Vehicle Routing Problem - VRP

Emma Szigeti, Flaviu Sălnicean

Octombrie 2025

1 Introduction

The Vehicle Routing Problem (VRP) is about planning routes for vehicles to deliver goods to all customers while minimizing total distance or cost. In this project, we focus on the Capacitated VRP (CVRP), where each vehicle has a limited capacity, so routes must be designed to serve all customers efficiently without exceeding these limits.

To solve this problem, we use Genetic Algorithms, which are inspired by natural evolution. They create a population of possible solutions, and then improve them step by step using selection, crossover, and mutation. This way, the algorithm gradually finds better and more efficient routes.

We chose to work on the Capacitated Vehicle Routing Problem (CVRP) because it is directly related to real-world logistics and delivery problems. Other classic optimization problems, like the Travelling Salesman Problem, the N-Queens problem, or the Ant Colony problem, are interesting theoretically, but they are either too simplified or too abstract.

CVRP allows us to apply algorithms in a realistic scenario where vehicles have capacity limits and customers must be served efficiently. It combines optimization and practical constraints, making it both challenging and meaningful for a project.

2 Difference between the Vehicle Routing Problem and the Travelling Salesman Problem

The Travelling Salesman Problem can be considered, at least from my point of view, as a simplified version of the Vehicle Routing Problem.

The Travelling Salesman Problem (TSP) consists in finding the shortest possible route that passes through a given set of points (theoretically, cities), each visited exactly once, with known distances between them, and the route must end at the starting point.

The Vehicle Routing Problem (VRP), in its basic form, consists in finding a set of routes with minimal total cost for a fleet of vehicles (theoretically, delivery trucks) starting from the same depot, such that each customer is visited exactly once. Various additional constraints can be added later.

So, what is the difference? In the Travelling Salesman Problem, there is only one person who must complete the shortest possible tour through all points, while in the Vehicle Routing Problem there are multiple vehicles, each requiring an optimal route that covers a subset of points so that, together, all points are visited.

3 Types of VRP

There are several variations of the Vehicle Routing Problem, including:

- CVRP (Capacitated Vehicle Routing Problem)
- VRPP (Vehicle Routing Problem with Profits)
- VRPTW (Vehicle Routing Problem with Time Windows)
- VRPWT (Vehicle Routing Problem with Transfers)
- VRPMT (Vehicle Routing Problem with Multiple Trips)
- VRPPD (Vehicle Routing Problem with Pickup and Delivery)
- VRPB (Vehicle Routing Problem with Backhauls)
- PVRP (Periodic Vehicle Routing Problem)
- SVRP (Stochastic Vehicle Routing Problem)
- SDVRP (Split Delivery Vehicle Routing Problem)
- MDVRP (Multiple Depot Vehicle Routing Problem)
- IRP (Inventory Routing Problem)
- OVRP (Open Vehicle Routing Problem)
- OP (Orienteering Problem)
- TOP (Team Orienteering Problem)
- CTOP (Capacitated Team Orienteering Problem)
- TOPTW (Capacitated Team Orienteering Problem with Time Windows)
- EVRP (Electric Vehicle Routing Problem)

4 The Meaning of Each Variant

CVRP is the simplest and most common form of VRP. Each vehicle has a limited carrying capacity, and the goal is to find the set of routes with minimal total cost.

VRPP is a variant of VRP in which each customer provides a certain profit, while each trip has an associated cost. The objective is to maximize the total profit.

VRPTW introduces time windows for customers: each must be visited within a specific time interval. The goal remains to minimize the total cost.

VRPWT allows vehicles to leave or pick up goods at transfer points. The objective is to find routes with minimal total cost.

VRPMT allows each vehicle to perform multiple trips per day (for instance, returning to the depot to reload). The goal is again to minimize total travel cost.

VRPPD considers customers who both receive and send deliveries. When a vehicle visits a customer, it delivers a package and picks up another one for delivery elsewhere. The goal is to minimize the total cost.

VRPB distinguishes between two types of customers: those who only receive goods and those who only send them. The objective is to minimize total cost.

PVRP extends VRP over a planning horizon (for example, a week or a month). The goal is to decide on which days and with which vehicles deliveries will be made to each customer.

