Faculty of Engineering and Technology.

Department of Electrical and Computer Engineering.

ENCS3340- Artificial Intelligence

Project #1: Packages-Vehicles Routing

Prepared by: Noura Khdour.

ID:1212072.

Instructor: Dr.Yazen Abu Farha

Section:4

Date:4/5/2025

- **The formulation of the problem.**

1-Genatic Algorithm:

- Initializing Chromosome: A dictionary with list where the key is the vehicle and the value is a list of assigned packages.
- Fitness: Inverse of cost (try to find the min cost)
- Crossover: Copy parents and swap some chromosomes
- Mutation: Randomly swap packages between vehicles or reorder rout.
- Selection: use Tournament selection: Pick best of random subset.

2- Simulated Annealing

- Solution State: Represent the solution as *"self.assignments*"
- Initial Solution: Randomly assign packages to vehicles
- Neighborhood: Generate a new solution by swapping packages between vehicles or reassigning a package or reorder package
- Cost Function: Use *"compute_cost"* to evaluate the solution (total distance + penalties for capacity/priority).
- Acceptance: Accept better solutions; accept worse ones with probability exp(-(new_cost - old_cost)/temp).
- Cooling: Gradually decrease temperature

- **The heuristics that used.**

1-Genatic Algorithm

- Random Initialization with Capacity Preference
- Sort each vehicle's route by package priority (ascending) to ensure high-priority packages are delivered first.

2-Simulated Annealing

- Neighborhood Search
- Greedy Acceptance: (exp(-(new_cost - old_cost)/temp))
- Cooling Schedule: Gradually reduces temperature (e.g., temp *= cooling_rate)

- **How to handled constraints violations**

- Capacity Constraint: Add a penalty to the cost if a vehicle's total package weight exceeds its capacity.
- Priority Constraint: Incorporate package priority (e.g., 1 is highest, 4 is lowest) into the cost, penalizing lower-priority undelivered packages less or rewarding higher-priority deliveries.
- Distance-Based Cost: Base cost on route distances (e.g., Euclidean distance between package coordinates and depot).

- **Effects of parameters tuning**

1-Genatic Algorithm (Mutation Rate :Current: 0.04)

Controls the probability of reassigning packages or reordering routes.it preserves good solutions, refining feasible assignments, but it limits exploration, trapping GA in local optima with violations.

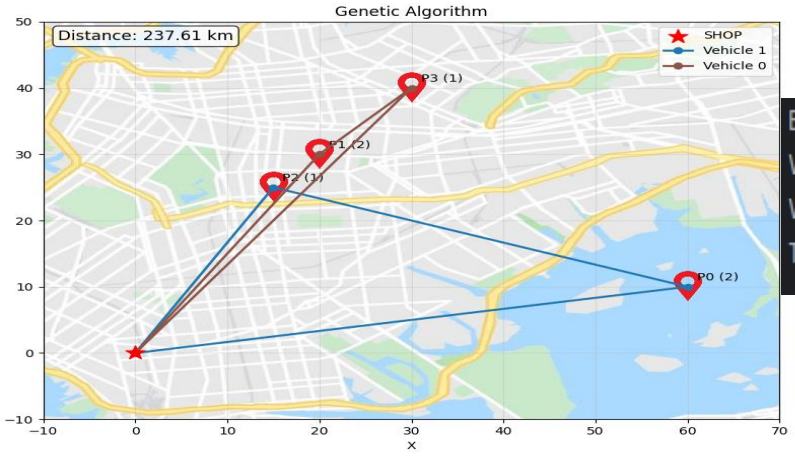
2-Simulated Annealing (Cooling Rate: 0.95)

A smaller cooling rate (e.g., 0.95) slows temperature decrease, allowing more exploration and potentially better solutions but increasing runtime. A larger rate (e.g., 0.99) speeds up convergence but may miss optimal assignments.

