Study and Implementation of Sweep Heuristic for Capacitated Vehicle Routing Problem

Team Name: #No Name

Team Members:

1) Neeraj Naik (22m1508)
2) Vishal (22m1515)
3) Kalal Karthik (22m1529)

Abstract

Vehicle Routing Problem aims to find an optimal schedule for a fleet of vehicles to deliver goods to customers. In order to solve the Vehicle Routing Problem, we propose two variants alongside the sweep heuristic which comes under the family of cluster-first, route-second approaches and Clarke and Wright savings heuristic. For cluster-first, route-second algorithms the customers are divided into subsets in the first stage so that each set has the maximum number of customers that can be served within a tour. In the following step, all assigned clusters' optimal tours are calculated. The two proposed variants differ in the clustering phase. For routing phase MTZ formulation is used for all the three methods. We compare the performance of the savings and sweep heuristics and two distance based variants of sweep that are considered in this paper.

1. Introduction

Solving a Vehicle Routing Problem with large number of customers to optimality takes a large amount of time. In practice various heuristics are adopted to solve the problem within a reasonable time and produce delivery schedules. One such heuristic approach which comes under cluster-first route-second method is Sweep Algorithm. The depot is positioned in the middle of the plane, like the hands of a clock. Then, a clock hand moves across the plane. If a feasible insertion is possible, the hand angle advances while a customer is added to the current cluster; otherwise, a new cluster is initiated from the current angle of the hand.

2. Objective

- To study and implement Sweep Heuristic to solve the CVRP.
- To minimize the number of vehicles used and the distance travelled by it in each subset.

3. Problem Definition

- Here we are considering an instance of dataset which contains 250 demand points with origin as depot in x-y plane.
- Demand at each location is available.
- The capacity of each vehicle was fixed and equal to 50 units.
- We must come up with optimal routes for meeting demand requirements with the capacity of vehicle being a constraint.

4. Sweep Strategy

The sweep algorithm is based on the polar coordinate angles of the customers C, with the depot d located in the grid's centre and the angle component of a customer a denoted by θ(a) ∈ [0,2π]. We choose the zero angle, or the starting angle, so that it divides the two adjacent customers with the maximum angle gap, i.e., $max_{a,b \in C} \quad \theta(a) - \theta(b)$.

Sweep heuristic is of type cluster first route second. We suggest the following general approach, containing following steps to find answers for a specific VRP instance:

1. Use one of the techniques to get a feasible clustering of C.
2. Compute the optimal route of each cluster.

Clustering:

We apply the following variants of clustering algorithms in Step 1.

- Classic Sweep (Based on polar coordinate angles)
- Variants based on distance.
    - Farthest first then Nearest Cluster Algorithm
    - Nearest first then Nearest Cluster Algorithm

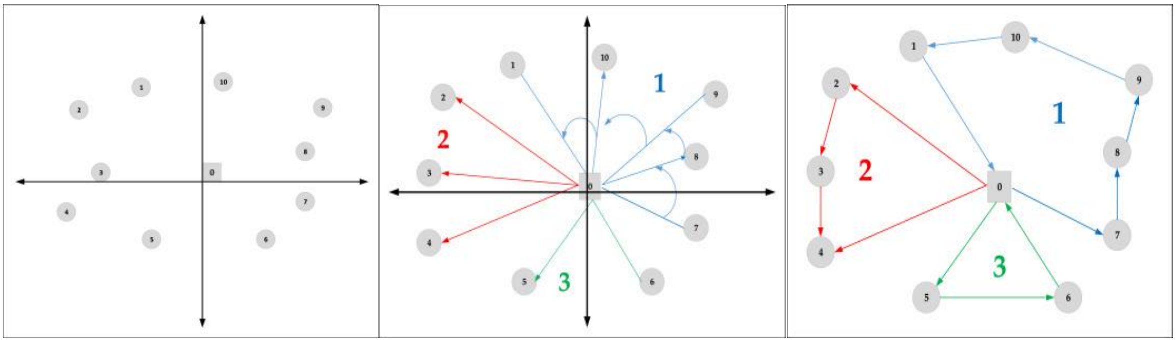
Routing:

In Step 2, a tour for each cluster is obtained by solving the TSP-MILP. MTZ formulation is being used to find the optimal routes of the clusters.

