MS-E2121 - Linear optimization **DEADLINE:**
Homework 1 **31.01 @23:55**

Please submit your answers here before the given deadline.

You are requested to submit:

(1) A short report in .pdf format with your answers to items (1.1) and (1.2). If you scan your handwritten notes, please make sure the quality is good.
(2) The completed Jupyter notebook.

You can download the skeleton notebook and the LaTeX template from the MyCourses page.

Problem 1.1: Deterministic capacity expansion problem

Suppose that we are interested in designing an optimised production system for the next $|T|$ periods. Let I be a set of supply sources and J a set of demand points, each with demand D_{jt} at each time period $t \in T$, for all $j \in J$. The objective is to define minimum cost capacity levels x_i for each supplier $i \in I$ in advance so that the demands D_{jt} of all customers $j \in J$ can be optimally fulfilled when these demands are observed in the future.

Let p_{it} be the amount of product manufactured at the supplier $i \in I$ in period $t \in T$ at the cost of M_i per unit, and let e_{ijt} be the amount of product available at supplier $i \in I$ that is used to fulfil the demand of client $j \in J$ in period $t \in T$. Any amount of product that is not used can be stored at a cost of H_i per unit and time period, the amount of product in storage at the end of time period t is represented by variable k_{it}. The storage is empty in the beginning and there are no constraints for storage at the end of the planning period.

Let u_{jt} be the amount of demand of client $j \in J$ in period $t \in T$ that is not fulfilled. We assume that unfulfilled amounts u_{jt} are penalised at a unit cost Q_j for each client $j \in J$ in any period of time. Each unit of capacity x_i built for supplier $i \in I$ costs C_i, and F_{ij} is the unit cost to fulfil the demand of client $j \in J$ using supplier $i \in I$.

This problem can be formulated as follows:

$$
\begin{aligned}
\text{min.} \quad & \sum_{i \in I} C_i x_i + \sum_{i \in I} \sum_{t \in T} (H_i k_{it} + M_i p_{it}) + \sum_{i \in I} \sum_{j \in J} \sum_{t \in T} F_{ij} e_{ijt} + \sum_{j \in J} \sum_{t \in T} Q_j u_{jt} && (1) \\
\text{s.t.:} \quad & p_{it} \leq x_i, \ \forall i \in I, \forall t \in T && (2) \\
& p_{it} + k_{i(t-1)} = \sum_{j \in J} e_{ijt} + k_{it}, \ \forall i \in I, \forall t \in T && (3) \\
& \sum_{i \in I} e_{ijt} = D_{jt} - u_{jt}, \ \forall j \in J, \forall t \in T && (4) \\
& k_{i0} = 0, \ \forall i \in I && (5) \\
& x_i \geq 0, \ \forall i \in I && (6) \\
& p_{it}, k_{it} \geq 0, \ \forall i \in I, \forall t \in T && (7) \\
& e_{ijt} \geq 0, \ \forall i \in I, \forall j \in J, \forall t \in T && (8) \\
& u_{jt} \geq 0, \ \forall j \in J, \forall t \in T. && (9)
\end{aligned}
$$

Table 1: List of variables in the problem (1) – (9).

Variable	Description
Parameters	
$i \in I$, $j \in J$, $t \in T$	Supply sources, demand points and time periods
C_i	Unit cost of building capacity at supplier i
H_i	Storage cost at supplier i
M_i	Unit cost for manufacturing at supplier i
F_{ij}	Transportation cost between supply i and demand j
Q_j	Penalty for unmet demand of point j
D_{jt}	Demand for customer j at time period t
Decision variables	
x_i	Capacity of supplier i
k_{it}	Amount of product in storage at supplier i at time period t
p_{it}	Amount of product manufactured at supplier i at time period t
e_{ijt}	Amount sent from supply i to demand point j at time period t
u_{jt}	Unmet demand of demand point j at time period t

Implement and solve the problem (1) – (9) with `Julia` using `JuMP` using the data provided in this skeleton Jupyter notebook.

Hint: Remember the boundary condition for storage: $k_{i0} = 0$, assuming $T = \{1, ..., |T|\}$. Disclaimer: depending on your approach, you might not need to explicitly include this constraint or the corresponding variables for initial storage.

Problem 1.2: Stochastic capacity expansion problem

Now, consider that the future demand at each client $j \in J$ is uncertain. Assume that the only an expected value for the first period demand is known, namely $\overline{D}_{j1}$, and that a predictor is available in the form of

$$D_{j1} = (1 + \sigma\varepsilon)\overline{D}_{j1} \tag{10}$$

$$D_{jt} = (1 + \mu + \sigma\varepsilon)D_{j(t-1)}, \forall t \in T \setminus 1 \tag{11}$$

where, for each demand point $j \in J$, μ is an expected demand growth, σ is the maximum expected variability and $\varepsilon \sim N(0, 1)$.

To represent the variability associated with the demand, we will consider an extension of the problem formulated in **(1.1)** based on *two-stage stochastic programming* (2SSP). The main idea behind 2SSP models is that decisions are represented taking into account a collection of scenarios and as being made in terms of optimising against the average behaviour of the system.

In this case, we consider that our first-stage decisions, i.e., those that have to hold regardless of the observed scenarios, are the capacity expansion decisions. On the other hand, all other decisions are assumed to be made once the demand for all upcoming periods are known.

Let us assume that a set of scenarios $s \in S$ is generated by (Monte Carlo) sampling, following the random process described in (11). Because of the sampling process, we assume that each scenario has a probability $P_s = 1/|S|$. Each scenario $s \in S$ represents a demand series D^s_{jt} for each location, that is the demand observed in each time periods. Notice that we must redefine the variables in the model to accommodate the respective scenario realisation, with exception of the variables representing the capacity decisions, which are made prior to the uncertainty realisation and thus do not depend on specific scenarios.

(a) Modify the deterministic problem into the stochastic variant described above and write out the model in your report.

(b) Implement and solve the stochastic version with `Julia` using `JuMP` and the data provided in the skeleton Jupyter notebook.

(c) Compare the solutions obtained from each model. How they differ in terms of the optimal capacities? And how about computational requirements (i.e., solution times). NOTE: before you write the report, please restart your notebook and run all cells (for Jupyterlab: Kernel $->$ Restart kernel and run all cells). This way, the random data used in your solution is the same as in our model solutions.

Hint: You can use the model solution for Exercise 1.5 (given on the MyCourses page) as a reference to guide the adaptation of the deterministic model developed in (1.1) into a stochastic model.