CS202 Design and Analysis of Algorithms, Spring 2025
Project Handout
version 0.7

Code Due: Sunday, 6 April, 2025, 11:59PM
Slides Due: Monday, 7 April, 2025, 8AM
Report Due: Sunday, 13 April, 2025, 1PM

1 Introduction

Fig. 1: Image of an autonomous vehicle

Singapore's government is trying to deploy autonomous vehicles (AEs) for public transit and other purposes.[1] The design of the AEs' routes is important and challenging. Your task is to design the algorithms to generate the routes given the different instances, which is called Vehicle Route Problem (VRP).

The Capacitated Vehicle Routing Problem (CVRP) is a fundamental challenge in logistics and operations research. It involves determining the optimal set of routes for a fleet of vehicles, each with a limited carrying capacity, to deliver goods from a central depot to a set of customers with known demands. The objective is to minimize the total travel distance while ensuring that each customer's demand is met, and no vehicle exceeds its capacity.

2 Objective

In this project, you will develop an algorithm to solve the Capacitated Vehicle Routing Problem (CVRP).

Input to the CVRP consists of n locations (a depot and a set of n - 1 customers), an n × n symmetric matrix D

[1] https://www.channelnewsasia.com/singapore/self-driving-public-bus-191-400-autonomous-vehicles-lta-trial-4898326

specifying the distance (or some other cost) to travel between each pair of locations, a quantity qi that specifies the demand for some resource by each customer i, and the maximum quantity, Q, of the resource that a vehicle can carry. A feasible solution to the CVRP consists of a set of routes that begin and end at the depot, such that each customer is visited on exactly one route and the total demand by the customers assigned to a route does not exceed the vehicle capacity Q. An optimal solution for CVRP is a feasible solution that minimizes the total combined distance of the routes. It is assumed that there are no restrictions to the number of routes in a solution. [2]

In short, your solution should do the following:

1. Generate a set of routes starting and ending at the depot.
2. Ensure each customer is visited exactly once.
3. Ensure the total demand on each route does not exceed the vehicle's capacity.
4. Minimize the total travel distance across all routes.

2.1 Inputs and Outputs

Input: the distance matrix D (node 0 is the depot), the demand vector q, the capacity of vehicle Q
Output: a list of routes where each route is a sequence of the nodes (both starting and ending at the depot node)

All inputs are read from `stdin` and outputs should be written to `stdout`.

2.1.1 Sample Input

```
10
75
0 12 22 15 30 45 50 55 60 70
12 0 14 18 25 40 47 52 58 67
22 14 0 10 20 35 42 47 55 63
15 18 10 0 17 32 38 44 50 60
30 25 20 17 0 20 28 35 42 50
45 40 35 32 20 0 15 22 30 38
50 47 42 38 28 15 0 12 20 28
55 52 47 44 35 22 12 0 10 18
60 58 55 50 42 30 20 10 0 15
70 67 63 60 50 38 28 18 15 0
0 10 15 12 8 10 15 18 20 22
```

Explanation:

- The first line contains an integer indicating the total number of locations, including the depot. In this example, N = 10
- The second line contains Q indicating the capacity of each vehicle. Here, Q = 75.
- Then follows a distance matrix D, where node 0 is the depot.
- Finally, we have the demand vector q, indicating the capacity demanded at each location.

[2] More information on the CVRP can be found at https://dmac.rutgers.edu/programs/challenge/vrp/cvrp/

2.1.2 Sample Output

```
0 1 2 3 4 5 6 0
0 7 8 9 0
```

Explanation: Each line indicates a single route, which starts and ends at the depot (node 0).

3 Code Template

A code template is provided on eLearn, in the file `vrp.py`.

4 Constraints and Limits

Inputs: $0 < N \leq 1000$, $0 < q[i] \leq 50$, $0 < Q \leq 100$
Resources: 1 minute per test case, 1024Mb memory
Submission Limits: No more than 10 submissions per day per team. **Teams that exceed this limit will be penalized.**

5 Online Judge

We will be using the Online Judge to automatically evaluate your solutions. The Online Judge for the Project can be found at `https://10.0.104.119/d/project_sp25/`. Note that this is different from the link for submitting Assignments. You will soon be receiving login information for the Project Online Judge via email - this account will be shared among members of your team.

Once you submit your code to the Online Judge, it will be run automatically against our hidden test cases, and you will receive a score denoting your total distance travelled across all test cases, and the total runtime across all test cases. In this project, we want to minimize distance travelled, so the lower the score, the better.

There is a project leaderboard covering all teams across 4 sections; the total distance and runtime from the latest submission of each team will be displayed. Due to system limitations, the distance is sorted from high to low, but do remember that the lower the distance, the better!

Please take note of the daily submission limit - your team should make no more than 10 submissions per calendar day to the Online Judge.

6 Project Deliverables

6.1 Source Code

You should submit well-documented code implementing your CVRP solution to the Online Judge.

6.2 Report

A concise report (3-5 pages) detailing:

- Your approach and methodology.
- Challenges encountered and how they were addressed.
- Performance analysis, including computational efficiency and complexity analysis.

6.3 Presentation

A brief presentation (15 minutes) summarizing your findings and demonstrating your solution. This is also an opportunity to show how your solution stands out.

7 Evaluation

We define a metric that balances solution quality (total route distance) and computational efficiency (runtime), rewarding solutions that achieve an optimal tradeoff between minimizing distance and execution speed.

7.1 Scoring Metric

Given a student's solution S, we define the score as:

$$score(S) = 0.6 \times Q_{\text{norm}} + (1 - 0.6) \times T_{\text{norm}} \tag{1}$$

where:

- Q_{norm} is the **normalized solution quality score** (scaled based on total distance).
- T_{norm} is the **normalized runtime score** (scaled based on execution time).

7.1.1 Solution Quality Score (Q_{norm})

We define Q_{norm} relative to the best student solution:

$$Q_{\text{norm}} = \frac{D_{\text{best}}}{D_{\text{student}}} \tag{2}$$

where:

- D_{student} = total route distance of the student's solution.
- D_{best} = total route distance of the best student solution.

Range: $0 < Q_{\text{norm}} \leq 1$ (higher is better).
Note: on the Online Judge, total distance is summed across all test cases.

7.1.2 Runtime Score (T_{norm})

To normalize runtime, we define:

$$T_{\text{norm}} = 1 - \frac{T_{\text{student}} - T_{\text{min}}}{T_{\text{max}} - T_{\text{min}}} \tag{3}$$

where:

- T_{student} = runtime of the student's algorithm, in milliseconds.
- T_{min} = fastest runtime among all submissions, in milliseconds.
- T_{max} = slowest runtime among all submissions, in milliseconds.

Range: $0 \leq T_{\text{norm}} \leq 1$ (higher is better).
Note: on the Online Judge, runtime is summed across all test cases.

8 Grading

Your project is graded on the following components:

- **Solution Quality (80%)**
  The quality of your solution. Your solution will score a 0 if the problem's constraints are not met.
- **Report and Presentation (20%)**
  Clarity, completeness, and professionalism in conveying your approach and results.

9 Suggestions

Here are some suggestions for what you can try to begin with, just to get the ball rolling.

- Start early! Leave yourself plenty of time to work on the problem.
- Start simple. Begin with a basic feasible solution (e.g. assigning each customer to a separate route), and iteratively improve upon it.
- Explore the use of heuristics and optimization techniques to generate and refine solutions.