4.1 Classic Sweep

Customers are assumed as points in a plane with Euclidean distances to depot located at center of the grid (Origin) as cost. The polar coordinates of each customer with respect to depot are computed and the customers are sorted in increasing polar angle. Starting angle (Customer) is chosen such that it splits the two neighbouring customers with the largest angle gap.

1. Start with an empty cluster.
2. Assign customers to clusters from the sorted set until it is capacity feasible.
3. Otherwise initialize new cluster and start assigning customers.
4. Repeat step 2 & 3 until all customers are assigned to a cluster.
5. After getting feasible clusters, compute the optimal route for each cluster.



The above figure describes a toy example with depot located at origin and 10 customers to be served. Based on capacity feasibility 3 clusters were formed as shown and then optimal route is found by solving travelling salesman problem for each of the cluster.

4.2 Farthest First then Nearest Cluster

1. Starting customer is chosen such that it is farthest from depot and customers that are closest to this farthest customer are added to cluster until the vehicle is capacity feasible.
2. Repeat the procedure until all customers are assigned to a cluster.
3. After getting feasible clusters, compute the optimal route for each cluster.

4.3 Nearest First then Nearest Cluster

1. Starting customer is chosen such that it is nearest from depot and customers that are closest to this nearest customer are added to cluster until the vehicle is capacity feasible.
2. Repeat the procedure until all customers are assigned to a cluster.
3. After getting feasible clusters, compute the optimal route for each cluster.

5. MTZ Formulation

The Miller-Tucker-Zemlin (MTZ) formulation of the TSP is used to find the optimal routes of the clusters and is described below:

Label the cities with 1, …, n and define the decision variable:

Decision Variables:

$x_{i,j} = 1$ if path goes from city i to city j

$x_{i,j} = 0$ otherwise

For $i = 1 \ldots n$, let u_i be the variable which is the position of city i in the tour.

Then TSP can be written as the following integer linear programming problem:

$$min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}$$

$$x_{ij} \in \{0,1\} \quad i, j = 1, \ldots n;$$

$$u_i \in Z \quad i = 1, \ldots n;$$

$$\sum_{i=1, j \neq i}^{n} x_{ij} = 1 \quad j = 1, \ldots \ldots n;$$

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \quad i = 1, \ldots \ldots n;$$

$$u_i - u_j + n x_{ij} \leq n - 1 \quad 2 \leq i \neq j \leq n;$$

$$0 \leq u_i \leq n - 1 \quad 2 \leq i \leq n;$$

Each city must be reached from exactly one other city in order to satisfy the first set of equalities, and each city must depart from exactly one other city in order to satisfy the second set of equalities. The final restrictions require that there be only one trip that visits every city, rather than two or more separate trips.

6. Clarke and Wright Savings Heuristic

Clarke and Wright Savings heuristic often called as classic vehicle routing problem is also considered. This algorithm's foundation is a notion of so-called savings. The distance between each pair of to-be-visited cities is listed in the distance matrix. If the transportation costs between every pair of cities are known, the cost can be used in place of distance. Euclidean distance is used to calculate the distance dij on a grid between the points i with coordinates (xi, yi) and j with coordinates (xj, yj):

1. One consumer is served by each of the n vehicles as a starting solution.

2. Use the formula $S_{i,j} = C_{o,i} + C_{o,j} - C_{i,j}$ to determine the savings for linking the cycles using edge $[i, j]$ for all node pairings (i, j) $, i \ldots j$.

