Artificial Intelligence Project

(VRP: Vehicle Routing Problem)

Team Member:

ID	Name	Level	Department
20220180	Ziad Adel Farghaly Mehran	3	AI
20220179	Ziad Tarek Galal Elsayed	3	AI
20220191	Ziad Waleed Mohamed	3	AI
20220181	Ziad Atef Ali Abdelsalam	3	AI
20220167	Rawan Hesham Mostafa Abdelhafez	3	AI
20220104	Bassel Waleed hamd Mohamed Ibrahim	3	CS

Repo Link: https://github.com/ZiadWaleed2003/VRP-Vehicle-Routing-Problem-with-Genetic-Algorithms-Differential-Evolution

INTRODUCTION

Overview:

The Vehicle Routing Problem (VRP) is a fundamental challenge in combinatorial optimization and integer programming, widely studied in the fields of logistics and transportation. It seeks to determine the most efficient set of routes for a fleet of vehicles to deliver goods or services to a predefined set of customers. The problem extends the Traveling Salesman Problem (TSP) by involving multiple vehicles, depots, and additional operational constraints.

Delivery company operates a fleet of vehicles from one or more depots (Location that start from or end at), to service customer demands spread across a road network. The objective is to construct routes for each vehicle, starting and ending at their respective depots, while ensuring that customer requirements are met, operational constraints such as vehicle capacity are adhered to, and the overall transportation cost is minimized. This cost could represent monetary expenditure, total travel distance, or other resource-related metrics.

VRP plays a crucial role in optimizing supply chains, improving delivery efficiency, and reducing operational expenses, making it a cornerstone of modern logistics and transportation planning.

Table of Contents

Application On Project:

Jumia, a leading e-commerce platform, exemplifies the practical application of the Vehicle Routing Problem (VRP) in its logistics operations. Jumia connects vendors and customers through its online platform and relies on a robust delivery network to ensure that orders reach customers efficiently. To achieve this, the platform incorporates various functionalities and features to address the challenges of route optimization, customer demand, and operational constraints.

- Functionalities/Features and How They Work:
  - Orders placed by customers are grouped based on their geographic location, urgency, and delivery time windows.
  - Using VRP algorithms, Jumia optimizes delivery routes in real-time to minimize costs, reduce delivery times, and accommodate last-minute changes such as new orders or cancellations.
  - Routes are recalculated dynamically if there are traffic updates or unexpected delays, ensuring maximum efficiency.
  - Jumia operates from several warehouses or hubs (depots) to reduce the distance to customers.
  - The VRP framework helps determine the optimal depot for each order and designs vehicle routes accordingly.
  - The system integrates these preferences into its routing algorithms to ensure on-time delivery.

- How It Works:

  The VRP algorithms underpinning Jumia's logistics system analyze multiple variables simultaneously: customer locations, vehicle constraints, depot positions, traffic conditions, and delivery schedules. The system continuously refines the routes to meet both business objectives and customer satisfaction goals, ensuring that products are delivered in a timely, cost-effective manner.

By integrating advanced VRP-based solutions, platforms like Jumia streamline their operations, reduce costs, and enhance the customer experience, setting a high standard for modern e-commerce logistics.

References and Citation:

1. The Vehicle Routing Problem: State-of-the-Art Classification and Review.
   URL: https://www.mdpi.com/2076-3417/11/21/10295#

2. Review of Research on Vehicle Routing Problems.
   Citation: Daoriao Hao, Xinhua Mao, Yingtao Wei, Lu Sun, and Yang Yang "Review of research on vehicle routing problems", Proc. SPIE 13018, International Conference on Smart Transportation and City Engineering (STCE 2023), 130180Y (14 February 2024).

3. Joint Approach for Vehicle Routing Problems Based on Genetic Algorithm and Graph Convolutional Network.
   Citation: https://doi.org/10.3390/math12193144

4. An Improved Genetic Algorithm for Vehicle Routing Problem with Time-Window.
   URL: https://ieeexplore.ieee.org/document/9698698