- **Test cases**

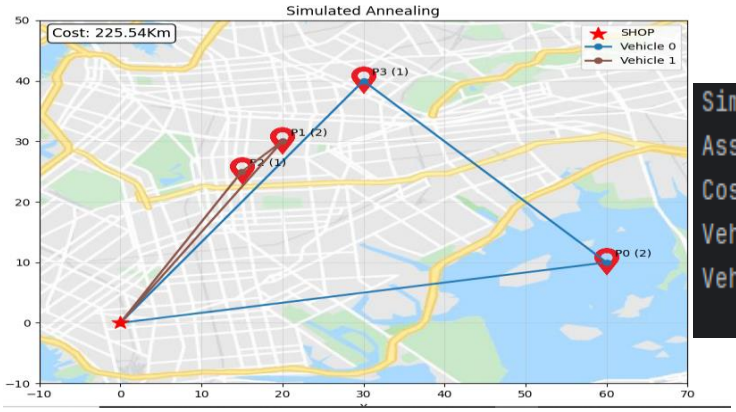
1-Ensure that the system can assign packages to vehicles without exceeding their capacities. 2 vehicles, each with a capacity of 100 kg. Packages: 4 packages with the following weights: 30 kg, 40 kg, 50 kg, and 60 kg.

- **In GA**




```
Best Solution:
Vehicle 1:Route = [2, 0], Weight: 80/100 kg
Vehicle 0:Route = [3, 1], Weight: 100/100 kg
Total Distance: 237.61 km
```

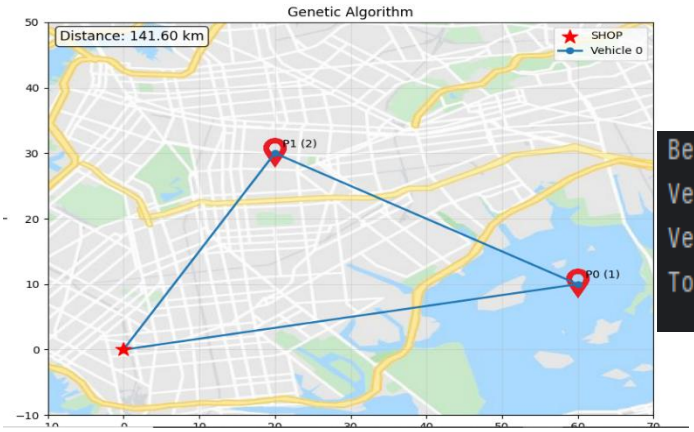
- **In SA**




```
Simulated Annealing Result:
Assignments: {0: [3, 0], 1: [2, 1]}
Cost(Total Distance): 225.54 Km
Vehicle 0: Total Weight = 90 kg / Capacity = 100 kg
Vehicle 1: Total Weight = 90 kg / Capacity = 100 kg
```

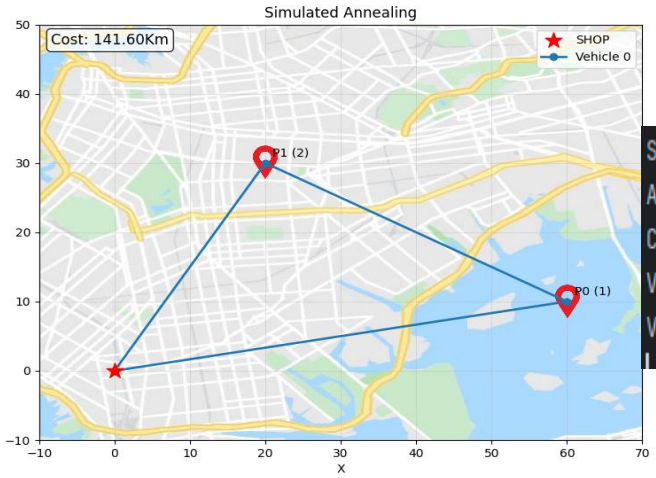
2-Verify that higher-priority packages are delivered before lower-priority ones when possible. Vehicles: 1 vehicle with a capacity of 100 kg.Packages:Package A: 50 kg, Priority 1,Package B: 50 kg, Priority 2,Package C: 50 kg, Priority 3

- **In GA**




```
Best Solution:
Vehicle 0:Route = [0, 1], Weight: 100/100 kg
Vehicle 1:Route = [], Weight: 0/0 kg
Total Distance: 141.60 km
```

