كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

AI Project

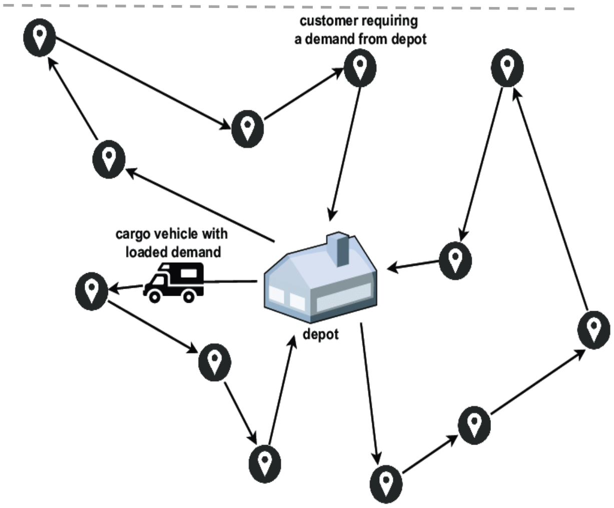
Solving VRP

"Solving Vehicle Routing Problem" using both Genetic Algorithms & Differential Evolution

CONTENTS

Solving VRP Using Both Genetic Algorithms & Differential Evolution

Introduction and Overview:



The project focuses on solving the Vehicle Routing Problem (VRP), a combinatorial optimization problem that aims to find the optimal set of routes for a fleet of vehicles to deliver to a given set of customers. The VRP generalizes the Traveling Salesman Problem (TSP) and is crucial for optimizing the delivery operat ions of a company.

Similar Applications:

Existing applications addressing VRP include [Google Maps and Google Route Planner, Route4Me, OptimoRoute, WorkWave Route Manager], which optimize vehicle routes for efficient delivery operations. These applications typically employ various optimization algorithms to minimize transportation costs.

Literature Review:

In researching our project, we consulted a range of academic publications:

I. **A genetic algorithm for the vehicle routing problem**
   - A genetic algorithm for the vehicle routing problem - ScienceDirect

II. **A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants**
   - A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants - ScienceDirect

III. **Ant colony optimization for real-world vehicle routing problems**
   - Ant colony optimization for real-world vehicle routing problems | Swarm Intelligence (springer.com)

IV. **Differential Evolution: A Survey and Analysis**
   - Applied Sciences | Free Full-Text | Differential Evolution: A Survey and Analysis (mdpi.com)

V. **Solving vehicle routing problem by using improved genetic algorithm for optimal solution**

- Solving vehicle routing problem by using improved genetic algorithm for optimal solution - ScienceDirect

Proposed Solution

The Project aims to solve the VRP by determining a set of routes (S) for vehicles, starting, and finishing at their respective depots, to fulfill customer demands while minimizing the overall transportation cost.

Main Functionalities/Features:

The primary features include:

- Genetic Algorithm-based optimization
- Differential Evolution-based optimization
- Visualization of optimized routes

Use-Case Diagram:

Include a use-case diagram illustrating the interaction between different system components and user actions.

Applied Algorithms:

