BEE 4750/5750 Homework 3

Akanksha Srivastava (as2752)

2022-10-15

Problem 1

Problem 1.1

For this problem, the decision variables are:

1. The installed capacity (MW) of generator type g: x_g
2. The production (MW) from generator type g in period t: $y_{g,t}$
3. The nonserved load (MW) in each period t: nse_t

Here, generators types include geothermal, coal, CCGT, CT, wind, and solar. Periods include every hour from 1 to 24.

We can begin to implement our model in Julia:

```
julia> using JuMP

julia> using HiGHS

julia> gencap = Model(HiGHS.Optimizer);

julia> generators = ["geo","coal","ccgt","ct","wind","solar"];

julia> G = 1:length(generators);

julia> T = 1:length(hours);

julia> @variable(gencap,x[G]>=0);

julia> @variable(gencap,y[G,T]>=0);

julia> @variable(gencap,ns[T]>=0);
```

Problem 1.2

The objective function is to minimize the total cost of generation, which is the sum of the investment cost and the operating cost. If C_g^{INV} is the investment cost for generator g, C_g^{OP} is the operating cost for generator g, and L_t is the length of time period t:

$$minZ = \sum_g C_g^{INV} x_g + \sum_g \sum_t L_t C_g^{OP} y_{g,t} + \sum_t NSE_{cost} nse_t$$

In Julia, this becomes:

```
julia> @objective(gencap,Min,investment_cost'*x + 365*(22*sum(y[2,:]) +
35*sum(y[3,:]) + 45*sum(y[4,:]) + 1000*sum(ns)));
```

Problem 1.3

Derive all relevant constraints (you don't need to write them all out, but they should all be represented through your notation). Make sure to include any needed justifications or derivations. Why is your set of constraints complete?

The constraints of this problem include installed capacity and availability, the need to serve the load in all time frames, and nonnegativity. These can be written out as follows:

1. Installed Capacity and Availability: In a given time period, our $y_{g,t}$ cannot exceed the installed capacity multiplied by the capacity factor, as it is not possibly for the generator to produce more than that quantity. When CF_g is the capacity factor for a given generator,

$$y_{g,t} \leq CF_g * x_g$$

2. Serving Load: The sum of the generated electricity and the nonserved load should equal the demand for that time period, D_t.

$$\sum_g y_{g,t} + nse_t = D_t$$

3. Nonnegativity: Our installed capacities, productions, and nonserved loads cannot be negative (this is physically not possible).

$$x_g \geq 0, y_{g,t} \geq 0, nse_t \geq 0$$

.

In Julia, this translates to:

```
julia> # Availability Constraints
       avail = zeros(length(G),length(T));

julia> avail[1,:] = thermal_cf[1]*ones(length(T));
```

```
julia> avail[2:4,:] = thermal_cf[2:4]*ones(length(T))';

julia> avail[5,:] = wind_cf;

julia> avail[6,:] = solar_cf;

julia> @constraint(gencap,availability[g in G, t in T], y[g,t] <=
avail[g,t]*x[g]);

julia> # Load Constraints
       @constraint(gencap,load[t in T], sum(y[:,t]) + ns[t] == demand[t]);
```

Problem 1.4

```
julia> optimize!(gencap)
Presolving model
156 rows, 162 cols, 420 nonzeros
156 rows, 162 cols, 420 nonzeros
Presolve : Reductions: rows 156(-12); columns 162(-12); elements 420(-24)
Solving the presolved LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 24(60321.5) 0s
        120     9.1214221224e+08 Pr: 0(0) 0s
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Simplex   iterations: 120
Objective value     :  9.1214221224e+08
HiGHS run time      :          0.00
```

Problem 1.5

Answering the relevant questions using the following block of code:

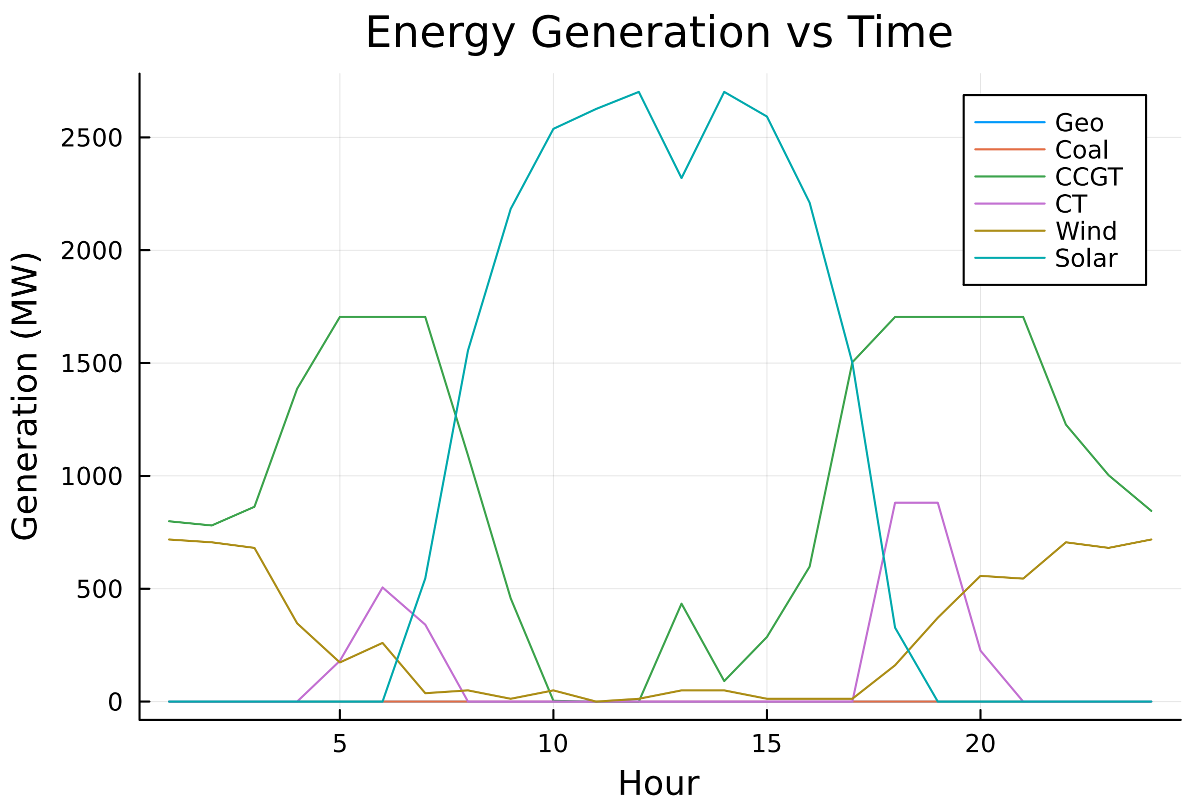
```
julia> # Installed Capacity for Each Generating Plant
       installed_capacity = value.(x);

julia> # Total Cost
       total_cost = objective_value(gencap);

julia> # Nonserved Energy
       nonserved_energy_total = sum(value.(ns));
```

Therefore we get that the installed capacities for geothermal, coal, CCGT, CT, wind and solar are 0 MW, 0 MW, 1704 MW, 881 MW, 1238 MW, and 2729 MW respectively. The total cost for the expansion in this case would be 912,142,212 dollars, and the nonserved load would be 0 MW.

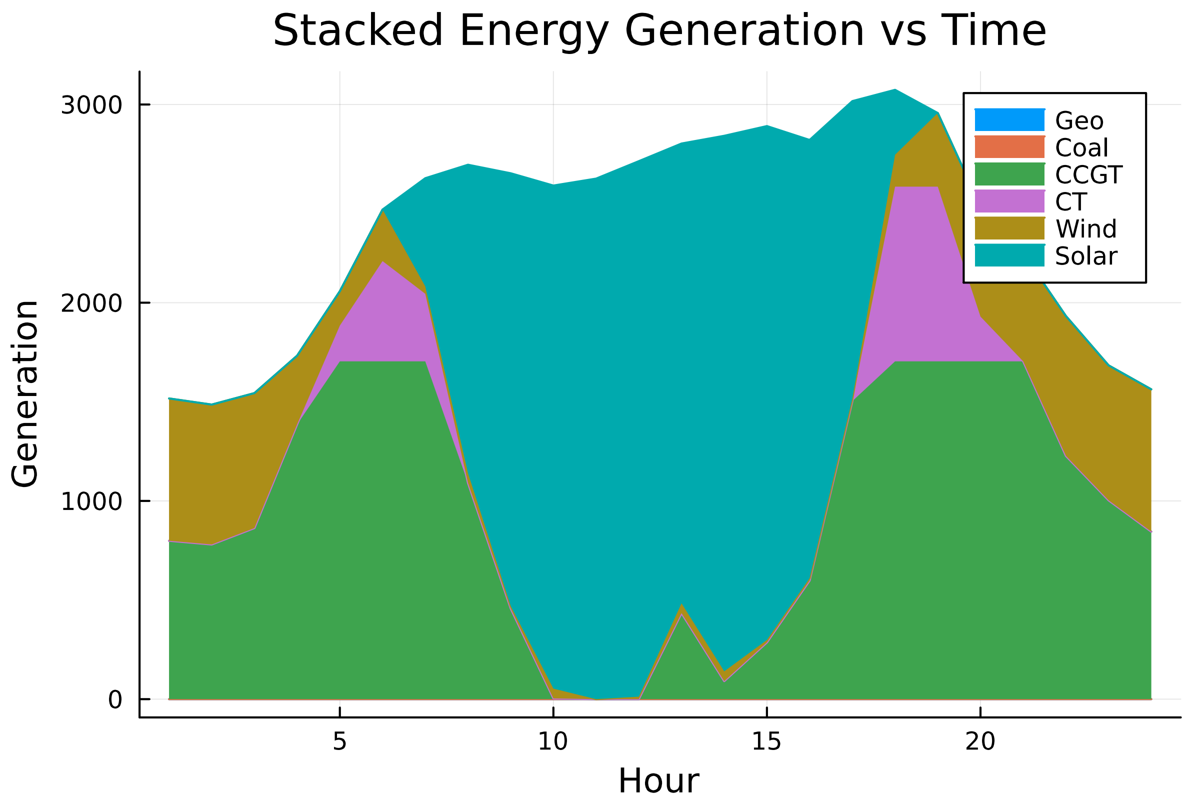
Problem 1.6

```
julia> # Assign generations to generator type
```

```
        served_load = ones(length(G),length(T));
served_load[:,:]=value.(y)[:,:];

julia> using Plots

julia> # Plotting Line Plots for Raw Amounts

plot(hours,served_load[1,:],label="Geo",xlabel="Hour",ylabel="Generation
(MW)",title="Energy Generation vs Time");

julia> plot!(hours,served_load[2,:],label="Coal");

julia> plot!(hours,served_load[3,:],label="CCGT");

julia> plot!(hours,served_load[4,:],label="CT");

julia> plot!(hours,served_load[5,:],label="Wind");

julia> plot!(hours,served_load[6,:],label="Solar")
```




```
julia> # Plotting Area Plot for Overall Contributions

areaplot(hours,served_load',label=permutedims(["Geo","Coal","CCGT","CT","Wind","Solar"]),xla
Energy Generation vs Time")
```



Notably, geothermal and coal are not providing any energy to the mix at any time. When solar has a high CF (i.e. in the middle of the day) it is producing the majority of the electricity. When it does not, then the CCGT generator, which is not dependent on time of day, is taking up the majority of the generation.

Problem 2

Problem 2.1

Here, the only change that would be required would be adding an additional constraint on the total amount of emissions. The following Julia code block creates another model where the only change is this constraint.

```
julia> gencap_CO2 = Model(HiGHS.Optimizer);

julia> @variable(gencap_CO2,x_CO2[G]>=0);

julia> @variable(gencap_CO2,y_CO2[G,T]>=0);

julia> @variable(gencap_CO2,ns_CO2[T]>=0);

julia> @objective(gencap_CO2,Min,investment_cost'*x_CO2 +
365*(22*sum(y_CO2[2,:]) + 35*sum(y_CO2[3,:]) + 45*sum(y_CO2[4,:]) +
1000*sum(ns_CO2)));

julia> @constraint(gencap_CO2,availability[g in G, t in T], y_CO2[g,t] <=
avail[g,t]*x_CO2[g]);
```

```
julia> @constraint(gencap_CO2,load[t in T], sum(y_CO2[:,t]) + ns_CO2[t] ==
demand[t]);

julia> #Adding in CO2 Constraint
       @constraint(gencap_CO2,emission_limit, 0.365*sum(co2_emissions'*y_CO2)
<= 10^3);
```

Problem 2.2

```
julia> optimize!(gencap_CO2);
Presolving model
157 rows, 162 cols, 492 nonzeros
157 rows, 162 cols, 492 nonzeros
Presolve : Reductions: rows 157(-12); columns 162(-12); elements 492(-24)
Solving the presolved LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 24(41167) 0s
         93     1.1334512774e+09 Pr: 0(0) 0s
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Simplex   iterations: 93
Objective value     :  1.1334512774e+09
HiGHS run time      :          0.00
```

Problem 2.3

Answering the relevant questions using the following block of code:

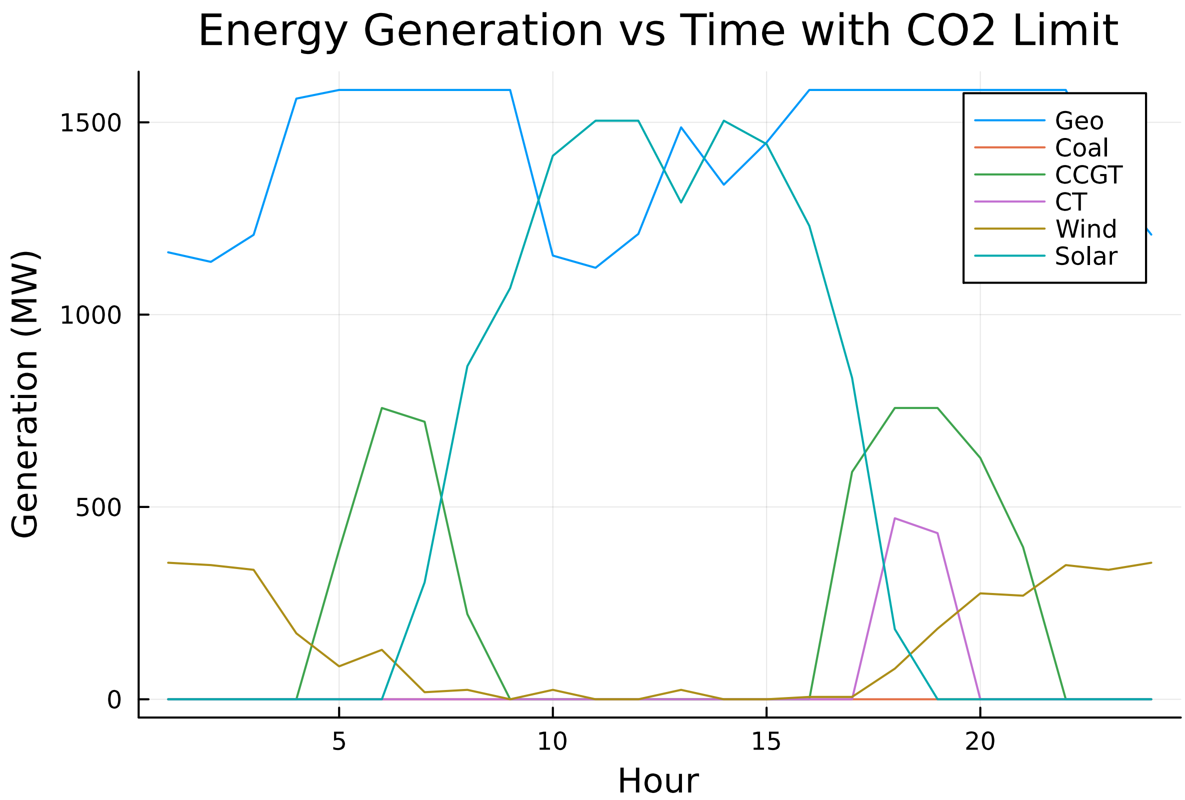
```
julia> # Installed Capacity for Each Generating Plant
       installed_capacity_CO2 = value.(x_CO2);
```

With the CO2 limit, the installed capacities for geothermal, coal, CCGT, CT, wind and solar are 1668 MW, 0 MW, 757 MW, 470 MW, 611 MW, and 1519 MW respectively. This is notably different from the model in Problem 1 because the geothermal installed capacity becomes nonzero (and quite high), and the CCGT and CT capacities decrease significantly. The wind and solar capacities also decrease.

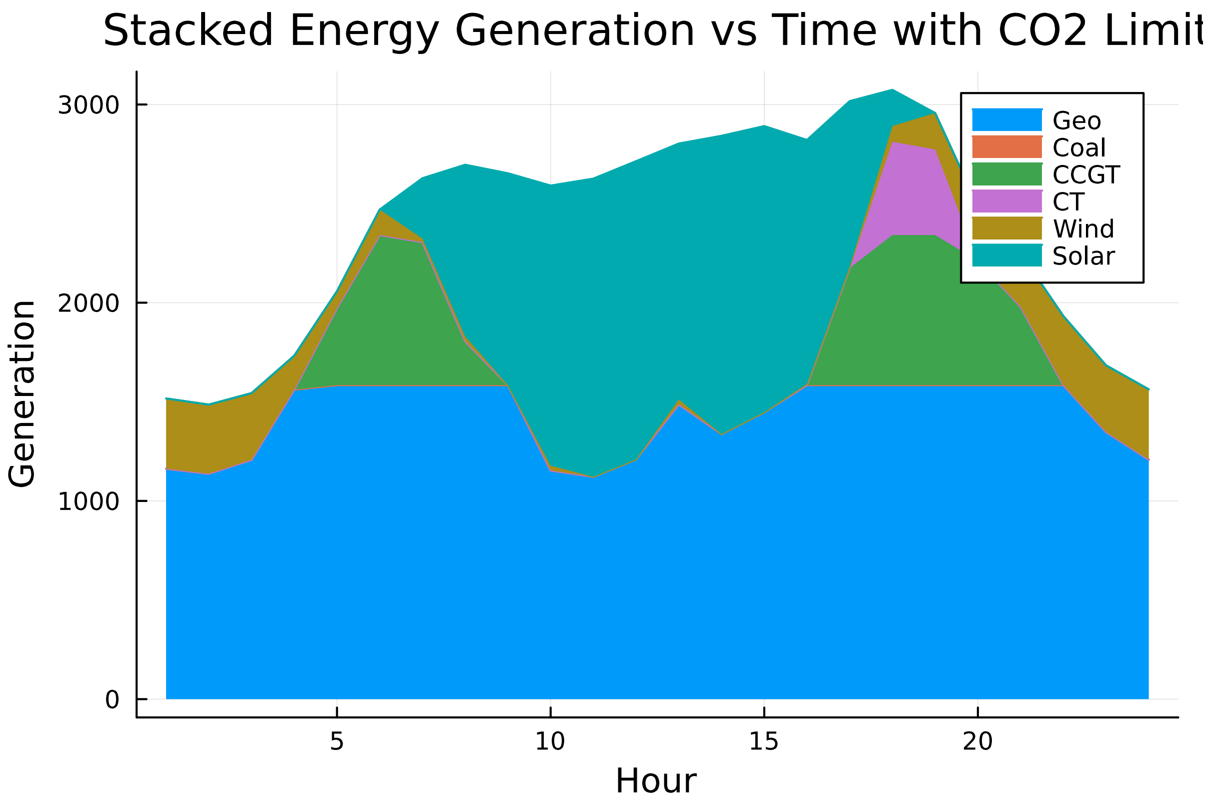
Problem 2.4

```
julia> # Assign generations to generator type
       served_load_CO2 = ones(length(G),length(T));
served_load_CO2[:,:]=value.(y_CO2)[:,:];

julia> using Plots

julia> # Plotting Line Plots for Raw Amounts

plot(hours,served_load_CO2[1,:],label="Geo",xlabel="Hour",ylabel="Generation
(MW)",title="Energy Generation vs Time with CO2 Limit");
```

```
julia> plot!(hours,served_load_CO2[2,:],label="Coal");

julia> plot!(hours,served_load_CO2[3,:],label="CCGT");

julia> plot!(hours,served_load_CO2[4,:],label="CT");

julia> plot!(hours,served_load_CO2[5,:],label="Wind");

julia> plot!(hours,served_load_CO2[6,:],label="Solar")
```




```
julia> # Plotting Area Plot for Overall Contributions

areaplot(hours,served_load_CO2',label=permutedims(["Geo","Coal","CCGT","CT","Wind","Solar"])
Energy Generation vs Time with CO2 Limit")
```



The key difference is the contribution of the geothermal generator. Of the remaining generation types, the breakdown or fairly similar (though not the same). This is why all generations but the coal (which stayed at 0 MW) decreased.

Problem 2.5

We can determine the value to the utility of allowing it to emit an additional 1000 tCO2/yr by implemented the following code in Julia. Note that in order to find the added value of 1000 tCO2/yr instead of 1 tCO2/yr, we had to alter the CO2 constraint so that the limit was in terms of 1000 tCO2/yr.

```
julia> shadow_price.(emission_limit);
```

This tells us that the added value (i.e. the total cost would decrease by) 156,844.22 dollars if the limit was increased by 1000 tCO2/yr.

References

1. Using areaplot(): https://docs.juliaplots.org/dev/api/#Plots.areaplot