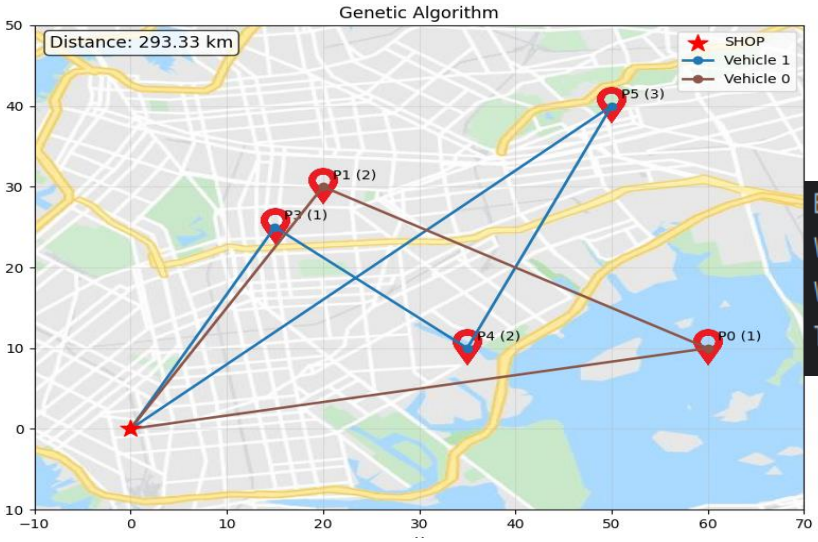
- **In SA**




```
Simulated Annealing Result:
Assignments: {0: [0, 1], 1: []}
Cost(Total Distance): 141.60 Km
Vehicle 0: Total Weight = 100 kg / Capacity = 100 kg
Vehicle 1: Total Weight = 0 kg / Capacity = 0 kg
```

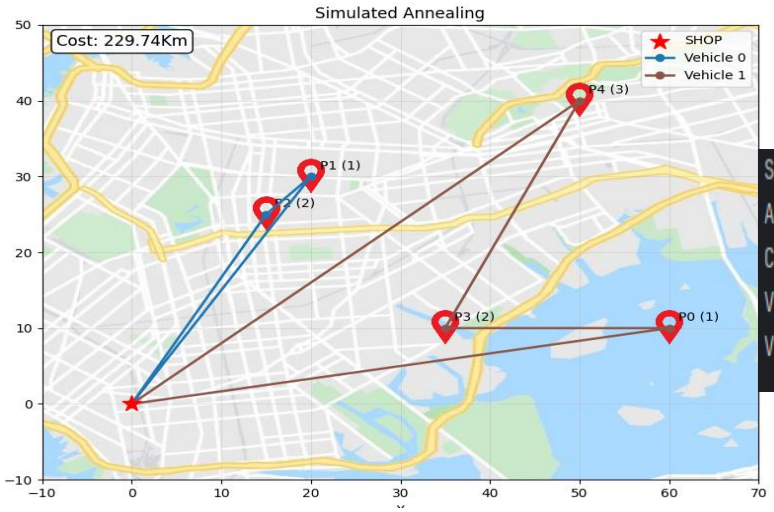
3- Ensure that the system minimizes the total distance travelled by all vehicles. : 2 vehicles, each with a capacity of 100 kg.Packages: 6 packages located at varying distances from the depot.

- **In GA**




```
Best Solution:
Vehicle 1:Route = [3, 4, 5], Weight: 90/100 kg
Vehicle 0:Route = [0, 1], Weight: 60/100 kg
Total Distance: 293.33 km
```

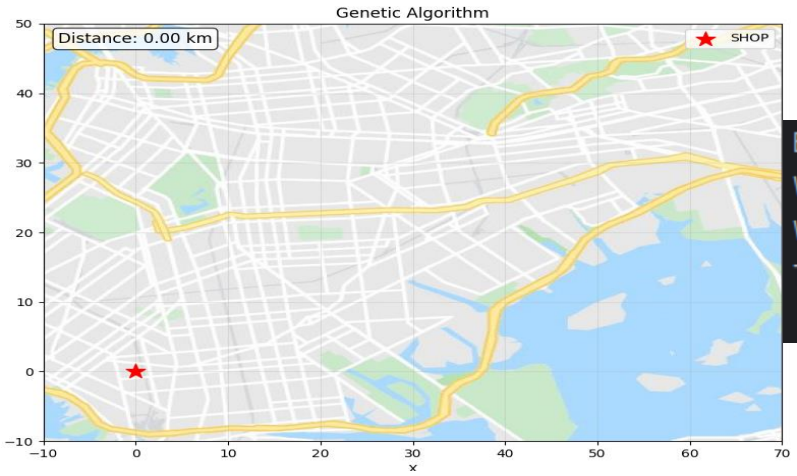
- **In SA**




```
Simulated Annealing Result:
Assignments: {0: [1, 2], 1: [0, 3, 4]}
Cost(Total Distance): 229.74 Km
Vehicle 0: Total Weight = 50 kg / Capacity = 100 kg
Vehicle 1: Total Weight = 100 kg / Capacity = 100 kg
```