I. **Genetic Algorithm:**

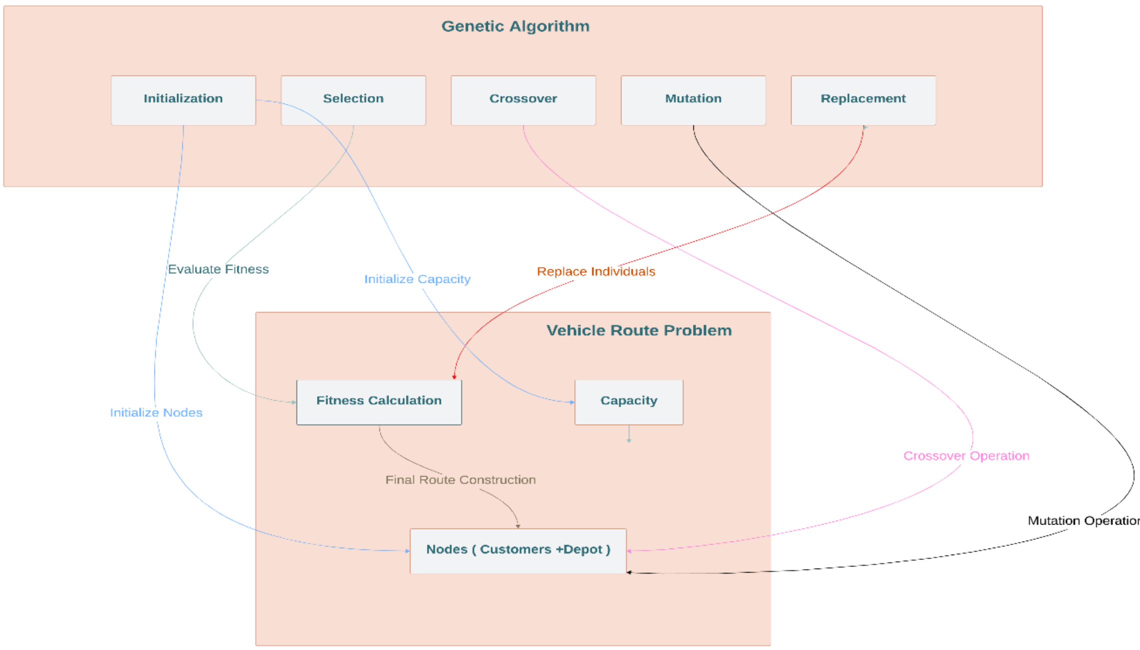
The Genetic Algorithm is employed to evolve a population of potential solutions iteratively. It involves processes such as selection, crossover, and mutation to generate optimized routes for the vehicles.

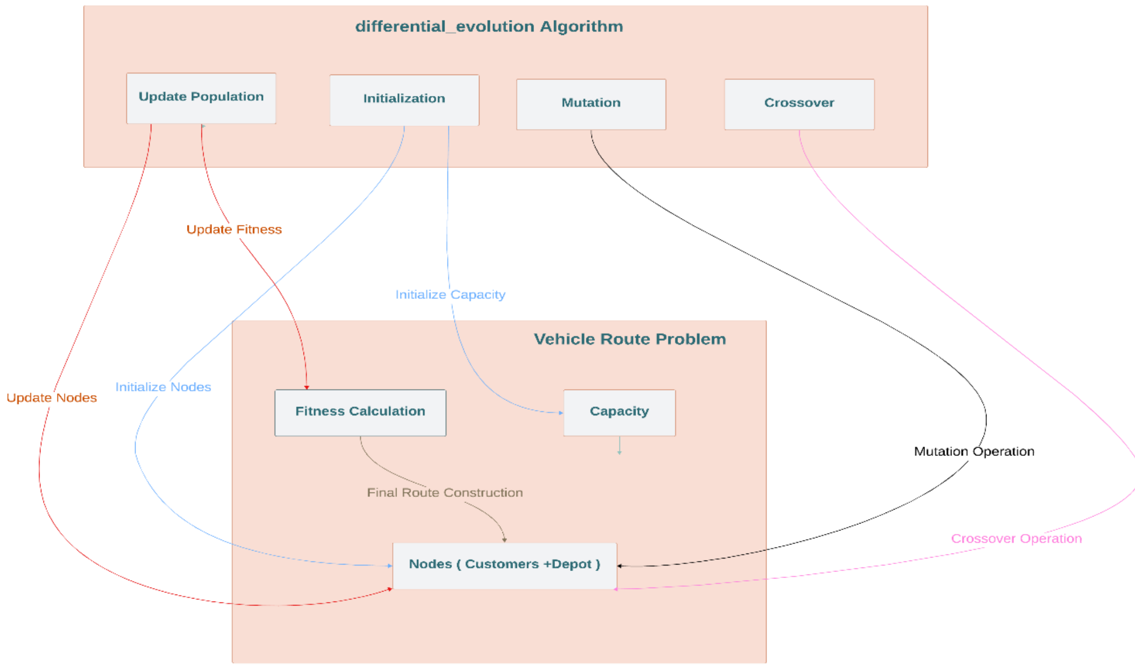
II. **Differential Evolution:**

Differential Evolution is utilized as an alternative optimization algorithm. It operates by combining the information from different candidate solutions to create new potential solutions, iteratively improving the overall quality of the routes.

