Capacitated Vehicle Routing Problem (CVRP) using Simulated Annealing

Belmadoui Moncef Group 8

Zakaria Mokhtari Group 2

Merouane Abdelhalim Arroudj Group 2

1 - Abstract :

This report details the development and implementation of a Simulated Annealing (SA) approach for solving the Capacitated Vehicle Routing Problem (CVRP). The CVRP requires determining optimal routes for a fleet of vehicles—each with limited capacity—to serve a set of customers with known demands.

2 - Introduction :

The Vehicle Routing Problem (VRP) is a complex combinatorial optimization challenge that involves finding optimal routes for a fleet of vehicles to serve a set of customers, while respecting various constraints. In this project, we tackle the Capacitated Vehicle Routing Problem (CVRP), where each vehicle has a limited capacity and each customer has a specific demand that must be fulfilled. This document outlines our approach to solving the CVRP using Simulated Annealing (SA), a powerful metaheuristic optimization technique. We present the problem formulation, solution methodology, implementation details, and results analysis.

3 - Data Overview :

We are working with two tables:

- **Vehicles**
- **Nodes (Locations with demands)**

All vehicles start at the **same location (40, 50)**.

Capacities range from **41 to 46 units**.

Node 0 is the **depot** (same location as the vehicles' start).

Remaining nodes are **customers** with different demands ranging from **3 to 40 units**.

The demands must be served without exceeding each vehicle's **individual capacity**.

4 - Problem Definition :

The Capacitated Vehicle Routing Problem can be defined as follows:

- A fleet of vehicles with known capacities is stationed at a central depot
- A set of customers with known locations and demands must be served
- Each customer must be visited exactly once by exactly one vehicle
- Each vehicle starts and ends its route at the depot
- The total demand on any route cannot exceed the capacity of the assigned vehicle
- The objective is to minimize the total distance traveled by all vehicles The CVRP is NP-hard, meaning that finding optimal solutions for large problem instances is computationally intractable. This complexity justifies our choice of a metaheuristic approach.

4.1 - Mathematical Model:

The Capacitated Vehicle Routing Problem (CVRP) can be formulated as follows:

Sets:

- V = {0, 1, 2, ..., n}: Set of nodes, where 0 is the depot and {1, 2, ..., n} are customers
- K = {1, 2, ..., m}: Set of available vehicles

Parameters:

- c_{ij} : Distance (or cost) from node **i** to node **j**
- D_i: Demand of customer **i** (D_0 = 0 for the depot)
- Q_k: Capacity of vehicle k

Decision Variables:

- x_{ijk}=$\{1, \text{if vehicle } k \text{ travels directly from node } i \text{ to node } j \ 0, \text{otherwise}$

- y_{ik}=$\{1, \text{if customer } i \text{ is served by vehicle } k \ 0, \text{otherwise}$

Objectif function:

Minimize total distance traveled :

$$\min Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk}$$

Constraints:

1. **Each customer is visited exactly once:**

$$\sum_{k \in K} y_{ik} = 1, \quad \forall i \in V \setminus \{0\}$$

2. **Each vehicle starts and ends at the depot:**

   Each vehicle must start from the depot and return to the depot. This ensures that each vehicle makes exactly one route.

$$\sum_{j \in V} x_{0jk} = \sum_{i \in V} x_{i0k} = 1, \quad \forall k \in K$$

3. **Flow conservation (route continuity):**

   This ensures route continuity - if a vehicle enters a customer node, it must also exit that node. This constraint connects the x_{ijk} variables with the y_{ik} variables.

$$\sum_{i \in V} x_{ijk} = \sum_{i \in V} x_{jik} = y_{jk}, \quad \forall j \in V \setminus \{0\}, \forall k \in K$$

4. **Vehicle capacity constraints:**

   The total demand of customers served by each vehicle cannot exceed the vehicle's capacity.

$$\sum_{i \in V \setminus \{0\}} d_i y_{ik} \leq Q_k, \quad \forall k \in K$$

5. **Subtour elimination constraints:**

   These constraints prevent the formation of subtours (cycles that don't include the depot). Without these constraints, solutions could include isolated cycles of customers not connected to the depot.

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ijk} \leq |S| - 1, \quad \forall S \subseteq V \setminus \{0\},\ |S| \geq 2,\ \forall k \in K$$

5 - Methodology

5.1 - Simulated Annealing

We chose **Simulated Annealing (SA)** for solving this problem due to its key advantages:

1. **Escaping local optima**: SA's probabilistic acceptance of worse solutions enables it to explore the search space more broadly and avoid becoming trapped in local minima.

