Faculty of Engineering & Technology

Electrical & Computer Engineering Department

ENCS3340, ARTIFICIAL INTELLIGENCE

Report. Optimization Strategies for Local Package Delivery Operations

---

Prepared by:

Roa Makhtoob 1221636 & Rand Saleh 1221124

Section: 1 **Section:** 4

Instructor:

Dr. Yazan Abu Farha

Date: 4th.May.2025

Abstract:

This project focuses on optimizing the operations of a local package delivery shop by solving the package-to-vehicle assignment problem. The objective is to minimize the total travel distance of all delivery vehicles while respecting vehicle capacity constraints and giving preference to high-priority deliveries when feasible. Two local search algorithms—**Simulated Annealing and Genetic Algorithm**—were implemented to tackle this optimization challenge. The entire system was developed using **Python** and relevant libraries, enabling efficient design, execution, and testing. Both algorithms were successfully tested and produced valid, efficient solutions. Simulated Annealing provided more consistent priority handling, while the Genetic Algorithm offered faster exploration and greater solution diversity. The results demonstrate a trade-off between delivery quality and computational efficiency, reflecting practical considerations in real-world logistics.

Contents

Table of Figures:

1. Problem Formulation

A package delivery shop's vehicle routing optimization scenario is the problem formulated in this project. Packages must be delivered daily by a number of vehicles with constrained carrying capacities to various locations denoted by (x, y) coordinates on a 2D plane. Every package has a priority level (1 being the highest, 5 being the lowest) and a specific weight. The goal is to:

- Assign packages to cars while keeping in mind their capacity restrictions.
- Reduce the overall distance that all vehicles travel.
- Delivering higher priority packages earlier in the delivery sequence is preferred, but if prioritization significantly increases distance, flexibility may be allowed.

Two distinct optimization methods were used: Genetic Algorithm (GA) and Simulated Annealing (SA).

2. Heuristics Used

2.A Simulated Annealing (SA):

- Start with a valid random assignment of packages to vehicles.
- Neighbor generation was based on **small local modifications**: moving a package from one vehicle to another, swapping packages between vehicles, or reordering delivery sequences.
- Acceptance of a neighbor depended on either reducing the total distance or improving the delivery priority order. Worse solutions could still be accepted probabilistically, especially at higher temperatures, to allow escaping local minima.

2.B Genetic Algorithm (GA):

To implement the GA algorithm of package delivery shop the following flow chart were used

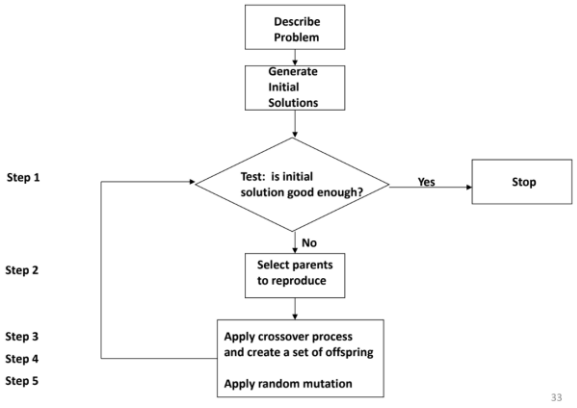

Figure 1 GA Flow chart

❖ Greedy Initialization

Step 1 in the flowchart: "Generate Initial Solutions", a greedy heuristic assigns packages to vehicles in a way that fills each vehicle up to its capacity. This ensures feasible starting individuals in the population, which increases the algorithm's efficiency in later steps.

❖ Fitness Function

In the evaluation stage ("Test: is initial solution good enough?"), each solution is judged based on a fitness function that combines:

- **Minimized total travel distance**,
- **Penalty for unassigned packages** (e.g., due to vehicle capacity limits),
- **Reward for delivering high-priority packages**.

This multi-objective fitness function helps balance logistical efficiency with service quality.

❖ Elitism and Selection

When the current solution isn't good enough (decision point in the flowchart), the algorithm selects the best-performing individuals ("Select parents to reproduce"). Elitism ensures top solutions are always carried into the next generation, improving convergence.