Block Diagram:

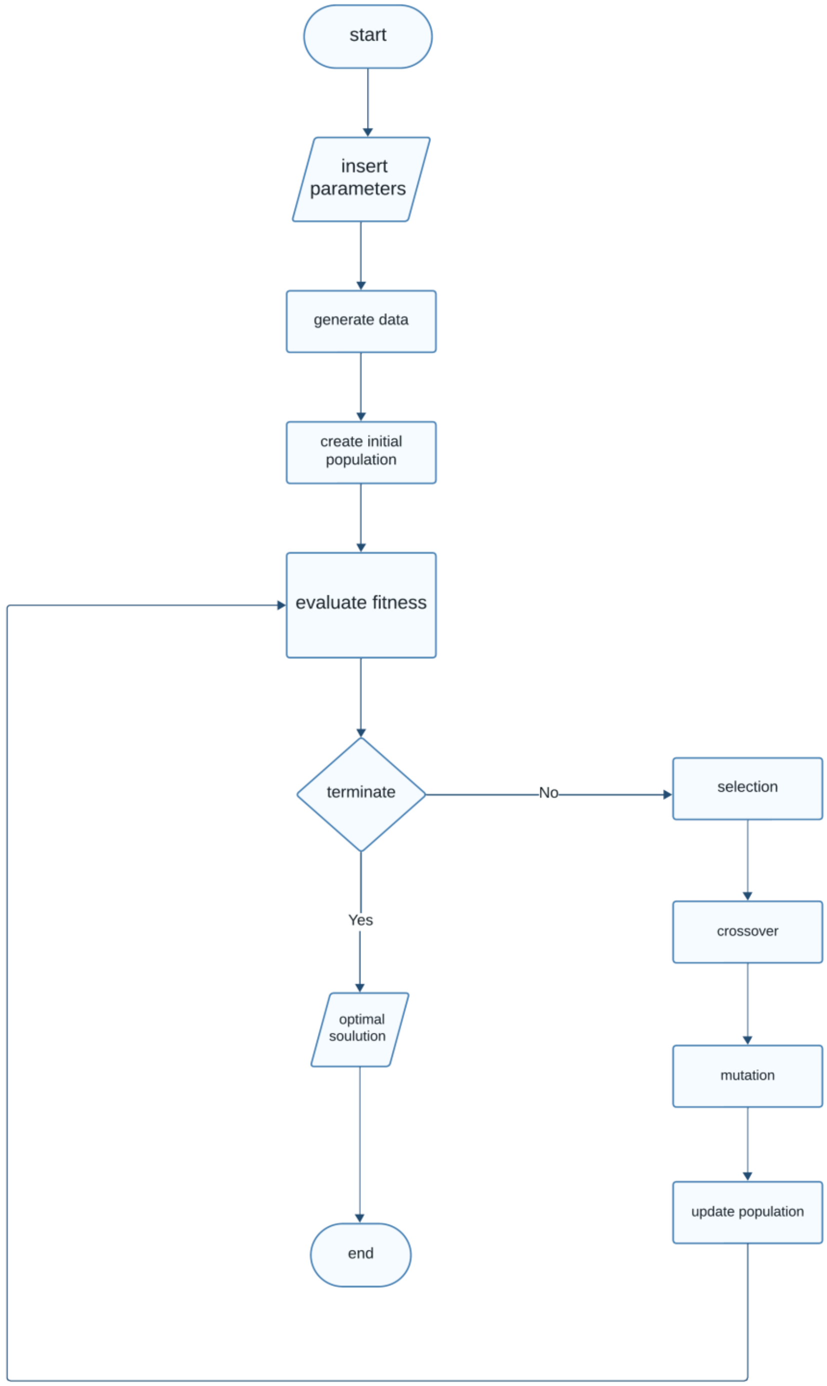
Genetic:



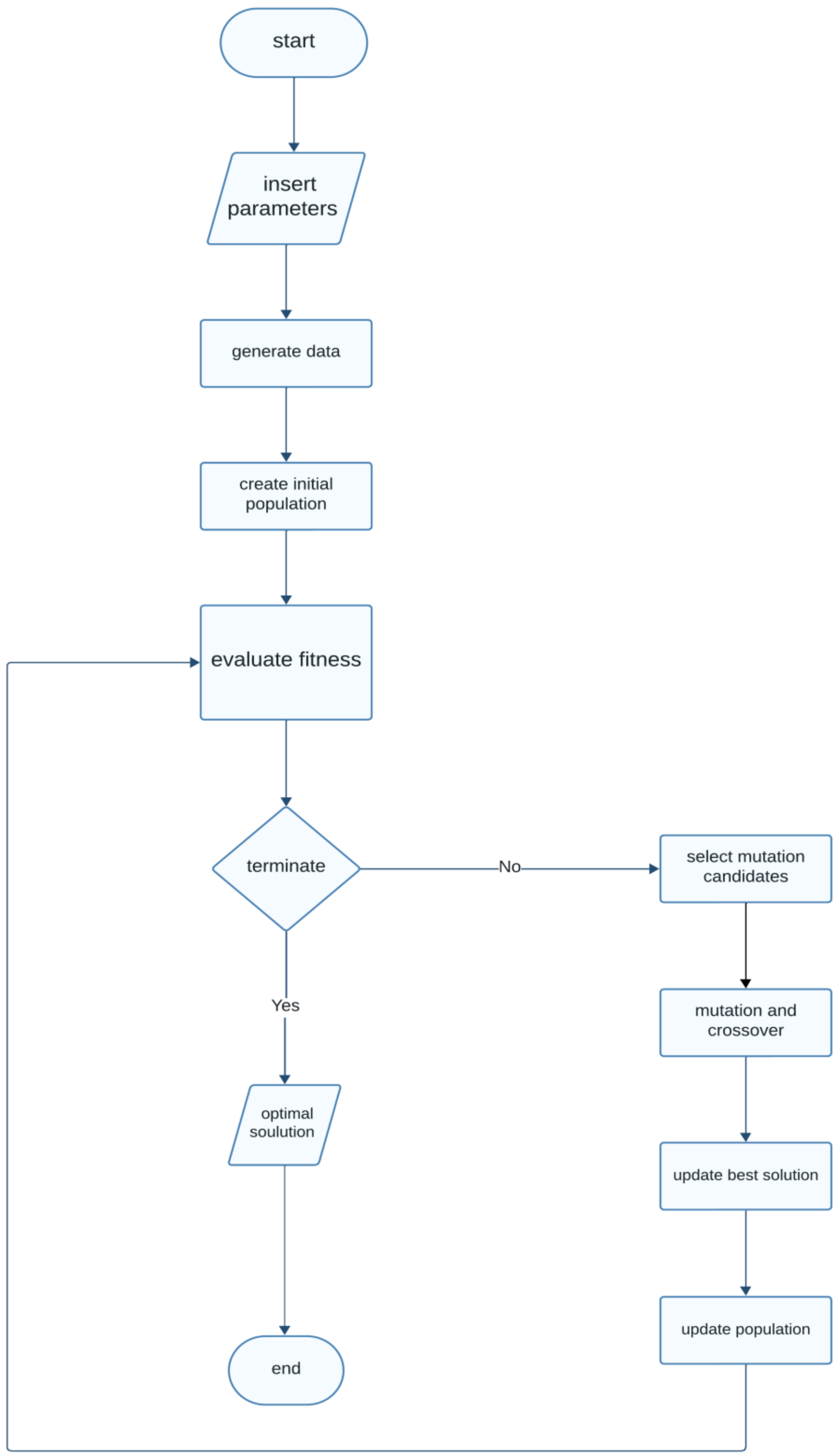


Flowchart Diagram:

Genetic:


start
insert parameters
generate data
create initial population
evaluate fitness
terminate
No
Yes
selection
crossover
mutation
update population
optimal soulution
end

Differential:



Analysis, Discussion, and Future Work:

Analysis of Results:

The analysis revealed that both Genetic Algorithms and Differential Evolution perform well in optimizing vehicle routes. Each algorithm has its strengths and weaknesses, and the choice between them may depend on specific constraints and objectives.

