Huawei Delivery Optimisation Competition

Department name: Ireland Research Center - CT Lab

Author's name:

Date: December 2021

Overview

1. You are in charge of Huawei’s delivery service that takes care of deliveries of consumables from a depot to customers.

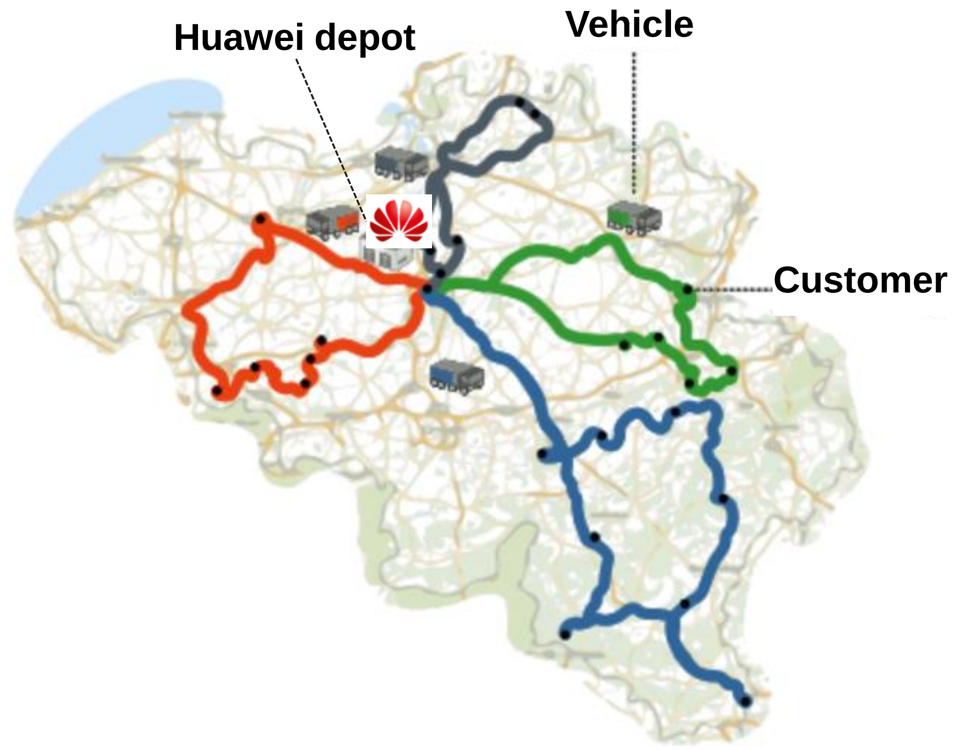
2. The depot has a given set of home **vehicles**, each with a certain **capacity** for carrying consumables.

3. **Customers** are represented as **nodes in a graph**, in which edges have an associated **transportation cost** equal to the distance between the nodes.