SVRP introduces randomness: some parameters are uncertain (stochastic), such as traffic conditions, weather, or customer demand. The objective is to minimize expected total cost while accounting for uncertainty.

SDVRP allows customer demands to be split among several vehicles, meaning that deliveries can be divided. The goal is to minimize total cost.

MDVRP involves multiple depots from which deliveries can be made to customers. The goal is to minimize total transportation cost.

IRP integrates inventory management. Each customer has a minimum and maximum stock level, and the objective is to plan routes that maintain proper inventory levels while minimizing total cost.

OVRP assumes that vehicles do not return to the depot at the end of the day. The objective remains to minimize total cost.

OP (Orienteering Problem) associates each customer with a profit, and a single vehicle has a limited distance or time available. The goal is to select which customers to visit to maximize total profit.

TOP (Team Orienteering Problem) generalizes the OP to multiple vehicles. The goal is to select customer visits that maximize total profit.

CTOP adds capacity constraints to the TOP, where each vehicle has limited capacity. The objective remains to maximize total profit.

TOPTW extends CTOP by adding time windows for customers. The goal is to select customer visits that maximize total profit while respecting time constraints.

EVRP models routes for electric vehicles with limited driving range and recharging requirements. The goal is to minimize total cost while considering energy consumption and charging time.

5 Chosen Variant

For this project, we chose to work with the CVRP (Capacitated Vehicle Routing Problem) variant, as it is the easiest to understand and to implement.

6 Mathematical formulation for the Capacitated Vehicle Routing Problem (CVRP)

Data:

- $V = 0, 1, \ldots, n$ — set of nodes, where 0 is the depot and $1, \ldots, n$ are the customers.
- $c_{ij} \geq 0$ — cost (or distance) from i to j.
- $d_i > 0$ — demand of customer i, for $i = 1, \ldots, n$; $d_0 = 0$.
- $Q > 0$ — capacity of each vehicle.
- K — maximum number of available vehicles.

Decision variables:

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V,\ i \neq j$$

$$f_{ij} \geq 0, \quad \forall i, j \in V,\ i \neq j$$

Interpretation: $x_{ij} = 1$ if the arc (i, j) is used in a route; f_{ij} is the flow of goods transported on arc $i \to j$.

Objective function:

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} c_{ij}\, x_{ij}$$

Constraints:

$$\sum_{\substack{j\in V\\ j\neq i}} x_{ij} = 1, \qquad \forall i \in \{1,\dots,n\} \qquad \text{(each customer is visited exactly once)} \tag{1}$$

$$\sum_{\substack{i\in V\\ i\neq j}} x_{ij} = 1, \qquad \forall j \in \{1,\dots,n\} \qquad \text{(each customer is left exactly once)} \tag{2}$$

$$\sum_{\substack{j\in V\\ j\neq 0}} x_{0j} \le K, \quad \sum_{\substack{i\in V\\ i\neq 0}} x_{i0} \le K \qquad \text{(maximum } K \text{ vehicles)} \tag{3}$$

$$0 \le f_{ij} \le Q\, x_{ij}, \qquad \forall i,j \in V,\ i \neq j \qquad \text{(linking flow and arc)} \tag{4}$$

$$\sum_{\substack{j\in V\\ j\neq i}} f_{ji} + d_i = \sum_{\substack{j\in V\\ j\neq i}} f_{ij}, \qquad \forall i \in \{1,\dots,n\} \qquad \text{(flow conservation at customers)} \tag{5}$$

$$\sum_{\substack{j\in V\\ j\neq 0}} f_{0j} = \sum_{i=1}^{n} d_i \qquad \text{(total flow leaving the depot)} \tag{6}$$

Domains:

$$x_{ij} \in \{0,1\}, \qquad f_{ij} \ge 0$$

7 Fitness function and penalized cost for CVRP

In genetic algorithms, the fitness function evaluates the quality of a candidate solution (set of routes). Because not all generated solutions satisfy capacity or vehicle-number constraints, a penalized cost is used to integrate these violations into the total cost.

Let $C(S)$ be the unpenalized total cost of a solution S (set of routes):

$$C(S) \;=\; \sum_{r\in S} \sum_{(i,j)\in r} c_{ij},$$

where the inner sum is over successive arcs in route r (including the arc from depot 0 to the first customer and the arc from the last customer back to 0).

