Decision-making in Transport and Logistics (1CM110)

ASSIGNMENT 2

Deadline: Dec 15th, 16:59

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

In order to increase the performance of the Dutch railway system, the government has decided to split up the network in several parts, and invite train operating companies to compete for these subnetworks through a public tender. Bids will be judged on their expected level of passenger service, and their expected costs.

TRAINU/e is one of the companies developing a bid for the so-called *A2-corridor*, which involves five intercity train lines that connect cities along the major axis from Maastricht to Den Helder, and some branches, see Figure 1. In this assignment, you are asked to contribute to this bid by constructing a feasible timetable and rolling stock plan for the Noord-Oost case.

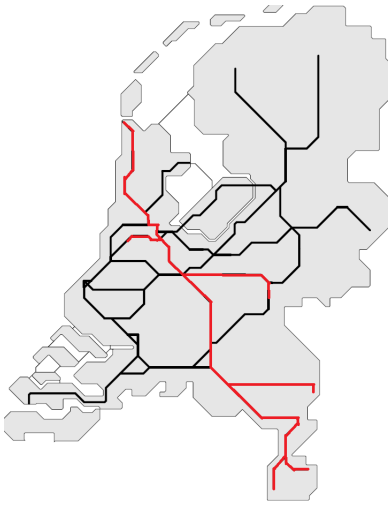

Figure 1: Lines in the A2-corridor (red).

1 Timetabling

In Table 1, you can find the name, frequency per hour and stops for the lines in the A2-corridor (note that we do not consider all stops for simplicity). All lines are operated in both directions. In `a2_part1.xlsx`, you can find the lines, the running times between stations and how stations are abbreviated.

Table 1: Lines in the A2-corridor.

Name	Frequency	Stops					
800	2	Amr	Asd	Ut	Ehv	Std	Mt
3000	2	Hdr	Amr	Asd	Ut	Nm	
3100	2	Shl	Ut	Nm			
3500	2	Shl	Ut	Ehv	Vl		
3900	2	Ehv	Std	Hrl			

The tender specifies that the following constraints must be taken into account:

- **Periodicity** The timetable should repeat itself every 30 minutes.
- **Headway constraints**
  The minimum headway between each pair of trains on the same track section is 3 minutes.
- **Dwell time constraints**
  The dwell time at all relevant stations is at least 2 minute, and at most 8 minutes.
- **Running time constraints**
  The running times should be exactly the values specified in the sheet "Travel Times".
- **Synchronisation constraints**
  On all parts of the networks where there are four trains running per hour per direction, these trains should be separated by exactly 15 minutes. For example, the 3000 and 3100 line should be synchronized between Utrecht and Nijmegen.
- **Transfer constraints**
  Passengers traveling from Heerlen to Utrecht should have a transfer opportunity at Eindhoven. The same holds for passengers traveling from Utrecht to Heerlen. The minimum transfer time is 2 minutes, the maximum transfer time is 5 minutes.

To ensure that the timetable of this network fits into the timetable in the western part of the country, the departure time of the 3500 at Schiphol is fixed at .09.

1.1. Basic Model (27 points)

(a) (2 points) Describe how this problem can be formulated as a Periodic Event Scheduling Problem. Provide the mathematical formulation. What is the value of T in this case?

(b) (6 points) The necessity of headway constraints depends on the railway infrastructure. For example, at station Utrecht Centraal, all northbound trains must use the same track, so headway constraints are required between the corresponding events (note that whenever there are four tracks, it is assumed that the middle tracks are used for sprinter trains). On the other hand, trains departing at Utrecht towards Eindhoven do not interfere with trains departing at Utrecht towards Nijmegen, so no headway constraints are required between these corresponding events.

Determine and explain between which events headway constraints should be added. You can examine the infrastructure at sporenplan.nl (Click on "Sporenplannen" in the top left corner, then on NL, then on "Actuele tekeningen - simpel"). There is only one station at which headway constraints between arrival events are required. Which one?

(c) (4 points) How many events do we have in the event-activity network? How many activities of each type are there?

(d) (2 points) Many timetables satisfy all described constraints. However some timetables are preferred above others. Come up with an appropriate objective function for the PESP formulation? Provide the formula and explain how it relates to passenger satisfaction.

(e) (10 points) Implement the model in Python and Gurobi, solve it and present the results in a meaningful way in your report. Your code should also output the timetable to the console in a comprehensible manner (e.g. for every event the assigned time). Name the Python file `Exercise_1.1e.py`.

