Vehicle Routing Problem

Computational Inteligence course project

Anja Cvetković mi20127@alas.matf.bg.ac.rs
Stefan Jevtić mi20187@alas.matf.bg.ac.rs

2024

Contents

1 Introduction

Vehicle Routing Problem (abbreviated as VRP) is a term used for the problem of establishing routes with a set of vehicles based on one or more warehouses and customers to which goods are to be delivered. This is a major problem in distribution and logistics even today, and was first described in 1959 by Danzig and Ramser. It can be seen as a generalization of the problem of the traveling salesman, where the minimization of the traveled distance is most often done, and nowadays the minimization of fuel consumption or time is also done.

2 Classic VRP

Let $X = \{x_i | i = 1, ..., N\}$ be a set of N customers and x_0 be the depot.
Let $V = \{v_j | j = 1, ..., M\}$ be a set of vehicles inside of depot. Let time be T.

Each x_i may have the following requirements:

1. Quantity of q_i products for delivery
2. Time u_i for which the vehicle is required to deliver the goods
3. Priority δ_i

Vehicle v_j has the following characteristics:

1. Maximum cargo capacity O_j
2. Work period from the moment T_j^s to the moment T_j^k
3. Fixed price C_k

We will assume that the price of the shortest path between customers x_i and x_j with c_{ij} and the corresponding time t_{ij} is given.

Goal:

Design least cost routes for each vehicle so that:

1. Each customer is visited only once by exactly one vehicle
2. Each route starts and ends at a warehouse
3. Satisfying additional constraints

3 Vehicle Routing Problem With Time Windows

VRPTW is defined by a set of vehicles V and a set of customers C, $|C| = n$.
In this paper, the set of vehicles is treated as homogeneous - the vehicles are identical with the given capacity q. Each customer i is characterized by the quantity of requested products d_i) as well as the time interval $[a_i, b_i]$ of goods receipt.
The vehicle must visit the customer before b_i, and in case of arrival before a_i it must wait.

Each route starts and ends at the same depot, specifically at node 0 and node $n + 1$ which are equivalent, and its cost is equivalent to the time to go around it with additional delays accounted for in each node. The time to get from customer i to customer j, in the label t_{ij} is equivalent to the Euclidean distance between those two nodes - which is in accordance with Solomon's data set that was used in this paper, we will introduce the label c_{ij} as the cost of the path between the nodes i and j.

3.1 Mathematical model

Let's define the variable x_{ijk} for a pair of nodes i and j, where $i \neq j$, $i \neq n+1$, $j \neq 0$, and vehicle k as:

$$x_{ijk} = \begin{cases} 1, & \text{if the vehicle } k \text{ goes directly from node } i \text{ to } j, \\ 0, & \text{otherwise} \end{cases}$$

The variable s_{ik} is defined for each vehicle k and each node c_i as the time when the vehicle k started servicing the customer at the node i. In case the vehicle k does not go around the node i then this value is irrelevant.

$$\min \sum_{k \in V} \sum_{i \in C} \sum_{j \in C} c_{ij} x_{ijk} \tag{1}$$

$$\sum_{k \in V} \sum_{j \in C} x_{ijk} = 1 \qquad \forall i \in C \tag{2}$$

$$\sum_{i \in C} d_i \sum_{j \in C} x_{ijk} \leq q \qquad \forall k \in V \tag{3}$$

$$\sum_{j \in C} x_{0jk} = 1 \qquad \forall k \in V \tag{4}$$

$$\sum_{i \in C} x_{ihk} - \sum_{j \in C} x_{hjk} = 0 \qquad \forall h \in C, \forall k \in V \tag{5}$$

$$\sum_{i \in C} x_{i(n+1)k} = 1 \qquad \forall k \in V \tag{6}$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \qquad \forall i, j \in C, \forall k \in V \tag{7}$$

$$a_i \leq s_{ik} \leq b_i \qquad \forall i \in C, \forall k \in V \tag{8}$$

The objective function (1) minimizes the total cost of the trip. Constraint (2) ensures that each node is visited exactly once and (3) indicates that the load of each vehicle does not exceed the maximum capacity q. Furthermore, (4), (5) and (6) ensure that each route starts and ends at a depot and after unloading goods at one node, the vehicle must proceed to the next. Inequality (7) tells about the order of route visits, and (8) confirms compliance with time window constraints. Note that a vehicle that is unused can be modeled by route (0, n+1).
An upper limit on the number of vehicles can also be introduced:

$$\sum_{k \in V} \sum_{j \in C} x_{0jk} \leq |V| \tag{9}$$

4 Genetic algorithm

The principles of genetic algorithms (GA) are already well known. A population of solutions is maintained, on the basis of which the individuals that will leave offspring are selected using selection operations. The offspring have the characteristics of both parents. The fitness function corresponds to the objective function, which in this case is the total time to travel all routes. Analogous to biological processes, individuals that are better adapted have a greater chance of surviving and leaving offspring, with the aim of finding individuals with a better value of the fitness function through the imitation of evolution.

Algorithm 1 Genetic algorithm

```
Create initial population
Evaluate population
while Termination criterion is not met do
    Select good individuals for reproduction
    Perform crossover operation on said individuals
    Perform mutation operation on children individuals with probability P_m
    Evaluate new population
end while
```

4.1 Chromosome representation

The basic structure of a route can be constructed as a list of integer values that starts and ends with zero (depot) and in between nodes that are visited in a given order. The structures of all routes can be concatenated to obtain the final structure, and the zeros can be removed to increase readability.
Vehicle markings can be inserted as delimiters between individual routes.
Namely, if we label the nodes representing customers with values from 1 to n, and at the same time we have m vehicles at our disposal, we will label the vehicles with values from n+1 to n+m, with the fact that the vehicle label for the last route can be omitted.

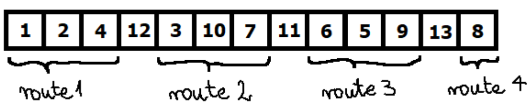

Figure 1: Chromosome representation

4.2 Initial population

The first step of the genetic algorithm is the generation of an initial population of admissible solutions. A modified nonarrest neighbor heuristic with a randomization parameter was used, which enables the creation of random admissible solutions.

Algorithm 2 Chromosome initialization

```
Create remaining customers list from available customers
routes = []
last visited = Depot
current route = []
while Remaining customers list is not empty do
    Create list of feasible customers and sort it according to distance from last visited customer
    if List of feasible customers is empty then
        last visited = Depot
        Append current route to routes list
        current route = []
        continue
    end if
    if random[0, 1] ≤ randomizing parameter then
        Choose a random feasible customer and add it to current route
        Remove said customer from feasible customers list
        Update last visited customer
    else
        Choose a nearest customer from feasible customer list and add it to current route
        Remove said customer from feasible customers list
        Update last visited customer
    end if
end while
```

4.3 Selection operators

The selection operator selects individuals from the population that will leave offspring. We will list the algorithms that were used.

Random selection represents the selection of a random individual from the population.

Tournament selection selects a sample from the population, and then the individual with the best value of the fitness function is selected from that sample.

Roulette selection involves calculating the proportion of fitness values for each individual, and then selecting an individual based on those proportions.

Rank selection is similar to roulette, but instead of basing selection probability on fitness values, it is based on the ranking of individuals.

4.4 Crossover operators

Order Crossover (OX) is specifically designed for use over a list without duplicates. First, two random positions are selected from the parent and the corresponding subfile is copied to the child, between the selected positions. After that, passing through the second parent, we copy the corresponding elements to the child, in order, if they are not already present.

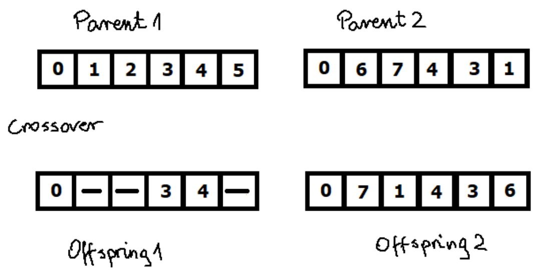

Figure 2: Order Crossover

Partially Mapped Crossover (PMX) children are originally deep copies of their parents. Two positions are randomly selected and the corresponding submissions are exchanged. Elements at the appropriate positions are mapped, removing cycles and transitivity in mappings. For example, the mappings $1 \rightarrow 2$ and $2 \rightarrow 1$ are ignored, and $1 \rightarrow 2$ and $2 \rightarrow 3$ are treated as $1 \rightarrow 3$. After normalizing the mappings, the elements from the children are exchanged according to the mappings. For example, if $\mathrm{a} \rightarrow \mathrm{b}$, element a from the first child is replaced by element b from the second child, outside the selected subarray.

