LINMA2491: Project

Battery trading on the electricity market

***Released**: March 19, 2025*
***Due**: May 16, 2025 (11:59pm)*

Context

The firm Engie is currently building one of Europe's biggest Battery Energy Storage System (BESS) in Vilvoorde, Belgium. The battery park will be directly connected to the transmission grid. When fully operational, the park will have a storage capacity of 800 MWh of power to release to the grid for four hours.

Figure 1: The future battery energy storage system (BESS) plant in Vilvoorde, Belgium, one of the largest in Europe. Image ©ENGIE

Engie is currently preparing its operational strategy. To this end, they have hired your team of independent consultants to come up with an algorithm to trade on the electricity market. The goal of the algorithm is to decide when to charge and discharge the BESS in order to maximize the profit, taking into account the dynamic electricity price on the Belgian wholesale market.

Modeling and Data

The battery has given capacity (800MWh) and maximal rate of charge/discharge (200MW). At every time-step, we can either sell (by discharging energy from the battery) or buy electricity (by charging energy to the battery). For our analysis, we use real data from the European spot market EPEX, in the year 2024, provided in the file `belpex_price.txt`. The spot price is sampled at an hourly time-step during 1 day (giving a total of 24 time-steps). The battery has a charging efficiency and a discharging effiency of 90%. That is, if you buy 100MWh on the market, only 90MWh get stored in the battery. And if you sell 100MWh, 111.11MWh get discharged.

We use a classical stochastic model for the energy price. The given price vector encodes the average energy price $\overline{\lambda}_t$ (a.k.a. expected value). The log-deviation from the expected value is modeled by a stochastic process $\xi_t(\omega)$, here modeled as a stationary Markov chain. The model writes, for all $t = 1, \dots, T$,

$$\log(\lambda_t(\omega)) = \log(\overline{\lambda}_t) + \xi_t(\omega)$$

or, equivalently,

$$\lambda_t(\omega) = \overline{\lambda}_t \bullet \exp \xi_t(\omega)$$

At each time step the uncertainty $\xi_t(\omega)$ can take N distinct values $\xi_1, \dots, \xi_N$. The transition between ξ_t and ξ_{t+1} are encoded by the following conditional probabilities:

$$\mathbb{P}[\xi_{t+1} = \xi_j \mid \xi_t = \xi_i] = p_{ij}$$

The values and transition probabilities are stored as text files `markov_support_N.txt` and `markov_weights_N.txt` for $N \in \{4, 8, 16, 32\}$. Unless stated otherwise, let $N = 4$ (for questions 1.1 until 3.3 included).

Questions

Part 1: Deterministic model

1. Plot the evolution of the given expected price and comment. How can the battery make profit ? How is it useful for the system ?

2. Generate 10 different price scenarios over the 24 hours and plot them against the expected price.

3. Formulate the deterministic problem.

4. Solve the deterministic model of profit maximization over the 24 hours and plot the evolution of the energy stored in the battery.

For Part 2 and Part 3, assume that in the root node of the scenario tree, the system is in markov state 1 (no uncertainty in the root node).

Part 2: Two-stage stochastic model

Before considering the full multistage stochastic programming problem, let us transform the problem into multistage problem that can be equivalently formulated as a **two-stage stochastic programming problem**. In doing so, assume that the price in the first F hours of the day is perfectly known and equal to the expected price. Then, generate K scenarios that describe the evolution of the price for the next $T - F$ hours. Take $F = 5$ and $K = 10$. The scenario tree for this formulation is shown in Figure 2.

1. Formulate the two-stage-equivalent multistage stochastic programming problem.

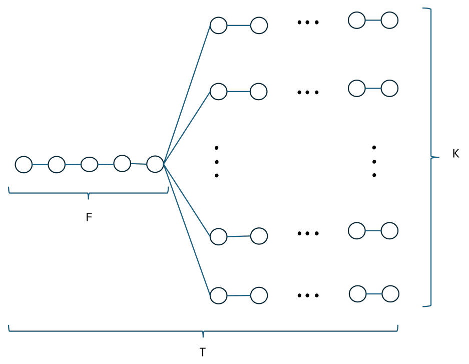

Figure 2: Scenario tree of the two-stage-equivalent multistage stochastic programming problem.

2. Explain why this problem is equivalent to a two-stage stochastic program.
3. Implement the L-Shaped and the multicut L-Shaped algorithm for solving this problem. For both algorithms, plot the evolution of the lower and upper bounds.

Part 3: Multistage stochastic model

Now, consider the full multistage stochastic model.

1. How many nodes are there in the full scenario tree ?
2. Implement both the extended formulation of the multistage stochastic program and the scenario tree formulation for H < T. Solve both formulations for H = 1, 2, ... until your reach the memory limit of your computer. What is the maximum $\overline{H}$ that your computer can handle ?
3. Plot the evolution of the number of variables and constraints for both formulations as H increases from 1 to $\overline{H}$.

We will now solve the stochastic model using the SDDP algorithm. You can either implement the algorithm yourself or use a package such as `SDDP.jl` (recommended).

4. Solve the full stochastic model using SDDP for $N \in \{4, 8, 16, 32\}$ and report for each instance a confidence interval on the solution.
5. Report the Expected Value of Perfect Information and the Value of the Stochastic Solution for $N \in \{4, 8, 16, 32\}$. Explain how you generalize these quantities to the multistage case.
6. For each N, provide spaghetti plots and ribbon plots for the energy content of the BESS. Plot the value function of state $\omega = 3$ at stage $t = 22$. See here for a description of these plots.
7. Discuss the results.

Handout

You are asked to submit your results as a PDF file, either in the form of a report or as a slide deck to be presented to the client, as is customary in consulting. The form of the report will have no influence on the grade, but you may find it interesting to try the slide deck approach.

Please also submit your code as separate files together with the submission. The submission must be made on the moodle page of the course in the dedicated section. The project must be realized by groups of 3 students. One submission per group is sufficient.