Advantages/Disadvantages

Genetic Algorithm (GA):

Advantages:

- ✓ **Global Optimization: GAs explore a diverse solution space, increasing the likelihood of finding global optima.**
- ✓ **Adaptability: GAs can handle various problem types and constraints, making them versatile.**
- ✓ **Parallel Processing: They can be parallelized, allowing for efficient computation on modern hardware.**

Disadvantages:

- ✓ **Computational Complexity: GAs can be computationally demanding, especially for large-scale problems.**
- ✓ **Parameter Sensitivity: The performance is sensitive to parameter settings, requiring careful tuning.**
- ✓ **Premature Convergence: There's a risk of converging to suboptimal solutions prematurely.**

Differential Evolution (DE):

Advantages:

- ✓ **Efficiency: DE is known for its simplicity and efficiency in finding solutions for optimization problems.**
- ✓ **Fewer Parameters: DE typically has fewer parameters to tune compared to GAs, simplifying the optimization process.**
- ✓ **Robustness: DE is often robust to parameter settings, making it easier to implement and use.**

Disadvantages:

- ✓ **Limited Exploration: DE may not explore the solution space as extensively as GAs in some cases.**

- ✓ **Dependency on Initialization: The choice of the initial population can influence DE's performance.**
- ✓ **Less Effective for Discrete Problems: DE is originally designed for continuous optimization and may need adaptations for discrete problems.**

Rational Behind Algorithm Behavior

Genetic Algorithm (GA):

1. **Population-Based Exploration:**
    - **Rationale:**
      GAs operate with a population of potential solutions (chromosomes). Through processes like selection, crossover, and mutation, the algorithm explores a diverse set of solutions. This allows GAs to perform a global search across the solution space, increasing the likelihood of finding optimal or near-optimal solutions.
2. **Genetic Diversity:**
    - **Rationale:**
      The genetic diversity introduced through crossover and mutation operations helps prevent premature convergence. It encourages the exploration of different regions of the solution space, promoting the discovery of diverse and potentially better solutions.
3. **Fitness-Based Selection:**
    - **Rationale:**
      GAs employ a fitness function to evaluate the quality of solutions. This selection mechanism guides the algorithm toward areas of the solution space that exhibit better fitness values, steering the optimization process in the direction of improved solutions.
4. **Iterative Improvement:**
    - **Rationale:**
      The iterative nature of GAs allows for continuous improvement over generations. As the algorithm progresses, it refines the population, focusing on solutions with higher fitness values and converging towards optimal or near-optimal solutions.

Differential Evolution (DE):

1. **Vector Difference Operations:**
    - **Rationale:**
      DE operates by creating new candidate solutions through vector difference operations (mutation and crossover). This differential strategy enables the

algorithm to combine information from different solutions, facilitating efficient exploration of the solution space.

2. **Population Dynamics:**
   - **Rationale:**
     DE maintains a population of solutions, and its dynamics are influenced by the competition between the current solutions and the newly generated trial solutions. This competitive interaction helps drive the optimization process towards regions of the solution space with better fitness values.
3. **Parameter Control:**
   - **Rationale:**
     DE often has fewer tunable parameters compared to GAs. The simplicity of parameter control contributes to the algorithm's ease of use. The algorithm's robustness to parameter settings is a result of its ability to adapt effectively to diverse problem instances.
4. **Versatility in Problem Types:**
   - **Rationale:**
     DE is designed to be versatile and is effective in solving a wide range of optimization problems. Its simplicity and efficiency make it a popular choice, especially in scenarios where computational resources are limited.

Future Work:

Future work may involve:

- Further parameter tuning for improved performance.
- Integration of additional metaheuristic algorithms.
- Real-world testing and validation with live delivery data.

Conclusion:

In conclusion, the application successfully addresses the VRP using Genetic Algorithms and Differential Evolution. The comprehensive documentation provided serves as a guide for users and developers to understand the project's inner workings. The insights gained contribute to the ongoing research and application of optimization algorithms in solving real-world logistics challenges.

