Treball de Fi de Màster

Màster universitari en enginyeria industrial

Anàlisi de la situació energètica espanyola al 2040

ANNEX

Autor: Sara Lujan Tutusaus
Director: Jordi Freixa Terradas

Convocatòria: Juliol 2019

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

1. Codi desenvolupat per trobar la situació òptima de mix energètic

```
#!/usr/bin/env python3

"""
-------------------------------------------------------------------------
------------------
Programa per calcular el mix d'energia produïda a partir de diferents
condicions meteorologiques i de contorn del sistema
-------------------------------------------------------------------------
------------------
"""

import numpy as np
import matplotlib.pyplot as plt
import xlwt

"""
-------------------------------------------------------------------------
------------------
Importació i definició de les condicions del contorn del sistema
-------------------------------------------------------------------------
------------------
"""

data = np.loadtxt('data.dat')
nuclear = data[0]*0.0 # kW per capita
hydro = data[1] # kW
Ppumpmax = data[2] # kW
Pturbmax = data[3] # kW
Estorageaverage = data[4] # kW
Estoragemax = data[5] # kW
costsolar = data[6] # euros/kW
costwind = data[7] # euros/kW
Pinstcombcycle = data[8] #kW
Pinstsolar = data[9] #kW
Pinstwind = data[10] #kW
population = data[11]
investment = data[12] # euros per capita
CO2factor = data[13] #kgCO2/kWhcombcycle

Pdemand = np.loadtxt('demand5v.dat') #kW
print('demand =', sum(Pdemand))

"""
-------------------------------------------------------------------------
------------------
Inicialització de les variables auxiliars
-------------------------------------------------------------------------
------------------
```

```
"""
dim = (len(Pdemand))
Pwind = [0.0]*dim
Psolar = [0.0]*dim
Pinstmilltotal = [0.0]*dim
Pinstsolartotal = [0.0]*dim
Pnuclear = [nuclear*population]*dim
Phydro = [hydro]*dim
Ppump = [0.0]*dim
Pturbine = [0.0]*dim
Pgen = [0.0]*dim
Pdif = [0.0]*dim
Pexport = [0.0]*dim
Pcombcycle = [0.0]*dim
Estorageaverage=[Estorageaverage]*dim
Prenewfinal=0.0
gasenergy=[0.0]*101
"""
-------------------------------------------------------------------------
------------------
 Importació de les dades meteorológiques
-------------------------------------------------------------------------
------------------
"""
meteo = np.loadtxt('meteo.dat')

"""
-------------------------------------------------------------------------
------------------
Càlcul del mix òptim entre energia solar i eòlica
-------------------------------------------------------------------------
------------------
"""

for i in range(0,100):
    Pwind = [0.0]*dim
    Psolar = [0.0]*dim
    for e in range(0,dim):

Pinstwindtotal=Pinstwind+((i/100)*investment*population)/costwind
        Pwind[e]=Pinstwindtotal*meteo[e,1]
        Pinstsolartotal=Pinstsolar+(((100-
i)/100)*investment*population)/costsolar
        Psolar[e]=Pinstsolartotal*meteo[e,0]
        Pgen[e]=Pnuclear[e]+Psolar[e]+Pwind[e]+Phydro[e]
        Pdif[e]=Pgen[e]-Pdemand[e]
        if Pdif[e]>0.0:
           Ppump[e]=min(abs(Pdif[e]),Ppumpmax)
           newEstorage=Estorageaverage[e-1]+Pdif[e]*1
           Ppump[e]=Ppump[e]*(min(Estoragemax,newEstorage)-
Estorageaverage[e-1])/(newEstorage-Estorageaverage[e-1])
           Estorageaverage[e]=min(newEstorage,Estoragemax)
           Pexport[e]=Pdif[e]-Ppump[e]
           Pdemand[e]=Pdemand[e]+Ppump[e]
        else:
```

```
            Pturbine[e]=min(abs(Pdif[e]),Pturbmax)
            newEstorage=Estorageaverage[e-1]+Pdif[e]*1
            Pturbine[e]=Pturbine[e]*(Estorageaverage[e-1]-
max(0.0,newEstorage))/(Estorageaverage[e-1]-newEstorage)
            Estorageaverage[e]=max(newEstorage,0.0)
            Pcombcycle[e]=-Pdif[e]-Pturbine[e]
            Phydro[e]=Phydro[e]+Pturbine[e]
        Pgen[e]=Pnuclear[e]+Psolar[e]+Pwind[e]+Phydro[e]
        Pdif[e]=Pgen[e]-Pdemand[e]
        if Pdif[e]<-1e-7:
            gasenergy[i]=gasenergy[i]+Pdif[e]
optim=max(gasenergy)
best=gasenergy.index(optim)

"""
------------------------------------------------------------------------
------------------
Càlcul de les potències dels diferents recursos
------------------------------------------------------------------------
------------------
"""
Pexport = [0.0]*dim
Pcombcycle = [0.0]*dim
Ppump = [0.0]*dim
Pturbine = [0.0]*dim
Pgen = [0.0]*dim
Pdif = [0.0]*dim
gasenergy2=0.0
Phydro = [hydro]*dim
Pdemand = np.loadtxt('demand5v.dat') #kW



for e in range(0,dim):
    Pinstsolartotal=Pinstsolar+(((100-
best)/100)*investment*population)/costsolar
    Psolar[e]=Pinstsolartotal*meteo[e,0]

Pinstwindtotal=Pinstwind+((best/100)*investment*population)/costwind
    Pwind[e]=Pinstwindtotal*meteo[e,1]
    Pgen[e]=Pnuclear[e]+Psolar[e]+Pwind[e]+Phydro[e]
    Pdif[e]=Pgen[e]-Pdemand[e]
    if Pdif[e]>0.0:
       Ppump[e]=min(abs(Pdif[e]),Ppumpmax)
       newEstorage=Estorageaverage[e-1]+Pdif[e]*1
       if Estorageaverage[e-1] != newEstorage:
           Ppump[e]=Ppump[e]*(min(Estoragemax,newEstorage)-
Estorageaverage[e-1])/(newEstorage-Estorageaverage[e-1])
       Estorageaverage[e]=min(newEstorage,Estoragemax)
       Pexport[e]=Pdif[e]-Ppump[e]
       Pcombcycle[e]=0.0
       Pdemand[e]=Pdemand[e]+Ppump[e]
    else:
        Pturbine[e]=min(abs(Pdif[e]),Pturbmax)
        newEstorage=Estorageaverage[e-1]+Pdif[e]*1
```

```
        if Estorageaverage[e-1] != newEstorage:
            Pturbine[e]=Pturbine[e]*(Estorageaverage[e-1]-
max(0.0,newEstorage))/(Estorageaverage[e-1]-newEstorage)
        Estorageaverage[e]=max(newEstorage,0.0)
        Pexport[e]=0.0
        Pcombcycle[e]=-Pdif[e]-Pturbine[e]
        Phydro[e]=Phydro[e]+Pturbine[e]


"""-----------------------------------------------------------------
--------------------
Càlcul dels valors finals
-----------------------------------------------------------------------
------------------
"""

Etotal=sum(Pdemand)+sum(Pexport)
Xsolar=sum(Psolar)/Etotal*100
Xwind=sum(Pwind)/Etotal*100
Xcombcycle=sum(Pcombcycle)/Etotal*100
Xhydro=sum(Phydro)/Etotal*100
Xnuclear=(sum(Pnuclear))/Etotal*100
kgCO2=CO2factor*sum(Pcombcycle)
print('Investment % wind mills =', best)

