

Homework 3

Spencer Egan

March 17, 2023

1 Problem Formulation

1.1 Required Sets and Parameters

For this greenfield capacity expansion problem, we wish to determine the optimal levels of installed capacity γ_g (in MW) for each type of generator $g \in \mathcal{G}$ where \mathcal{G} is the set of five generator types as described in the provided csv file.

Given an annualized per-MW investment cost σ_g for each generator type we can calculate the total annualized system investment cost as $\sum_{g \in \mathcal{G}} \gamma_g \sigma_g$. Given the per-MW annualized fixed O&M costs ξ_g for each generator type, we can calculate the total fixed O&M cost for our capacity mix as $\sum_{g \in \mathcal{G}} \gamma_g \xi_g$.

The remaining system costs are variable i.e. they are dependent on the generation level of each generator type at each time period. While the purpose of capacity expansion modeling is to determine the optimal capacity mix of each generator type, we also need to optimize the dispatch of each generator type given a capacity mix γ . Given a planning horizon \mathcal{T} with discrete hourly timesteps $t \in \mathcal{T}$ we will consider the power output $P_{g,t}$ in MW for each generator type g at timestep t .

Variable per-MWh O&M costs for each generator type χ_g can be used to determine the total variable O&M cost of a given capacity mix and dispatch schedule as $\sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} P_{g,t} \chi_g$. Similarly, given per-MWh fuel costs ν_g , the total fuel cost over the planning horizon is given by $\sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} P_{g,t} \nu_g$.

For each time step t we need to ensure that each generator type does not produce more than its available capacity. For thermal plants, this is equal to the type's installed capacity γ_g at all time steps, while for wind and solar plants this is equal to $A_{g,t} \gamma_g$ where $A_{g,t}$ is the resource availability (described as a capacity factor in the assignment) for generator type g at time step t .

Besides the generator output constraints described above, we also need to impose a power balance constraint for all times $t \in \mathcal{T}$. We can write this constraint as

$$\sum_{g \in \mathcal{G}} P_{g,t} = d_t \quad \forall \quad t \in \mathcal{T}.$$

where d_t is the system demand at time step t .

1.2 Decision Variables

The decision variables of this problem are the installed capacities of each generator type γ_g and the power output of each generator at each time step $P_{g,t}$.

1.3 The Objective function

Before we state the objective function for this problem, we will state the problem parameters and variables described in section 1 in vector-matrix form for a concise formulation.

$$\begin{aligned} n &= |\mathcal{G}| \\ s &= |\mathcal{T}| \\ \sigma &\in \mathbb{R}^n \\ \xi &\in \mathbb{R}^n \\ \chi &\in \mathbb{R}^n \\ v &\in \mathbb{R}^n \\ d &\in \mathbb{R}^s \\ A &\in \mathbb{R}^{s \times n} \\ \gamma &\in \mathbb{R}^n \\ P &\in \mathbb{R}^{s \times n} \end{aligned}$$

From this we can formulate our objective function $f(\gamma, P)$ as a minimization of total system costs in vector-matrix form as

$$f(\gamma, P) = (\sigma + \xi)^\top \gamma + 1_s^\top P(\chi + v)$$

where $1_s \in \mathbb{R}^s$ is the vector of all ones.

1.4 Constraints

We have two sets of constraints for this first problem. The first is the min and max generator output constraint, and the second is the power balance constraint, both described in the Required Sets and Parameters subsection. In vector-matrix form we can write these constraints as

$$\begin{aligned} 0_{s \times n} &\leq P \leq A \mathbf{diag}(\gamma) \\ P 1_n &= d \end{aligned}$$

where $0_{s \times n} \in \mathbb{R}^{s \times n}$ is a matrix of all zeros, and $1_s \in \mathbb{R}^s$ is a vector of all ones. Here we set the columns of A representing the thermal plants as all ones representing their availability at full capacity at all times.