(f) (3 points) How many travel options per hour are there for passengers traveling from Nijmegen or Eindhoven to Amsterdam or Schiphol? Would adding transfer constraints at Utrecht improve the level of service?

1.2. High-Frequency Service (23 points)

TRAINU/e believes it can offer a higher level of passenger service by extending the 3900 line to Amsterdam (with a stop at Utrecht), such that there are 6 trains per hour between Eindhoven and Amsterdam in both directions. The transfer constraints at Eindhoven are dropped, as all passengers can travel directly.

(a) (4 points) Explain why it is not possible to operate exactly one train every 10 minutes between Utrecht and Amsterdam and exactly one train every 15 minutes between Utrecht and Schiphol.

(b) (9 points) Explain how you can slightly relax the synchronicity requirements between Amsterdam and Eindhoven. Implement the additional constraints. Solve the model. What is the impact on the objective? Name the Python file `Exercise_1.2b.py`. The code should again output the timetable to the console.

(c) (10 points) Present and analyze the timetables at Amsterdam, Eindhoven and Utrecht. For which OD pairs does the new timetable with the increased frequencies actually decrease the level of service?

2 Rolling Stock Scheduling

Now we have determined the timetable that will be operated, we consider the question how much rolling stock of every available type TRAINU/e should buy in order to serve all trips. Since the required amount of rolling stock is determined by the peak hours, we will look at the most crowded period of the day and solve the so-called *8 o'clock rolling stock assignment problem* (see Abbink et al (2004)). In this problem, rolling stock needs to be assigned to all train services operated at the busiest time of the day. It is assumed that shunting is not possible during trips, so every trip should be assigned one composition that is operated during the entirety of the trip.

TRAINU/e can choose between two types of rolling stock, namely type PL3 and type PL4. Longer trains can be formed by coupling different trains units to each other. It is possible to couple an PL3 unit with an PL4 unit. Table 2 contains relevant data of both types.

Table 2: Costs, capacity and length of the rolling stock types.

Name	Fixed costs/year (in euros)	Capacity (in seats)	Length (in meters)
PL3	315,000	400	80
PL4	385,000	600	110

In the provided file `a2_part2.xlsx` you can find the minimum number of seats every train should have, depending on the line and direction. An additional constraint is that the rolling stock manufacturer requires that the number of units between the two types does not differ too much. In particular, the difference between the number of units of the most desired type and the number of units of the least desired type cannot be more than 25% of the number of units of the least desired type. Finally, the railway operator has to take into account the maximum length of a train. Trains should not be longer than the smallest platform along their route, which is 300 meters in general. However, the trains of line 3900 cannot be longer than 200 meters.

In the remaining questions, use the timetable specified in `a2_part2.xlsx` for all calculations, **do not** use the timetable you computed yourself in part 1.

2.1. Basic Model (30 points)

(a) (14 points) Assume that the periodic timetable is operated from 05:00 until 24:00. Determine the set of all trips that are running at 08:00 (include trips that depart or arrive exactly at 08:00). These trips are referred to as the *cross-section trains*. How many are there?

(b) (8 points) Formulate the problem as an integer linear programming problem. Make use of the formulation given by Abbink et al. (2004), with the decision variables $N_{u,t}$, which represents the number of train units of type u allocated to cross-section train t.

(c) (8 points) Implement the model using Python and Gurobi. What are the optimal costs? How much of each type should TRAINU/e buy? Your code should output the solution for every trip in a comprehensible way. Name the Python file `Exercise_2.1c.py`.

2.2. **Composition Model** (20 points)

Alternatively, we can model the problem using the concept of *compositions*, sequences of train units. Examples of compositions are PL3-PL3 (two PL3 units coupled together) or PL3-PL4 (one PL3 coupled to an PL4).

(a) (4 points) Explicitly write out the set P containing all compositions that satisfy the maximum length constraint. Note that since we do not consider shunting in this problem, the order is not relevant.

(b) (8 points) Provide an alternative formulation, with decision variables $X_{t,p}$, which is a binary variable equal to 1 if composition $p \in P$ is used for cross-section train t, and 0 otherwise.

(c) (8 points) Implement the composition formulation and compare its runtime to the basic formulation. Your code should output the assigned composition for every trip. Name the Python file `Exercise_2.2c.py`.Which formulation do you prefer and why?

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., `Assignment2_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).

References

Abbink, E., Van den Berg, B., Kroon, L., & Salomon, M. (2004). Allocation of railway rolling stock for passenger trains. Transportation Science, 38(1), 33-41.