3. Arrange the savings in descending order.

4. Remove the top savings item, edge $[i, j]$. Join two distinct cycles together along the edge $[i, j]$ if

   a) The nodes are members of distinct cycles, and
   b) The vehicle's carrying capacity is not exceeded.
   c) The first and last customers on their respective cycles are i and j.

Repeat step 4 as necessary to finish off the savings list or until the capacities are full.

7. Dataset

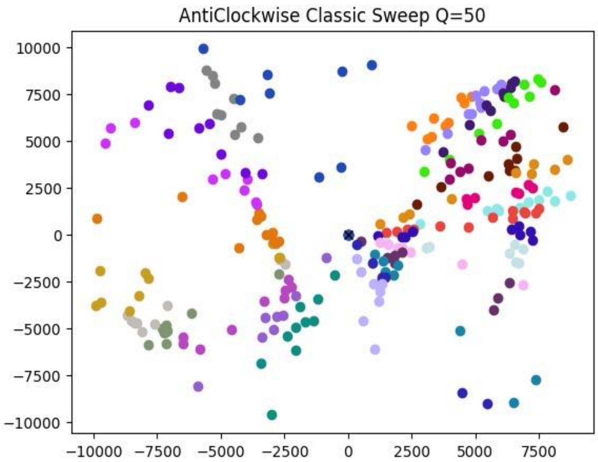
The same dataset is being used for the study, which was mentioned in the paper we referred, and a brief description of the dataset is given below.

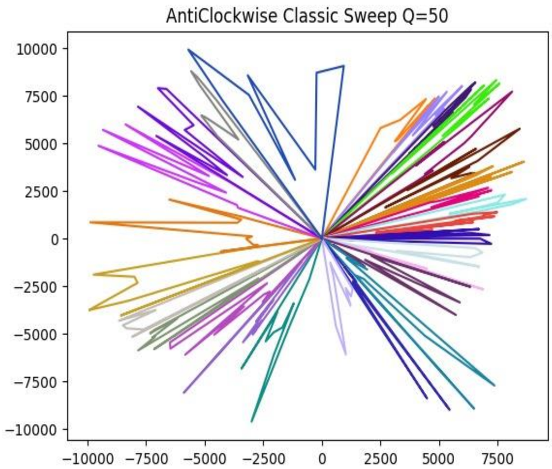
A grid of 20 km x 20 km squares, or about the same area as Vienna, Austria, is used to distribute the customers. Additionally, 80% of the customers are distributed among randomly chosen clusters and only 20% are uniformly distributed on the grid. The distance between two consumers is calculated using the Euclidean distance.