❖ Crossover and Mutation

As shown in Steps 3–5 of the flowchart:

- **Crossover** mixes genes (routes and assignments) from parent solutions to create new offspring.
- **Mutation** introduces random swaps of packages within routes to maintain diversity and escape local optima.

❖ Repair Heuristic for Constraint Handling

Since crossover and mutation may create invalid solutions (e.g., overloaded vehicles, duplicate packages), a **repair function** is applied after mutation. This function:

- Removes any duplicate assignments,
- Reassigns leftover packages (favoring higher priorities),
- Ensures all vehicle capacities are respected.

3. Constraint Handling

Both algorithms enforce the following constraints during solution generation and modification:

- **Capacity Constraint**: No vehicle's load (sum of package weights) may exceed its capacity.
- If a move, swap, or mutation would cause a vehicle to overload, it is rejected or not performed.

In Simulated Annealing, every local move checks vehicle capacities before accepting package movements. In Genetic Algorithm, feasible individuals are created or corrected to satisfy capacity constraints before being allowed into the population.

In Genetic Algorithm solution is first created using a method that assigns packages to vehicles without going over their capacity limits. If any issues come up later—like a vehicle being overloaded or a package assigned more than once (if the cross over and the mutation caused any problem)—a repair step is used to fix them. This step removes duplicates and reassigns leftover packages to vehicles that still have space. This ensures that all solutions stay valid throughout the process.

4. Parameter Tuning

Simulated Annealing:

- **Cooling Rate** was selected as the tunable parameter.
- Values between 0.90 and 0.99 were explored.
- A cooling rate of **0.95** provided a good balance between exploration and convergence speed. Higher rates like 0.99 produced slightly better solutions but took significantly longer to converge.

Genetic Algorithm:

One of the most important parameters in a GA is the **mutation rate**, which decides how often random changes happen in the solutions. the mutation rate is set to **0.1**, meaning there's a **10%** chance of introducing changes in each generation. This will:

- help the algorithm avoid getting stuck too early on one solution.
- keep the search process active by exploring new possibilities over time.
- allow room for improvement without damaging good solutions too much.

Other important parameters include the **population size (30)** and **number of generations (100)**

→ **Larger populations** offer more solution variety but take more time to process.

→ **More generations** improve results but increase runtime.

5. Test Case and Observations

From the following test cases we will notice that Simulated Annealing produced valid routes with lower distances and gave preference to high-priority packages when possible. Genetic Algorithm explored solutions faster and generated more diverse results but sometimes needed repair steps to ensure feasibility. Overall, SA maintained more consistent priority handling, while GA offered broader exploration.

Test Case 1 Configuration:

```
vehicles = [
    {'id': 1, 'capacity': 70, 'assigned_packages': [], 'current_load': 0},
    {'id': 2, 'capacity': 80, 'assigned_packages': [], 'current_load': 0},
    {'id': 3, 'capacity': 60, 'assigned_packages': [], 'current_load': 0}
]
packages = [
    {'id': 1, 'x': 10.0, 'y': 20.0, 'weight': 10.0, 'priority': 2},
    {'id': 2, 'x': 30.0, 'y': 40.0, 'weight': 20.0, 'priority': 1},
    {'id': 3, 'x': 70.0, 'y': 80.0, 'weight': 15.0, 'priority': 3},
    {'id': 4, 'x': 15.0, 'y': 60.0, 'weight': 25.0, 'priority': 2},
    {'id': 5, 'x': 50.0, 'y': 10.0, 'weight': 5.0, 'priority': 4},
    {'id': 6, 'x': 60.0, 'y': 30.0, 'weight': 20.0, 'priority': 2},
    {'id': 7, 'x': 5.0, 'y': 90.0, 'weight': 10.0, 'priority': 5},
    {'id': 8, 'x': 85.0, 'y': 25.0, 'weight': 12.0, 'priority': 3},
    {'id': 9, 'x': 25.0, 'y': 5.0, 'weight': 8.0, 'priority': 1},
    {'id': 10, 'x': 95.0, 'y': 90.0, 'weight': 18.0, 'priority': 2}
]
```