Route-specific definitions

For each route $r \in S$, define:

$$Q_r \;=\; \sum_{i \in r} d_i \qquad \text{(total load on route r } r\text{)},$$

$$\Delta_r \;=\; \max\bigl(0,\; Q_r - Q\bigr) \qquad \text{(capacity violation for rout } r\text{)}.$$

If there is a limit K on the number of vehicles, also define:

$$\Delta_K \;=\; \max\bigl(0,\; |S| - K\bigr) \qquad \text{(number of vehicles in excess)}.$$

Penalized cost

Then the penalized cost for solution S is:

$$C_{\text{pen}}(S) \;=\; C(S) \;+\; \alpha \sum_{r \in S} \Delta_r \;+\; \beta\,\Delta_K,$$

where $\alpha, \beta > 0$ are penalty coefficients (e.g., α — cost per unit of exceeded load; β — fixed cost per vehicle in excess).

Transformation into a fitness function

$$\text{fitness}(S) \;=\; \frac{1}{1 + C_{\text{pen}}(S)}.$$

So, higher fitness values correspond to solutions with lower penalized cost (i.e., shorter and more feasible routes).

8 Representation of the Individual for CVRP

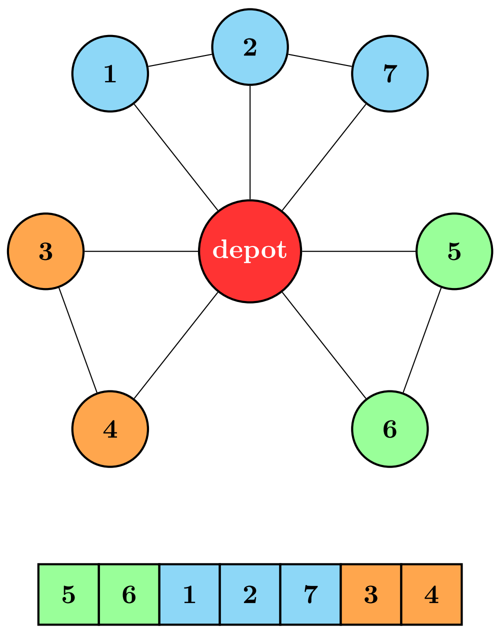
In the Genetic Algorithm framework, each individual encodes a possible solution to the Capacitated Vehicle Routing Problem (CVRP). Each solution consists of a set of routes that start and end at the central depot, visiting every customer exactly once.

Array-based sequential representation

In this simplified representation, the chromosome contains only customer IDs, without using 0 or any explicit separator to mark route boundaries. Each customer appears exactly once.

The CVRP routes are reconstructed dynamically: the decoding algorithm reads the sequence from left to right and assigns customers to the current vehicle as long as the load does not exceed the vehicle capacity. When adding the next customer would violate capacity constraints, a new vehicle route is started automatically.

This representation is compact, avoids unnecessary depot markers, and allows the route structure to be generated implicitly based on capacity limits rather than explicit symbols.



(1) Visual grouping of customers by color and route

9 Selection Operators in the Genetic Algorithm

In a Genetic Algorithm (GA), the selection operator determines which individuals from the current population are chosen to produce offspring for the next generation. Its role is to favor individuals with higher fitness while still preserving diversity.

1. Roulette Wheel Selection

Roulette Wheel Selection assigns each individual a probability proportional to its fitness.

If the population contains m individuals and the fitness of individual S_i is f_i, then:

$$P(S_i) = \frac{f_i}{\sum_{k=1}^{m} f_k}$$

Advantages:

- Simple and widely used.
- Higher fitness $\Rightarrow$ higher probability of selection.

Disadvantages:

- Premature convergence if a few individuals dominate.
- Sensitive to large fitness differences.

2. Rank Selection

Instead of using raw fitness values, individuals are sorted by fitness. If the best individual has rank m and the worst has rank 1, then:

$$P(S_i) = \frac{r_i}{\sum_{k=1}^{m} k},$$

where r_i is the rank of individual i.

Advantages:

- Eliminates the negative effect of extreme fitness values.
- Maintains constant selective pressure.