The instance is available at http://dx.doi.org/10.13140/RG.2.2.20934.60480

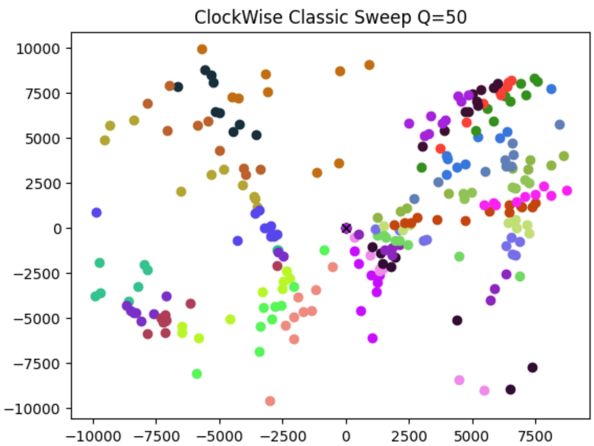
8. Results

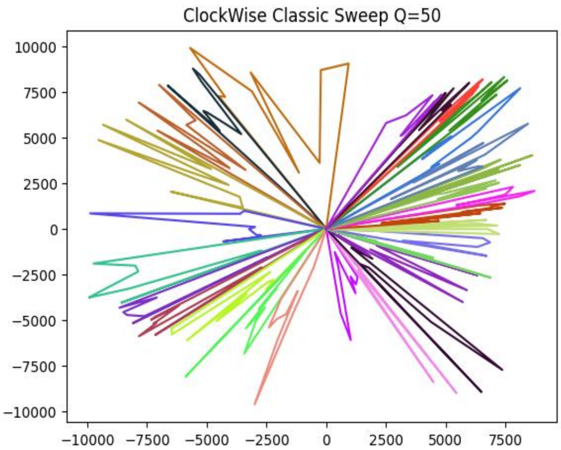
Anticlockwise Sweep



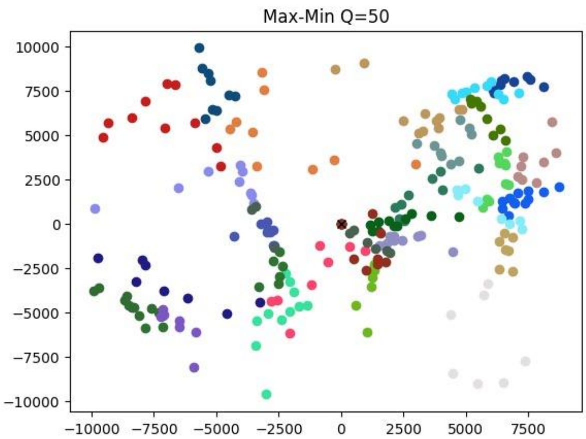


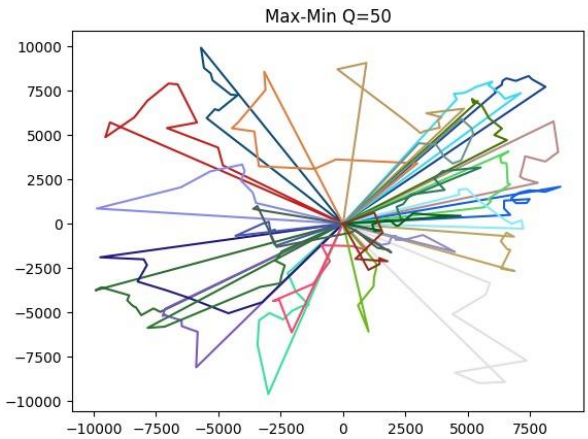
Clockwise Sweep



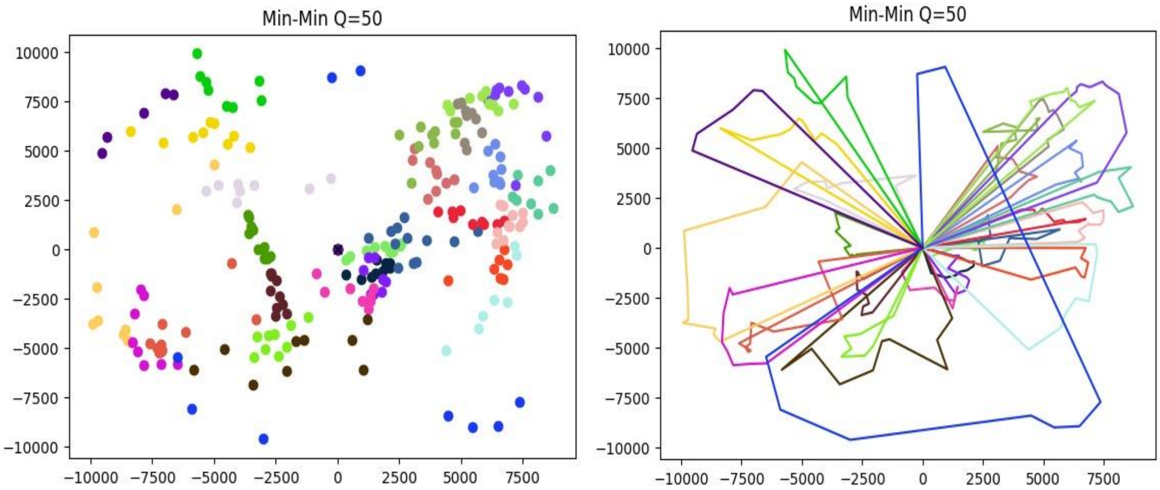


Farthest First then Nearest Cluster

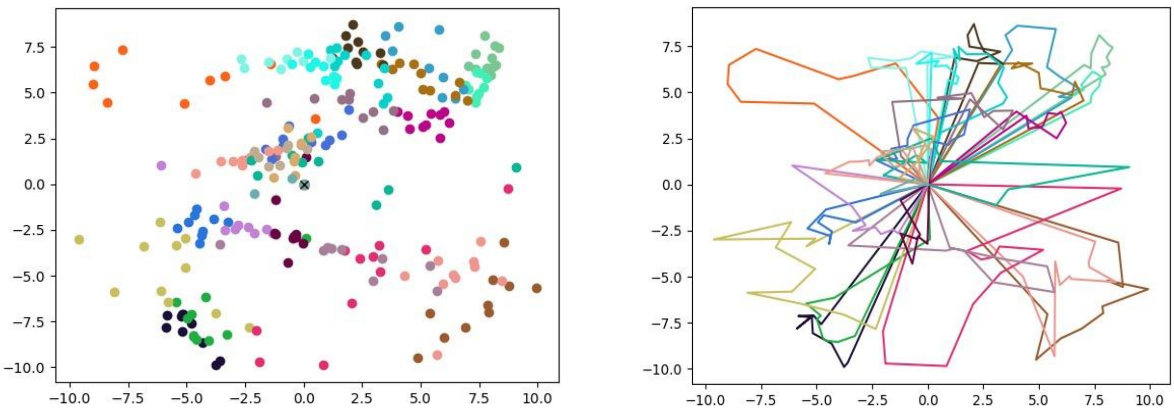




Nearest First then Nearest Cluster

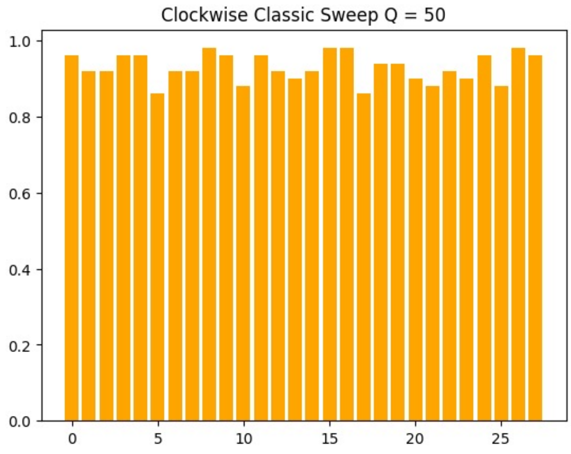


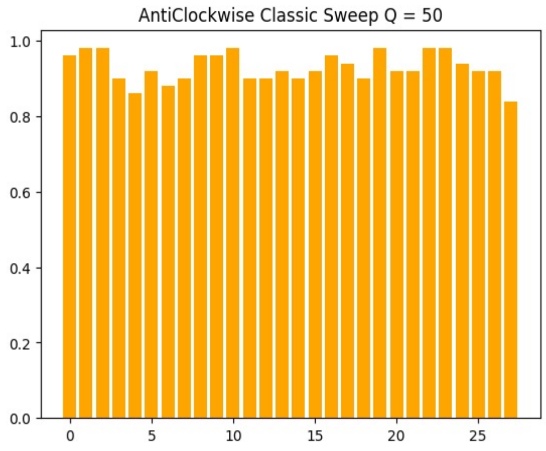
Clarke and Wright Savings

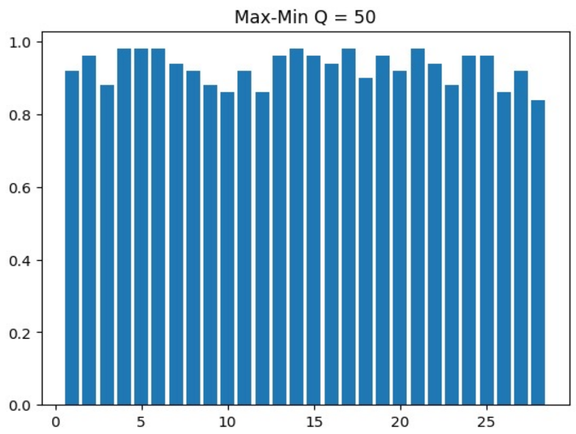


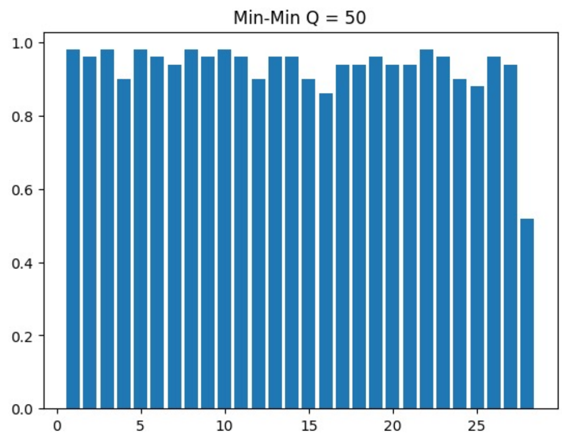
The left side figure in all algorithms shows the feasible cluster sets, points with identical colours represent a cluster set. Feasible routes for these cluster sets are shown in right side figure.

Utilization









Algorithm	Clusters	Cost (Km)	Average Utilization	Time(s)
Anticlockwise Sweep	28	553.80	0.9292	220.24
Clockwise Sweep	28	555.58	0.9292	1084.92