Development Platform

Programming Language:

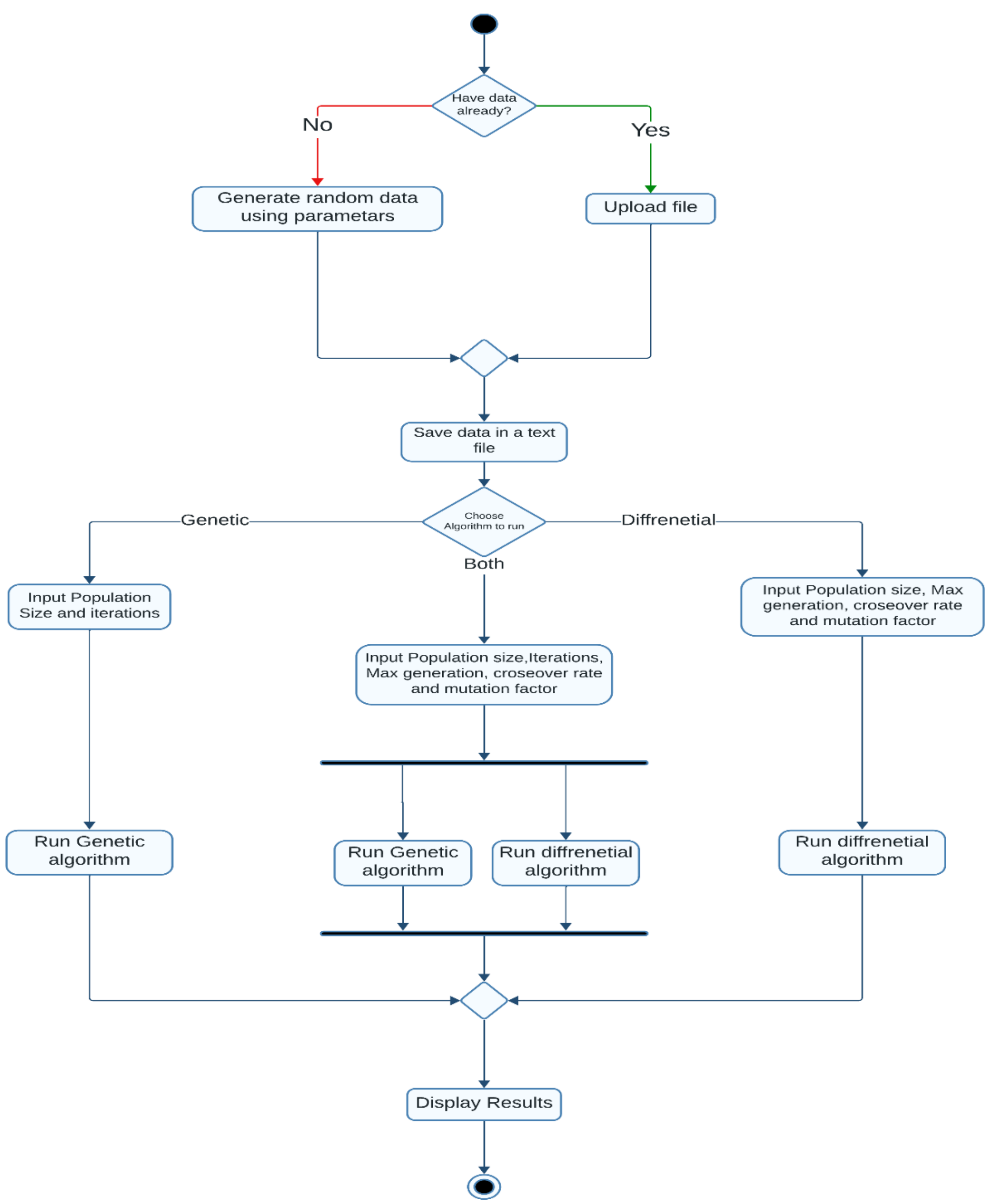
Use Python as the primary programming language.

