DECISION ANALYTICS

Assignment 2: Linear Programming

DUE DATE

This assignment should be submitted to Canvas before 11:59pm on **Friday 17/12/2021**.

Please submit a single ZIP file with your student number and name in the filename. Your submission should contain **exactly 2 files**:

- A detailed documentation of all code you developed, including the tests and evaluations you carried out. Please make sure that you include a .pdf document with every result you produce referencing the exact subtask and lines of code it refers to.
- All Python code you developed in a single .py file that can be executed and that generates the outputs you are referring to in your evaluation. The file needs to be readable in a plain text editor, please do NOT submit a notebook file or link. Please also make sure that you clearly indicate in your comments the exact subtask every piece of code is referring to.

Please do NOT include the input files in your submission.

You can achieve a total of 50 points as indicated in the tasks.

TASK 1 (supply chain, 22 points)

In this task you will optimise the cost of sourcing raw material from different suppliers, manufacturing products in different factories and delivering these products to customers. The input data for this task is contained in the Excel file *Assignment_DA_2_Task_1_data.xlsx* and can be downloaded from Canvas. The file contains 8 sheets:

- Supplier stock
  A table indicating how many units of each raw material each of the suppliers has in stock.
- Raw material costs
  A table indicating how much each of the suppliers is charging per unit for each of the raw materials.
- Raw material shipping
  A table indicating the shipping costs per unit of raw material (the units for each material are the same) from each supplier to each factory

- <u>Product requirements</u>
  A table indicating the amount of raw material required to manufacture one unit of each of the products.
- <u>Production capacity</u>
  A table indicating how many units of each product each of the factories is able to manufacture.
- <u>Production cost</u>
  A table indicating the cost of manufacturing a unit of each product in each of the factories.
- <u>Customer demand</u>
  A table indicating the number of units of each product that have been ordered by the customers
- <u>Shipping costs</u>
  A table indicating the shipping costs per unit for delivering a product to the customer.

Factories can order suppliers from multiple suppliers and products can be delivered to customers from multiple factories.

The goal of this task is to develop and optimise a Liner Programming model that helps decide what raw material to order from which supplier, where to manufacture the products, and how to deliver the manufactured products to the customers so that the overall cost is minimised.

A. Load the input data from the file *Assignment_DA_2_Task_1_data.xlsx* [1 point]. Note that not all fields are filled, for example Supplier C does not stock Material A. Make sure to use the data from the file in your code, please do not hardcode any values that can be read from the file.
B. Identify and create the decision variables for the orders from the suppliers [1 point], for the production volume [1 point], and for the delivery to the customers [1 point] using the OR Tools wrapper of the GLOP_LINEAR_PROGRAMMING solver.
C. Define and implement the constraints that ensure factories produce more than they ship to the customers [1 point].
D. Define and implement the constraints that ensure that customer demand is met [1 point].
E. Define and implement the constraints that ensure that suppliers have all ordered items in stock [1 point].
F. Define and implement the constraints that ensure that factories order enough material to be able to manufacture all items [1 point].
G. Define and implement the constraints that ensure that the manufacturing capacities are not exceeded [1 point].
H. Define and implement the objective function. Make sure to consider the supplier bills comprising shipping and material costs [1 point], the production cost of each factory [1 point], and the cost of delivery to each customer [1 point].
I. Solve the linear program and determine the optimal overall cost [1 point].

J. Determine for each factory how much material has to be ordered from each individual supplier [1 point].
K. Determine for each factory what the supplier bill comprising material cost and delivery will be for each supplier [1 point].
L. Determine for each factory how many units of each product are being manufactured [1 point]. Also determine the total manufacturing cost for each individual factory [1 point].
M. Determine for each customer how many units of each product are being shipped from each factory [1 point]. Also determine the total shipping cost per customer [1 point]
N. Determine for each customer the fraction of each material each factory has to order for manufacturing products delivered to that particular customer [1 point]. Based on this calculate the overall unit cost of each product per customer including the raw materials used for the manufacturing of the customer's specific product, the cost of manufacturing for the specific customer and all relevant shipping costs [2 points].

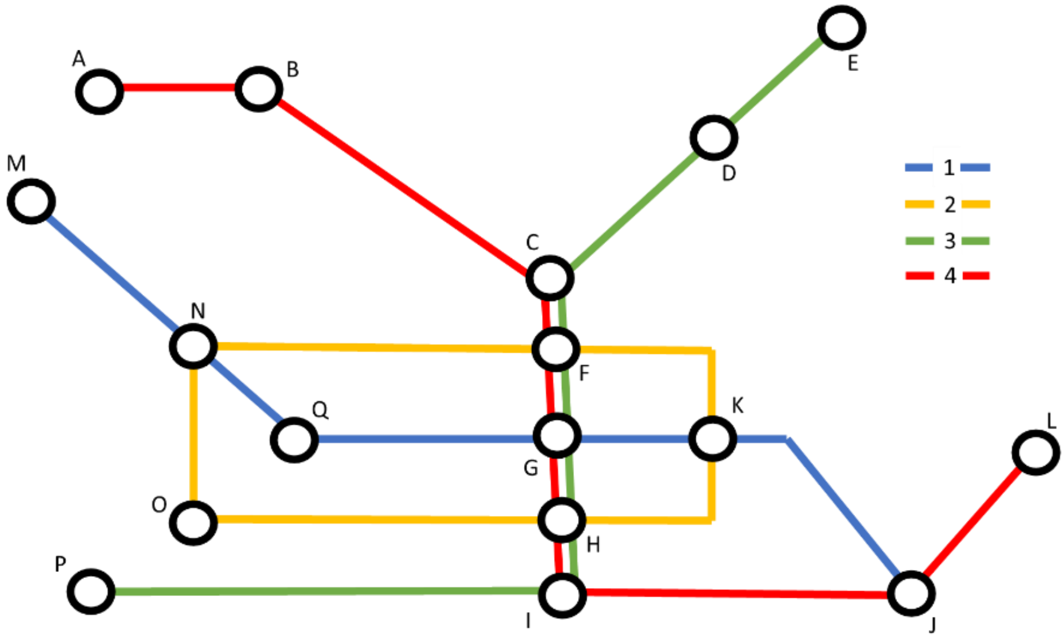
TASK 2 (delivery driver, 8 points)