Figure 2 Test Case 1

Using simulated Annealing first solution → 1st run

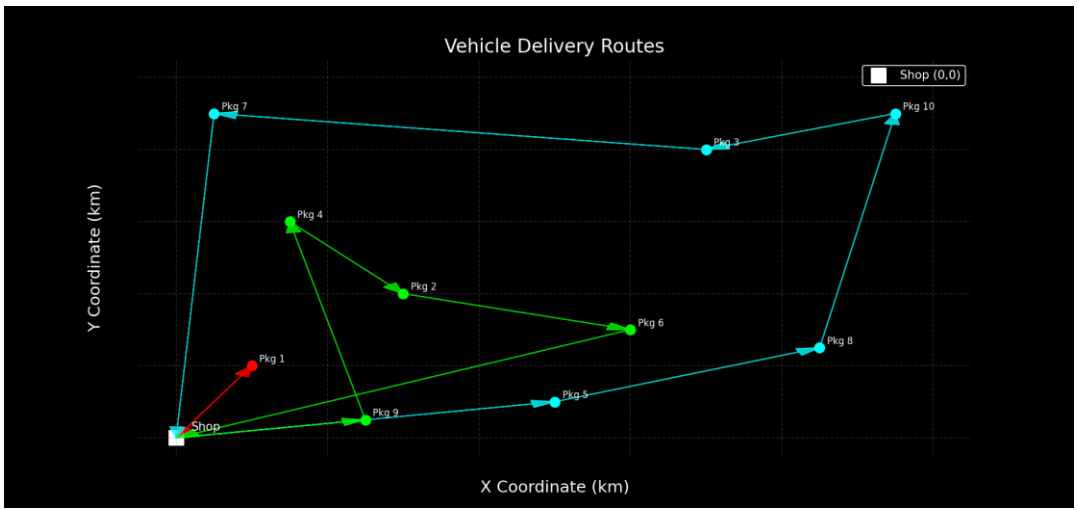

Figure 3 simulated annealing output1

```
Initial total traveled distance: 835.87 km
Initial total priority score: 63

Initial Assignment:
Vehicle 1 (Load: 18.0 / 70 kg)
  Package 10 at (95.0,90.0) weight=18.0 priority=2

Vehicle 3 (Load: 55.0 / 60 kg)
  Package 5 at (50.0,10.0) weight=5.0 priority=4
  Package 6 at (60.0,30.0) weight=20.0 priority=2
  Package 7 at (5.0,90.0) weight=10.0 priority=5
  Package 8 at (85.0,25.0) weight=12.0 priority=3
  Package 9 at (25.0,5.0) weight=8.0 priority=1

Vehicle 2 (Load: 70.0 / 80 kg)
  Package 1 at (10.0,20.0) weight=10.0 priority=2
  Package 2 at (30.0,40.0) weight=20.0 priority=1
  Package 3 at (70.0,80.0) weight=15.0 priority=3
  Package 4 at (15.0,60.0) weight=25.0 priority=2
```

Figure 4 simulated Annealing output

```
Best total traveled distance: 587.49 km
Best total priority score: 71

Final Best Solution After Simulated Annealing:
Vehicle 1 (Load: 10.0 / 70 kg)
  Package 1 at (10.0,20.0) weight=10.0 priority=2

Vehicle 3 (Load: 60.0 / 60 kg)
  Package 5 at (50.0,10.0) weight=5.0 priority=4
  Package 8 at (85.0,25.0) weight=12.0 priority=3
  Package 10 at (95.0,90.0) weight=18.0 priority=2
  Package 3 at (70.0,80.0) weight=15.0 priority=3
  Package 7 at (5.0,90.0) weight=10.0 priority=5

Vehicle 2 (Load: 73.0 / 80 kg)
  Package 9 at (25.0,5.0) weight=8.0 priority=1
  Package 4 at (15.0,60.0) weight=25.0 priority=2
  Package 2 at (30.0,40.0) weight=20.0 priority=1
  Package 6 at (60.0,30.0) weight=20.0 priority=2
```

