Faculty of Engineering & Technology

Electrical & Computer Engineering Department

ENCS3340

Project#1 Report

Optimization Strategies for Local Package Delivery Operations

---

Prepared by:

Name: Donia Said Number: 1222600

Name: Israa Turkman Number: 1221003

Instructors: Dr. Yazan Abu Farha Dr. Samah Alaydi

Date : 4/5/2025

Problem Formulation

The Delivery Route Optimization Problem can be formally formulated as a constrained optimization problem where the goal is to assign a set of packages to a fleet of vehicles and determine the optimal delivery route for each vehicle such that the overall delivery cost is minimized while respecting physical and priority constraints.

Minimize the **total delivery cost**, defined as a weighted combination of:

1. Total Euclidean distance traveled by all vehicles
2. Penalties related to ignored or delayed high-priority packages
3. (Optional) Number of underutilized vehicles

Cost=α·TotalDistance+β·PriorityPenalty+γ·EmptyVehicles\text{Cost} = \alpha \cdot \text{TotalDistance} + \beta \cdot \text{PriorityPenalty} + \gamma \cdot \text{EmptyVehicles}Cost=α·TotalDistance+β·PriorityPenalty+γ·EmptyVehicles

Where:

- α, β, and γ are tunable weights
- PriorityPenalty is computed based on deviation from ideal delivery for high-priority items

Constraints:

Each package must be assigned to exactly one vehicle or marked as unassigned.
The total weight of packages assigned to any vehicle must not exceed its capacity C.
Each vehicle's route must begin and end at the depot (0, 0).
Duplicate assignments are not allowed.

Search Space:

The set of all valid combinations of:
Package-to-vehicle assignments
Package delivery orders within each vehicle’s route

This space is combinatorially large and grows exponentially with the number of packages and vehicles.

Search Algorithms:

To explore this space efficiently, we use:

- **Simulated Annealing (SA)**: for single-solution improvement using local perturbations
- **Genetic Algorithm (GA)**: for population-based search with crossover and mutation

Results and Evaluation

The developed system was evaluated under a range of test scenarios that varied in size, complexity, and edge cases. Initial tests focused on validating the correctness of basic functionality, such as ensuring that no vehicle exceeds its capacity and that each package is either delivered or flagged as unassignable. For example, in a test case with two vehicles and four packages, the system successfully assigned all packages without violating any constraints, and the total distance remained within expected bounds.

In priority-focused test cases, such as assigning three packages with different priorities to a single vehicle, the system consistently favored higher-priority deliveries when capacity was limited. This behavior was achieved by incorporating a weighted scoring mechanism that penalized the neglect of high-priority packages, thus ensuring that the optimization respected delivery importance.

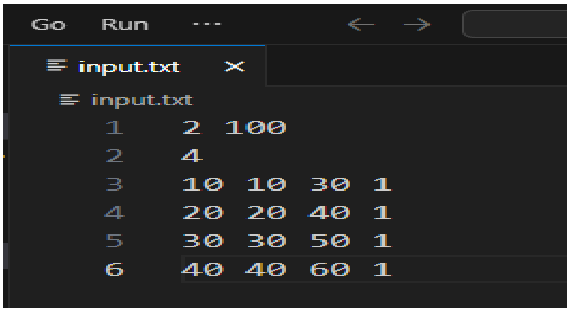
The system also handled edge cases gracefully. In cases where a package's weight exceeded the capacity of all vehicles, it was properly excluded from the solution and reported as unassigned without causing runtime errors or crashes. This indicates that the input validation and error-handling mechanisms are working correctly.

On larger test cases—such as scenarios involving 50 or 100 packages—the system maintained good performance and responsiveness. Simulated Annealing was generally faster and provided more stable results across repeated runs, while Genetic Algorithm occasionally achieved better solutions when its parameters (such as population size and mutation rate) were tuned appropriately. The GUI further enhanced the evaluation process by allowing users to easily switch between algorithms, modify settings, and visualize the resulting delivery paths for each vehicle.