Best Route Better Adjustment (BRBAX) copies half of the best routes from the first parent to one child and fills in the remaining routes from the second parent, keeping the vehicle positions fixed within the child.

4.5 Mutation operators

A swap mutation randomly selects two positions in an individual and replaces the elements at those positions.

An inversion mutation randomly selects a subarray within an individual and changes the order of elements in that subarray.

A shaking mutation randomly selects a subarray within an individual and shuffles the order of elements in that subarray.

4.6 Results

By combining different operators, over Solomon's R101 data set, it was found that the best value of the fitness function is given by the following operators:

$$\begin{aligned} \text{Selection operator} &= \text{Random Selection} \\ \text{Crossover operator} &= \text{Best Route Better Adjustment} \\ \text{Mutation operator} &= \text{Swap Mutation} \end{aligned}$$

For this combination of operators, the best values of the parameters of the genetic algorithm were found:

$$\begin{aligned} \text{Population size} &= 700 \\ \text{Number of generations} &= 35 \\ \text{Elitism size} &= 70 \end{aligned}$$

For these parameters, the best value of the normalized fitness function obtained in relation to the length of individual routes is 541,695.

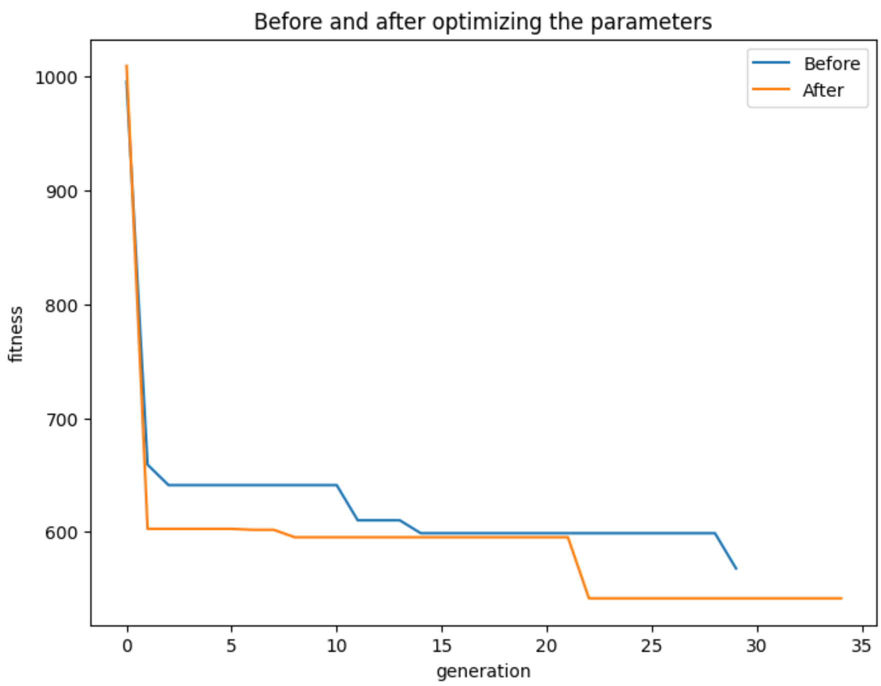

Figure 3: Fitness of the best individual through generations

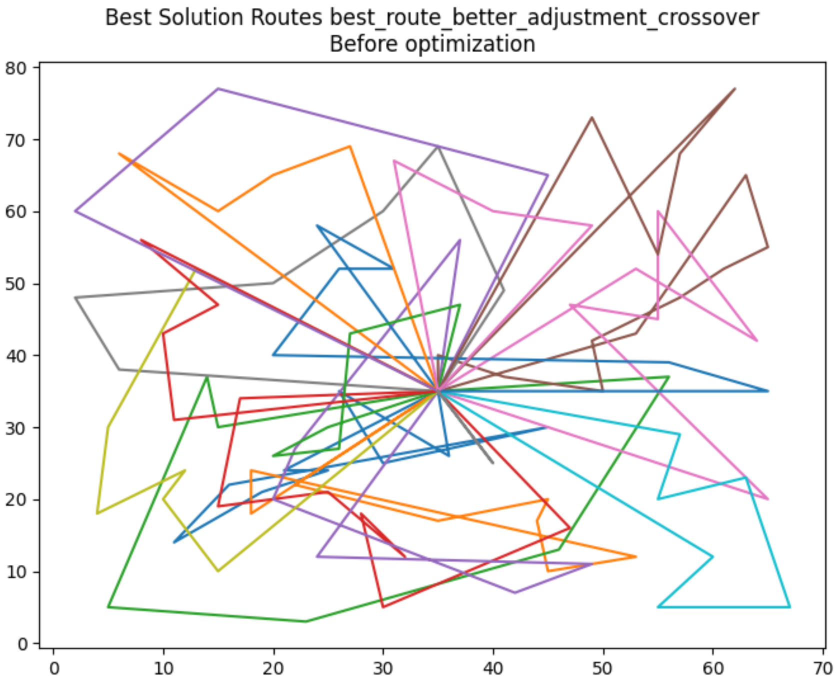

Figure 4: Before optimization

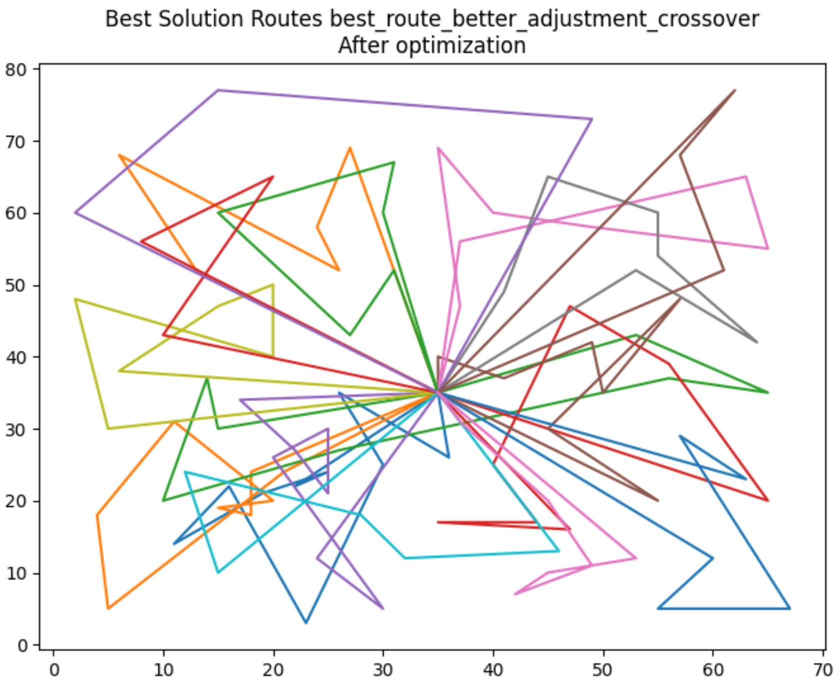

Figure 5: After optimization

5 References

Wikipedia
Vehicle Routing Problem

Lin Cheng, USA 2005.
A GENETIC ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Anagnostopoulou Afroditi, Maria Boile, Sotirios Theofanis, Eleftherios Sdoukopoulos, Dimitrios Margaritis, Spain 2014.
Electric Vehicle Routing Problem with Industry Constraints: Trends and Insights for Future Research

Sam R. Thangiah, USA 1995.
Vehicle Routing with Time Windows using Genetic Algorithms

Brian Kallehauge, Jesper Larsen, Oli B. G. Madsen, Denmark 2000.
Lagrangean Duality Applied on Vehicle Routing with Time Windows

Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds) Column Generation. Springer, USA 2005.
A Primer in Column Generation

Joanna Ochelska-Mierzejewska, Aneta Poniszewska-Maranda, Witold Maranda, Poland 2021.
Selected Genetic Algorithms for Vehicle Routing Problem Solving

Marius M. Solomon, USA 2005.
VRPTW Benchmark Problems

Hari Kurnia, Elyza Gustri Wahyuni, Elang Cergas Pembrani, Syifa Tri Gardini, Silfa Kurnia Aditya, Indonesia 2018.
Vehicle Routing Problem Using Genetic Algorithm with Multi Compartment on Vegetable Distribution