# Decision-making in Transport and Mobility (1CM110)

# ASSIGNMENT 3

## Deadline: January 19th, 13:00

Dr. Rolf van Lieshout, Dr. Bart van Rossum, Thao Nguyen

1. **Read the entire assignment carefully before you start.** Familiarize yourself with all tasks and questions to estimate the required effort and plan your work effectively.

2. **Work collaboratively within your group.** Do not divide tasks such that individual members are responsible only for specific parts (e.g., one person writing the report and another coding). Every group member should contribute to all components and be able to explain every part of the code and analysis.

3. **Collaboration across groups is strictly prohibited.** Any indication of inter-group cooperation or sharing of material will result in penalties for all involved parties.

4. **Use AI tools responsibly.** You may use artificial intelligence tools (such as ChatGPT or Copilot) to support your learning process—e.g., for debugging, clarifying concepts, or improving readability—but not to generate complete solutions or analyses. You must always understand and be able to explain any content produced with AI assistance. If AI tools were used, briefly describe their role in your workload report

5. **Verify your results.** Test your implementations on small, self-constructed instances to confirm correctness. It is good practice to write auxiliary verification code to check the validity of your solutions—do not rely solely on solver outputs.

6. **Ensure model linearity.** All Gurobi models must be linear. Note that Gurobi does not automatically detect non-linear formulations; you are responsible for ensuring this requirement is met.

7. **Write clean and well-documented code.** Your code should be easy to read and understand. Points will be deducted if your implementation or logic is unclear.

8. **Seek clarification when needed.** Use the Canvas discussion board and Q&A sessions to ask questions or request clarifications about the assignment.

Goals of this assignment:

1. Deepen your understanding of the theoretical concepts discussed in the course by applying them to realistic routing and facility location problems.
2. Develop your proficiency in Python and enhance your ability to formulate and solve small-scale optimization models using the commercial solver Gurobi.
3. Improve your ability to interpret solver output, analyze model performance, and derive meaningful managerial insights from quantitative results.
4. Practice presenting and discussing actionable recommendations based on sensitivity analyses and modeling outcomes.

---

# Introduction

This assignment consists of two parts. In the first part of the assignment, we will study the concepts of fairness and robustness in the context of a capacitated vehicle routing application. In the second part of the assignment, we will explore both cooperative and non-cooperative optimization models within the context of a charging facility location problem. You may impose a computing time limit of 10 minutes for all optimization models in this assignment.

# 1 Fair and Robust Vehicle Routing

We have extensively studied the classical capacitated vehicle routing problem (CVRP), where the goal is to find a minimum-cost set of routes that cover all customer demand. Now, we investigate how we can incorporate fairness and robustness into this traditional routing problem. To do so, we consider a small problem instance featuring 25 customers and 5 vehicles.

1.1. **Range Minimization** (30 points)

We start by incorporating fairness into the CVRP. In particular, we will try to optimise the range fairness measure, and aim to construct a set of routes for which the difference between the longest and shortest route is small. In other words, both the cost and payoff of a route are equal to its distance.

In all the questions of part 1.1, we will work with route-based formulations. In particular, you can assume that the full set of feasible routes $R$ is given by the routes in `routes.txt`. Each line in this file corresponds to a possible route, in the following format: `distance load depot firstCustomer secondCustomer ... lastCustomer depot`.

(a) (5 points) Propose, implement, and solve a route-based formulation of the CVRP. Report the running time, final optimality gap, and the costs and range of the best solution.

(b) (5 points) Implement the vehicle-index model for range minimization as discussed in the lecture. Set $\epsilon = 0.05$ and $z^*$ equal to the costs you found in (a). In other words, the total route cost may be at most 105% of that of your solution to (a). Report the same metrics as before, and discuss all differences you find with (a).

