Decision-making in Transport and Logistics (1CM110)

ASSIGNMENT 1

Deadline: Nov 27th, 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 a realistic routing problem.

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

Heat It B.V. is a company that specializes in the transportation and installation of heat pumps in residential homes. The company operates from a central depot, from which a fleet of vehicles departs each day to deliver and install heat pumps at customer locations. Each customer is assigned a specific time window during which the driver must arrive, as well as a service time that represents the expected duration of the installation process. In addition, each customer request includes the volume of the heat pump to be delivered.

To ensure efficient operations and high service quality, Heat It B.V. seeks to optimize its daily routing schedule. The goal is to minimize overall operational costs, primarily by reducing total driving distance, while adhering to delivery time windows, service durations, and vehicle capacity constraints.

For a single day of operation, the file `customers.xlsx` contains detailed information for each customer, including their location (in kilometer coordinates), demand volume, assigned time window (in minutes from the start of the planning horizon), and service time (in minutes). All customers are to be served from a central depot located at $(40, 50)$, using vehicles with a maximum capacity of 200 units. Assume Euclidean distances (rounded to the closest integer) and a constant driving speed of 60 km/h, implying that traveling 1 kilometer takes 1 minute.

1 Formulating and Solving the Basic Routing Problem

In the first part of the assignment, we start with a simplified version of the problem. We temporarily ignore vehicle capacity, time windows, and service times, and aim only to minimize the total distance traveled.

1.1. TSP Formulations (20 points)

In the lecture, we discussed two different formulations for the TSP, the *Miller-Tucker-Zemlin* formulation (MTZ) and the *Dantzig-Fulkerson-Johnson* formulation (DFJ). Both have their advantages and disadvantages.

(a) (4 points) Provide both mathematical models for both MTZ and DFJ.

(b) (6 points) Implement the TSP with the MTZ formulation in Python and Gurobi.

(c) (6 points) Implement the TSP with the DFJ formulation in Python and Gurobi.

(d) (4 points) Try solving the formulations for the first 5, 10, 15, 20 and 25 customers. Use a time limit of 10 minutes per instance for building and solving the model. Which formulation performs better? Does this depend on the instance size? Explain.

1.2. Improving DFJ (20 points)

Next, we develop a cut generation algorithm to solve the DFJ formulation.

(a) (5 points) Explain and implement a method that, for a given solution, determines if it contains subtours.

(b) (8 points) Extend your implementation of the DFJ formulation such that it only eliminates those subtours that actually emerge for a given instance.
(Hint: You can let Gurobi solve the instance entirely and then iteratively add those constraints that are violated. This will earn you full points. Do not use callbacks or similar since you will have less insight into what happens.)
(Hint: If you were not able to solve question 1.2(a), you can manually separate violated constraints and still earn points here.)

(c) (3 points) Plot the objective value per iteration for the largest instance that you are able to solve, and explain what you see.

(d) (4 points) How does the runtime of the new implementation of DFJ compare with the runtime of your old implementation of DFJ, and the implementation of MTZ? Why? Does this differ between instance sizes?

2 Extending the Model: Capacity and Time Windows

In this section, we extend the basic routing models to incorporate real-world considerations, including vehicle capacity limits and customer time windows. You may assume access to an unlimited number of vehicles. You will also explore trade-offs between cost, service quality, and resource utilization.

2.1. Practical Constraints (18 points)

(a) (5 points) Extend the MTZ or the DFJ formulation to account for capacities and time windows.

(b) (5 points) Solve the model and compare the solution to the one in question 1.1.

(c) (8 points) Besides driven distance, the company also wants to minimize the number of vehicles. How can you account for this in the model? Then, analyze the trade-off between the number of vehicles and the driven distance.

2.2. Improving Customer Service (30 points)

Customer satisfaction is strongly influenced by the length of the delivery time windows. Narrower time windows improve predictability for customers but may increase operational costs, while wider time windows offer flexibility but reduce convenience. In this section, we analyze this trade-off and assess how changes to time window width affect the optimal routing plan and overall performance. For this analysis, Ignore vehicle capacities.

(a) (5 points) Explain why the TSP formulations developed in question 1.1 provide a lower bound on the optimal solution for every time window width.

(b) (5 points) Analyze the trade-off between the distance and the time window width.

(c) (12 points) Develop an alternative mathematical formulation that uses a time-expanded network.

(d) (3 points) How fine should the time-discretization be to find the exact solution?

(e) (5 points) Which formulation performs better? How does this depend on the time window width? Explain why.

2.3. **Driver Constraints** (12 points)

Although the company aims to minimize operational costs, it also prioritizes fair and sustainable working conditions for its employees. The management wants to ensure that no driver's total working day—including driving, service and waiting times—exceeds a given maximum duration of 7 hours. In this part, you will incorporate this constraint into your model and analyze its impact on routing decisions and overall efficiency.

(a) (8 points) Provide a mathematical formulation for the duration constraint. *(Hint: you need to define additional decision variables)*

(b) (4 points) Provide some insights in how this affects the solutions.

3 Workload reporting

3.1. **Workload reporting** (1 point)

Please truthfully report

(a) who did which work

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

(c) if and how you used AI

(this is a bonus point)

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., `Assignment1_Group3_vanLieshout_vanRossum_Nguyen.pdf` and `Assignment1_Group03_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, the names and student IDs of every group member. The final report cannot exceed 8 pages (including pictures etc., but excluding the front page).