4-Test the system's behaviour when a package exceeds the capacity of all available vehicles. Vehicles: 2 vehicles, each with a capacity of 100 kg.Packages: 1 package weighing 150 kg.

- **In GA**



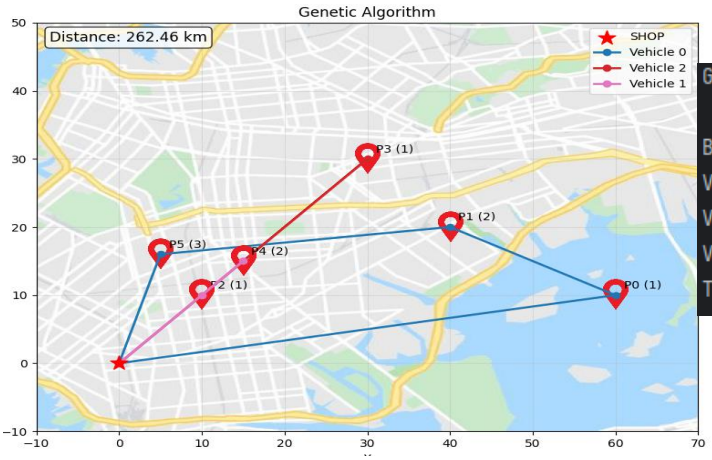

```
Best Solution:
Vehicle 1:Route = [], Weight: 0/100 kg
Vehicle 0:Route = [], Weight: 0/100 kg
Total Distance: 0.00 km
```

- **In SA**


```
Simulated Annealing Result:
Assignments: {0: [], 1: []}
Cost(Total Distance): 0.00 Km
Vehicle 0: Total Weight = 0 kg / Capacity = 100 kg
Vehicle 1: Total Weight = 0 kg / Capacity = 100 kg
```

5-Compare the performance of both algorithms in terms of solution quality and computation time.Vehicles: 3 vehicles, each with a capacity of 100 kg.Packages: 6 packages with varying weights and priorities

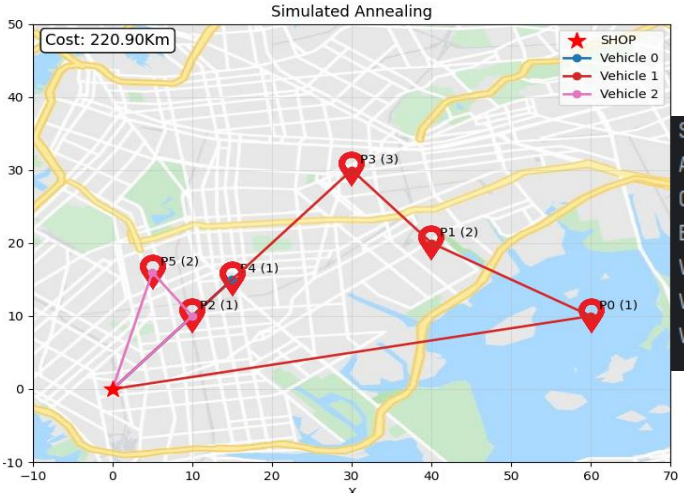
- **In GA**




```
GA Execution Time: 4.01 seconds

Best Solution:
Vehicle 0:Route = [0, 1, 5], Weight: 70/100 kg
Vehicle 2:Route = [3], Weight: 35/100 kg
Vehicle 1:Route = [2, 4], Weight: 75/100 kg
Total Distance: 262.46 km
```

