Lecture X - Intermission: Exam Preparation

Energy System Optimization with Julia

Dr. Tobias Cors

Intermission

Today's lecture

- Today's lecture is a **little bit different**
- Manage your expectations
- Give you a **better idea** of what to expect from the exam
- We will go through **some examples together!**

Energy System Optimization Exam

- **Duration**: 90 minutes
- **Total Points**: 60 points
- **Structure**: Three parts (Modeling, Theory, Programming)
- **Allowed**: One handwritten DIN A4 sheet

. . .

> ❗ Important
>
> You can take a **handwritten** DIN A4 sheet of paper with you!
> Each point corresponds to approximately 1.5 minutes of work!

Exam Structure

- **Part I**: Energy System Modeling (30 points)
- **Part II**: Theory and Concepts (15 points)

- **Part III**: Julia Programming (15 points)

Exam Preparation Checklist

> 💡 Before starting the exam:
>
> ☐ Review modeling notation conventions
> ☐ Practice writing sets, parameters, and variables
> ☐ Review common constraint patterns
> ☐ Practice Julia syntax
> ☐ Prepare your DIN A4 cheat sheet

Part I

Energy System Modeling

1.a (8 Points)

A power system operator needs to optimize the economic dispatch of thermal generators to meet electricity demand. The system consists of:

- Multiple thermal generators with different variable costs
- Wind turbines with forecasted power output
- A single aggregated demand that must be met

Each generator has minimum and maximum power output limits. The wind turbines have zero variable cost but their output is limited by the forecast.

Define all sets, parameters, and variables required to model this Economic Dispatch problem. Use clear notation and explicitly state which elements are sets, parameters, and variables.

Solution

Sets:

- $g \in \mathcal{G}$: Set of thermal generators
- $w \in \mathcal{W}$: Set of wind turbines

Parameters:

- c_g^{var} : Variable cost of generator g [€/MWh]
- $p_g^{\min}$: Minimum power output of generator g [MW]
- $p_g^{\max}$: Maximum power output of generator g [MW]
- p_w^f : Forecasted power output of wind turbine w [MW]
- d^f : Forecasted demand [MW]

Variables:

- p_g : Power output of thermal generator g [MW]
- p_w : Power output of wind turbine w [MW]

1.b (4 Points)

Based on your notation from 1.a, write the objective function for the Economic Dispatch problem.

Solution

$$\text{min} \quad \sum_{g \in \mathcal{G}} c_g^{var} p_g$$

Note: Wind turbines have zero variable cost, so they don't appear in the objective function.

1.c (6 Points)

Write all necessary constraints for the Economic Dispatch problem using your notation from 1.a.

Solution

Power Balance:

$$\sum_{g \in \mathcal{G}} p_g + \sum_{w \in \mathcal{W}} p_w = d^f$$

Generator Limits:

$$p_g^{\min} \leq p_g \leq p_g^{\max} \quad \forall g \in \mathcal{G}$$

Wind Limits:

$$0 \leq p_w \leq p_w^f \quad \forall w \in \mathcal{W}$$

1.d (6 Points)

Now extend the Economic Dispatch model to include Unit Commitment decisions. The generators can be turned on/off and have additional start-up costs.

Define the additional sets, parameters, and variables needed for the Unit Commitment problem.

Solution

Additional Sets:

- $t \in \mathcal{T}$: Set of time periods

Additional Parameters:

- c_g^{start} : Start-up cost of generator g [€]

Additional Variables:

- $u_{g,t}$: Binary variable indicating if generator g is on at time t
- $v_{g,t}$: Binary variable indicating if generator g starts up at time t

1.e (6 Points)

Write the (a) objective function and (b) start-up variable definition for the Unit Commitment problem using your notation from 1.d. In addition, consider that (c) generator 1 and 2 use the same grid connection which is currently limited so that they cannot be on at the same time. An additional constraint is that (d) due to personnel reasons the startup of generator 3 has to be at least 3 timesteps away from startup of generator 4.

Solution

(a) Objective Function:

$$\text{min} \quad \sum_{t \in \mathcal{T}} \left(\sum_{g \in \mathcal{G}} (c_g^{var} p_{g,t} + c_g^{start} v_{g,t}) \right)$$

(b) Start-up Variable Definition:

$$v_{g,t} \geq u_{g,t} - u_{g,t-1} \quad \forall g \in \mathcal{G}, t \in \mathcal{T}$$

(c) Grid Connection Limits:

$$u_{1,t} + u_{2,t} \leq 1 \quad \forall t \in \mathcal{T}$$

(d) Startup of Generator 3 has to be at least 3 timesteps away from startup of generator 4:

$$\sum_{\tau \in [t, \min(t+3, |\mathcal{T}|)]} v_{3,\tau} \leq (1 - v_{4,t})M \quad \forall t \in \mathcal{T} \quad \text{with} \quad M = |\mathcal{T}|$$

$$\sum_{\tau \in [t-1, \max(t-3, 1)]} v_{3,\tau} \leq (1 - v_{4,t})M \quad \forall t \in \mathcal{T} \quad \text{with} \quad M = |\mathcal{T}|$$

Part II

Theory and Concepts

2.a (3 Points)

What is the main difference between Economic Dispatch and Unit Commitment problems? Explain in 2-3 sentences.