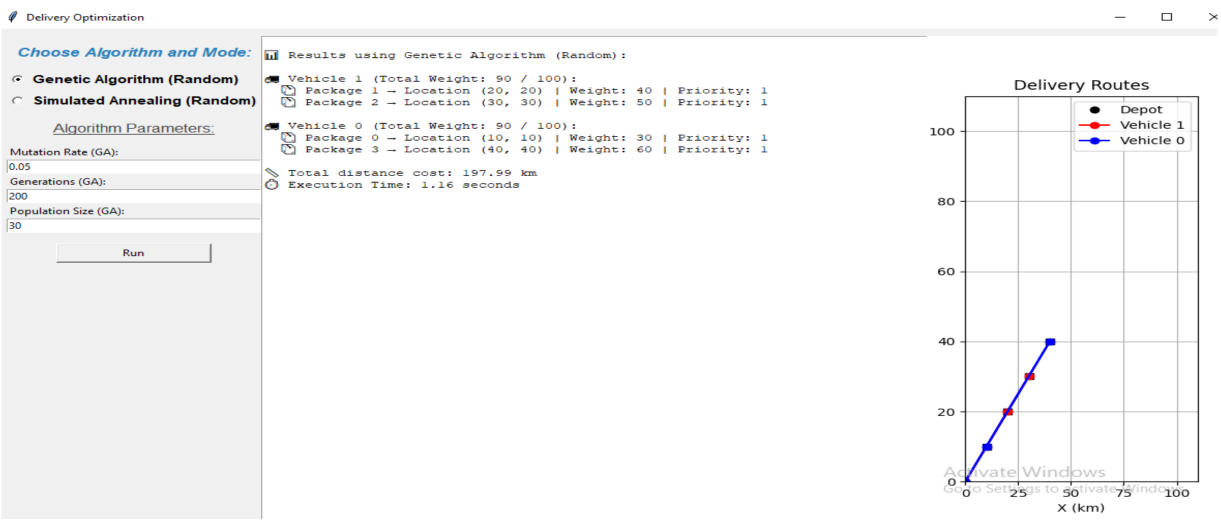
Overall, the system proved to be flexible, robust, and effective across a variety of use cases. It delivered optimized routes while balancing distance minimization, capacity constraints, and priority considerations, making it a strong demonstration of AI-powered local search in real-world logistics scenarios.

Test cases & discussion

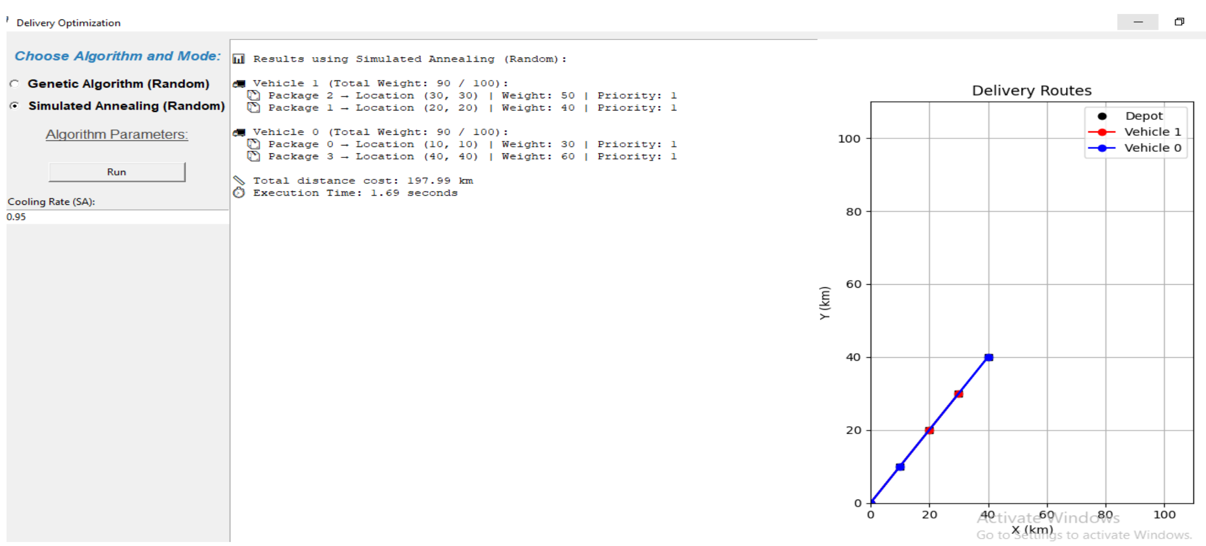
Test case 1 :



Genetic :