print('Psolarinst =', max(Psolar)-6751000)
print('Peolicinst =', max(Pwind)-22961000)
print('Pccinst =', max(Pcombcycle)-24600000)
print('Energy generated with combined cycle [kWh]=', sum(Pcombcycle))
print('Exported energy (%) =', 100*sum(Pexport)/Etotal)
print('CO2 emissions (kgCO2) =', kgCO2)
print('Psolar =', sum(Psolar))
print('Peolic =', sum(Pwind))

"""
-----------------------------------------------------------------------
------------------
    Printing and ploting final values:
-----------------------------------------------------------------------
------------------
"""

# energy mix pie
labels = 'Nuclear','Hydro','Gas','Wind','Solar'
sizes=[Xnuclear,Xhydro,Xcombcycle,Xwind,Xsolar]
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
ax1.axis('equal')

# energy mix pie
labels2 = 'Hydro','Wind','Solar'
sizes2=[Xhydro,Xwind,Xsolar]
fig2, ax2 = plt.subplots()
ax2.pie(sizes2, labels=labels2, autopct='%1.1f%%', startangle=140)
ax2.axis('equal')
```

```
#Exporto a Excel
book = xlwt.Workbook(encoding="utf-8")
sheet1 = book.add_sheet("Sheet 1")
sheet1.write(0, 0, "Psolar")
sheet1.write(0, 1, "Pwind")
sheet1.write(0, 2, "Phydro")
sheet1.write(0, 3, "Pnuclear")
sheet1.write(0, 4, "Pcombcycle")
sheet1.write(0, 5, "Pdemand")
sheet1.write(0, 6, "Pexport")
sheet1.write(0, 7, "Estorageaverage")

for e in range(0,dim):
    sheet1.write(e+1, 0, Psolar[e])

for e in range(0,dim):
    sheet1.write(e+1, 1, Pwind[e])

for e in range(0,dim):
    sheet1.write(e+1, 2, Phydro[e])

for e in range(0,dim):
    sheet1.write(e+1, 3, Pnuclear[e])

for e in range(0,dim):
    sheet1.write(e+1, 4, Pcombcycle[e])

for e in range(0,dim):
    sheet1.write(e+1, 5, Pdemand[e])

for e in range(0,dim):
    sheet1.write(e+1, 6, Pexport[e])

for e in range(0,dim):
    sheet1.write(e+1, 7, Estorageaverage[e])

book.sav
```

2. Codi per trobar la inversió òptima

```
#!/usr/bin/env python3

"""
------------------------------------------------------------------------
------------------
Programa per calcular el mix d'energia produïda a partir de diferents
condicions meteorologiques i de contorn del sistema
------------------------------------------------------------------------
------------------
"""

import numpy as np
import matplotlib.pyplot as plt
import xlwt

"""
------------------------------------------------------------------------
------------------
Importació i definició de les condicions del contorn del sistema
------------------------------------------------------------------------
------------------
"""

data = np.loadtxt('data.dat')
nuclear = data[0]*0.0 # kW per capita
hydro = data[1] # kW
Ppumpmax = data[2] # kW
Pturbmax = data[3] # kW
Estorageaverage = data[4] # kW
Estoragemax = data[5] # kW
costsolar = data[6] # euros/kW
costwind = data[7] # euros/kW
Pinstcombcycle = data[8] #kW
Pinstsolar = data[9] #kW
Pinstwind = data[10] #kW
population = data[11]
investment = data[12] # euros per capita
CO2factor = data[13] #kgCO2/kWhcombcycle

Pdemand = np.loadtxt('demand5v.dat') #kW
print('demand =', sum(Pdemand))

"""
------------------------------------------------------------------------
