Vehicle Routing Problem

Advanced Algorithms, Spring 2020

Due Tuesday April 17th @ 1:30pm EST

1 Introduction

Vehicle routing problems are the class of problems related to routing a fleet of vehicles to visit a set of clients. In the Capacitated Vehicle Routing Problem (CVRP) the goal is to optimally route a fleet of vehicles to deliver a set of packages to clients subject to vehicle capacity. In this case, Emily's furniture business has built a lot of furniture and is ready to deliver it to their clients. In the CVRP, we are given an origin o, a destination d, and a set of clients $N = 1, 2, ..., n$ each with demand q_i (for simplicity we set $q_o = q_d = 0$). We represent this set using a complete, directed graph $G = (V, A)$ where $V = N \cup \{o, d\}$ (aka all the nodes including the destination and the source) and let the arc cost $c_{(i,j)}$ be the travel cost from i to j for all $(i, j) \in A$. Emily has K vehicles that can deliver furniture to clients. All the vehicles need to start at the origin and end at the destination. Also, Emily is well known for providing the best quality of service to her clients. Therefore, she will need to make sure that each client is visited by at least one vehicle while working hard to minimize her company's total travel costs. Further, each vehicle has a fixed capacity Q; if a vehicle visits the subset of clients, $S \subseteq N$, then we require $\sum_{i \in S} q_i <= Q$ (the sum of demands from all the places the vehicle visits must be less than or equal to the capacity of that vehicle) so that Emily can deliver all the requested furniture to her clients.

Summary of variables in the problem:

$$\text{origin} = o$$

$$\text{destination} = d$$

$$\text{clients} = N = \{1, 2, ...n\}$$

$$\text{number of vehicles} = K$$

$$\text{capacity of each vehicle} = Q$$

$$\text{demand for each client (i)} = q_i$$

$$\text{graph} = G = (V, A)$$

$$\text{all nodes} = V = N \cup \{o, d\}$$

$$\text{travel cost from i to j (for all (i,j) in A)} = c_{(i,j)}$$

2 The Problem

Part A

Below is an integer program for the CVRP, where $\delta^+(B)$ is the set of arcs (i, j) such that i is not in B and j is in B where B is an arbitrary subset of nodes. Similarly, $\delta^-(B)$ is the set of arcs (i, j) such that i is in B and j is not in B. x_a is a variable that represents the number of routes that use arc $a \in A$. u_i is a variable representing the cumulative quantity of goods delivered between the origin o and node i along the chosen route.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{(i,j)} x_{(i,j)} \\
\text{s.t.} \quad & \sum_{a\in\delta^-(\{o\})} x_a <= K \\
& \sum_{a\in\delta^+(\{d\})} x_a <= K \\
& \sum_{a\in\delta^+(\{d\})} x_a = \sum_{a\in\delta^-(\{o\})} x_a \\
& \sum_{a\in\delta^+(\{i\})} x_a = 1 \qquad \forall i \in N \\
& \sum_{a\in\delta^-(\{i\})} x_a = 1 \qquad \forall i \in N \\
& u_i - u_j + Q x_{(i,j)} <= Q - q_j \qquad \forall (i,j) \in A \\
& x_a >= 0, \text{ integer} \qquad \forall a \in A \\
& q_i <= u_i <= Q \qquad \forall i \in V
\end{aligned}
$$

Explain why this is a valid integer program for the problem. That is, show that any feasible solution to the IP corresponds to a valid solution to the CVRP of equal cost and vice versa.

Part B

As you may remember from your lab 0 assignment, picos is a Python package that can solve linear programs. Guess what? It can solve integer programs too! Your task is to use picos to write a function that takes in an n x n matrix of edge costs, an integer number of vehicles K, a vector of n demands, and a capacity Q and returns the total travel cost of the solution to the above IP. (Hint: We recommend using either the gurobi solver or the cplex solver which can be installed with 'pip install cplex'). To solve this problem, you should fill in the function called cvrp_ip in solver.py. Your solution should return the minimum objective value and the x matrix (which gives values for the number of routes that use each of the arcs in the graph) in the form (objective value, x matrix). It should pass all of the provided test cases which can be found in vrp_tests.py.

Part C (OPTIONAL/BONUS/EXTRA SPECIAL - but still fun!)

Instead of solving CVRP optimally using integer programming, we can also develop a heuristic algorithm to find a good solution. Design a local search algorithm that iteratively tries to improve the current solution using local moves (e.g., one move might be removing a client from a route and reinserting it at the optimal place among all K routes). Compare the algorithm's performance to that of the IP. Is there a way to improve the performance of the heuristic? If so, what is the trade-off in run time? (Note that given the optional nature of this problem, we did not write any test cases or anything, but it should be fairly easy for you to test it on your own if you are so inclined)

References

[1] Alice Paul and Susan Martonosi. Operations Research, in Nathan Carter (ed.), Data Science for Mathematicians, 2020.

[2] Toth, Paolo Vigo, Daniele. (2014). Vehicle Routing - Problems, Methods, and Applications (2nd Edition) - 1.2.1 Problem Statement. Society for Industrial and Applied Mathematics. Retrieved from https://app.knovel.com/hotlink/pdf/id:kt0119ZQS1/vehicle-routing-problems/problem-statement

[3] "Vehicle Routing - Advanced Topics: Part I." Coursera, www.coursera.org/learn/discrete-optimization/lecture/QB8JE/vehicle-routing.