In this task you will optimise the travel distance for a delivery driver. The input data for this task is contained in the Excel file *Assignment_DA_2_Task_2_data.xlsx* and can be downloaded from Canvas. It contains the distances between cities and towns in Ireland. Your task is to find the shortest route starting and finishing in Cork and going through Dublin, Limerick, Waterford, Galway, Wexford, Belfast, Athlone, Rosslare, and Wicklow.

A. For each pair of towns that need to be visited create a decision variable to decide if this leg should be included into the route [1 point]. Use the OR Tools wrapper of the CBC_MIXED_INTEGER_PROGRAMMING solver.
B. Define and implement the constraints that ensure that the delivery driver arrives in each of the towns that need to be visited [1 point].
C. Define and implement the constraints that ensure that the driver departs each of the towns that need to be visited [1 point].
D. Define and implement the constraints that ensure that there are no disconnected self-contained circles in the route [2 point]. To do this, enforce for all subsets of visited towns that within these subset there are strictly less journey legs than there are towns (for example, when you consider the subset {Dublin, Cork, Limerick, Galway} make sure that there cannot be more than 3 connections within this subset to prevent the circle Dublin -> Cork -> Limerick -> Galway -> Dublin).
E. Define and implement the objective function to minimise the overall distance travelled [1 point]. Use the distance data from the file *Assignment_DA_2_Task_2_data.xlsx*.
F. Solve the linear program and determine the overall distance that needs to be travelled to visit all towns [1 point]. Also output the optimal route starting and ending in Cork [1 point].

TASK 3 (train network, 20 points)

In this task you will optimise the number of trains required for the operation of the network depicted below. The network operates four lines, with the trains on the lines L1, L3, and L4 always going back and forth between the terminal stations and at least two trains operating on the circular line L2, one in clockwise direction the other in anti-clockwise direction.



The input data for this task is contained in the Excel file *Assignment_DA_2_Task_3_data.xlsx* and can be downloaded from Canvas. The file contains 4 sheets:

- Stops
  This table contains the stops for each line. Use this input data in your code, the picture above is for illustration only and does not constitute an input for this task.
- Distances
  This table contains the distances in minutes between the train stations.
- Passengers
  This table contains the number of passengers per hour that want to travel between two stations on the network. For example, there are 25 passengers per hour who want to travel from A to C, while there are 20 passengers per hour who want to travel from C to A.
- Trains
  This table contains the passenger capacity of each train on the different lines.

Assume that passengers are always travelling on the route to their destination that takes the shortest amount of time. To simplify the problem also assume that trains do not need any extra time in the train stations and that passengers can always change trains immediately and without losing any time either. The goal of this task is to determine the number of trains required on each line, which are necessary to meet passenger demand.

A. Load the input data from the file *Assignment_DA_2_Task_3_data.xlsx* [1 point]. Make sure to use the data from the file in your code, please do not hardcode any values that can be read from the file, in particular do not hardcode the train network from the image above.
B. For each pair of train stations determine the amount of time required to travel between the two using an Integer Linear Program:
    a. For each two stations connected by a train line create a decision variable to decide if this leg should be included into the route [2 point]. Use the OR Tools wrapper of the CBC_MIXED_INTEGER_PROGRAMMING solver.
    b. Define and implement the constraints that ensure the train station where the journey originated is included in the path [1 point], the destination train station is included in the path [1 point], and that there are no dead ends in the path [1 point].
    c. Define and implement the objective function to minimise the overall travel time [1 point].
    d. Solve the linear program and calculate the travel times [1 point], the optimal travel route through the network [2 point], and the train lines operating on each leg [1 point]. Output these results.
C. Now determine how many trains are required for each line to meet passenger demand using an Integer Linear Program:
    a. Create the decision variables to determine how many trains will be required for each line to meet passenger demand [2 point]. Use the OR Tools wrapper of the CBC_MIXED_INTEGER_PROGRAMMING solver.
    b. Define and implement the constraints that ensure passenger demand is met on the whole network [3 point]. If multiple lines operate between two stations (e.g. F-G) passengers will use the capacity available in either line, possibly changing train if necessary.
    c. Define and implement the objective function to minimise the number of trains operated on the network [1 point].
    d. Solve the linear program and determine how many trains are required [1 point]. Also determine how many trains are required per line [2 point], make sure to distinguish between clockwise and anti-clockwise trains for the circular line L2.