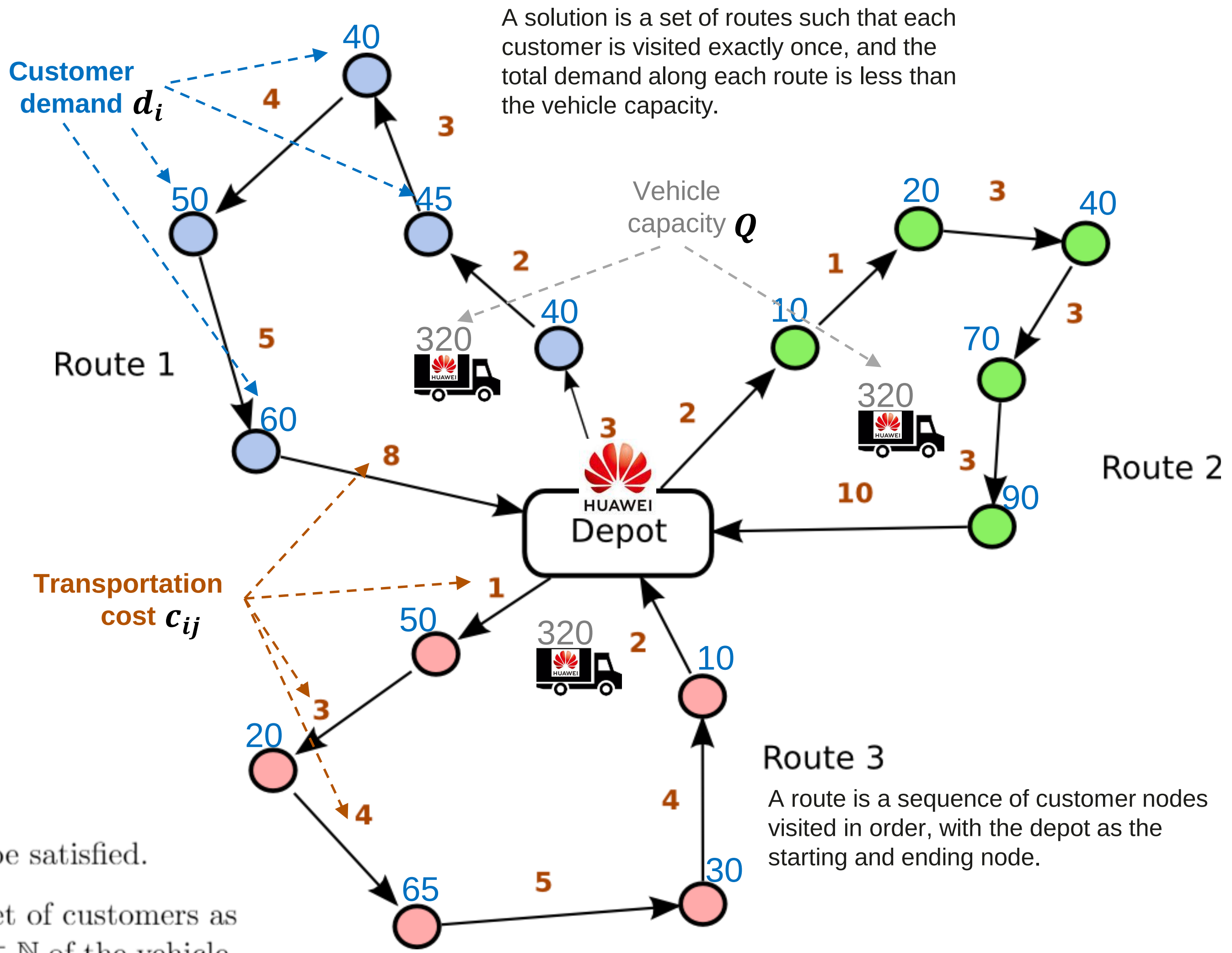
4. Each customer node has an associated **demand** for a consumable.

5. You need to **minimise the total transportation cost** by determining a set of routes (one route for each vehicle that must start and finish at the depot) such that each customer is visited exactly once, and the total demand along each route is less than the vehicle capacity.

Huawei Consumables Delivery Optimisation



Technical description



- There is one depot and a set of N customers.
- There is a fleet of M vehicles.
- Depot is represented by node 1, and has a demand of 0.
- Each customer node i, $i \in \{2, \dots, N+1\}$ has a demand $d_i \in \mathbb{N}$ to be satisfied.
- A vehicle, which always starts and ends at the depot, can serve a set of customers as long as the total customer demand does not exceed the capacity $Q \in \mathbb{N}$ of the vehicle. All vehicles have the same capacity.
- The transportation cost $c_{i,j}$ is the cost of a vehicle travelling from node i to j.
- The objective is to find a routing plan with minimal transportation cost for each of the M vehicles, which starts and finishes at the depot node, such that each customer is visited exactly once and all of customer demands are served without violating vehicle capacity constraints.