Figure 5Figure 4 simulated Annealing output

2nd sol

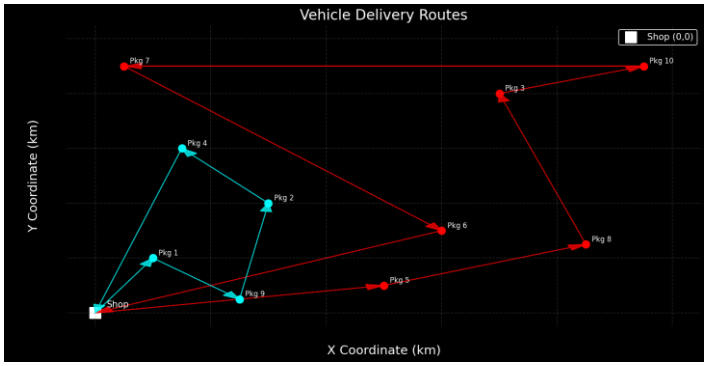

Figure 6 2nd sol graph

```
Initial total traveled distance: 865.13 km
Initial total priority score: 66

Initial Assignment:
Vehicle 3 (Load: 45.0 / 60 kg)
  Package 4 at (15.0,60.0) weight=25.0 priority=2
  Package 6 at (60.0,30.0) weight=20.0 priority=2

Vehicle 2 (Load: 31.0 / 80 kg)
  Package 5 at (50.0,10.0) weight=5.0 priority=4
  Package 9 at (25.0,5.0) weight=8.0 priority=1
  Package 10 at (95.0,90.0) weight=18.0 priority=2

Vehicle 1 (Load: 67.0 / 70 kg)
  Package 1 at (10.0,20.0) weight=10.0 priority=2
  Package 2 at (30.0,40.0) weight=20.0 priority=1
  Package 3 at (70.0,80.0) weight=15.0 priority=3
  Package 7 at (5.0,90.0) weight=10.0 priority=5
  Package 8 at (85.0,25.0) weight=12.0 priority=3
```

Figure 7 2nd sol output

```
Best total traveled distance: 577.26 km
Best total priority score: 79

Final Best Solution After Simulated Annealing:
Vehicle 3 (Load: 0.0 / 60 kg)

Vehicle 2 (Load: 80.0 / 80 kg)
  Package 5 at (50.0,10.0) weight=5.0 priority=4
  Package 8 at (85.0,25.0) weight=12.0 priority=3
  Package 3 at (70.0,80.0) weight=15.0 priority=3
  Package 10 at (95.0,90.0) weight=18.0 priority=2
  Package 7 at (5.0,90.0) weight=10.0 priority=5
  Package 6 at (60.0,30.0) weight=20.0 priority=2

Vehicle 1 (Load: 63.0 / 70 kg)
  Package 1 at (10.0,20.0) weight=10.0 priority=2
  Package 9 at (25.0,5.0) weight=8.0 priority=1
  Package 2 at (30.0,40.0) weight=20.0 priority=1
  Package 4 at (15.0,60.0) weight=25.0 priority=2
```

Figure 8 2nd sol output

Using Genetic Algorithm:

1st sol→1st Run

Figure 9 1st test case using genetic algorithm (plot) 1st run

```
Which algorithm do you want to run?
1 - Simulated Annealing
2 - Genetic Algorithm
Enter your choice (1 or 2): 2
Vehicle 1 (Load: 65.0 / 70 kg)
  Package 8 at (85.0,25.0) weight=12.0 priority=3
  Package 1 at (10.0,20.0) weight=10.0 priority=2
  Package 7 at (5.0,90.0) weight=10.0 priority=5
  Package 9 at (25.0,5.0) weight=8.0 priority=1
  Package 5 at (50.0,10.0) weight=5.0 priority=4
  Package 2 at (30.0,40.0) weight=20.0 priority=1

Vehicle 2 (Load: 78.0 / 80 kg)
  Package 4 at (15.0,60.0) weight=25.0 priority=2
  Package 3 at (70.0,80.0) weight=15.0 priority=3
  Package 10 at (95.0,90.0) weight=18.0 priority=2
  Package 6 at (60.0,30.0) weight=20.0 priority=2

Vehicle 3 (Load: 0 / 60 kg)

Total Traveled Distance: 716.66
Total Priority Score: 25
Plot saved as 'vehicle_routes.png' successfully
```