*Hint: Compute the optimal solution to the linear programming relaxation of the vehicle-index formulation. What do you observe?*

(c) (5 points) There exists an alternative formulation for the range minimization problem. For each route $r \in R$ and customer $i \in N \setminus \{0\}$, define the binary parameter $b_{ir}$ to equal one if customer $i$ is the last customer on route $r$ and zero otherwise. Moreover, let $M$ be an upper bound on the distance of any route. The last-customer formulation for the range minimization problem then reads:

$$
\begin{aligned}
\min \quad & \eta - \gamma && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{r \in R} a_{ir} x_r = 1 && \forall i \in N \setminus \{0\} && \text{(1b)}\\
& \sum_{r \in R} c_r x_r \leq (1+\epsilon) z^* && && \text{(1c)}\\
& \sum_{r \in R} x_r = |K| && && \text{(1d)}\\
& \sum_{r \in R} p_r b_{ir} x_r \leq \eta && \forall i \in N \setminus \{0\} && \text{(1e)}\\
& M\left(1 - \sum_{r \in R} b_{ir} x_r\right) + \sum_{r \in R} p_r b_{ir} x_r \geq \gamma && \forall i \in N \setminus \{0\} && \text{(1f)}\\
& x_r \in \{0, 1\} && \forall r \in R. && \text{(1g)}
\end{aligned}
$$

Explain why the last-customer formulation is a valid formulation for the range minimization problem. In particular, explain why constraints (1e) and (1f) correctly model the maximum distance $\eta$ and minimum distance $\gamma$.

(d) (5 points) Provide two reasons why, from a computational point of view, this formulation might be preferred over the vehicle-index formulation.

(e) (5 points) Implement the last-customer formulation. Then, solve it for all $\epsilon \in \{0.01, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15\}$. For each run, report the same metrics as in (a), and discuss your results.

(f) (5 points) Suppose now that we must solve a vehicle routing problem for each day of the week. Our goal is to assign the daily routes to drivers in a way that is fair over time, i.e., that minimizes the range of the total weekly distance driven by each driver. Which of the above models would be most appropriate for this purpose? How would you tackle this problem in practice?

### 1.2. Uncertain Demand (35 points)

For the second part of this question, we will discard fairness, but instead drop the assumption of deterministic customer demand. In particular, we will assume that, for each customer $i \in N \setminus \{0\}$, the true demand is uniformly distributed in $[0.9 \cdot q_i, 1.1 \cdot q_i]$, i.e., the true demand can deviate at most 10% from the nominal demand. The true demand is only revealed once the vehicle arrives at the corresponding customer; at the time of planning the routes, only the nominal demand is known.

In all questions of part 1.2, we will work with two-index formulations for the CVRP (as compared to the route-based formulation used in part 1.1). To this end, you should read the parameters of the CVRP instance in the file `instance.txt`. The first line of the file specifies the vehicle capacity $Q$. The second line contains $n$ values, where the $i$-th value represents the demand of customer $i$. The remaining $n+1$ lines define the distance matrix. The first of these lines gives the distances from the depot to all nodes, the second gives the distances from customer 1 to all nodes, and so on, with the final line giving the distances from customer $n$ to all nodes.

(a) (5 points) Implement and solve the classical two-index formulation to obtain a 'cost-efficient' solution. You can re-use your code from Assignment 1. Report the computing time, optimality gap, and costs.

(b) (10 points) We perform a simulation to assess the robustness of the cost-efficient solution against uncertain demand. In particular, we analyse the number of routes featuring a capacity violation, i.e., for which the sum of true customer demands exceeds the vehicle's capacity.

Repeat the following procedure for $k = 1,000$ iterations:

1. For each customer $i$, draw (integer) demand $\tilde{q}_i \in [0.9 \cdot q_i, 1.1 \cdot q_i]$.
2. For each selected route $r$, compute the capacity violation as $\max\{0, \sum_{i \in r} \tilde{q}_i - Q\}$
3. Compute the total capacity violation by summing capacity violation over all routes

Report the total number of capacity violations, average capacity violation, and maximum capacity violation across all simulated demand realisations.

(c) (10 points) Implement the scenario-based two-index formulation of the CVRP as discussed in the lecture.

Then, perform five iterations of the following cutting-plane algorithm.

1. Solve the scenario-based model with scenario set $S$.
2. Simulate $k = 1,000$ demand realisations as in (b).
3. Add to $S$ the demand realisation featuring the highest total capacity violation (summed over all vehicles), if any.

You should initialise the scenario set $S$ to with only the deterministic demand $q$.

For each iteration of this algorithm, report the computing time, optimality gap, routing costs, simulated robustness of the incumbent solution as in (a), and the customer demands of the most violating scenario (if any). Discuss your results.