2. **Simplicity and effectiveness**: SA strikes a good balance between ease of implementation and solution quality, making it suitable for complex combinatorial problems.

3. **Flexible neighborhood design**: The method allows for the integration of various neighborhood moves, enhancing its ability to explore diverse parts of the solution space.

4. **Controlled convergence**: The temperature schedule gradually shifts the algorithm's behavior from exploration to exploitation, helping refine the solution over time.

Solution Representation

Solutions are represented as a list of routes, where each route is a list of customer nodes:

```
Example: [[1, 3, 5], [2, 4], [6, 7, 8], [9, 10]] represents 4 vehicle
routes.
```

This representation offers several benefits:

- **Clarity**: It provides an intuitive understanding of how customers are grouped into routes.

- **Computational efficiency**: It simplifies the calculation of route distances and demands.

- **Modular structure**: It supports the straightforward implementation of various neighborhood operations for solution improvement.

5.2 - Neighborhood Operators

We implemented three neighborhood operators to generate diverse candidate solutions:

1. **Swap**: Exchange the positions of two customers, either within the same route or across different routes.

2. **Relocate**: Move a customer from one route to another, potentially balancing loads or reducing costs.

3. **Two-opt**: Reverse the order of a segment within a single route to improve route efficiency by eliminating crossovers.

These operators provide a balanced mix of local adjustments and larger structural changes, enabling thorough exploration of the solution space.

5.3 - Algorithm Implementation

Our Simulated Annealing algorithm follows the conventional framework:

1. **Initialization**:
    - Construct an initial feasible solution using a greedy heuristic.
    - Set the initial temperature and define cooling parameters.

2. **Main Loop**:
    - Generate a neighboring solution by applying one of the neighborhood operators.
    - Verify feasibility of the neighbor (e.g., capacity constraints).
    - Compute the acceptance probability based on the difference in solution quality and current temperature.
    - Accept or reject the neighbor solution probabilistically.
    - Update the best solution found so far if the neighbor is better.
    - Reduce the temperature according to the cooling schedule.

3. **Termination**:
    - The algorithm stops when the temperature drops below a predefined minimum or when the maximum number of iterations is reached.

5.4 - Parameter Selection

The performance of Simulated Annealing (SA) is highly sensitive to its parameter configuration. After careful experimentation, the following parameters were selected:

- **Initial temperature**: 10 — high enough to encourage broad exploration early on
- **Cooling rate**: 0.99999 — a slow decay that allows for a thorough search
- **Minimum temperature**: 1 — the stopping criterion
- **Maximum iterations**: 100,000 — ensures sufficient time for convergence

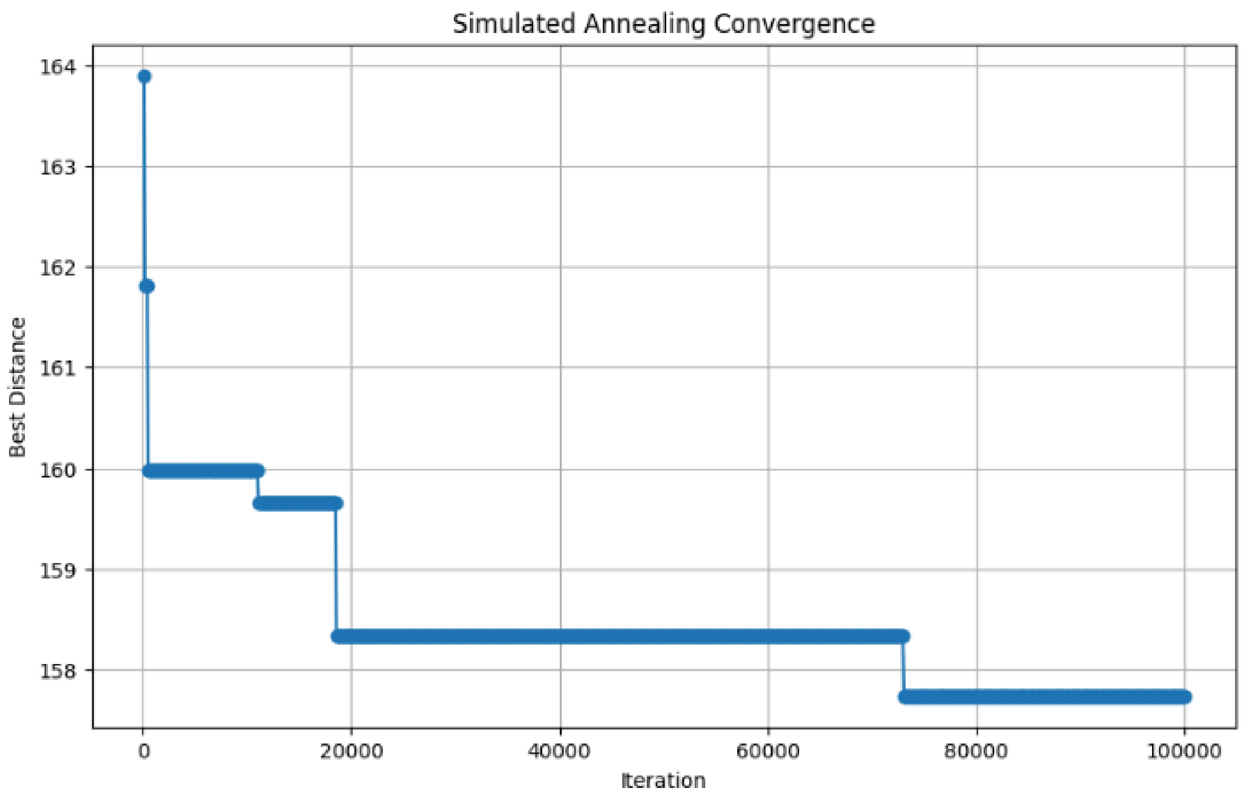
These parameters offer a strong balance between exploration and exploitation, enabling the algorithm to progressively refine its solutions.