Figure 10 genetic algorithm output 1st run

2nd sol → 2nd Run

```
/AiProject/main.py
Which algorithm do you want to run?
1 - Simulated Annealing
2 - Genetic Algorithm
Enter your choice (1 or 2): 2
Vehicle 1 (Load: 63.0 / 70 kg)
  Package 1 at (10.0,20.0) weight=10.0 priority=2
  Package 2 at (30.0,40.0) weight=20.0 priority=1
  Package 5 at (50.0,10.0) weight=5.0 priority=4
  Package 6 at (60.0,30.0) weight=20.0 priority=2
  Package 9 at (25.0,5.0) weight=8.0 priority=1

Vehicle 2 (Load: 80.0 / 80 kg)
  Package 8 at (85.0,25.0) weight=12.0 priority=3
  Package 3 at (70.0,80.0) weight=15.0 priority=3
  Package 4 at (15.0,60.0) weight=25.0 priority=2
  Package 7 at (5.0,90.0) weight=10.0 priority=5
  Package 10 at (95.0,90.0) weight=18.0 priority=2

Vehicle 3 (Load: 0 / 60 kg)

Total Traveled Distance: 634.19
Total Priority Score: 25
Plot saved as 'vehicle_routes.png' successfully
```

Figure 11 genetic algorithm output 2nd run

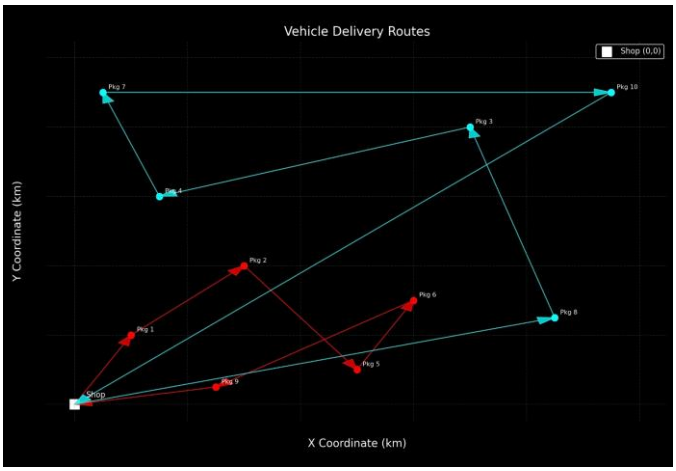

Figure 12 1st test case using genetic algorithm (plot) 2nd run

Test Case 2 Configuration:

```python
#test case 2
vehicles = [
    {'id': 1, 'capacity': 60, 'assigned_packages': [], 'current_load': 0},
    {'id': 2, 'capacity': 75, 'assigned_packages': [], 'current_load': 0}
]
packages = [
    {'id': 1, 'x': 5.0, 'y': 10.0, 'weight': 10.0, 'priority': 1},
    {'id': 2, 'x': 20.0, 'y': 25.0, 'weight': 15.0, 'priority': 2},
    {'id': 3, 'x': 35.0, 'y': 15.0, 'weight': 5.0, 'priority': 3},
    {'id': 4, 'x': 40.0, 'y': 30.0, 'weight': 12.0, 'priority': 1},
    {'id': 5, 'x': 55.0, 'y': 5.0, 'weight': 8.0, 'priority': 4},
    {'id': 6, 'x': 60.0, 'y': 40.0, 'weight': 18.0, 'priority': 2},
    {'id': 7, 'x': 10.0, 'y': 50.0, 'weight': 9.0, 'priority': 5}
]
```