------------------
Inicialització de les variables auxiliars
------------------------------------------------------------------------
------------------
"""
dim = (len(Pdemand))
```

```
Pwind = [0.0]*dim
Psolar = [0.0]*dim
Pinstmilltotal = [0.0]*dim
Pinstsolartotal = [0.0]*dim
Pnuclear = [nuclear*population]*dim
Phydro = [hydro]*dim
Ppump = [0.0]*dim
Pturbine = [0.0]*dim
Pgen = [0.0]*dim
Pdif = [0.0]*dim
Pexport = [0.0]*dim
Pcombcycle = [0.0]*dim
Estorageaverage=[Estorageaverage]*dim
Prenewfinal=0.0
gasenergy=[0.0]*101
"""
------------------------------------------------------------------------
------------------
 Importació de les dades meteorológiques
------------------------------------------------------------------------
------------------
"""
meteo = np.loadtxt('meteo.dat')

"""
------------------------------------------------------------------------
------------------
Càlcul del mix òptim entre energia solar i eòlica
------------------------------------------------------------------------
------------------
"""

for i in range(0,100):
    Pwind = [0.0]*dim
    Psolar = [0.0]*dim
    for e in range(0,dim):

Pinstwindtotal=Pinstwind+((i/100)*investment*population)/costwind
        Pwind[e]=Pinstwindtotal*meteo[e,1]
        Pinstsolartotal=Pinstsolar+(((100-
i)/100)*investment*population)/costsolar
        Psolar[e]=Pinstsolartotal*meteo[e,0]
        Pgen[e]=Pnuclear[e]+Psolar[e]+Pwind[e]+Phydro[e]
        Pdif[e]=Pgen[e]-Pdemand[e]
        if Pdif[e]>0.0:
           Ppump[e]=min(abs(Pdif[e]),Ppumpmax)
           newEstorage=Estorageaverage[e-1]+Pdif[e]*1
           Ppump[e]=Ppump[e]*(min(Estoragemax,newEstorage)-
Estorageaverage[e-1])/(newEstorage-Estorageaverage[e-1])
           Estorageaverage[e]=min(newEstorage,Estoragemax)
           Pexport[e]=Pdif[e]-Ppump[e]
           Pdemand[e]=Pdemand[e]+Ppump[e]
        else:
            Pturbine[e]=min(abs(Pdif[e]),Pturbmax)
            newEstorage=Estorageaverage[e-1]+Pdif[e]*1
```

```
            Pturbine[e]=Pturbine[e]*(Estorageaverage[e-1]-
max(0.0,newEstorage))/(Estorageaverage[e-1]-newEstorage)
            Estorageaverage[e]=max(newEstorage,0.0)
            Pcombcycle[e]=-Pdif[e]-Pturbine[e]
            Phydro[e]=Phydro[e]+Pturbine[e]
        Pgen[e]=Pnuclear[e]+Psolar[e]+Pwind[e]+Phydro[e]
        Pdif[e]=Pgen[e]-Pdemand[e]
        if Pdif[e]<-1e-7:
            gasenergy[i]=gasenergy[i]+Pdif[e]
optim=max(gasenergy)
best=gasenergy.index(optim)

"""
------------------------------------------------------------------------
------------------
Càlcul de les potències dels diferents recursos
------------------------------------------------------------------------
------------------
"""
Pexport = [0.0]*dim
Pcombcycle = [0.0]*dim
Ppump = [0.0]*dim
Pturbine = [0.0]*dim
Pgen = [0.0]*dim
Pdif = [0.0]*dim
gasenergy2=0.0
Phydro = [hydro]*dim
Pdemand = np.loadtxt('demand5v.dat') #kW


for e in range(0,dim):
    Pinstsolartotal=Pinstsolar+(((100-
best)/100)*investment*population)/costsolar
    Psolar[e]=Pinstsolartotal*meteo[e,0]

Pinstwindtotal=Pinstwind+((best/100)*investment*population)/costwind