2 Implementation

We will implement the optimization problem outlined above in Julia using the versatile JuMP package.

```

using CSV, LinearAlgebra, DataFrames, PrettyTables
using JuMP, Gurobi
root = dirname(@__FILE__)

plants_path = joinpath(root, "plant_characteristics.csv")
timeseries_path = joinpath(root, "timeseries.csv")
plants = CSV.read(plants_path, DataFrame)
plants.lifetime = [30,30,60,30,30]
sdr = 0.03 #social discount rate
pmt(capex, yrs) = sdr*capex/(1-(1+sdr)^(-yrs))
transform!(plants, [:cap_cost, :lifetime] => ByRow(pmt) => :ann_capex)
timeseries = CSV.read(timeseries_path, DataFrame)

function capacity_expansion(
    plants::DataFrame,
    timeseries::DataFrame
)
    plant_types = plants.plant_char

    timestep_hours = 1.0
    timeseries_years = (timestep_hours * nrow(timeseries))/8760

    n = nrow(plants)
    s = nrow(timeseries)
     $\sigma$  = plants.ann_capex * 1000 * timeseries_years
     $\xi$  = plants.fom_cost * 1000 * timeseries_years
     $\chi$  = plants.vom_cost
     $v$  = plants.fuel_cost
    d = timeseries.load_MW

    A = ones((s,n))
    A[:, 4] = timeseries.solar_cf
    A[:, 5] = timeseries.wind_cf

    capacityexp = Model(Gurobi.Optimizer)

    @variable(capacityexp,  $\gamma[1:n] \geq 0$ )
    @variable(capacityexp, P[1:s, 1:n]  $\geq 0$ )

    @objective(capacityexp, Min, ( $\sigma + \xi$ )'* $\gamma$  + ones(s)'*P*( $\chi + v$ ))

    @constraint(capacityexp, P . $\leq$  A * Diagonal( $\gamma$ ))
    @constraint(capacityexp, P*ones(n) . $=$  d)

    optimize!(capacityexp)
    optimal_fleet = DataFrame(plant = plant_types, Capacity_GW = value( $\gamma$ ) / 1000.0)
    optimal_cost = objective_value(capacityexp)
    return optimal_fleet, optimal_cost
end

```

Note that absent additional information in the problem, we are amortizing the capital costs of the plants over their technical lifetimes as described in the NREL [ATB database](#), assuming a rate equal to the "middle-of-the-road" social discount rate of 3% listed in the February 2021 [US Interagency Report on the Social Cost of Greenhouse Gases](#). This should give reasonable results to first approximation.

2.1 Base Case Solution

The objective function is a measure of total annualized costs in the optimal solution including all annual capital, fixed, and variable costs. The primal solution value gives us this cost.

```
optimal_fleet, optimal_cost = capacity_expansion(plants, timeseries)

println("Optimal annual cost = \$ ", round(optimal_cost/1e9; digits=1), " B")
```

Optimal annual cost = \$ 24.1 B

For our system with previously stated amortization parameters, the total cost appears to be about 24.1 billion dollars.

The optimal solution, shown below, provides the least-cost fleet mix. We see that given the cost assumptions provided and assumed, the optimal solution includes no coal or nuclear, lots of CCGT and wind, and a bit of solar. The capacity of each plant type is shown in GW.

```
pretty_table(optimal_fleet, nosubheader = true)
```

plant	Capacity_GW
coal	0.0
gas-cc	70.4977
nuclear	0.0
solar	1.73436
wind	38.5635

If we only consider a portion of the year, say the first 720 hours, the optimal fleet mix will be different.

```
ts_jan = timeseries[1:720, :]
fleet_jan, cost_reduced = capacity_expansion(plants, ts_jan)

pretty_table(fleet_jan, nosubheader = true)
```

plant	Capacity_GW
coal	0.0
gas-cc	58.598
nuclear	0.0
solar	0.0
wind	0.0

Clearly wind and solar don't have the same value to the system in January than they do over the whole year, as the solution for just this period calls for all gas! Let's take a look at a one month period in summer:

```
ts_summer = timeseries[5000:5720, :]
fleet_summer, cost_reduced = capacity_expansion(plants, ts_summer)

pretty_table(fleet_summer, nosubheader = true)
```

plant	Capacity_GW
-------	-------------

coal	0.0
gas-cc	68.1517
nuclear	0.0
solar	13.9523
wind	2.37564

Clearly solar has much more value in the summer months than it's average value to the system over the course of the whole year.

We note that these results are very sensitive to the discount rate assumed. When using a discount rate of 0, the solution called for quite a bit of nuclear, and more solar than wind when optimizing over the whole year. This makes sense as both solar and nuclear are capex heavy, and therefore look much more favorable when no interest is required to be paid.

2.2 Adding a CO₂ Cap

We can add an annual CO₂ cap to our model quite easily. Given the CO₂ emissions rate (measured in tons of CO₂ per MWh) for each plant type ρ_g , and an annual CO₂ cap c (measured in tons), we need to ensure that $\sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} P_{g,t} \rho_g \leq c$. We can express the constraint in vector-matrix form as $\mathbf{1}_s^\top P \rho \leq c$.

We implement this constraint in the code below:

```
function capacity_expansion_CO2(
    plants::DataFrame,
    timeseries::DataFrame,
    ts_cap::AbstractFloat
)
    plant_types = plants.plant_char

    timestep_hours = 1.0
    timeseries_years = (timestep_hours * nrow(timeseries))/8760

    c = ts_cap

    n = nrow(plants)
    s = nrow(timeseries)
    σ = plants.ann_capex * 1000 * timeseries_years
    ξ = plants.fom_cost * 1000 * timeseries_years
    ρ = plants.emit_rate
    χ = plants.vom_cost
    v = plants.fuel_cost
    d = timeseries.load_MW

    A = ones((s,n))
    A[:, 4] = timeseries.solar_cf
    A[:, 5] = timeseries.wind_cf

    capacityexp = Model(Gurobi.Optimizer)

    @variable(capacityexp, γ[1:n] ≥ 0)
    @variable(capacityexp, P[1:s, 1:n] ≥ 0)

    @objective(capacityexp, Min, (σ + ξ)'*γ + ones(s)'*P*(χ + v))

    @constraint(capacityexp, P .≤ A * Diagonal(γ))
```

```

@constraint(capacityexp, pbal, P*ones(n) .== d)
@constraint(capacityexp, ccap, ones(s)'*P*ρ ≤ c)

optimize!(capacityexp)
optimal_fleet = DataFrame(plant = plant_types, Capacity_GW = value.(γ) / 1000.0)
optimal_cost = objective_value(capacityexp)
cprice = dual(ccap)
return optimal_fleet, optimal_cost, cprice
end

