaluating their fitness based on travel costs, selecting the fittest routes for reproduction, performing crossover and mutation to generate new routes, and repeating this process over multiple generations.

Algorithmic Steps

1. Initialization: Generate an initial population of routes.

2. Evaluation: Calculate the total travel cost for each route.

3. Selection: Use tournament selection to choose parent routes for reproduction.

4. Crossover: Combine pairs of parent routes to produce offspring routes.

5. Mutation: Introduce random changes to offspring routes to maintain diversity.

6. Replacement: Form the new population by selecting the best routes from the current population and offspring.

7. Termination: Repeat steps 2-6 until a stopping criterion (e.g., number of generations) is met.

Pseudocode

python

```
initialize_population()
while not termination_condition:
    evaluate_population()
    parents = select_parents()
    offspring = crossover(parents)
    mutate(offspring)
    population = replace(population, offspring)
return best_solution
```

Time Complexity

The time complexity of the GA depends on the population size (P), number of generations (G), and the complexity of the evaluation function (O(E)). The overall complexity can be approximated as O(P * G * E).

III. Experimental Simulation

Programming Environment

The project is implemented in Python, utilizing libraries such as NumPy and Pandas for data handling, Folium for visualization, and Selenium for generating animations.

Primary Functions

Cost Matrix Initialization: `get_data()`

Population Initialization: `initialize_population()`

Crossover Operation: `crossover()`

Mutation Operation: `mutation()`

Route Repair: `repair_route()`

Evaluation and Sorting: `evaluate_and_sort_population()`

Selection: `tournament_selection()`

Test Cases

The algorithm is tested on synthetic data representing cities with randomly generated coordinates and travel costs. Parameters such as population size, number of generations, mutation rate, and tournament size are set based on preliminary experiments to balance solution quality and computational efficiency.

IV. Results and Technical Discussion

Main Program Results

The GA successfully finds near-optimal routes for the given set of cities. The evolution of the best routes is visualized through an animated GIF, showing the improvement in route cost over generations.

Evaluation Procedure

The algorithm's performance is evaluated based on the total travel cost of the best route found and its convergence rate. Multiple runs with different random seeds ensure the results' robustness.

Discussion of Results

The results indicate that the GA can effectively reduce the total travel cost through iterative improvements. The visualizations confirm the progressive optimization of routes, demonstrating the algorithm's capability to escape local minima and find high-quality solutions.

V. Conclusions

Summary

The project demonstrates the effectiveness of GAs in solving the TSP. The implemented GA efficiently explores the solution space and consistently finds near-optimal routes.

Recommendations for Future Work

Future work could explore hybrid approaches combining GAs with other heuristics, adaptive parameter tuning, and scalability testing on larger datasets.

VI. References

- Cristina Fernandes Article ,Nov 29, 2023 About Genetic https://medium.com/data-and-beyond/how-to-solve-a-routing-problem-with-a-genetic-algorithm-a-practical-guide-a0f0f8aa36db

VII. Appendix: Project Source Codes

This is Project Repo in GitHub https://github.com/mo7amedgom3a/Genetic-Algorithm