Disadvantages:

- Requires sorting the population at each generation.

3. Tournament Selection

Tournament Selection works by repeatedly choosing a small random subset of individuals and selecting the best from that subset.

The procedure is:

1. Select t individuals uniformly at random.
2. Choose the one with the highest fitness.

3. Repeat until enough parents are selected.

Typical value: $t = 2, 3$ or 4.
Advantages:

- Very simple and efficient.
- Selective pressure is easy to control through t.
- Works well even with noisy or irregular fitness distributions.

Disadvantages:

- Large t reduces population diversity.

4. Elitism
Elitism means that the best individuals automatically survive to the next generation.
Let:
$E =$ number of elite individuals
Typical values: $E = 1$ or $E = 2$.
Advantages:

- Guarantees that the best solution is not lost.
- Increases convergence speed.

Disadvantages:

- Too much elitism can cause premature convergence.

Selection Method Used in This Project
In this project, we use:

- **Roulette Wheel Selection**, combined with
- $E = 1$ elite individual.

Reasons:

- Roulette Wheel Selection is simple and effective for optimization problems like VRP.
- It favors individuals with higher fitness while still allowing diversity in selection.
- Elitism ensures that the best individual found so far is preserved.

10 Mutation Operators for CVRP

Mutation introduces small random changes in an individual, helping the genetic algorithm avoid premature convergence and explore new regions of the search space. In CVRP, mutation must preserve the constraint that each customer appears exactly once in the solution.

1. Swap Mutation

Swap Mutation selects two customer positions at random and exchanges them.

Example:

$$[0, 5, 7, 0, 3, 4, 0] \Rightarrow [0, 5, 4, 0, 3, 7, 0]$$

Advantages:

- Simple and effective.
- Preserves feasibility automatically (no duplicate customers).

Disadvantages:

- Produces only small local changes.

2. Insertion Mutation

This operator removes a customer from one position and reinserts it elsewhere in the sequence.

Example:

$$[0, 5, 7, 3, 4, 0] \Rightarrow [0, 5, 3, 7, 4, 0]$$

Procedure:

1. Choose a random customer.
2. Remove it from its current position.
3. Insert it at another random position (not necessarily in the same route).

Advantages:

- More flexible than swap.
- Can create significantly different route structures.

Disadvantages:

- Can disrupt good partial routes.

3. Inversion Mutation (Reversal)

Inversion selects a random sub-sequence and reverses its order.

Example:

$$[0, 5, \underline{7, 3, 4}, 0] \Rightarrow [0, 5, \underline{4, 3, 7}, 0]$$

Advantages:

- Very effective for route optimization (similar to 2-opt).
- Useful for reducing route length.

Disadvantages:

- Mutation effect can be large depending on interval length.

4. 2-opt Local Search Mutation

This mutation operator transforms a route into a shorter one by removing two edges and reconnecting the route in a different order.

Given customers i and j in a route, 2-opt replaces:

$$(i, i+1), (j, j+1) \; with \; (i, j), (i+1, j+1).$$

Effect: the segment between i+1 and j is reversed.

Advantages:

- Strong local optimization operator.
- Often improves route quality significantly.

Disadvantages:

- More computationally expensive.
- Typically applied with a small probability.

5. Route Splitting Mutation (Optional)

This operator selects two random cut points and reshuffles route boundaries.

Example:

$$[0, 5, 7, 0, 3, 4, 0] \Rightarrow [0, 5, 7, 3, 0, 4, 0]$$

Useful when the algorithm tends to produce too many or too few vehicles.

Advantages:

- Encourages exploration between routes.
- Helps escape local optima.

Disadvantages:

- Can easily violate route capacity constraints (fixed later using penalty or repair).

Mutation Strategy Used in This Project

For this CVRP implementation, we use an **adaptive mutation strategy** to balance exploration and exploitation:

- **Primary mutation:** Swap Mutation with base probability 0.7

- **Secondary mutation:** Inversion Mutation with base probability 0.3
- **Adaptive rate:** If the best fitness does not improve for a certain number of generations (e.g., 20), the mutation rates are temporarily increased (doubled or tripled) for a few generations. This helps the GA escape local optima and explore new solutions.