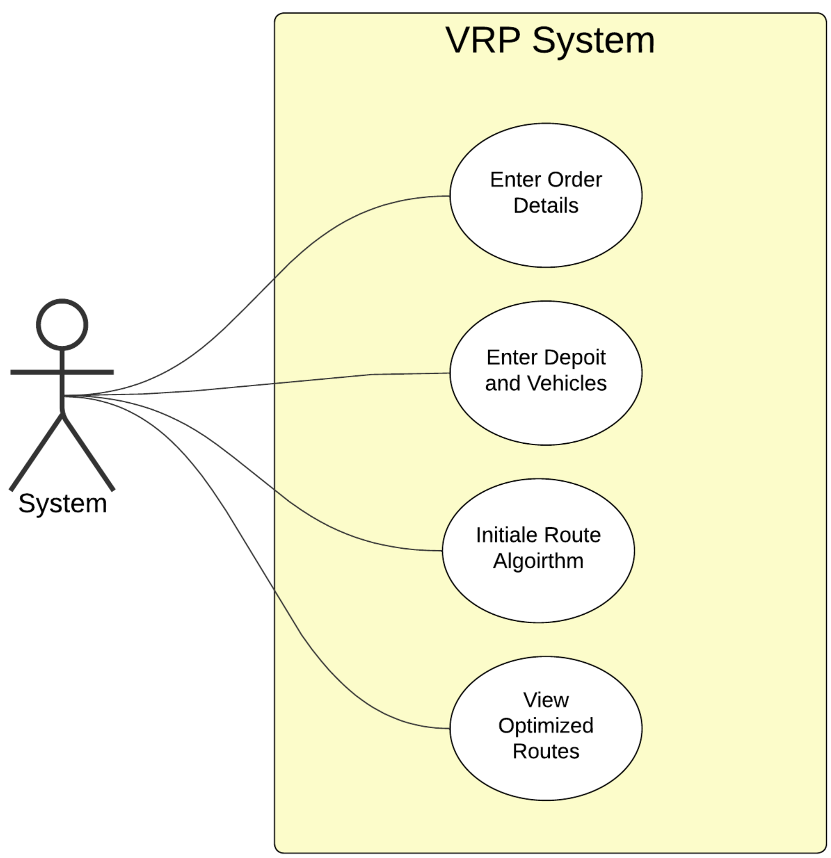
5. Solving the Capacitated Vehicle Routing Problem with a Genetic Algorithm.
   Citation: https://doi.org/10.1016/j.apm.2011.08.010
6. The Differential Evolution Algorithm for Solving Vehicle Routing Problems.

   Citation: https://doi.org/10.1016/j.swevo.2023.101245

Proposed Solution

Main functional:

The proposed Solution aims to optimize vehicle routing by minimizing delivery costs while meeting constraints like vehicle capacity, customer demands, and time windows.



Algorithms

Genetic Algorithm:

Genetic algorithms are a type of evolutionary algorithm inspired by natural selection and biological evolution. They are used to solve optimization and search problems by mimicking processes such as selection, crossover, mutation, and survival of the fittest.

Principles of Genetic Algorithm that will be inspired from:

1. **Population:** A group of potential solutions (chromosomes).
2. **Selection:** Choose the best individuals to be parents based on a fitness function.
3. **Crossover:** Combine two parents to create offspring by exchanging genetic information.
4. **Mutation:** Introduce random changes to maintain diversity and explore new solutions.
5. **Fitness Function:** Measures the "quality" of a solution, such as minimizing cost (distance).

Vehicle Routing Problem (VRP) involves finding optimal routes for a fleet of vehicles to deliver to customers while satisfying constraints (Capacity of vehicles, Min distance, Demand of each customer). Genetic algorithms can solve VRP by encoding potential routes as (Solution) chromosomes and applying evolutionary strategies to find the most efficient delivery routes.

Steps for Our GA Algorithm:

1. Define the initial population:

    - The initial population consists of randomly generated solutions (routes). Each solution represents a set of routes assigned to the vehicles.

- Routes are created using **permutation techniques** to ensure valid delivery sequences while adhering to constraints like vehicle capacity.

2. Crossover:

   - Combines genetic information from two parent solutions to produce new offspring.
   - Select two parent routes (solutions).
   - Retain the order of cities from the first parent while integrating a subset of the second parent's route.

3. Mutation:

   - Random changes to offspring solutions to maintain diversity and explore new areas of the solution space by Swap.

4. Evolve (Selection and Replacement):
   - Choose parents for the next generation based on their fitness score (solutions with shorter routes are better).
   - Repeated until reach maximum number of Iteration.