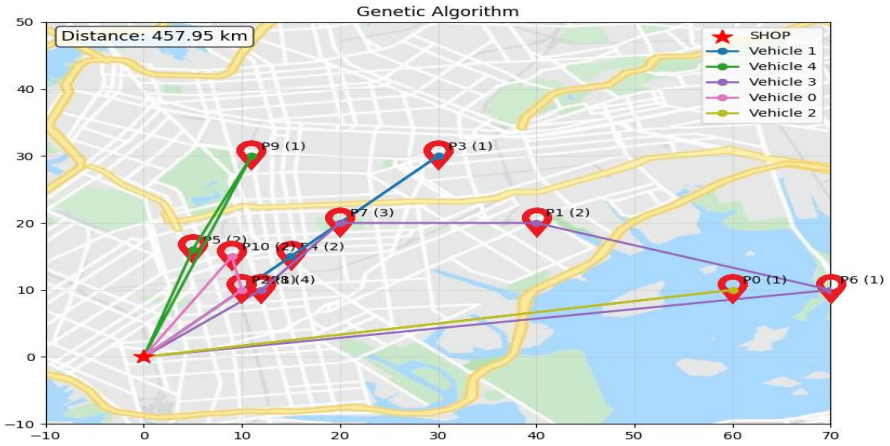
- **In SA**




```
Simulated Annealing Result:
Assignments: {0: [4], 1: [0, 1, 3], 2: [2, 5]}
Cost(Total Distance): 220.90 Km
Execution Time: 0.05 seconds
Vehicle 0: Total Weight = 60 kg / Capacity = 100 kg
Vehicle 1: Total Weight = 65 kg / Capacity = 100 kg
Vehicle 2: Total Weight = 55 kg / Capacity = 100 kg
```

6- Assess the system's performance with a large number of packages and vehicles,Vehicles: 5 vehicles, each with a capacity of 100 kg.Packages: 11 packages with random weights and priorities

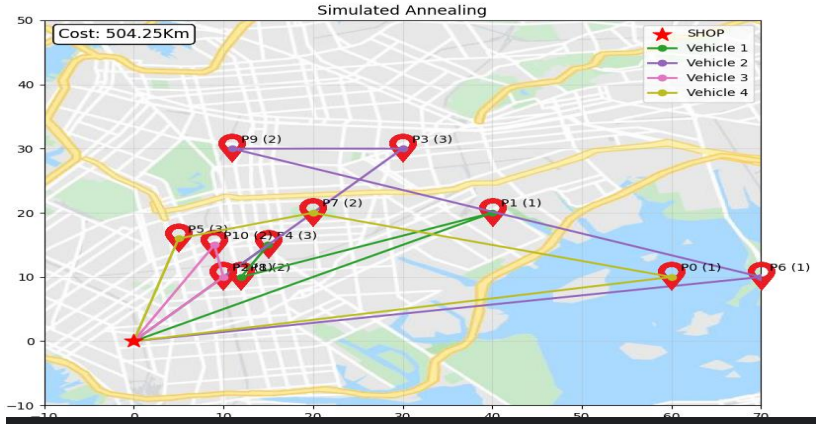
- **In GA**




```
**GA Execution Time: 6.27 seconds

Best Solution:
Vehicle 1:Route = [3, 4], Weight: 95/100 kg
Vehicle 4:Route = [9, 5], Weight: 85/100 kg
Vehicle 3:Route = [6, 1, 7, 8], Weight: 70/100 kg
Vehicle 0:Route = [2, 10], Weight: 70/100 kg
Vehicle 2:Route = [0], Weight: 20/100 kg
Total Distance: 457.95 km
```

- **In SA**




```
Simulated Annealing Result:
Assignments: {0: [], 1: [1, 8, 4], 2: [6, 9, 3], 3: [2, 10], 4: [0, 7, 5]}
Cost(Total Distance): 504.25 Km
Execution Time: 0.09 seconds
Vehicle 0: Total Weight = 0 kg / Capacity = 100 kg
Vehicle 1: Total Weight = 75 kg / Capacity = 100 kg
Vehicle 2: Total Weight = 95 kg / Capacity = 100 kg
Vehicle 3: Total Weight = 70 kg / Capacity = 100 kg
Vehicle 4: Total Weight = 100 kg / Capacity = 100 kg
```