Libraries

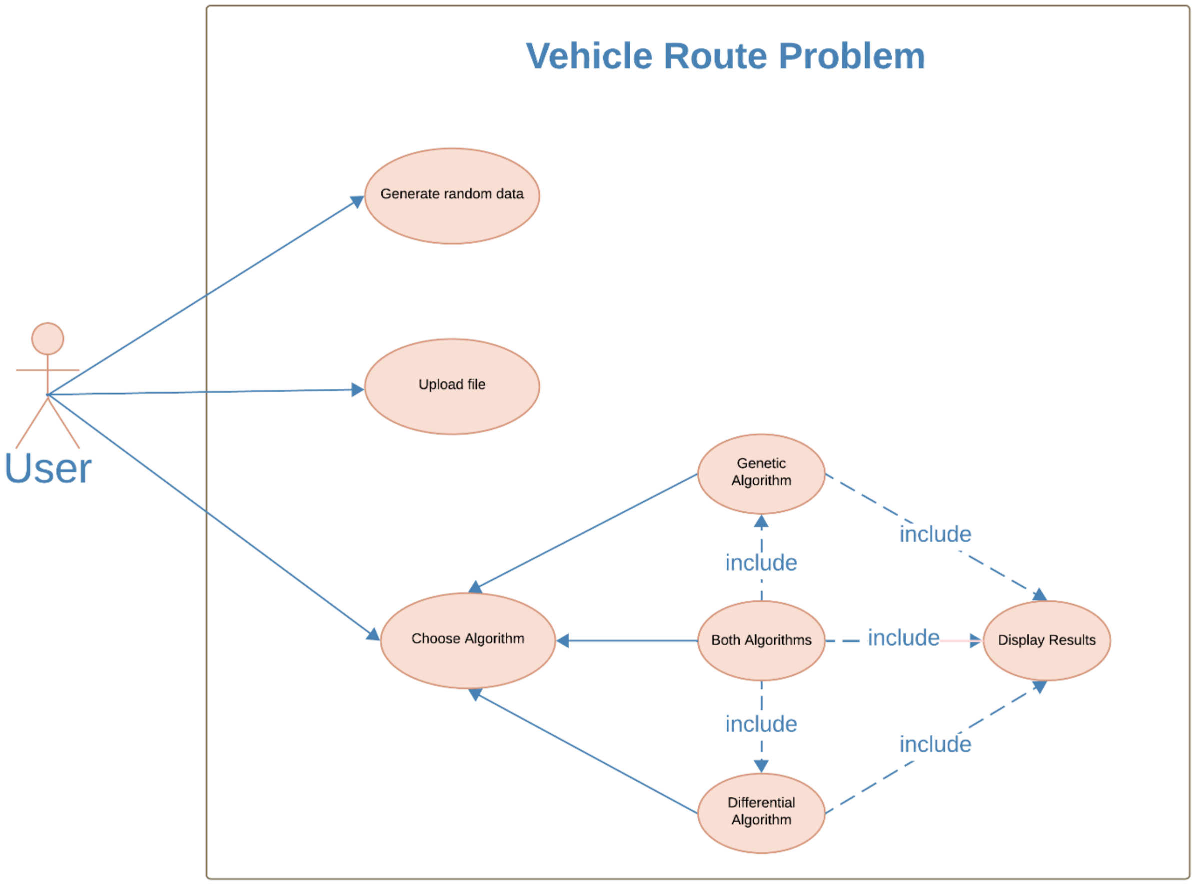
- NumPy: for numerical operations.
- Matplotlib: for visualization (optional but helpful for debugging and analysis).
- **Matplotlib: For plotting and visualizing the routes.**
- **sys:** For system-related operations (though it may not be heavily used in this context).
- **Random:** For generating random numbers.
- **Math:** For mathematical functions and operations.
- **Matplotlib.patches:** Specifically for drawing shapes in Matplotlib plots.

UML Diagram:

Activity:



Use Case:



Team Members:

ID	NAME
20210715	مازن اسلام صبحى
20210694	كيرلس سامى عطا الله
20210695	كيرلس سليمان وهيب
20210712	مارينا عطاس
20210669	فام فايز
20210710	مارينا ايمن حنا
20210679	فيلوباتير عزت رزق الله

Code Repository:

https://github.com/N1ghtHunter/vrp_solver