6 - Results and Analysis

6.1 - Performance Metrics

The SA algorithm yielded the following performance indicators:

- **Total distance traveled**: 157.74 units
- **Number of routes**: 4
- **Execution time**: 1.91 seconds
- **Vehicle utilization**: Ranged between 78.05% and 100%



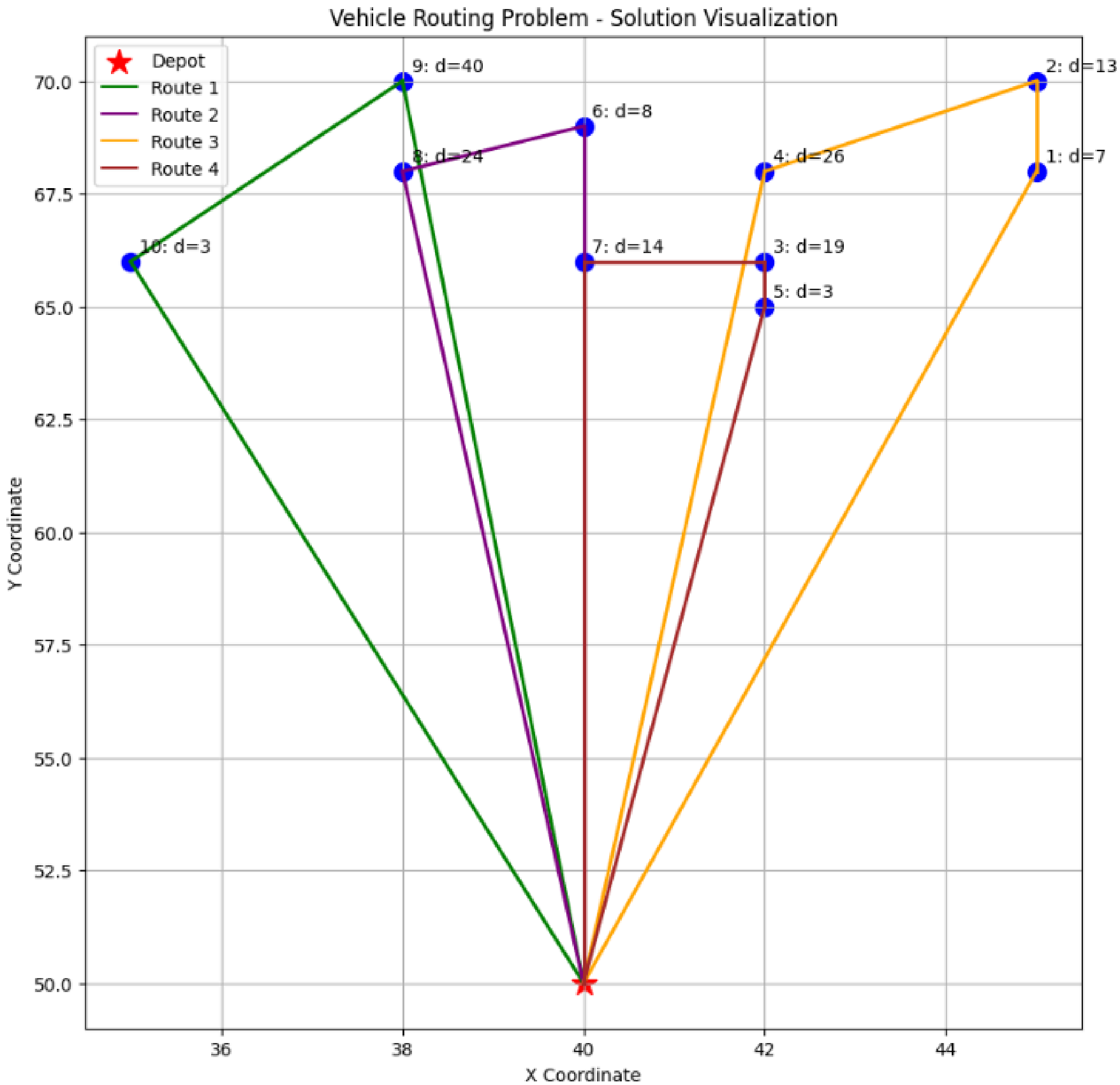
6.2- Route Details

The final solution includes four well-optimized routes:

1. **Route 1**: Depot → 10 → 9 → Depot
    - Capacity: 45
    - Demand: 43
    - Distance: 41.86
    - Utilization: 95.56%
2. **Route 2**: Depot → 5 → 3 → 7 → Depot
    - Capacity: 42
    - Demand: 36
    - Distance: 34.13

    - Utilization: 85.71%

3. **Route 3**: Depot → 4 → 2 → 1 → Depot
    - Capacity: 46
    - Demand: 46
    - Distance: 42.40
    - Utilization: 100.00%
4. **Route 4**: Depot → 6 → 8 → Depot
    - Capacity: 41
    - Demand: 32
    - Distance: 39.35
    - Utilization: 78.05%



6.3 - Analysis of Results

Several key observations can be drawn:

1. **Strong vehicle utilization**: The average utilization of 89.83% indicates efficient resource deployment, with some vehicles operating near full capacity.

2. **Well-balanced routes**: The distance across all routes is fairly uniform (ranging from 34.13 to 42.40), which ensures balanced workloads.

3. **Effective convergence**: The solution quality improves rapidly in the early stages and gradually refines over time, characteristic of an effective SA schedule.

4. **Fast computation**: With a runtime of just 1.91 seconds, the algorithm demonstrates strong efficiency for medium-sized problems.

6.4 - Algorithm Efficiency

The Simulated Annealing algorithm proved efficient in several aspects:

1. **Rapid convergence**: High-quality solutions emerged early in the process

