COMP 9057 - DECISION ANALYTICS

ASSIGNMENT 2 - Linear Programming

SUBMISSION & DUE DATE

This assignment should be submitted to Canvas before 11:59pm on **Friday 8/12/2023**. The usual late submission penalties apply in accordance with MTU's marks and standards policy.

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 for the submission as indicated in the tasks. For each subtask you are given full marks for correct answers in your submission, 70% for minor mistakes, and 35%, 15%, or 0% for major mistakes or omissions depending on severity.

TASK 1 (supply chain, 21 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
- Product requirements
  A table indicating the amount of raw material required to manufacture one unit of each of the products.
- Production capacity
  A table indicating how many units of each product each of the factories is able to manufacture.
- Production cost

A table indicating the cost of manufacturing a unit of each product in each of the factories.

- Customer demand
  A table indicating the number of units of each product that have been ordered by the customers
- Shipping costs
  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 [1 point].

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 (investment portfolio, 21 points)

In this task you will optimise an investment portfolio over different stock indices. 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 3 sheets:

- <u>USD</u>
  This table contains monthly data of four major US stock indices, the S&P 500, the Dow Jones Industrial Average, the NASDAQ Composite, and the Russell 2000 index, for the past five years. All values in this table are in US dollars.
- <u>EUR</u>
  This table contains monthly data of four major European stock indices, the EuroStoxx50, the CAC, the DAX, and the ISEQ20 index, for the past five years. All values in this table are in Euro.
- <u>Currency</u>
  This table contains the Euro to US dollar exchange rates for the past five years.

The goal of this task is to determine an optimal investment strategy into these eight stock indices and to evaluate the expected reward of an optimal risk minimising portfolio. To simplify the equations, we assume that there are no dividends and that monthly profits/losses are immediately cashed/compensated and not re-invested.

A. Load the input data from the file *Assignment_DA_2_Task_3_data.xlsx* and convert all stock prices consistently into either Euro or USD. Create a parameter to decide which currency to use [2 points]. For each month t (except the first) calculate the monthly return for each investment position i from the position values S_i as follows [1 point]

$$R_i[t] = \frac{S_i[t]}{S_i[t-1]}$$

Under the assumption that profits/losses are not re-invested, the overall average monthly reward for a position can then be calculated by summing over all months:

$$r_i = \frac{1}{T}\sum_{t=1}^{T} R_i[t]$$

Determine and output the overall average monthly reward for each investment position [1 point].

B. Create a Linear Program to determine the reward that optimal timing the market could have achieved over the past five years using the OR Tools wrapper of the GLOP_LINEAR_PROGRAMMING solver.
   a. For each month create decision variables that indicate the percentage of each position held as well as the percentage of cash not invested during this month [1 point].
   b. Identify and create the implicit constraints to ensure that the investment portfolio always adds up to 100% [1 point].
   c. Investing everything into one single position is not good practice. Therefore, identify and create constraints that ensure that no single investment position is ever more than 30% of the overall portfolio [1 point].
   d. Identify and implement an objective function that maximises the overall reward of the portfolio by summing all respective monthly returns [1 point].
   e. Solve the Linear Program and output the portfolio allocation over time [1 point] as well as the overall average monthly reward that would have been achievable by optimally timing the market over the past 5 years [1 point].

C. Timing the markets is of course impossible without knowing what is going to happen in the future, therefore an optimal investment portfolio should aim at minimising the investment risk instead. Create another Linear Program to determine such an optimal portfolio that minimises the investment risk using the OR Tools wrapper of the GLOP_LINEAR_PROGRAMMING solver.
   a. In this approach the portfolio if fixed over the whole period of five years, i.e. the money allocation does not change and there is no cash position. Create decision variables that indicate the percentage of each position held in the portfolio during the entire investment period [1 point].
   b. Create the implicit constraint that the investment portfolio always adds up to 100% [1 point].
   c. Identify and create constraints to ensure that no single investment position is ever more than 30% of the overall portfolio [1 point].
   d. Even if risk minimisation is the primary goal of the portfolio, a minimum reward should be expected. Therefore, create a constraint to ensure that the overall average monthly reward of the portfolio is at least 0.5% over the five-year investment period [1 point].
   e. The risk of a position can be controlled by making sure that the monthly return never deviates more than some amount $y_t \geq 0$ from the average monthly return, i.e. if w_i is the investment into a position then

$$-y_t \leq \sum_i w_i(R_i[t] - r_i) \leq y_t$$

Create these additional variables y_t for each month and implement the necessary constraints for bounding the deviation between average and actual monthly reward of the investment portfolio [3 point].

f. Create an objective function that minimises the average of the risk bounds implemented in the previous sub-task [1 point].
g. Solve the Linear Program and output the portfolio allocation [1 point] as well as the overall average monthly reward of the portfolio over the past five years [1 point]. Compare the results between using EUR or USD as basis for the investments [1 point].