Lab 2: Stochastic expansion modeling

MAE573: Optimization Methods for Energy Systems Engineering

Jesse Jenkins, Jonah Langlieb, Godwin Obi

Fall 2025

In this lab session we will work with a full complex capacity expansion model and use stochastic optimization to understand how to model parametric uncertainty. By the end of this lab session, students will be comfortable working with a basic capacity expansion model and gain hands-on experience with two-stage stochastic optimization

Please work through the steps below in groups of 2-3, making sure to discuss as a team when possible.

Both team members should do these first steps individually on their own computers, to be ready for HW5:

1. Download and install VSCode, and the Julia extension, if you don't have it already.
2. Download and install the [Gurobi solver](), making sure to obtain an academic license, if you don't have it already.
3. Test that your Gurobi license is working using [these instructions]().
4. Install the Gurobi package in Julia by going to the package manager and typing `] add Gurobi`

Section 1: **Modify the CEM in the notebook provided to become a two-stage stochastic model, with operational decisions and constraints and parameters indexed over scenarios s in S**.
Section 2: Load in new input data for several scenarios. Model the optimal strategy under stochastic optimization with equal probabilities for each scenario. Then model the optimal strategy for deterministic optimization under each of the scenarios (one at a time). Compare results and discuss.
Section 3: Identify which of the scenarios had the worst outcome (highest objective function cost) in the set of deterministic scenario analysis cases in Section 2. Increase the weight of this scenario in the stochastic optimization model to a higher level (up to you) to make the results more robust to this outcome. How does the optimal strategy change as you increase the weighting? How much does the cost reduce in the worst-case scenario as you increase this weight? What is the tradeoff in expected cost assuming equal probability for all scenarios? (edited)

The remainder of the steps should be done as a team. For the questions which require you to propose changes and analyze results, please type up your responses in a separate document and submit your responses to the Canvas assignment (one per team). Please include the names of all team members on this document.

- **Refresh your course repository to the latest version.**
  Then, open the `extra_labs/Lab3/Lab3.ipynb` file in the course repo.
  We will work through the different cells throughout this lab to experiment with stochastic optimization within a complex capacity expansion model.
- **Install `CairoMakie`**
  While we discuss, install `CairoMakie` (see the top of the notebook) as it will take some time to compile
- **Familiarize yourself `generate_stochastic_model`**
  This is the main workhorse function for this lab and is more complicated than previous JuMP models seen so far.
  As with any sort of complicated code, make a few passes, getting a sense of:
  - the main indices
  - the different types of constraints we've discussed in class (e.g. storage, existing/new-built generators, ramping limits, etc).
  - how stochastic programming modifies the structure (hint: look for the scenario indices `s ∈ Sc`)
  - where does the 'planning' half end and the 'operations' half begin

  Then to check your understanding:
  Pick one of the operational constraints and walk through it in detail, discussing each part

  Explain why the objective and `eOperationCosts` looks the way it does

- **Understand the setup.**

  The notebook includes **five scenarios (`s1–s5`)**, each representing a different combination of load, fuel prices, and renewable generation profiles. Each scenario can be solved both **deterministically** (as if it is the only possible future) and **stochastically** (when a subset of the futures are considered together with assigned probabilities).

Section 2 — Deterministic and Stochastic Runs

- **Run the stochastic model with equal probabilities.**

Run the cell which has three equal-weight scenarios. This will solve the two-stage stochastic model, where first-stage capacity decisions are made before knowing which scenario occurs.

- **Run the deterministic models.**
  Now resolve where you run each scenario individually, examining the results.
- **Compare deterministic and stochastic outcomes.**
  - How do the stochastic optimization results compare with the deterministic results?
  - What are the expected total system costs? How does this compare with the system costs for the deterministic case?
  - How does the capacity mix compare to the deterministic results?
  - Which scenario has the highest system cost within the stochastic solutions?
- **Re-run the stochastic model with higher weight in the worst case (scenario with the highest system cost).**

  Increase the probability-weight of the worst-case scenario.

  How does the optimal strategy change as you increase the weighting?

  How much does the cost reduce in the worst-case scenario as you increase this weight? What is the tradeoff in expected cost assuming equal probability for all scenarios?

If have extra time:

- Play around with different probabilistic weights and different scenario multipliers
- The generator data itself can be modified in `../../complex_expansion_data/10_days/` (or with julia at `base_inputs[:generators]`).
  How does modifying the initial conditions and various costs change the outputs
- How could you model deep uncertainty here