EVOLUTIONARY ALGORITHMS

PROJECT

BY:
Ziad Hany 20210380
Ziad Mohamed 20210372
Aya Hassan Sayed 20210205
Shrouk Mohamed 20210449
Ziad Abdlalem 20210365

TABLE OF CONTENTS

PROJECT IDEA

Large-Scale Route Optimization Problem using an Evolutionary
Algorithm

The customer is a
manufacturing company. They have warehouses in different locations. When they receive orders from their clients, a planner needs to plan the item-to-truck assignment for order delivery. The planner also needs to decide the route of each delivery truck, namely, the sequence of the stops to deliver orders to different destinations. A delivery assignment has its associated cost determined by the type of the assigned delivery truck and the corresponding travelling distance. The optimization objective here is to minimize the overall delivery cost.
This is a variant of the vehicle routing problem (VRP). The constraints modelled are:
- There are different types of trucks from which we can choose. A truck has a capacity limit on both area and weight. (We assume that there is no limit on the number of trucks for each type)
- An item is only available by a specific time. A truck can start only when all items assigned to it are available.

- The available time difference between the earliest and last available items in the same truck should be less than a user-defined limit (e.g., 4 hours).
- All items need to be delivered to their destinations before their deadlines.
- Depending on their properties, some products can be put in the same truck, but some cannot.
- A truck can have N stops at most, where N is a user-defined number.
- A truck must stay at each stop for M hours to unload the items, where M is a user-defined number. Each stop will incur a fixed cost in addition to the delivery cost

MAIN FUNCTIONALITIES

01

Data Integration and Preprocessing: Integrate and preprocess the Large-Scale Route Optimization dataset, including information on waste collection points, road networks, vehicle capacities, and time constraints.

02

Route Optimization Algorithms: Implement and compare various optimization algorithms such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) to find optimal waste collection routes.

03

Machine Learning Models: Develop machine learning models to predict waste generation patterns based on historical data, weather conditions, and other relevant factors.

04

Visualization: Visualize optimized waste collection routes on interactive maps, highlighting key metrics such as distance traveled, number of vehicles used, and collection time.

05

Performance Evaluation: Evaluate the performance of different optimization algorithms and machine learning models based on objective metrics such as route efficiency, resource utilization, and environmental impact.

ction time.

SIMILAR APPLICATIONS

VEHICLE ROUTING PROBLEM (VRP)

VRP involves optimizing the routes of a fleet of vehicles to serve a given set of customers while minimizing costs
Evolutionary algorithms like Genetic Algorithms (GA), Genetic Programming (GP), or Particle Swarm Optimization (PSO) are commonly employed to find near-optimal solutions for VRP variants like Capacitated VRP (CVRP)

TRAVELING SALESMAN PROBLEM (TSP)

TSP aims to find the shortest possible route that visits each city exactly once and returns to the origin city.
Evolutionary algorithms, particularly Genetic Algorithms and Ant Colony Optimization (ACO), are used to solve large-scale instances of TSP where exact algorithms become impractical.

THE DATASET EMPLOYED

Route Optimization - A Real World Scenario

The dataset contains information relevant to route optimization problems, including geographic coordinates of waste collection points, road network data, vehicle capacities, and time constraints.

DETAILS OF THE ALGORITHM

There are two types of algorithms that have been used

GENETIC ALGORITHMS (GA)

1. **Initialization:**
   - Generate an initial population of candidate solutions
2. **Fitness Evaluation:**
   - Evaluate the fitness of each individual in the population
3. **Selection:**
   - Select individuals from the current population to serve as parents for the next generation.
4. **Recombination (Crossover):**
   - Create offspring individuals by combining genetic material from selected parents.
5. **Mutation:**
   - Introduce random changes in offspring individuals to maintain genetic diversity and explore new regions of the search space.
6. **Replacement:**
   - Form the next generation population by replacing some individuals from the current population with offspring individuals.
7. **Termination:**
   - Repeat the selection, crossover, mutation, and replacement steps for multiple generations until a termination condition is met.
8. **Result:**
   - The final population after termination contains individuals with solutions that approximate the optimal or near-optimal solutions to the optimization problem.

PARTICLE SWARM OPTIMIZATION (PSO)

1. **Initialization :**

- Initialize a population of particles in the search space.

2. **Fitness Evaluation:**

- Evaluate the fitness of each particle based on its position in the search space.

3. **Updating Particle Velocity:**

- Update the position of each particle based on its current position and velocity.

4. **Updating Particle Position:**

- Update the personal best position (individual memory) for each particle if its current position yields a better fitness value than its previous best.

5. **Termination:**

- Repeat steps 2 to 5 for multiple iterations (generations) until a termination condition is met. Termination conditions can include reaching a maximum number of iterations, achieving a satisfactory solution quality, or stagnation in improvement over successive iterations.

RESULT OF THE EXPERIMENT

- result of the experim :

  Programming Language: Python

  Libraries: SciPy, NumPy, Pandas, Matplotlib , Scikit-learn

  Tools: Jupyter Notebook for development and experimentation, GitHub for version control and collaboration.

- **DEVELOPMENT PLATFORM:**

  PROGRAMMING LANGUAGE: PYTHON

  LIBRARIES: SCIPY, NUMPY, PANDAS, MATPLOTLIB , SCIKIT-LEARN

  TOOLS: JUPYTER NOTEBOOK FOR DEVELOPMENT AND EXPERIMENTATION, GITHUB FOR VERSION CONTROL AND COLLABORATION.