Week 2 Assignment

1. Examine the balanced Transportation Problem

Download the file "*Week 2 Examples.ipynb*" from Canvas and open the file in Jupyter Notebook. Here, the balanced Transportation problem is described (copy from the textbook by Sierksma & Zwols), formulated and implemented as a function. Next, the problem instance as described in the textbook is solved.

Answer two follow-up questions as formulated by writing a few lines of Python code. You don't need to submit anything for this exercise.

2. Solve an unbalanced problem (Part A)

Download the file "*Week 2 Assignment.ipynb*" from Canvas. An unbalanced Transportation Problem can be transformed into an equivalent balanced problem, by adjusting the given data. Or more specific, by adding either a dummy supply node or a dummy demand node.

Apply this approach for answering **Problem W2.1** by writing a few lines of Python code. Report your findings in a pdf-file. Include a screenprint of the essential Python code.

3. Find the optimal selection for relay teams (Part B)

Solve **Problem W2.2** by changing the "reading data part": depending on the team, read the appropriate part(s) of the Excel file. For the Mixed team, also add additional constraint to the basic model.

Report your findings in a pdf-file. Include a screenprint of the essential Python code.

Note that these questions use the provided Jupiter Notebook. Please do not submit this file, only the relevant lines as screenshot.

Now, create a new and empty Jupiter Notebook for the final part of the assignment.

4. Formulate and solve an LO-model for two problems (Part C)

Formulate an LO-model for **Problem W2.3** and **Problem W2.4**. Implement both in Python. Report your findings (mathematical model, Gurobi model and solution) in a pdf-file. Hand-in the Jupyter Notebook separately.

PROBLEM W2.3 **(ANSWER QUESTIONS A AND B ON PAPER ONLY, USE GUROBIPY FOR QUESTION C)**

Consider the following LP model:

$$\begin{array}{lllr} \text{P: max} & z = 2y_1 + 3y_2 - 3y_3 + y_4 & & \\ \text{s.t.} & 2y_2 - y_3 + 3y_4 + y_5 & \leq 8 & (1) \\ & y_1 + y_2 - 2y_3 + y_4 & = 5 & (2) \\ & 5y_2 + 4y_4 - y_5 & \geq 10 & (3) \\ & y_1 & \geq 1 & (4) \\ & y_1 \in \mathbb{R},\ y_2, y_4, y_5 \geq 0; \quad y_3 \leq 0; & & (5) \end{array}$$

As should be clear right now, in order to obtain a proper set of constraints for the Simplex Method, the constraints are rewritten into a system of equations in terms of non-negative variables. To this end, a slack variable is added to every ≤constraint, an artificial variable to every =constraint and a surplus and artificial variable to every ≥constraint.

a) Determine according to this standard approach the first tableau in simplex form.
(Result: table with 5 rows, including the z-row, and twelve variables, including z).

b) Determine an equivalent but smaller starting tableau, after a short analysis of the problem P. It is not the intension to perform simplex calculations, rather analyse whether certain variables are useful, or can be replaced by something else. Finally, determine variables that are dominated by another variable and therefore will be equal to 0 in the optimal solution.

c) The rewriting and reducing steps are useful when solving models by hand. These are not needed when using software. In all packages a specified model is first analysed in a ProSolve phase. Solve the original model as given above (not your model from b) with Gurobipy and examine the effect of PreSolve (just set `Params.LogToConsole = 1` and analyse the output).

PROBLEM W2.4 Project Scheduling

An administrative project includes the following activities with corresponding duration (in weeks) and immediate predecessors that must be completed before the activities can begin. They are interested in the minimal total duration of the project. Or in other words, what is the minimal total number of weeks required to complete all activities. In Scheduling terms: the objective is to minimize the makepan (see the Note on Scheduling Terminology).

Activity/job	A	B	C	D	E	F	G	H	I	J	K	L	M
Duration	2	3	4	2	3	4	6	4	3	4	6	2	1
Predecessor(s)	-	-	A	B	B	E	E	E	D, F	C	C	H	G, I, J

a) Formulate a LO-model for this problem and solve it using Gurobipy.

Use python for the follow-up questions about the solution that you have just found.

b) Determine the *Critical Path(s)* of the project, i.e., a sequence of jobs that must start and finish as scheduled in order to complete the entire project within the makespan.

c) Check that the non-critical jobs have some slack, i.e., may start a little later without jeopardizing the makespan (ceteris paribus).

d) The main assumption for this problem is the unlimited availability of personnel to perform jobs simultaneously. Suppose each job requires one person, how many are needed for this project?