Figure 13 Test Case 2

using simulated Annealing: 1st Run

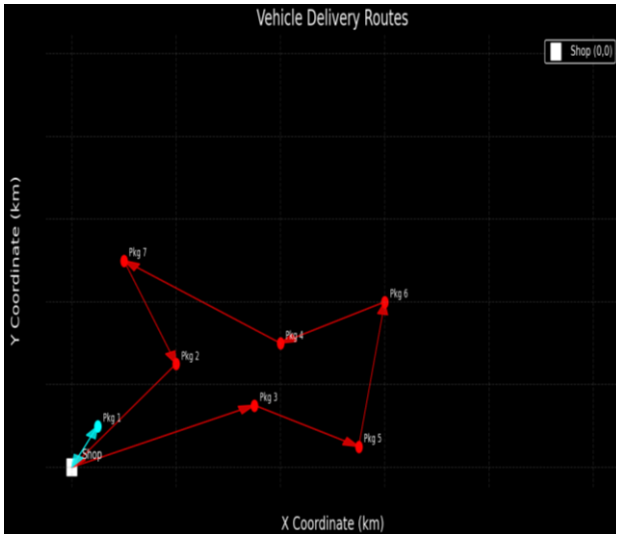

Figure 14 2nd Test Case output plot

```
Initial total traveled distance: 342.34 km
Initial total priority score: 56

Initial Assignment:
Vehicle 2 (Load: 54.0 / 75 kg)
  Package 1 at (5.0,10.0) weight=10.0 priority=1
  Package 2 at (20.0,25.0) weight=15.0 priority=2
  Package 4 at (40.0,30.0) weight=12.0 priority=1
  Package 5 at (55.0,5.0) weight=8.0 priority=4
  Package 7 at (10.0,50.0) weight=9.0 priority=5

Vehicle 1 (Load: 23.0 / 60 kg)
  Package 3 at (35.0,15.0) weight=5.0 priority=3
  Package 6 at (60.0,40.0) weight=18.0 priority=2
```

Figure 152nd run output SA

```
Best total traveled distance: 235.51 km
Best total priority score: 59

Final Best Solution After Simulated Annealing:
Vehicle 2 (Load: 67.0 / 75 kg)
  Package 3 at (35.0,15.0) weight=5.0 priority=3
  Package 5 at (55.0,5.0) weight=8.0 priority=4
  Package 6 at (60.0,40.0) weight=18.0 priority=2
  Package 4 at (40.0,30.0) weight=12.0 priority=1
  Package 7 at (10.0,50.0) weight=9.0 priority=5
  Package 2 at (20.0,25.0) weight=15.0 priority=2

Vehicle 1 (Load: 10.0 / 60 kg)
  Package 1 at (5.0,10.0) weight=10.0 priority=1
```

Figure 16 Figure 172nd run output SA

*Using GA: 1st Run

```
Enter your choice (1 or 2): 2
Vehicle 1 (Load: 58.0 / 60 kg)
  Package 4 at (40.0,30.0) weight=12.0 priority=1
  Package 3 at (35.0,15.0) weight=5.0 priority=3
  Package 2 at (20.0,25.0) weight=15.0 priority=2
  Package 5 at (55.0,5.0) weight=8.0 priority=4
  Package 6 at (60.0,40.0) weight=18.0 priority=2

Vehicle 2 (Load: 19.0 / 75 kg)
  Package 7 at (10.0,50.0) weight=9.0 priority=5
  Package 1 at (5.0,10.0) weight=10.0 priority=1

Total Traveled Distance: 334.10
Total Priority Score: 18
Plot saved as 'vehicle_routes.png' successfully
```

Figure 18

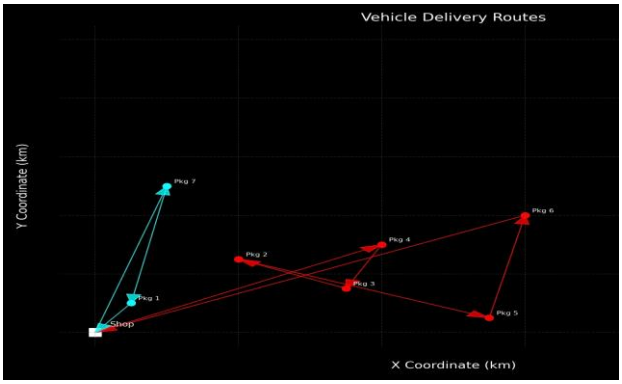

Figure 20

*Simulated Annealing 2nd Sol

```
Initial total traveled distance: 325.69 km
Initial total priority score: 49

Initial Assignment:
Vehicle 2 (Load: 32.0 / 75 kg)
  Package 1 at (5.0,10.0) weight=10.0 priority=1
  Package 3 at (35.0,15.0) weight=5.0 priority=3
  Package 5 at (55.0,5.0) weight=8.0 priority=4
  Package 7 at (10.0,50.0) weight=9.0 priority=5

Vehicle 1 (Load: 45.0 / 60 kg)
  Package 2 at (20.0,25.0) weight=15.0 priority=2
  Package 4 at (40.0,30.0) weight=12.0 priority=1
  Package 6 at (60.0,40.0) weight=18.0 priority=2
```