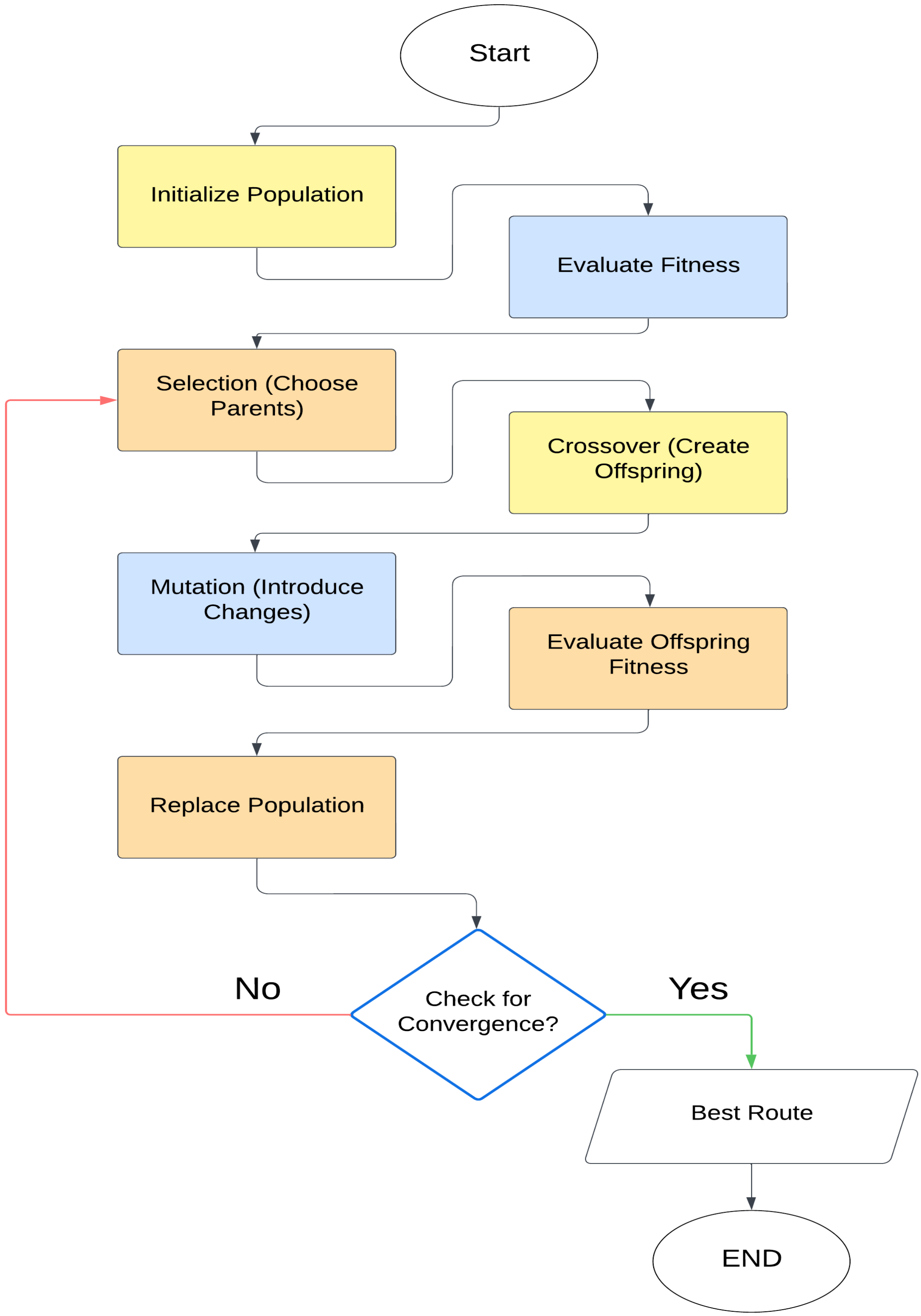
5. Fitness Function:

   - Minimize the total delivery cost, measured by the total distance traveled by all vehicles in their respective routes.

