Homework 5: Optimizing City Transport Routes Using Genetic Algorithms

Introduction

In this Homework, you will apply a Genetic Algorithm (GA) to optimize transportation routes in a fictional city. Your task is to design and implement a Genetic Algorithm to find the most efficient transport routes that connect various points of interest in a city while minimizing travel time and congestion.

Scenario and Data

You are given a set of locations in a fictional city, each representing a key point of interest like residential areas, business districts, schools, hospitals, and shopping centers. Your goal is to create routes that efficiently connect these locations.

Locations (Points of Interest)

- Residential Area: R1, R2, R3
- Business District: B1, B2
- Schools: S1, S2
- Hospital: H1
- Shopping Centers: SC1, SC2

Constraints

- Each route must connect at least one residential area to a different type of location.
- No route can exceed 10 units of distance.

Task

Develop a Genetic Algorithm to optimize the transport routes under the given constraints. Your GA should evolve solutions towards routes that connect more locations in less distance.

Genetic Algorithm Structure

- Representation (Chromosome): Represent a transport route as a chromosome, a sequence of locations (e.g., R1-B1-S1-H1).
- Initial Population: Randomly generate a set of routes adhering to the constraints.
- Fitness Function: Define a function that evaluates each route based on length and diversity of locations connected.
- Selection: Implement a method to select parent routes for breeding (e.g., tournament selection, roulette wheel selection).
- Crossover: Define a crossover method to produce new offspring (e.g., single-point, two-point).
- Mutation: Introduce random changes in the offspring to maintain diversity (e.g., swapping locations in a route).
- New Generation: Replace the old generation with the new generation of routes.
- Termination: Define a termination condition (e.g., number of generations, convergence criteria).

Requirements

- Algorithm Implementation: Implement the Genetic Algorithm as described.
- Optimization: Run the GA to optimize the transport routes.
- Documentation: Document your code, the algorithm's parameters, and the results.

Submission

Submit your code along with a description of your algorithm's design, the results it achieved, and your analysis of its performance.

Here's a code outline to help you get started:

1. Define Chromosome Structure:

   - Represent a transport route as a sequence of locations (e.g., R1-B1-S1-H1).

2. Initialize Population:

   - Randomly generate a set of routes adhering to the constraints.

3. Define Fitness Function:

   - Create a function to evaluate each route based on length and diversity of locations.

4. Selection Process:

   - Implement a method to select parent routes for breeding (e.g., tournament or roulette wheel selection).

5. Crossover:

   - Define a crossover method to produce new offspring (e.g., single-point or two-point crossover).

6. Mutation:

   - Introduce random changes in the offspring to maintain diversity (e.g., swapping locations in a route).

7. Create New Generation:

   - Replace the old generation with the new generation of routes.

8. Termination:

   - Define a termination condition (e.g., number of generations, convergence criteria).

9. Run the GA:

   - Loop through the process of selection, crossover, mutation, and creating new generations until termination criteria are met.

10. Analyze and Document:

   - Analyze the performance of your GA and document the results.

Hints:

- Focus on how your fitness function evaluates the efficiency of routes.
- Experiment with different selection, crossover, and mutation strategies to see which yields the best results.
- Make sure your routes adhere to the specified constraints (e.g., connecting different types of locations, distance limits).
- Pay attention to how your algorithm converges to ensure it is effectively optimizing the routes.