Figure 24 2nd sol (SA)

```
Best total traveled distance: 244.25 km
Best total priority score: 60

Final Best Solution After Simulated Annealing:
Vehicle 2 (Load: 67.0 / 75 kg)
  Package 3 at (35.0,15.0) weight=5.0 priority=3
  Package 5 at (55.0,5.0) weight=8.0 priority=4
  Package 4 at (40.0,30.0) weight=12.0 priority=1
  Package 6 at (60.0,40.0) weight=18.0 priority=2
  Package 7 at (10.0,50.0) weight=9.0 priority=5
  Package 2 at (20.0,25.0) weight=15.0 priority=2

Vehicle 1 (Load: 10.0 / 60 kg)
  Package 1 at (5.0,10.0) weight=10.0 priority=1
```

Figure 23 2nd sol SA output

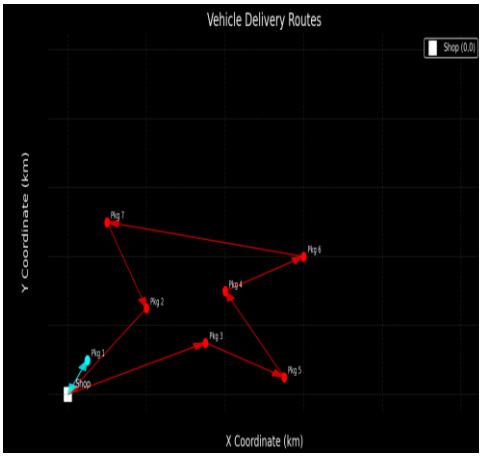

Figure 21SA output plot

GA 2nd Sol:

```
Enter your choice (1 or 2): 2
Vehicle 1 (Load: 58.0 / 60 kg)
  Package 2 at (20.0,25.0) weight=15.0 priority=2
  Package 3 at (35.0,15.0) weight=5.0 priority=3
  Package 4 at (40.0,30.0) weight=12.0 priority=1
  Package 5 at (55.0,5.0) weight=8.0 priority=4
  Package 6 at (60.0,40.0) weight=18.0 priority=2

Vehicle 2 (Load: 19.0 / 75 kg)
  Package 1 at (5.0,10.0) weight=10.0 priority=1
  Package 7 at (10.0,50.0) weight=9.0 priority=5

Total Traveled Distance: 304.96
Total Priority Score: 18
Plot saved as 'vehicle_routes.png' successfully
```

Figure 26 GA 2nd Sol output

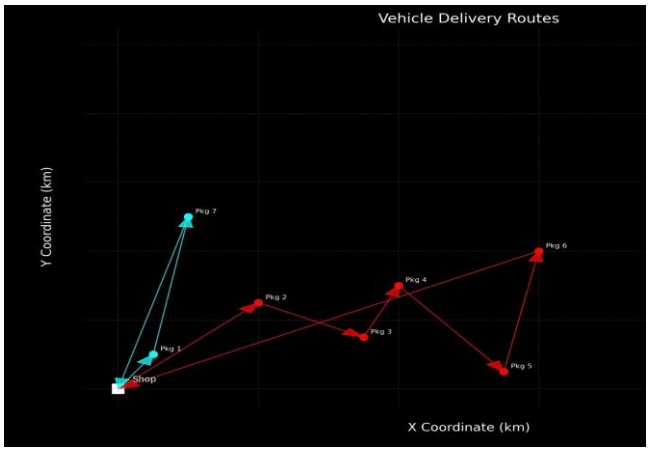

Figure 25 GA 2nd Sol Plot