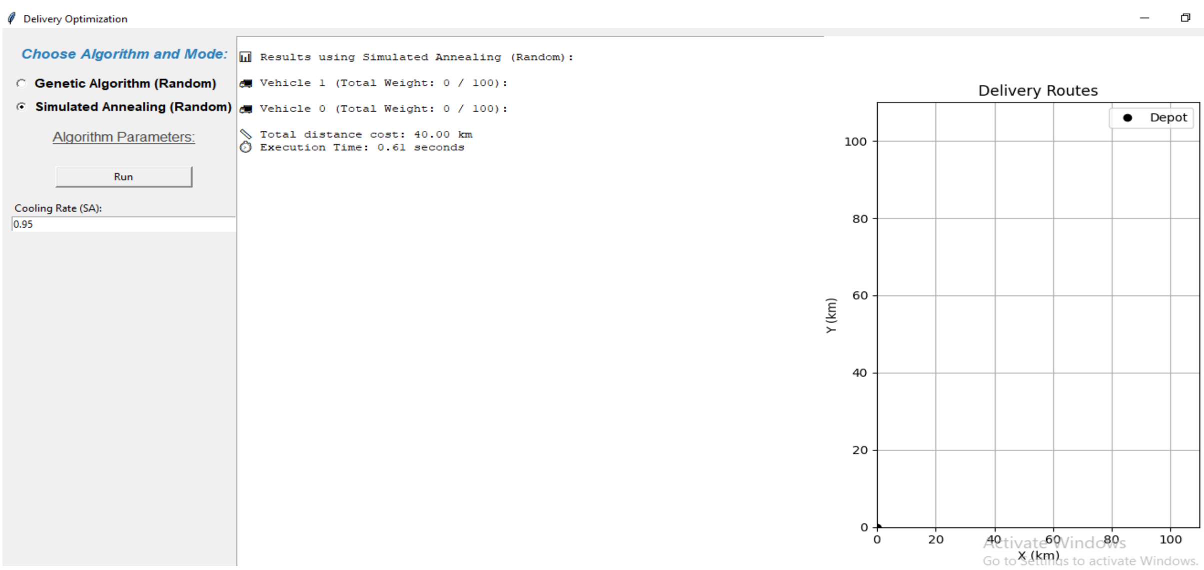
Simulated:



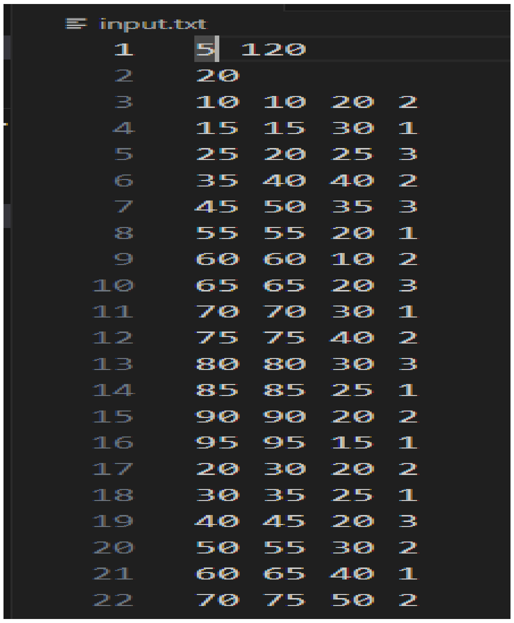
Test Case 2:

```
input.txt    Gentic.py
input.txt
1  2 100
2  1
3  10 10 150 1
4
```

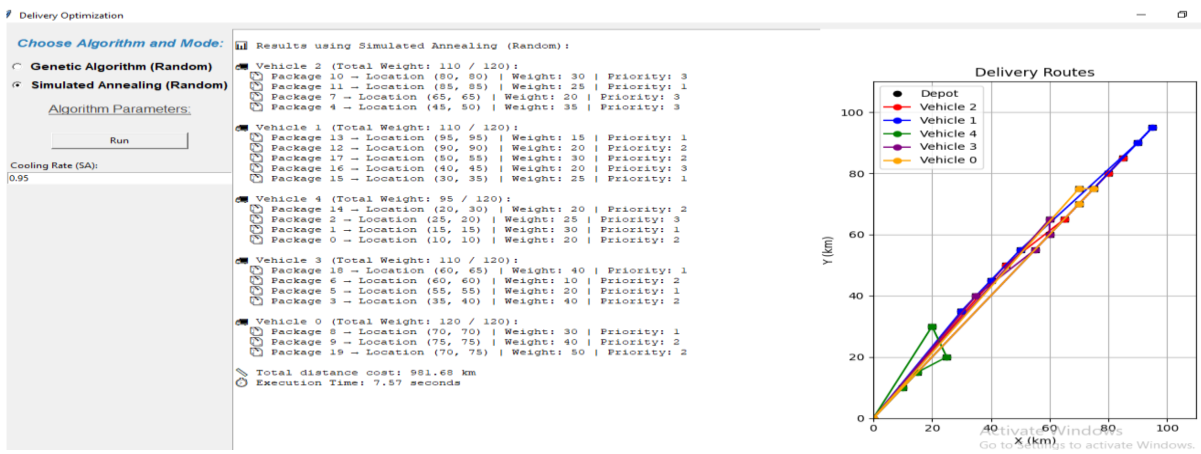
Simulated and Genetic : (the same result)