Datasets

The competition is based on six instances of the Huawei Consumables Delivery Optimisation problem of increasing difficulty. Problem specifications are given in the following table.

	number of customers	number of vehicles	vehicle capacity
problem 1	3	1	30
problem 2	21	1	210
problem 3	44	4	2,010
problem 4	71	4	30,000
problem 5	134	7	2,210
problem 6	1,000	43	131

Data for each problem $x \in \{1, 2, 3, 4, 5, 6\}$ is contained in a separate folder `problem_x` with:

1. `spec_x.txt` with problem information about number of customers, number of vehicles, and capacity of each vehicle.
2. `distance_x.txt` containing the symmetric distance matrix C of size $N + 1 \times N + 1$, where N is the number of customers, and $c_{i,j} \in \mathbb{R}$ is the distance between customer node i and j, where $i, j \in \{1, \ldots, N + 1\}$. The depot is represented by node 1.
3. `demand_x.txt` containing for each node $i \in \{1, \ldots, N + 1\}$ the demand $d_i \in \mathbb{N}_0$ for the node. Node 1 represents the depot, and has a demand of 0.

The data directory structure is shown on the right.

Data directory structure

```
data folder
├─ problem_1 folder
│  ├─ spec_1.txt
│  ├─ distance_1.txt
│  └─ demand_1.txt
├─ problem_2 folder
│  ├─ spec_2.txt
│  ├─ distance_2.txt
│  └─ demand_2.txt
├─ problem_3 folder
│  └─ ...
├─ problem_4 folder
│  └─ ...
├─ problem_5 folder
│  └─ ...
└─ problem_6 folder
   └─ ...
```

Solution

An example problem

Input: problem_x data

1) Problem spec (spec_x.txt)

N, M, Q

Vehicle capacity

Number of vehicles

Number of customers

2) Distance matrix (distance_x.txt)

Depot node — Customer nodes

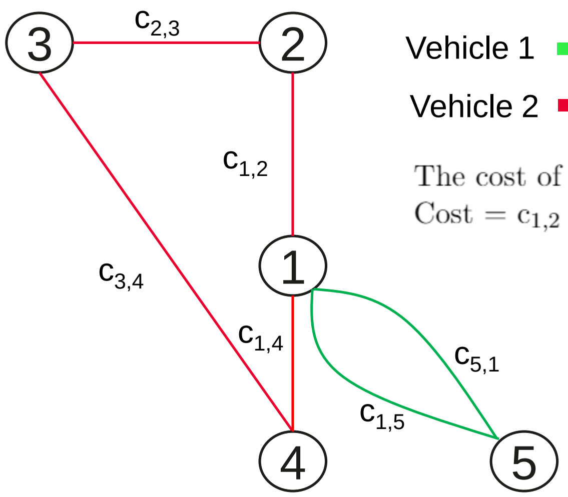
	Node 1	Node 2	Node 3	Node 4	Node 5
Node 1	0	$c_{1,2}$	$c_{1,3}$	$c_{1,4}$	$c_{1,5}$
Node 2	$c_{2,1}$	0	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$
Node 3	$c_{3,1}$	$c_{3,2}$	0	$c_{3,4}$	$c_{3,5}$
Node 4	$c_{4,1}$	$c_{4,2}$	$c_{4,3}$	0	$c_{4,5}$
Node 5	$c_{5,1}$	$c_{5,2}$	$c_{5,3}$	$c_{5,4}$	0

3) Demands (demand_x.txt)

node id, demand

1, 0
2, d_2
3, d_3
4, d_4
5, d_5

Output: solution_x.txt



Vehicle 1 (route 1)

Vehicle 2 (route 2)

The cost of the solution is the sum of distances travelled by each of the vehicles:
Cost $= c_{1,2} + c_{2,3} + c_{3,4} + c_{1,4} + c_{1,5} + c_{5,1}$

Deliverables:
For each problem $x \in \{1, 2, 3, 4, 5, 6\}$ submit a `solution_x.txt` file.
The format of the submission file is the following:

- Each line outputs the route for each of the vehicles as a comma-separated sequence of customer nodes visited in order, with the depot (node 1) as the starting and ending node. The example `solution_x.txt` file for `problem_x` is given bellow.