    Pwind[e]=Pinstwindtotal*meteo[e,1]
    Pgen[e]=Pnuclear[e]+Psolar[e]+Pwind[e]+Phydro[e]
    Pdif[e]=Pgen[e]-Pdemand[e]
    if Pdif[e]>0.0:
       Ppump[e]=min(abs(Pdif[e]),Ppumpmax)
       newEstorage=Estorageaverage[e-1]+Pdif[e]*1
       if Estorageaverage[e-1] != newEstorage:
           Ppump[e]=Ppump[e]*(min(Estoragemax,newEstorage)-
Estorageaverage[e-1])/(newEstorage-Estorageaverage[e-1])
       Estorageaverage[e]=min(newEstorage,Estoragemax)
       Pexport[e]=Pdif[e]-Ppump[e]
       Pcombcycle[e]=0.0
       Pdemand[e]=Pdemand[e]+Ppump[e]
    else:
        Pturbine[e]=min(abs(Pdif[e]),Pturbmax)
        newEstorage=Estorageaverage[e-1]+Pdif[e]*1
        if Estorageaverage[e-1] != newEstorage:
            Pturbine[e]=Pturbine[e]*(Estorageaverage[e-1]-
max(0.0,newEstorage))/(Estorageaverage[e-1]-newEstorage)
```

```
        Estorageaverage[e]=max(newEstorage,0.0)
        Pexport[e]=0.0
        Pcombcycle[e]=-Pdif[e]-Pturbine[e]
        Phydro[e]=Phydro[e]+Pturbine[e]


"""-------------------------------------------------------------------
--------------------
Càlcul dels valors finals
-------------------------------------------------------------------
-----------------
"""

Etotal=sum(Pdemand)+sum(Pexport)
Xsolar=sum(Psolar)/Etotal*100
Xwind=sum(Pwind)/Etotal*100
Xcombcycle=sum(Pcombcycle)/Etotal*100
Xhydro=sum(Phydro)/Etotal*100
Xnuclear=(sum(Pnuclear))/Etotal*100
kgCO2=CO2factor*sum(Pcombcycle)
print('Investment % wind mills =', best)
print('Energy generated with combined cycle [kWh]=', sum(Pcombcycle))
print('Exported energy (%) =', 100*sum(Pexport)/Etotal)
print('CO2 emissions (kgCO2) =', kgCO2)

"""
-------------------------------------------------------------------
-----------------
    Printing and ploting final values:
-------------------------------------------------------------------
-----------------
"""

# energy mix pie
labels = 'Hydro','Gas','Wind','Solar'
sizes=[Xhydro,Xcombcycle,Xwind,Xsolar]
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
ax1.axis('equal')

# energy mix pie
labels2 = 'Hydro','Wind','Solar'
sizes2=[Xhydro,Xwind,Xsolar]
fig2, ax2 = plt.subplots()
ax2.pie(sizes2, labels=labels2, autopct='%1.1f%%', startangle=140)
ax2.axis('equal')

#Exporto a Excel
book = xlwt.Workbook(encoding="utf-8")
sheet1 = book.add_sheet("Sheet 1")
sheet1.write(0, 0, "Psolar")
sheet1.write(0, 1, "Pwind")
sheet1.write(0, 2, "Phydro")
sheet1.write(0, 3, "Pnuclear")
sheet1.write(0, 4, "Pcombcycle")
```

```
sheet1.write(0, 5, "Pdemand")
sheet1.write(0, 6, "Pexport")
sheet1.write(0, 7, "Estorageaverage")

for e in range(0,dim):
    sheet1.write(e+1, 0, Psolar[e])

for e in range(0,dim):
    sheet1.write(e+1, 1, Pwind[e])

for e in range(0,dim):
    sheet1.write(e+1, 2, Phydro[e])

for e in range(0,dim):
    sheet1.write(e+1, 3, Pnuclear[e])

for e in range(0,dim):
    sheet1.write(e+1, 4, Pcombcycle[e])

for e in range(0,dim):
    sheet1.write(e+1, 5, Pdemand[e])

for e in range(0,dim):
    sheet1.write(e+1, 6, Pexport[e])

for e in range(0,dim):
    sheet1.write(e+1, 7, Estorageaverage[e])

book.save("output.xls")
```