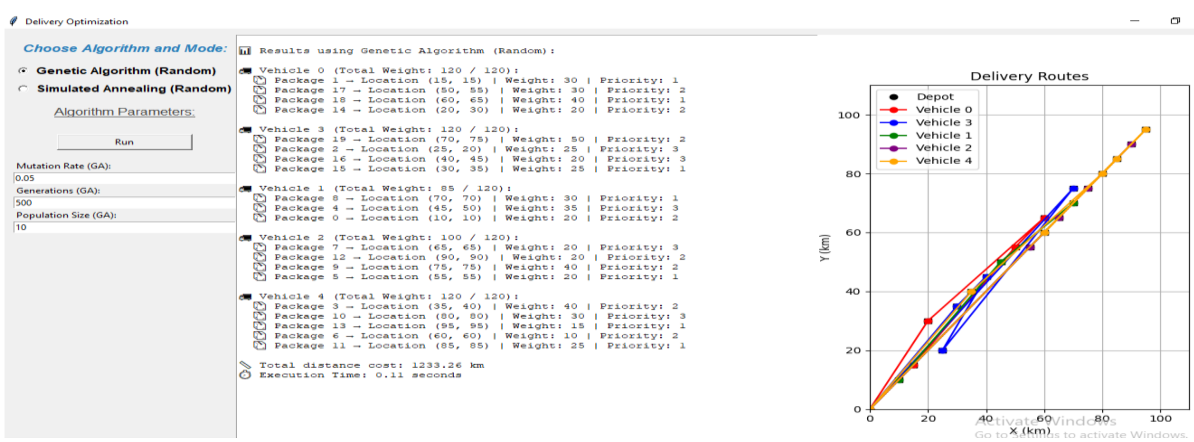
Test Case 3:



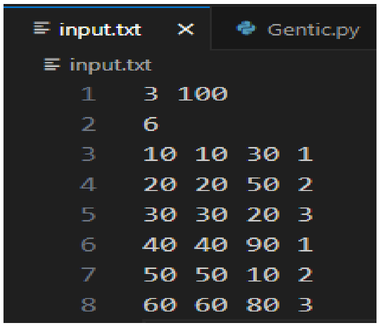
Simulated:



Genetic:

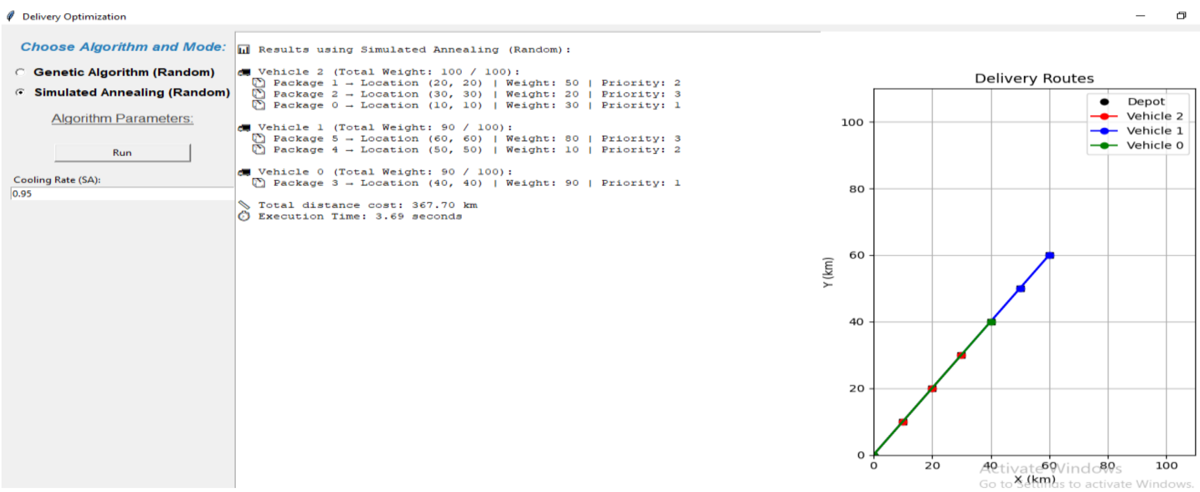


Test Case 4:



```
3 100
6
10 10 30 1
20 20 50 2
30 30 20 3
40 40 90 1
50 50 10 2
60 60 80 3
```

Simulated:



Genetic: