Energy Economics & Climate Policy Homework

by Mason Ross Hayes on 2022-02-26

for M2S2 EECP at Toulouse School of Economics

done in julia using Pluto.jl🎈

- View the code💻
- View the interactive notebook📓
- Download the PDF📂

📖 Table of Contents

The solution

- Weights $f(\theta) =$ (0.315, 0.651, 0.034) for $\theta = L, M, H$, respectively.
- $b^* =$ 5.17 gives weighted elasticity of 0.01 at price of 100€/MWh.
- The optimal $K^* \approx$ 83593.0 MW, which corresponds to a price of 272.147 €/MWh.
  - With CCGT of power 450 MW, this corresponds to 186.0 power stations.

Solving the exercise

Let's create a dataframe of the data we are given:

df =

	θ	a	hours	fuel_cost	cap_cost
1	"L"	40000.0	2760.0	72.0	6.85
2	"M"	60000.0	5700.0	72.0	6.85
3	"H"	85000.0	300.0	72.0	6.85

```
df = sort(
    DataFrame(
    θ = ["H", "M", "L"],
    a = [85000.0, 60000.0, 40000.0],
    hours = [300.0, 5700.0, 2760.0],
    fuel_cost = repeat([72.0], 3),
    cap_cost = repeat([6.85], 3)
    ), :a)
```

Now let's set the variable price and the parameter b to be flexible:

p = 100.0

Using the slider below, we can adjust b until the weighted elasticity equals 0.01 when price = 100 euros per MWh:

b = 5.17

weighted_elasticity = 0.00999

Case θ = 3

There are 3 potential cases: $\theta \in [L, M, H]$, corresponding to cases 1, 2, and 3, respectively. For each case, we have the demand constraint that the optimal capacity must be greater than the demand at the peak period.

Let's refer to any $\theta \in [L, M, H]$ as a *period*. Then, for any period, the demand in that period must be less than or equal to the total capacity. If this is satisfied in period 3, then it is always satisfied, since period 3 corresponds to $\theta = H$, the period of highest demand.

The optimal price

When $K = 83593.0$ and $\theta = 3$, then the optimal price is: 272.147

Is optimal capacity 83593.0 less than demand 84483.0?

true

The optimal K̂

```
K̂ =  [54147.0, 60825.0, 83593.0]
```

```
# the values were obtained by selecting Case θ and then adjusting K in the cell
below to solve the equation

K̂ = [54147., 60825., 83593.]
```

83593.0

Adjust K above so that the expected dual price of the capacity equals its fixed cost per hour (6.85€)

```
6.854
```

```
# ↑ adjust the value of K above ↑ to make this cell equal to 6.85
round(df.time_share[θ:3] .* (P((df.a[θ:3], b, K)) - df.fuel_cost[θ:3]) |> sum,
digits = 3)
```

Case 1

When the capacity constraint is binding even under low demand ($\theta = L$), then the optimal $K^* \approx$ 54147.0 MW. But this is higher than 39483.0, so cannot be optimal.

Case 2

So, we know we need more capacity than in Case 1. Let's consider case 2, under periods of medium demand ($\theta = M$). Then, the optimal $K^* \approx$ 60825.0.

But again, this is higher than the demand 59483.0.

Case 3

So, we know we need more capacity than in Case 2 as well. Now in case 3, under periods of peak demand ($\theta = H$). Then, the optimal $K^* \approx$ 83593.0 which lies within the range of medium demand and peak demand, $K \in$ (59483.0, 84483.0).

Functions and further calculations

```
using DataFrames   , StatsBase   , PlutoUI   , Dates
```

Define functions for demand, elasticity, surplus, and price.

```julia
begin
    D((a, b, p)) = a .- b.*p # Demand function
    ε_w((b,p,D,s)) = b.*p./D .* s # = weighted elasticity function
    S((a, b, p)) = 0.5 .* (a.^2 .- (b .* p).^2) # gross surplus
    P((a, b, K)) = (a .- K)/b # price function
end;
```

And now call the functions for demand and elasticity and store them in the data frame

```julia
begin # Set the time share = hours/total_hours; the demand function; and elasticity function
    df.time_share .= df.hours ./ sum(df.hours)
    df.D = D((df.a, b, p))
    df.elasticity .= ε_w((b,p,df.D, df.time_share))
end;
```

```
[20258.5, 30258.5, 42758.5]
```

```julia
begin
    df.served_demand .= S((df.a, b, p))
    df.VOLL = df.served_demand ./ df.D
end # returns VOLL for the chosen theta, prices, etc
```

```
9152.349086757991
```

```julia
# the mean weighted VOLL
mean((df.VOLL .- (df.fuel_cost .+ df.cap_cost)) .* df.time_share)
```