Capacitated Vehicle Routing Problem with precedence constraints and void volume penalty

Introduction

The Capacitated Vehicle Routing Problem (CVRP) is a fundamental logistics challenge focused on optimizing the delivery of goods or services to customers using a fleet of vehicles while considering constraints such as vehicle capacity and minimizing transportation costs. This report addresses a variant of the CVRP that includes precedence constraints and uses the Gurobi optimization tool to find an optimal solution.

In transportation and logistics, CVRP plays a crucial role in determining efficient routes for vehicles to deliver goods or services to customers. The variant considered here introduces precedence constraints that dictate the sequence in which specific customers should be visited. This additional complexity aims to model real-world scenarios where certain customers must be serviced before others due to time constraints, service requirements, or dependencies between customer orders.

Problem Statement

The objective of the CVRP variant with precedence constraints is to minimize the overall cost associated with routing vehicles while considering:

Distance Costs: Minimizing the total distance traveled by the vehicles, which directly affects transportation expenses.

Penalties for Void Volumes: Introducing penalties for orders with specific shapes, such as spheres, that might contain void space, impacting the overall vehicle load and, consequently, transportation efficiency.

Formulation

Sets and Indices:

- $N = \{1, 2, \ldots, n\}$: Set of clients, excluding the depot.
- $V = \{0\} \cup N$: Set of all nodes, including the depot (0) and clients.
- $A = \{(i, j) \mid i, j \in V, i \neq j\}$: Set of arcs between nodes.

Parameters:

- xc_i, yc_i: x and y coordinates of node i for $i \in V$.
- c_{ij}: Cost (distance) of traveling from node i to node j for $i, j \in V$.
- Q: Vehicle capacity.
- v_{vehicle}: Volume capacity of the vehicle.
- v_{orders}: Volume of orders for each customer node.
- q_i: Demand (quantity) for each customer node.
- $shapes_i$: Indicator for the shape of the container for each customer node.
- *Preceding_constraints:* Order of visiting clients

Variables:

- x_{ij}: Binary decision variable that equals 1 if the arc from node i to node j is traversed, 0 otherwise.
- u_i: Continuous decision variable representing the cumulative load at node i.
- v_i: Continuous decision variable representing the volume at node i.

- **Objective Function:**

  Minimize the total cost incorporating distances traveled and penalties for void volumes:

  $$\text{Minimize: } \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} + \text{penalty\_coefficient} \cdot \sum_{(i,j)\in A, j\neq 0, \text{shapes}_j=1} \text{void\_volume}_j \cdot x_{ij}$$

Constraints

- **Node Visitation:**
  - Ensure each node (excluding the depot) is visited exactly once:
    $\sum_{j \in V, j \neq i} x_{ij} = 1, \forall i \in N$
  - Leave each node (excluding the depot) exactly once:
    $\sum_{i \in V, i \neq j} x_{ij} = 1, \forall j \in N$

- **Flow Conservation:**
  - Maintain flow balance through each node:
    $\text{if } x_{ij} = 1, \text{ then } u_i + q_j = u_j, \forall (i, j) \in A, i \neq 0, j \neq 0$

- **Capacity Constraints:**
  - Ensure the cumulative load at each node doesn't exceed the vehicle capacity:
    $u_i \geq q_i, \forall i \in N$
  - Limit the cumulative load at each node by the vehicle's capacity:
    $u_i \leq Q, \forall i \in N$

- **Void Volume and Shape Constraints:**
  - Ensure the vehicle's volume capacity is not exceeded when carrying specific shapes:

$$\forall (i, j) \in A, i \neq 0, j \neq 0 \colon v[i] + v\_orders[j] + void\_volume[j] \leq v\_vehicle$$

$$\forall i \in N \colon v[i] \geq 0$$

- **Precedence Constraints:**
  - Sequentially visit clients based on precedence constraints (if provided):
    $x_{ij} = 1 \Rightarrow u_i \leq u_j, \text{for specified precedence constraints}$

Results and discussion

This optimization problem involves determining the most cost-effective routes for vehicles considering distances, penalties for void volumes, and fulfilling precedence constraints. The Gurobi optimization solver is utilized to find the most efficient routing plan that satisfies these constraints and minimizes the total cost incurred.

The problem parameters were set as follows:

Vehicle Capacity
Q=20 (units)

Volume of Vehicle:
v_vehicle=30 (units)

Volume of Orders:
v_orders= {1: 8, 2: 8, 3: 8, 4: 12, 5: 5, 6: 6, 7: 14, 8: 14, 9: 5, 10: 9} (Randomly generated for each customer)

Shapes of Orders:
shapes= {1: 1, 2: 1, 3: 0, 4: 1, 5: 0, 6: 1, 7: 0, 8: 0, 9: 0, 10: 0} (Randomly generated for each customer)

Demand quantity of each customer:
q={1: 5, 2: 6, 3: 6, 4: 7, 5: 9, 6: 5, 7: 2, 8: 5, 9: 9, 10: 2} (Randomly generated for each customer)

Precedence_constraints = [1,7,9]

Penalty coefficient = 2

X coordinates:
xc=[109.76270079, 143.03787327, 120.55267521, 108.9766366 , 84.73095987, 129.17882261, 87.51744225, 178.35460016, 192.7325521 , 76.68830377, 158.34500762]

Y coordinates:
yc=[52.88949198, 56.80445611, 92.55966383, 7.10360582, 8.71292997, 2.02183974, 83.26198455, 77.81567509, 87.00121482, 97.86183422, 79.91585642]

(xc[0],yc[0])=>coordinates of the depot

Calculations:

Void volumes computed for orders:
{1: 7.278874536821949, 2: 7.278874536821949, 3: 0, 4: 10.918311805232925, 5: 0, 6: 5.459155902616461, 7: 0, 8: 0, 9: 0, 10: 0}

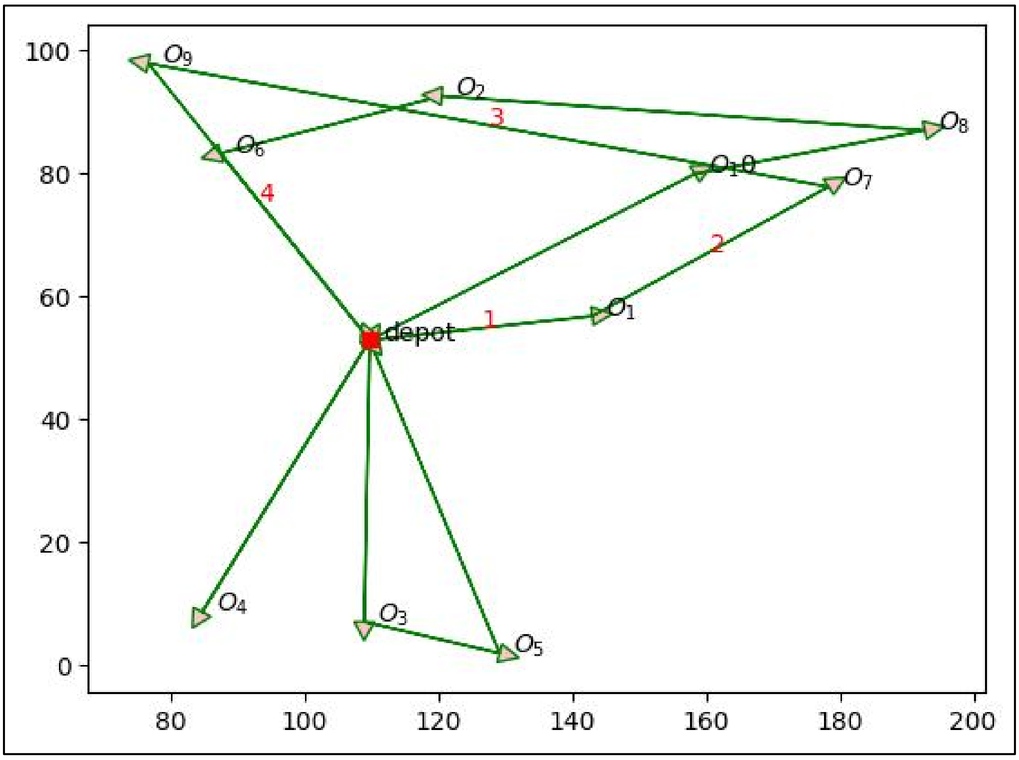
Optimisation results:

```
Set parameter MIPGap to value 0.1
Set parameter TimeLimit to value 500
Gurobi Optimizer version 11.0.0 build v11.0.0rc2 (linux64 - "Ubuntu 22.04.3 LTS")

CPU model: Intel(R) Xeon(R) CPU @ 2.20GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 1 physical cores, 2 logical processors, using up to 2 threads

Optimize a model with 144 rows, 140 columns and 326 nonzeros
Model fingerprint: 0xe66594e1
Model has 90 general constraints
Variable types: 30 continuous, 110 integer (110 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e+01, 2e+02]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 2e+01]
  GenCon rhs range [2e+00, 9e+00]
  GenCon coe range [1e+00, 1e+00]
Presolve removed 31 rows and 71 columns
Presolve time: 0.01s
Presolved: 113 rows, 69 columns, 399 nonzeros
Variable types: 25 continuous, 44 integer (44 binary)
Found heuristic solution: objective 1046.0713689
Found heuristic solution: objective 935.8790771
Found heuristic solution: objective 929.3619554

Root relaxation: objective 6.003191e+02, 34 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  600.31913    0   18  929.36196  600.31913  35.4%     -    0s
H    0     0                     909.1937744  600.31913  34.0%     -    0s
H    0     0                     812.0304608  600.31913  26.1%     -    0s
H    0     0                     803.7630967  600.31913  25.3%     -    0s
H    0     0                     794.0904209  600.31913  24.4%     -    0s
     0     0  647.48624    0   20  794.09042  647.48624  18.5%     -    0s
H    0     0                     753.6042091  666.53003  11.6%     -    0s
     0     0  666.53003    0   20  753.60421  666.53003  11.6%     -    0s
     0     0  666.53003    0   21  753.60421  666.53003  11.6%     -    0s
     0     0  671.90601    0   16  753.60421  671.90601  10.8%     -    0s

Cutting planes:
  Learned: 4
  Cover: 1
  Implied bound: 2
  Relax-and-lift: 3

Explored 1 nodes (92 simplex iterations) in 0.10 seconds (0.01 work units)
Thread count was 2 (of 2 available processors)

Solution count 8: 753.604 794.09 803.763 ... 1046.07

Optimal solution found (tolerance 1.00e-01)
Best objective 7.536042091300e+02, best bound 7.033147438748e+02, gap 6.6732%
```

Active arcs: where x(i,j)=1

```
active_arcs
```

```
[(0, 1),
 (0, 3),
 (0, 4),
 (0, 10),
 (1, 7),
 (2, 6),
 (3, 5),
 (4, 0),
 (5, 0),
 (6, 0),
 (7, 9),
 (8, 2),
 (9, 0),
 (10, 8)]
```



The precedence constraints were followed in the optimized routing plan. Specifically, the customers were visited in the sequence prescribed by the precedence constraints [1,7,9].