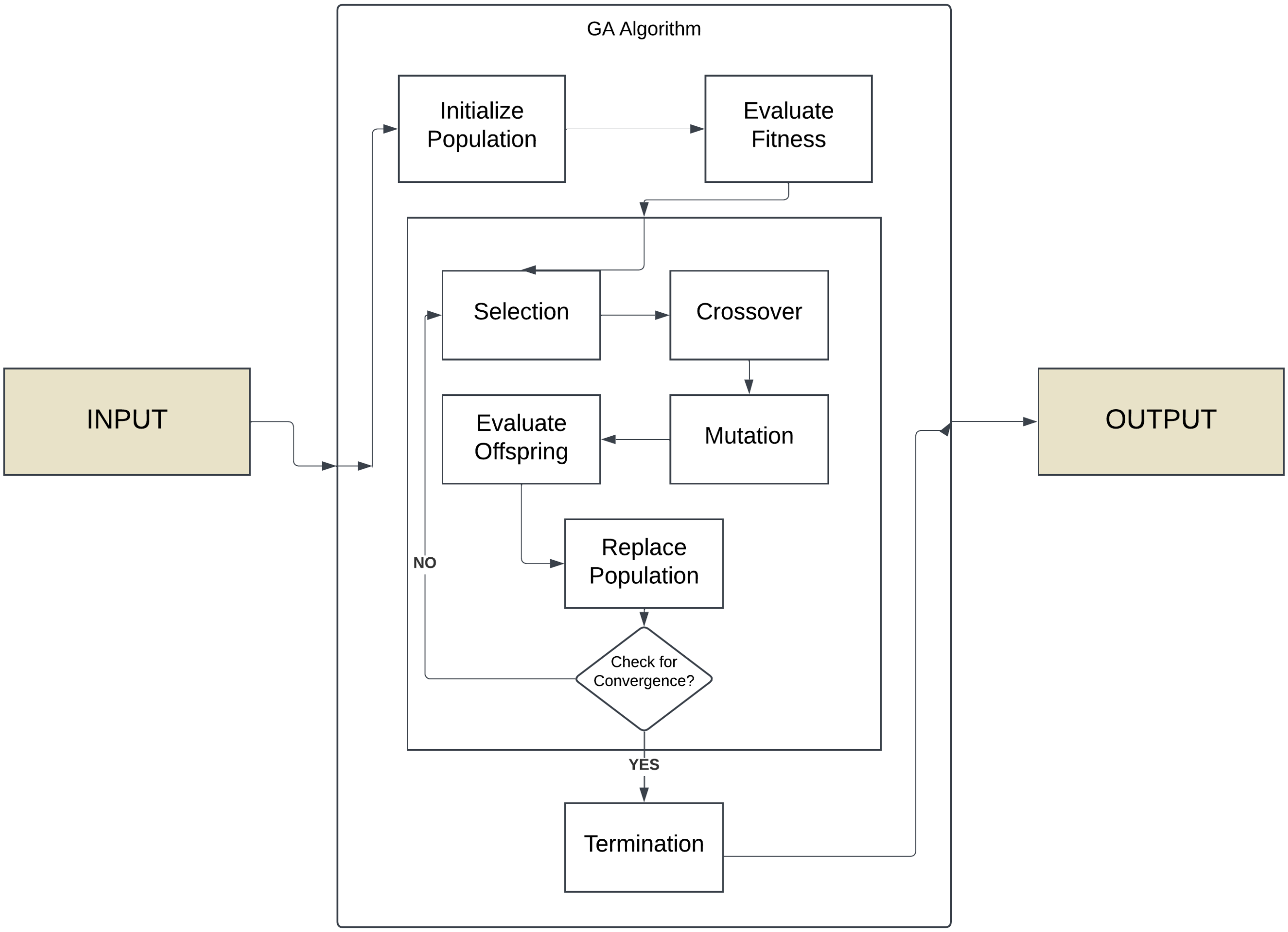
Class Diagram Genetic Algorithm:

Genetic Algorithm
- max_iter : int - num_customers : int - customers = [{ }] - num_vehicles : int - vehicle_capacity : int - customer_demand: int - population_size : int - mutation_rate : float - depot_location : [ ] - early_stop: int
- InitializingPopulation() - FitnessFunction() - CalculateDistance() - SelectParents() - order_crossover() - repair_solution() - SwapMutationFunction() - enforce_capacity() + Evolve()

Flow Chart Genetic Algorithm:



Block Diagram Genetic Algorithm:



Differential Evolution:

It is similar to Genetic Algorithms in that it uses a population of candidate solutions (individuals) to search for optimal solutions by applying operations like mutation, crossover, and selection.

DE is specifically powerful when dealing with multidimensional, noisy, and nonlinear search spaces. Unlike traditional genetic algorithms, DE directly works on the differences between randomly selected individuals in the population to produce new candidate solutions.

Principles of Differential Evolution:

1. **Initial Population**: Randomly generate feasible routes as candidate solutions for the vehicles.
2. **Mutation**: Use the differential mutation process to create new potential solutions.
3. **Crossover**: Combine the mutated solutions with existing solutions based on a crossover probability.
4. **Selection**: Select the best solutions based on their fitness score.
5. **Fitness Function**: Evaluate solutions based on the total distance traveled by the fleet of vehicles.
6. **Convergence**: Continue evolving the population until a satisfactory solution is found or a stopping condition is met.

Steps for Our DE Algorithm:

1. Define the initial population:

    - population of candidate solutions (routes) is randomly generated. Each solution (or individual) represents the routes assigned to each vehicle in the fleet.
    - Permutation Techniques: The customer orders for each vehicle must be distinct, and the initial population can be generated by randomly permuting customer orders while

ensuring that each route respects the vehicle capacity constraint and other VRP constraints.

2. Mutation:

   - Generates new candidate solutions (offspring) by combining solutions from the population. In DE, mutation is performed using the formula:

     Mutant vector = r1 + F × (r2 − r3)

   Where:

     - r1,r2,e3 are randomly chosen vectors (solutions) from the population.
     - F is the mutation factor, which controls the step size of the mutation (usually a value between 0 and 1).

3. Crossover:

   - Step to combines the information from the mutated vector and the parent vector to produce a new candidate solution. The crossover process is controlled by a **crossover probability** CR, which determines how much of the new solution comes from the mutant vector versus the original parent vector.

4. Evolve (Selection and Replacement):

   Selection determines whether the new candidate solution (offspring) should replace an existing one in the population. The new candidate solution is evaluated based on its fitness.