```

If this cap was prudently chosen by regulators to depress emissions (i.e. emissions under the business-as-usual case was higher than the proposed cap), then we would expect to see more wind, solar, and nuclear in the optimal fleet mix, as well as a higher overall cost.

```
fleet_CO2, cost_CO2, cprice = capacity_expansion_CO2(plants, timeseries, 100e6)
```

```
println("Optimal annual cost with 100 MMT annual emissions cap = \$ ",
round(cost_CO2/1e9; digits=1), " B")
pretty_table(fleet_CO2, nosubheader = true)
```

Optimal annual cost with 100 MMT annual emissions cap = \$ 24.2 B

plant	Capacity_GW
coal	0.0
gas-cc	65.3653
nuclear	5.11829
solar	1.72199
wind	39.8013

Comparing this solution to the solution without a cap, we see that it is optimal to replace about 5GW of gas plants with nuclear plants for firm "baseload." The amount of wind and solar built in each scenario is nearly the same. Also note how the total system price has only increase slightly over the no-carbon-cap solution.

We should also pause here to mention that the dispatch at each timestep would not necessarily follow from the usual merit order. There is effectively a carbon price that is applied to all timesteps that we can assess with the dual variable related to the carbon cap constraint. (Note that this dual variable is negative since inceasing the carbon cap decreases the system cost).

```
println("Shadow carbon price = \$", round(-1*cprice; digits = 2), " per ton")
```

Shadow carbon price = \$6.76 per ton

This price is pretty low, reflecting the fact that it was not too much more expensive to switch some gas to nuclear and bring emissions down under the cap.

2.3 Reflections on results

These results were not too suprising. All across the country we see coal retirements occuring at a rapid pace, with very few new coal plants being commissioned. There is much more wind in Texas than solar (though lookiong at interconnection queues, that might not be the case in a couple of years). Furthermore, very few nuclear plants are being built anywhere in the US right now, and there are none being built commercially in liberalized markets. However,

an interest in clean-firm resources has renewed interest in nuclear by private parties (i.e. not utilities with a captive ratebase), so it is not all that suprising that an emmissions cap brought some nuclear online in the optimal solution.

I was a bit suprised that the emmissions in the optimal scenario generated were not that much higher than 100 MMT of CO₂. A quick check with the [EIA](#) shows that in 2020, the electricity sector in the state of Texas emitted 179 MMT of CO₂. Because ERCOT covers slightly less than the whole state, and because that data is a couple of years old, *and* given the declining cost curves of wind and solar, we would expect the existing fleet to be more polluting than a greenfield capacity expansion solution. But it is still a bit suprising how easily we were able to get under this cap.

There are likely a couple of additional reasons emmissions were lower in our model relative to the actual ERCOT fleet. First, unit committment of thermal generators (nearly all of which are carbon emitters) often requires some uneconomic operation and therefore extra emmissions relative to the pure economic dispatch problem we are modeling. This is likely part of the reason why emmissions are lower in our solution, and incorporating start-up, shut-down, and minimum-run times of thermal generators into our problem (turning it from an LP to a more computationally-difficult MILP) could improve the accuracy of our model.

Second, it is well known that transmission constraints in Texas lead to the curtailment of wind and solar power and zero or negative LMPs. This model assumes a "copper plate" without any transmission constraints, and therefore the annual production of wind and solar in our model is likely meaningfully higher than what one might expect in the real world. Even a simplified transmission topology model with import and export zones could greatly improve our model with DCOPF equations incorporated into constraints.

Apart from these two improvements to the model, we could of course better incorporate the financing parameters of these systems, as well as any additional subsidies that might exist for certain types of generation. At the very least we could model the annualized capital costs using interest rates reflective of the risk profile of each generator type, as is commonly done in Net-CONE analyses for other ISO capacity market mechanisms.

A more accurate model would also incorporate reserve requirements in the hourly dispatch solutions and possibly also some sort of resource adequacy requirement either through a capacity reserve margin, or by limiting the expected unserved energy, or perhaps through the an analysis of the fleet's ELCC in a probabilistic model.

Finally, an additional refinement to this model could be to incorporate wind and solar availability profiles from several regions and allow for the solver to find the optimal capacity of each renewable type in each region. This could of course be incorporated into the transmission constraints with different resource profiles for different export zones.