This adaptive approach ensures that the algorithm maintains diversity and improves the chances of finding better solutions without disrupting convergence when fitness is improving steadily.

11 Survivor Selection Strategies

In this project, three survivor selection strategies are used. They correspond to classical evolutionary strategies, but are adapted to the CVRP genetic algorithm.

1. (μ, λ) **Selection (Parents Die, Many Offspring, Best Offspring Survive)**

In this strategy, all parents are removed after reproduction. Each parent produces several offspring, and then the next generation is created by selecting the best individuals among all offspring.

Let:

$$\mu = number\ of\ parents,\ \lambda = number\ of\ generated\ offspring.$$

Only the best μ offspring survive:

$$Next\ generation = best\ \mu\ individuals\ from\ the\ offspring.$$

Characteristics:

- Parents do not survive.
- Strong selective pressure on offspring quality.
- Useful for exploration when producing many children.

2. $(\mu + \lambda)$ **Selection (Parents Survive, Best Individuals from Both Sets)**

Here, parents are allowed to survive together with their offspring. Selection is performed on the union of parents and offspring:

$$Next\ generation = best\ \mu\ individuals\ from\ (parents + offspring).$$

Characteristics:

- Parents may remain in the population if they are good.

- Provides stability and prevents losing high-quality solutions.
- Maintains the same number of parents as the original population.

3. Fitness-Proportional Offspring Selection (Parents Die, Offspring Chosen Probabilistically)

In this survival strategy, parents are removed, and offspring are selected to form the next generation using probabilistic selection based on fitness.

Each offspring S_i has a probability of survival proportional to its fitness f_i:

$$P(S_i) = \frac{f_i}{\sum_{k=1}^{\lambda} f_k}$$

The algorithm repeatedly samples individuals according to these probabilities until μ survivors are chosen.

Characteristics:

- Parents die; only offspring can enter the next generation.
- Selection is randomized but biased toward fitter individuals.
- Helps maintain diversity compared to purely deterministic selection.

Selection Method Used in This Project

In this project, the $(\mu + \lambda)$ selection (Parents Survive, Best Individuals from Both Sets) strategy is used. This allows high-quality parents to remain in the population while still incorporating the best offspring, providing a balance between stability and exploration.

12 Crossover and Hybridization Operators

In genetic algorithms for the Capacitated Vehicle Routing Problem, crossover operators combine information from two parent solutions to generate new offspring. Hybridization refers to combining crossover with mutation or other local improvement methods to improve solution quality.

Random Cut-Point Crossover

This crossover operator works for permutation-based solutions. A random number n of cut points is selected, dividing the parent sequences into $(n + 1)$ segments. The offspring is created by alternating segments from each parent, while keeping each customer exactly once in the child.

Example Parents:

$$P_1 = [3, 5, 7, 2, 1, 6, 4, 8], \quad P_2 = [7, 1, 4, 6, 3, 8, 5, 2]$$

Randomly generated cut points: $\{2, 5, 7\}$

Split sequences:

$$P_1 : [3, 5] \mid [7, 2, 1] \mid [6, 4] \mid [8], \quad P_2 : [7, 1] \mid [4, 6, 3] \mid [8, 5] \mid [2]$$

Alternating segments to form offspring:

$$O_1 = [3, 5] \mid\mid [4, 6, 3] \mid\mid [6, 4] \mid\mid [2]$$

Final correction ensures no duplicates and all customers are present.

Visual Representation

$$\begin{array}{c}
\text{Cut points: } 2, 5, 7 \\
\begin{array}{cccccccccccc}
3 & 5 & | & 7 & 2 & 1 & | & 6 & 4 & | & 8 \\
7 & 1 & | & 4 & 6 & 3 & | & 8 & 5 & | & 2
\end{array} \\
\Downarrow \text{ alternate segments} \\
\begin{array}{cccccccccccc}
3 & 5 & | & 4 & 6 & 3 & | & 6 & 4 & | & 2
\end{array}
\end{array}$$

Hybridization: Crossover with Mutation

Hybridization improves exploration by combining crossover with mutation:

1. Generate offspring using random cut-point crossover.
2. Apply a mutation operator (Swap or Inversion) with a given probability.
3. Correct any duplicate or missing customers if necessary.