FFNC Algorithm	28	527.58	0.9292	34.33
NFNC Algorithm	28	517.01	0.9292	56.79
CW Savings	26	580.70	0.9596	---

Time in the above table considers only routing step as clustering step takes very few seconds. We can infer from the table that distance-based variants FFNC and NFNC algorithms take less than a minute time to compute the routes for the clusters.

The vehicles required as per CW savings algorithm is 26 whereas for sweep it is 28.

9. Conclusions

In this paper we considered Clarke and Wright savings heuristic and sweep heuristic cluster-first, route-second methods. For the clustering step along with classic sweep two other variants based on the distance from depot i.e., FFNC and NFNC are used to find the feasible cluster sets. For the routing step MTZ formulation is adopted to find the optimal routes of the clusters.

The experimental result on the dataset shows that the Nearest First Nearest Cluster algorithm has least cost compared to other heuristics and CR savings algorithm requires less number of vehicles as compared to other.

References

1. Armbrust, P., Maier, K., Truden, C. (2022). Sweep Algorithms for the Vehicle Routing Problem with Time Windows. In: Dorronsoro, B., Pavone, M., Nakib, A., Talbi, EG. (eds) Optimization and Learning. OLA 2022. Communications in Computer and Information Science, vol 1684. Springer, Cham. https://doi.org/10.1007/978-3-031-22039-5_11
2. Gillett, B.E., Miller, L.R.: A heuristic algorithm for the vehicle-dispatch problem. Oper. Res. **22**(2), 340–349 (1974)

Details of data files and code files

- "cVRPsTW250_1.csv" contains dataset used in this paper.
- IE716_sweep_hue_clustering.ipynb contains python code for clustering phase for classic sweep (Anticlockwise and Clockwise directions).
- CVRPTW_rand_dist.ipynb contains python code for clustering phase for distance based variants FFNC and NFNC.
- IE716_sweep_routing.ipynb contains python code for routing phase for all the four algorithms.
- IE716_savings.ipynp contains python code for Clarke and Write savings algorithm.