solution_x.txt

```
1,2,3,4,1
1,5,1
```

*Each route starts and ends with the depot node

Solution evaluation function

- We provide a Python implementation of a solution evaluation function (`Evaluation.py`) for problem x that takes four input files as parameters:
  - `solution_x.txt`, the solution file as described in previous slide.
  - `spec_x.txt`, the problem specification file.
  - `distance_x.txt`, the distance file.
  - `demand_x.txt`, the demand file.
- The evaluation function performs a number of checks on the syntactic validity of the `solution_x.txt` file as follows:
  1. The number of lines must be equal to the number of vehicles in the problem specification.
  2. Each line represents a route for a vehicle and needs to consist of a comma separated list of integers.
  3. Each route needs to start and end at the depot node 1.
  4. Each customer node must be served by one of the vehicles.
  5. Each customer node must be visited exactly once.
  6. Customer node ids in all routes must be compatible with the number of customers in the problem specification.
  7. Each route must satisfy vehicle capacity constraints.
- Once a solution passes all syntax validity checks, the cost of the solution is evaluated as the sum of the distances travelled by each vehicle in each of the routes as shown in the previous slide.

HUAWEI

Computing total team scores and deciding finalists (I)

For computing the team score, each problem $x \in \{1,2,3,4,5,6\}$ is assigned a weighting coefficient $w_x, \in \mathbb{R}$, which reflects the problem difficulty. The following table presents the weighting coefficients for the six problems:

	Problem 1	Problem 2	Problem 3	Problem 4	Problem 5	Problem 6
Weighting coefficient w_x	1.0	2.0	3.0	4.0	5.0	6.0

For each problem $x \in \{1,2,3,4,5,6\}$ each participant Team $i \in \{1,\ldots,T\}$ submits a solution file, and the associated cost $(cost_{i,x})$ is automatically computed. The score $(score_{i,x})$ for team i on problem x is then calculated as follows:

$$\text{score}_{i,x} = w_x * \frac{min_{1 \leq i \leq T} cost_{i,x}}{cost_{i,x}}$$

The team with the lowest cost for a particular problem attains a maximum score of w_x, while other teams with higher solution cost attain scores in $[0, w_x)$. If a team do not submit a solution for problem x then their score for this problem is 0.

Deciding on finalists:

- **Finalists will be determined based on total score across the six problems.**
- **Ties shall be broken by the jury panel based on presentations during the closing ceremony.**

Computing total team scores and deciding finalists (II)

An example of calculating scores is given bellow. For demonstration purposes, we are assuming two participating teams and three problems with weighting coefficients 1.0, 2.0 and 3.0 respectively. The submitted solution cost table is as follows:

Team	Solution 1 cost	Solution 2 cost	Solution 3 cost
team 1	31	43	21
team 2	45	24	*not submitted*

The score for each problem and its cumulative over all problems is computed as described in the previous slide, and is summarised in the following table. The winning team is team 1.

Team	Solution 1 score	Solution 2 score	Solution 3 score	Total score
team 1	$1.0 * \frac{31}{31} = 1.0$	$2.0 * \frac{24}{43} = 1.1$	$3.0 * \frac{21}{21} = 3.0$	$\boxed{1.0+1.1+3.0=5.1}$
team 2	$1.0 * \frac{31}{45} = 0.68$	$2.0 * \frac{24}{24} = 2.0$	0	$0.68+2.0+0.0 = 2.68$

Winning team with highest total score

HUAWEI

Timeline

ONCE YOU COMMENCE,
THERE'S NO GOING BACK!

6th November, 2021 at 9 AM!
Registrations open

4th December, 2021 at 9 AM!
Dataset Release

11th December, 2021 at 5 PM!
Knowledge Sharing

19th December, 2021 at 12 AM!
Final Submission

20th December, 2021 at 5 PM!
Closing Ceremony

Submission

The Isograd platform will be used to make submissions, the link is as follows:

https://www.isograd-testingservices.com/huaweiire/contestlogin

Submission deadline is on the 18th December at 23:59 GMT.

Submission items:

- Solution files (solution_x.txt) for each of the six problems.
- Zip file named “teamname.zip” containing:
    (i) the source code.
    (ii) slides describing the solution method and the main modules of the source code.

Good luck!

HUAWEI

Thank you.