Example Starting from offspring:

$$O_1 = [3, 5, 4, 6, 3, 6, 4, 2]$$

Apply Swap mutation (swap positions 2 and 4):

$$O_1' = [3, 6, 4, 5, 3, 6, 4, 2]$$

Final correction ensures each customer appears exactly once.

Advantages of Hybridization

- Combines exploration (crossover) and local improvement (mutation).
- Reduces risk of premature convergence.
- Increases diversity in the population while keeping good solutions.

Key Notes

- Random cut points make crossover flexible and unpredictable, which is good for exploring different routes.
- Hybridization is especially useful in vehicle routing because routes are sequences and small changes can create better solutions.
- After crossover and mutation, always ensure each customer appears exactly once.

13 Future Improvements and Optimization

Although the current implementation provides satisfactory results for small and medium-sized instances, several enhancements could further improve the algorithm's performance and scalability for larger datasets.

Heuristic Initialization

Currently, the initial population is generated randomly. A significant improvement would be to initialize a portion of the population (e.g., 50%) using heuristics such as the Nearest Neighbor algorithm or the Savings Algorithm (Clarke-Wright). This would provide the genetic algorithm with a higher-quality starting point, accelerating convergence.

Hybridization with Local Search (Memetic Algorithm)

To refine the solutions found by the genetic operators, a Local Search mechanism could be integrated. Applying a 2-Opt or 3-Opt operator on the best individuals of each generation (or periodically) would help eliminate crossing paths within routes and strictly minimize local travel distances, transforming the GA into a Memetic Algorithm.

Parameter Tuning via Grid Search

The current parameters (mutation rate, population size, crossover probability) are set based on empirical observations. A systematic Hyperparameter Tuning process (Grid Search or Bayesian Optimization) could dynamically determine the optimal configuration for a specific problem instance, balancing exploration and exploitation more effectively.

Elitism Management

While Elitism ensures the best solution is preserved, excessive elitism can lead to premature convergence. Implementing Island Models (running parallel populations that exchange individuals periodically) could maintain higher genetic diversity and prevent the algorithm from getting stuck in local optima.

14 Bibliografie

https://blog.locus.sh/vehicle-routing-problem/

https://en.wikipedia.org/wiki/Vehicle_routing_problem
https://tttp-au.com/wp-content/uploads/2024/06/Pages-from-TTTP-Vol-9_No-1_2024-WEB-8.pdf
https://ro.wikipedia.org/wiki/Problema_rut%C4%83rii_vehiculelor
https://ro.wikipedia.org/wiki/Problema_comis-voiajorului
https://developers.google.com/optimization/routing/cvrp
https://www.upperinc.com/glossary/route-optimization/vehicle-routing-problem-with-profits
https://developers.google.com/optimization/routing/vrptw
https://www.sciencedirect.com/science/article/abs/pii/S0305054825000085
https://www.sciencedirect.com/science/article/abs/pii/S0305054820301040
https://www.upperinc.com/glossary/route-optimization/vehicle-routing-problem-with-backhau
https://www.upperinc.com/glossary/route-optimization/periodic-vehicle-routing-problem-pv
https://www.upperinc.com/glossary/route-optimization/stochastic-vehicle-routing-problem-
https://www.sciencedirect.com/science/article/pii/S0305054814002159
https://www.sciencedirect.com/science/article/abs/pii/S0305054813001408
https://www.hexaly.com/templates/inventory-routing-problem-irp
https://www.upperinc.com/glossary/route-optimization/open-vehicle-routing-problem-ovrp/
https://www.sciencedirect.com/topics/mathematics/orienteering-problem
https://drops.dagstuhl.de/storage/01oasics/oasics-vol014-atmos2010/OASIcs.ATMOS.2010.142/OASIcs.ATMOS.2010.142.pdf
https://www.researchgate.net/publication/256022425_The_Capacitated_Team_Orienteering_Problem_A_Bi-Level_Filter-and-Fan_Method
https://www.praiseworthyprize.org/jsm/index.php?journal=irecos&page=article&op=view&path%5B%5D=18244
https://www.sciencedirect.com/science/article/pii/S2352146516000089
https://www.researchgate.net/figure/A-CVRP-solution-and-the-individual-encoding-represen
fig2_394965948