(d) (5 points) Propose a simple recourse policy that (i) ensures that all true customer demand is met, (ii) visits the customers in the same order as originally planned, and (iii) only makes use of information that is available to the vehicle, i.e., does not use the true demand of customers that have not yet been visited.

Choose a demand scenario and route with capacity violation, and illustrate your policy by applying it to this route.

(e) (5 points) Simulate the recourse policy $k = 1,000$ times for each of the five solutions encountered in the algorithm of (c), and report (i) average total routing costs before recourse and (ii) average total routing costs after performing recourse actions. Discuss your results.

# 2 Charging Facility Location

We now study cooperative and non-cooperative behaviour in the context of the charging facility location problem as discussed in the lecture. There are $|K|$ origin-destination pairs, representing volumes of travelers that wish to travel from their origin to their destination. All travelers have an electric vehicle with a fixed driving range of $R$. In other words, no traveler can travel a distance beyond $R$ without recharging somewhere. The goal of the charging facility location problem is to determine the lowest amount of charging facilities to open that allow all travelers to reach their destination without running out of charge.

## 2.1. Cost Sharing (20 points)

(a) (2 points) The formulation for the charging facility location problem as discussed in the lecture makes use of an implicit network $A$, that only contains arcs between nodes that can be reached without running out of range. Explain how one can efficiently compute the implicit network $A$ based on the road network and range $R$.

For the remainder of this exercise, you can use the implicit network given in the file `network.txt`, containing a single line for every arc in the implicit network $A$. Every line has the format `origin destination distance`.

(b) (3 points) Suppose that each charging location has a finite capacity $Q$, i.e., it can accommodate a passenger volume of at most $Q$. Extend the model from the lecture to incorporate this capacity restriction. In the remainder of this exercise, you should use $Q = 10$.

(c) (5 points) The companies `Amplify`, `Booster`, and `Charge`, abbreviated $A$, $B$, and $C$, respectively, all wish to build charging facilities for their customers. The origin-destination pairs of each company are given in the file `pairsX.txt`, where $X \in \{A, B, C\}$. Every line in these files has the format `volume origin destination`.

Implement the capacitated model from (b) and solve it for all three companies. For every company, report the computing time, optimal number of charging facilities, and optimal charging locations.

(d) (10 points) The companies realise that they can save costs by jointly building charging facility locations.

Use the Shapley value to propose a distribution of charging facility costs over the three companies. Report the optimal costs for each subset of companies, and the cost allocation according to the Shapley value. How much does each company save in this collaboration, compared to building facilities of their own?

## 2.2. Selfish Routing (15 points)

(a) (10 points) The three companies have assumed complete control over the routes that travelers take when building the charging stations. However, all travelers act in their self-interest and wish to reach their destination as fast as possible. Hence, it is safe to assume that all travelers traveling along the shortest, charge-feasible path from their origin to their destination.

Given the charging stations located in (d), describe and implement a method that computes the selfish routings of all travelers. Report the total travel time in the system-optimal solution and the total travel time under selfish routing. What is the effect that selfish routing has on the charging capacity constraint at each station? Report any capacity violations that occur.

(b) (5 points) To avoid capacity violations in the future, the companies wish to use an optimization model that finds cost-optimal charging locations while taking into account selfish routing behavioir. Propose and describe a method that the companies can use for this purpose. This can be a cutting-plane type model, something based on the lecture slides, or anything you come up with yourself. You do not have to actually implement the proposed method.

# 3 Workload reporting

### 3.1. Workload reporting (1 point)

For a bonus point, please truthfully report

(a) who did which work,

(b) how much work it was in total (in detail),

(c) whether and how you used AI.

# Reporting

Hand in the assignment via Canvas. Upload a **.pdf** file containing the complete report with answers to all questions (no MS Word documents). Furthermore, hand in a .zip file containing your full Python code. Make sure that your report (PDF) is not inside the zip archive and uploaded separately! Include assignment number, group number, and your last names in the filenames, e.g., `Assignment3_Group3_vanLieshout_vanRossum_Nguyen.pdf` and `Assignment3_Group3_vanLieshout_vanRossum_Nguyen.zip`. All your Python scripts must be executable irrespective of your computer, and must produce clear output. No Jupyter files are allowed. The front page of your report must state your group number as well as the name and student ID of every group member. The final report cannot exceed 8 pages (including results tables and figures, but excluding the front page).