2. **Diverse exploration**: Neighborhood operators enabled a broad and balanced search

3. **Feasibility handling**: The algorithm reliably produced feasible solutions, minimizing computational waste

7- Challenges and Solutions

Throughout development, several challenges were addressed:

1. **Low-quality initial solutions**: Poor starts can hinder performance. We resolved this by implementing a greedy initialization.

2. **Parameter sensitivity**: Fine-tuning was essential. Extensive testing helped identify optimal values for temperature and cooling.

3. **Exploration vs. exploitation**: Striking this balance is critical. Our temperature schedule gradually shifts focus as needed.

4. **Neighborhood design**: Solution quality heavily depends on the move operators. Our use of swap, relocate, and two-opt ensured diverse and effective exploration.

8 - Conclusion

This project demonstrates the effectiveness of Simulated Annealing for solving the Capacitated Vehicle Routing Problem (CVRP). Key takeaways include:

1. **High-quality solutions**: The algorithm consistently generated efficient routes with high utilization

2. **Fast runtime**: Practical execution times make the approach applicable to real-world scenarios

3. **Robustness**: The method handles constraints well and explores the solution space thoroughly

Future Directions

Potential improvements include:

- Adding constraints such as time windows or multiple depots

- Hybridizing with other metaheuristics for enhanced performance

- Parallelizing the algorithm for large-scale problems

This work highlights how a well-implemented metaheuristic can effectively address complex optimization challenges in transportation and logistics.