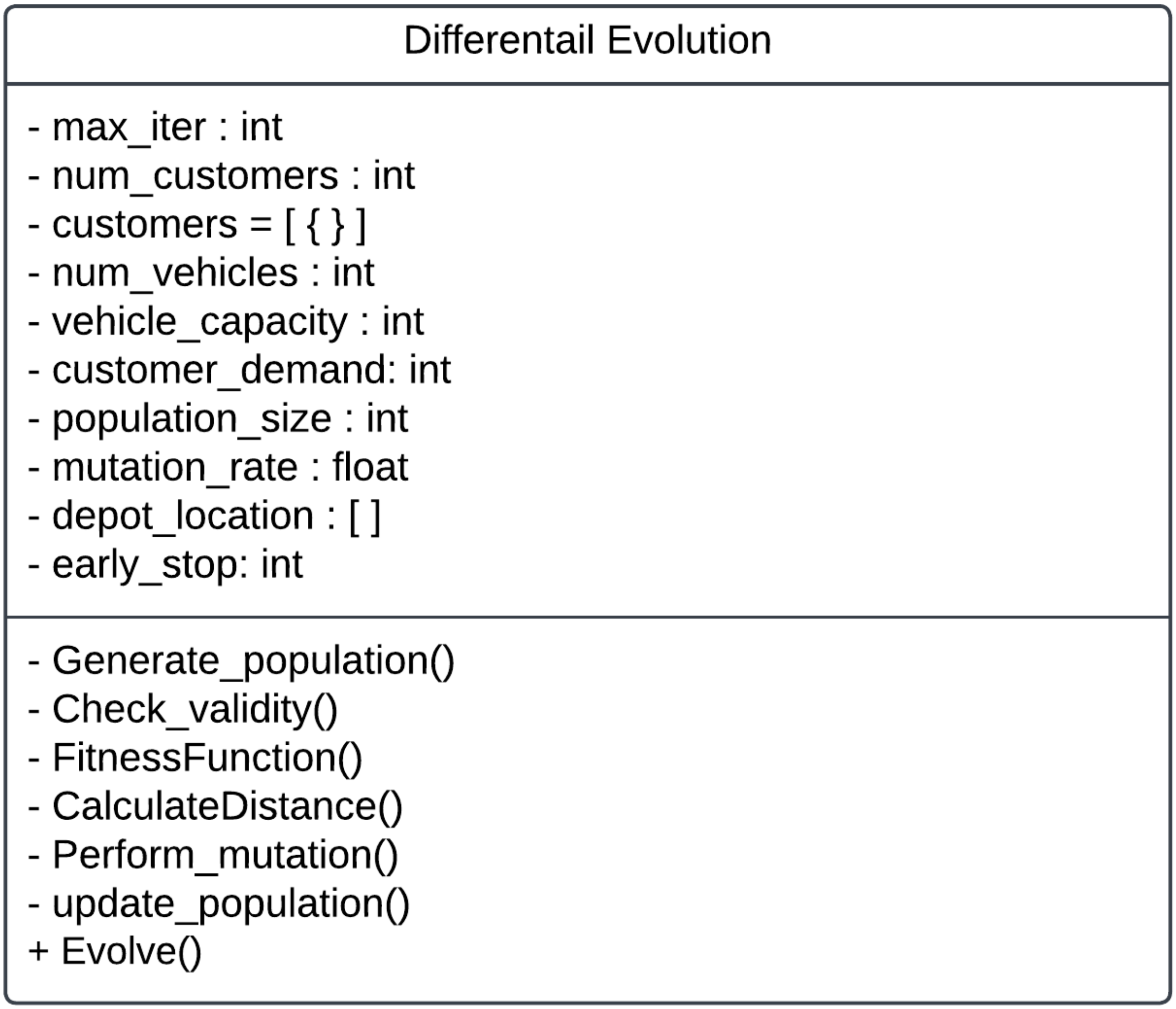
   - If the offspring has a better fitness score (lower total distance) than the parent solution, it replaces the parent in the next generation.
   - If the offspring has a worse fitness score, the parent is retained.

5. Fitness Function:

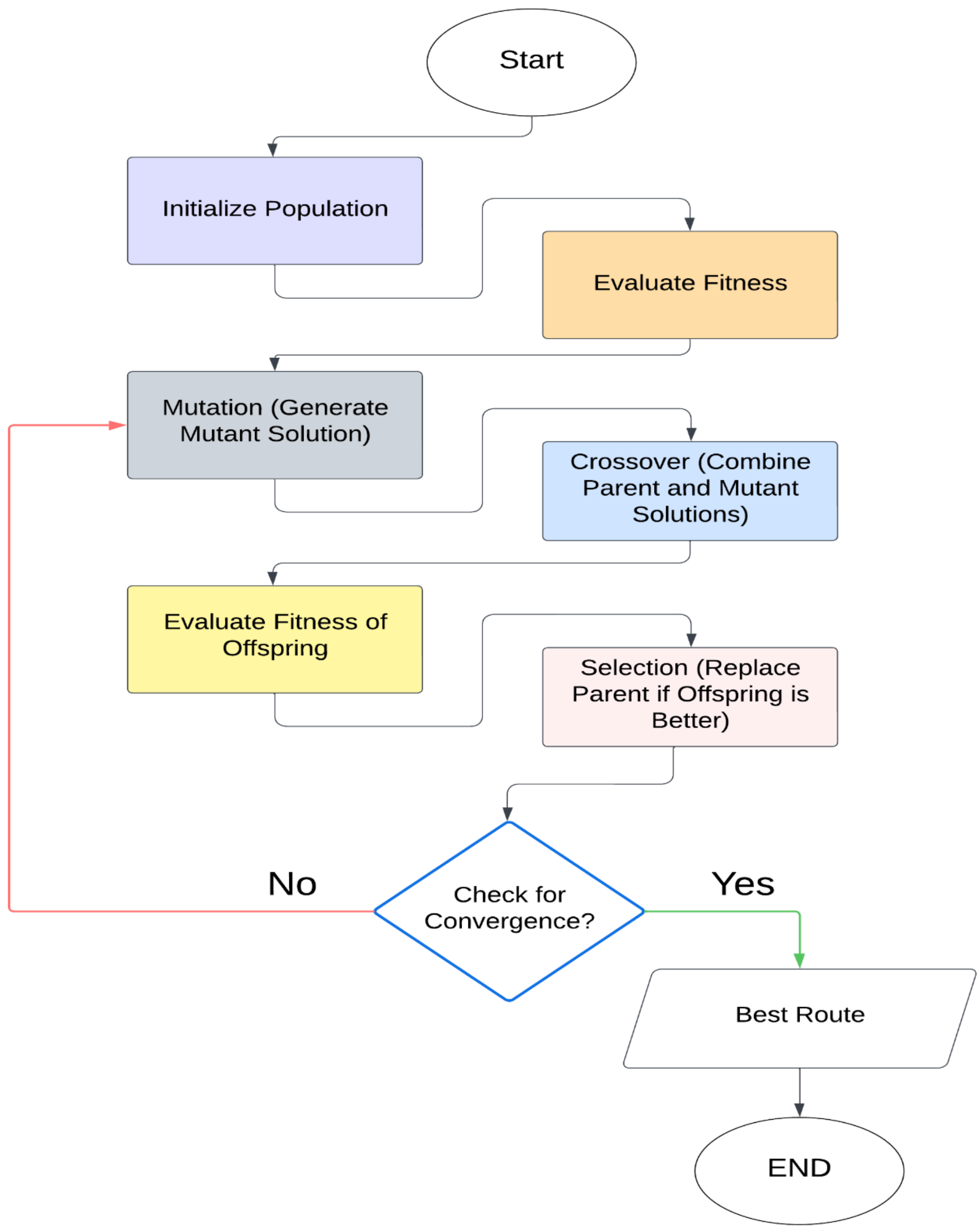
   - Minimize the total delivery cost, measured by the total distance traveled by all vehicles in their respective routes.

6. Evolve:
   - Once the mutation, crossover, and selection steps are performed, the population is updated with the best solutions, and the algorithm evolves. This process is repeated for several generations until convergence is reached or a stopping criterion (maximum number of generations) is met.

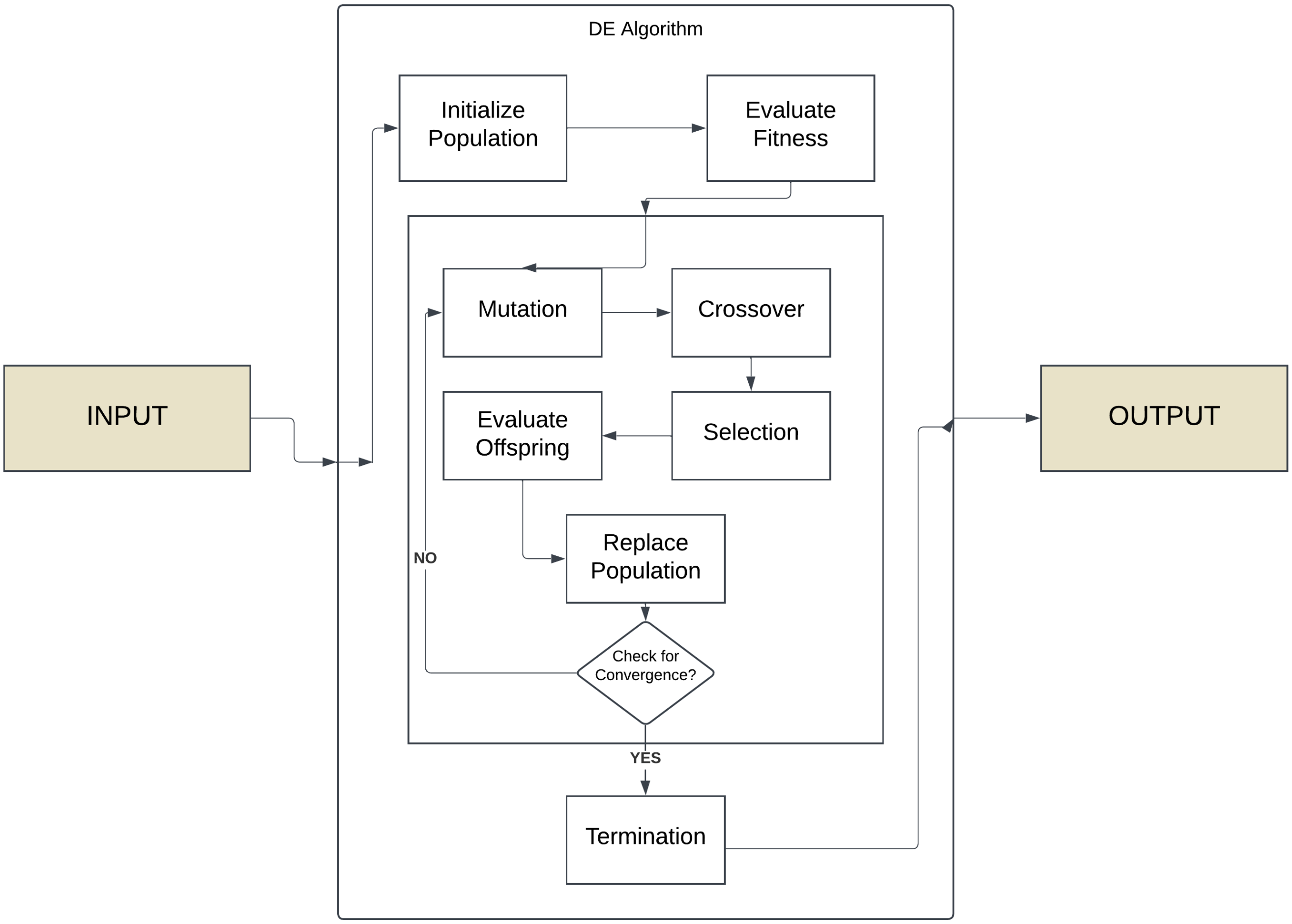
Class Diagram Deferential Evolution:



Flow Chart Deferential Evolution:



Block Diagram Deferential Evolution:



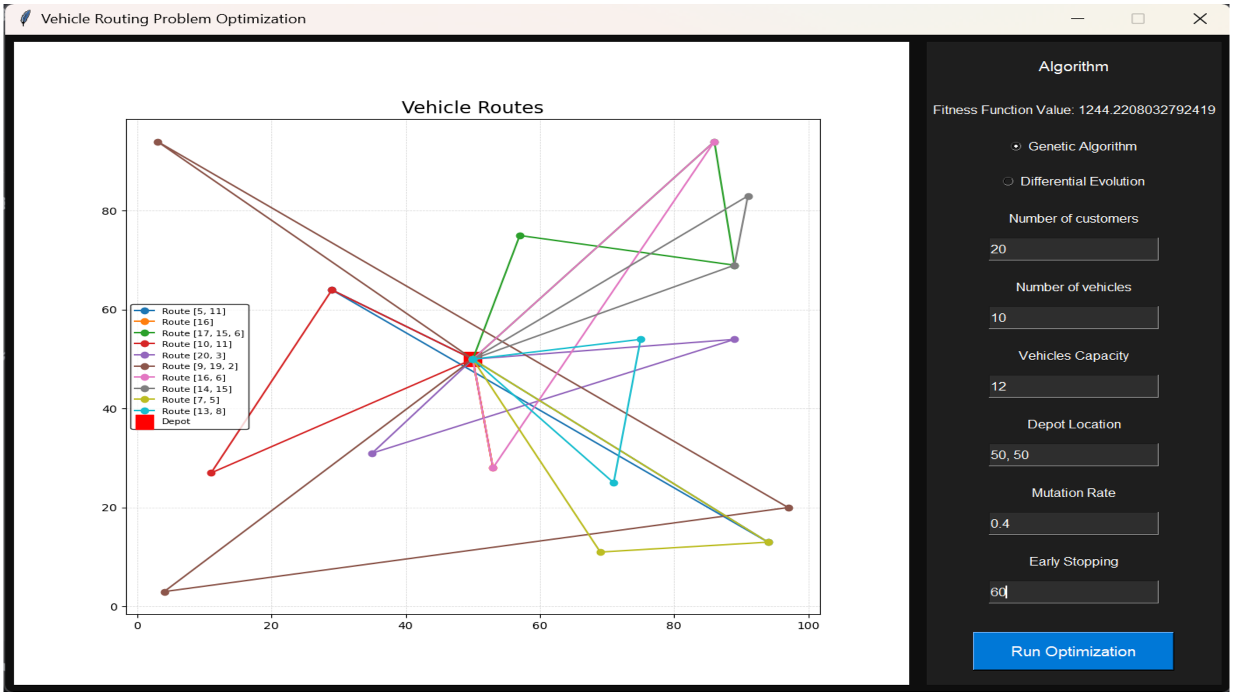
Development Tools:

Programming languages and Libraries:

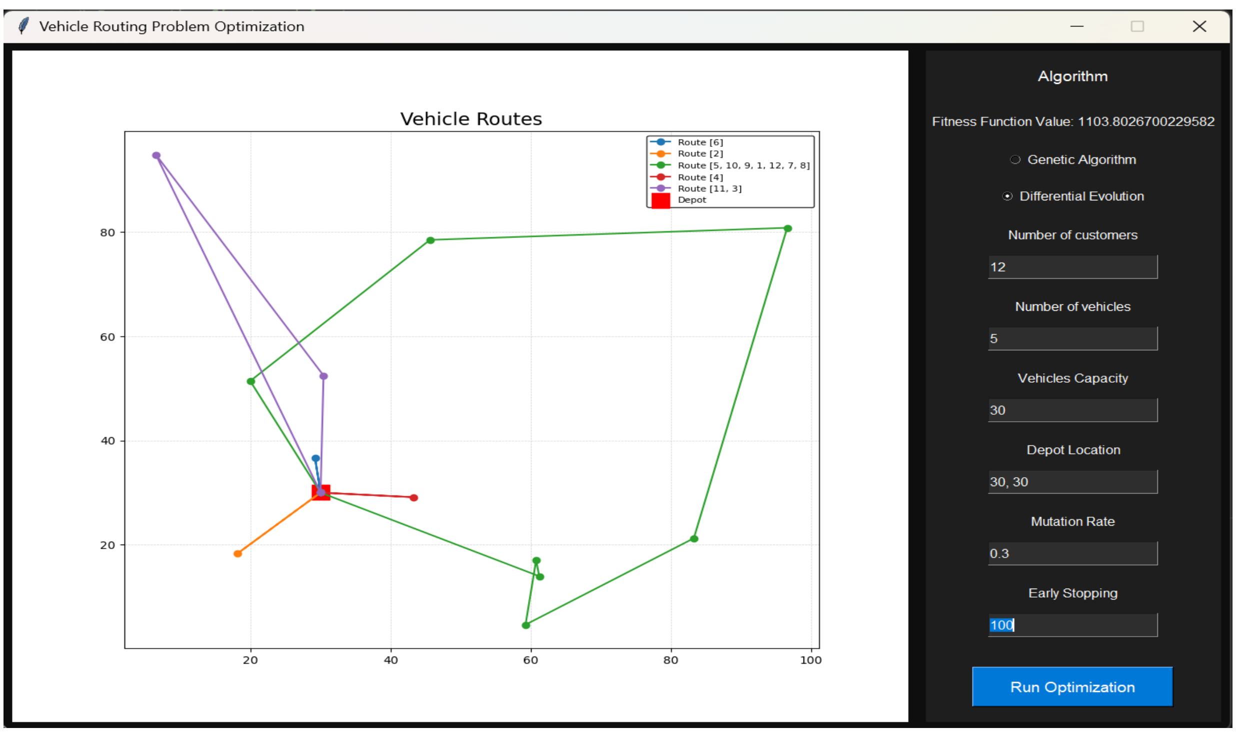
- Python (Classes Approach)
- Numpy – Random
- GUI: (Tkinter - Matplotlib)

Experiments & Results

Plot GA:



Plot DE:



Analysis & Recommendation

Analysis the result:

we observed that the best fitness value, representing the minimum total cost (total distance traveled by vehicles), is achieved after a certain number of iterations. To optimize computational efficiency, an early stopping mechanism was implemented to terminate the algorithm if the fitness value remained unchanged for a predefined number of iterations. This approach prevents unnecessary computations while ensuring convergence to an optimal or near-optimal solution.

Recommendation:

- Introduce a Diversity Function:
  - Implement a function to calculate the diversity of solutions. If the fitness value stagnates (indicating low diversity), dynamically adjust the mutation rate f f in the Differential Evolution algorithm to promote exploration and improve solution variety.

- More Fitness Metrics:
  - include additional constraints to reflect real-world scenarios more accurately.
    - **Vehicle Operating Costs:** Factor in fuel consumption, maintenance, and driver wages.
    - **Traffic Conditions:** Incorporate real-time or predicted traffic patterns into route planning.