Solution

Economic Dispatch optimizes power generation for a single time step assuming all generators are already committed to operation, focusing only on power output levels. Unit Commitment extends this by adding binary decisions for generator on/off status, considering multiple time periods, and including operational constraints such as start-up costs, minimum up/down times, and ramp rate limits. This makes Unit Commitment more realistic but computationally more complex.

2.b (3 Points)

Explain what a “tight formulation” means in the context of Mixed-Integer Linear Programming (MILP) problems like Unit Commitment.

Solution

A tight formulation in MILP has a small gap between the optimal solution of the linear programming (LP) relaxation and the integer solution. This is important because it helps solvers find optimal solutions faster, provides better lower bounds for branch-and-bound algorithms, and reduces the search space for integer solutions. For example, including both minimum and maximum power constraints for generators creates a tighter formulation than only including maximum constraints.

2.c (3 Points)

What is the purpose of storage systems in energy system optimization? Name three key constraints that are typically included in storage modeling.

Solution

Storage systems in energy system optimization serve to balance supply and demand over time, store excess renewable energy, and provide flexibility to the power system. Three key constraints typically included in storage modeling are:

1. **Energy balance constraint**: Links energy levels between consecutive time periods considering charging/discharging efficiency and self-discharge
2. **Energy capacity limits**: Ensures storage energy level stays within minimum and maximum bounds
3. **Power limits and mutual exclusion**: Prevents simultaneous charging and discharging while respecting maximum charging/discharging power rates

2.d (3 Points)

Explain the concept of “two-stage stochastic programming” in the context of energy system design problems.

Solution

Two-stage stochastic programming in energy system design separates decisions into two stages: first-stage (here-and-now) decisions made before uncertainty is revealed (e.g., investment in generation capacity, storage size), and second-stage (wait-and-see) decisions made after uncertainty is revealed (e.g., operational decisions like power dispatch, storage operation). This approach allows for robust investment decisions by considering multiple scenarios of renewable generation, demand patterns, and price variations, leading to more resilient energy system designs.

2.e (3 Points)

What are the main advantages of using Julia and JuMP for energy system optimization compared to other programming languages and modeling frameworks?

Solution

Julia and JuMP offer several advantages for energy system optimization: high performance comparable to compiled languages while maintaining ease of use, seamless integration with multiple solvers (HiGHS, Gurobi, etc.), excellent support for mathematical notation that closely resembles mathematical formulations, efficient handling of large-scale optimization problems, and strong ecosystem for data analysis and visualization. Additionally, JuMP's syntax is intuitive for expressing mathematical models, making it easier to translate theoretical formulations into working code.

Part III

Julia Programming

Hints

ℹ Programming Tips

- Pay attention to variable names and consistency
- Check for proper package imports
- Verify array indexing
- Remember to use the correct comparison operators
- Make sure to use proper JuMP syntax

3.a (8 Points)

The following Julia code contains four errors related to implementing an Economic Dispatch model. Identify and briefly describe each error.

```
# Load the necessary packages
using JuMP
using HiGHS

# Define the size of the problem instance
nrGenerators = length(generatorCosts)
nrWindTurbines = length(windForecast)

# Create model instance
dispatch = Model(HiGHS.Optimizer)

# Define variables
@variable(dispatch_model, p_g[g = 1:nrGenerators] >= 0)
@variable(dispatch_model, p_w[w = 1:nrWindTurbines] >= 0)

# Define objective
@objective(dispatch_model, Max,
    sum(generatorCosts[g] * p_g[g] for g in 1:nrGenerators)
    )

# Define the constraints
@constraint(dispatch_model,
    power_balance,
    sum(p_g[g] for g in 1:nrGenerators) + sum(p_w[w] for w in 1:nrWindTurbines) ===
     ↪   demand
    )

@constraint(dispatch_model,
    generator_limits[g=1:nrGenerators],
    p_g[g] <= maxPower[g]
    )

@constraint(dispatch_model,
    wind_limits[w=1:nrWindTurbines],
    p_w[w] <= windForecast[w]
    )

# Start optimization
solve_model(dispatch_model)
```

Solution

1. Model is called ‘dispatch’ but referenced as ‘dispatch_model’
2. Objective should minimize costs, not maximize
3. Power balance constraint uses ‘===’ instead of ‘==’
4. ‘solve_model’ should be ‘optimize!’

3.b (4 Points)

Write the Julia code to define a binary variable for generator commitment status in a Unit Commitment model. The variable should indicate whether generator g is on at time t.

Solution

```
@variable(model, u[g=1:nrGenerators, t=1:nrTimePeriods], Bin)
```

3.c (3 Points)

Write the Julia constraint that links generator power output to its commitment status, ensuring that if a generator is off (u[g,t] = 0), its power output must be zero.

Solution

```
@constraint(model, generator_limits[g=1:nrGenerators, t=1:nrTimePeriods],
    p_min[g] * u[g,t] <= p[g,t] <= p_max[g] * u[g,t])
```

Wrap Up

Key Takeaways

! Remember

- Time management is crucial (1.5 minutes per point)
- Read questions carefully
- Show your mathematical work clearly
- Use your cheat sheet strategically
- Double-check your Julia syntax

The end

ℹ Good luck with your exam!

This test exam covers the main concepts from the Energy System Optimization course. Make sure to review the course materials and practice implementing the models in Julia.

Questions?

Literature

Literature

For more information about energy system optimization and Julia programming, refer to the course materials and the literature list in the general section of this course.