Vehicle Routing Problem - VRP

Emma Szigeti, Flaviu Sălnicean

Octombrie 2025

1 Introduction

....

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:

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

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

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

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

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

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

Domains:

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

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 representation, each gene corresponds to a node: either the depot or a customer. The value 0 represents the depot (start or end of a route), while numbers $1, 2, 3, \ldots, n$ represent customers.

For example:

$$R_1 : 0 \to 2 \to 5 \to 7 \to 0,$$
$$R_2 : 0 \to 3 \to 4 \to 8 \to 0,$$
$$R_3 : 0 \to 1 \to 6 \to 9 \to 0.$$

Advantages

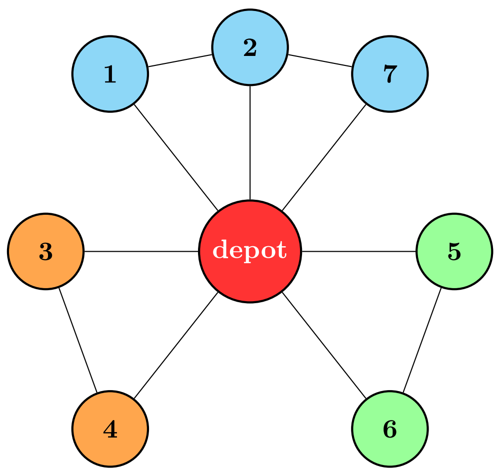
- Clear and intuitive structure: The representation directly shows the order of visits for each vehicle.
- Easy to verify route validity and vehicle capacity constraints: You can quickly compute the load of each vehicle.
- Single-array representation: All routes are represented in a single array, simplifying storage and manipulation.
- Flexible for different VRP variants: Works with CVRP, VRPTW, VRPPD, etc., by adjusting constraints or penalties.

Disadvantages

- Sequence interpretation can be less intuitive for very large instances: Long arrays with many vehicles and customers can be harder to read manually.
- Fixed depot representation: If multiple depots are involved, the array format needs modification.
- Limited direct information about route optimization: The representation shows order, but does not encode distance minimization explicitly.

Representation of a CVRP solution using numerical customer IDs

Each number represents a customer, and 0 represents the depot (start and end of each route). The sequence encodes the exact visiting order of customers for each route.



0	**5**	**6**	**0**	**1**	**2**	**7**	**0**	**3**	**4**	**0**

(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:
Tournament selection with size $t = 3$,
combined with:
$E = 1$ elite individual.
Reasons:

- Tournament selection performs well for routing problems such as VRP.
- It provides a good balance between exploration and exploitation.
- 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:

$$[0, 5, 7, 0, 3, 4, 0] => [0, 5, 4, 0, 3, 7, 0]$$

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:

$$[0, 5, 7, 3, 4, 0] => [0, 5, 3, 7, 4, 0]$$

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:

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

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:

$$[0, 5, 7, 0, 3, 4, 0] => [0, 5, 7, 3, 0, 4, 0]$$

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 a combination of two mutation operators:

$$Primary : Swap\ Mutation\ (probability\ 0.7)$$

$$Secondary : Inversion\ Mutation\ (probability\ 0.3)$$

- Swap is fast and maintains good diversity.
- Inversion helps improve route structure and reduce travel cost.
- Both preserve the permutation property (each customer appears exactly once).

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.

12 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