AUTUMN 2022

RESEARCH PROJECT

Project Report CS357

Economic emission dispatch Problem

Under the Guidence of : **Dr Kapil Ahuja**

Team Member

Umang Jain 200001076

Abhishek Jaiswal 200001001

Abstract

- The challenge of Economic Load Dispatch (ELD), which arises as a result of the increased energy needs, is brought on by the need to operate thermal power plants as inexpensively as possible.
- The challenge of ELD in the power system is to plan the power output for each committed generating unit in such a way that operating costs are reduced while load fulfilling demand and power operational restrictions.
- Power system designers are primarily concerned with the environment in the current situation since a rise in power generation has led to an increase in Emissions

- In addition to lowering operating costs, pollution management has grown to be a crucial operational goal for obtaining sustainable energy.
- The Economic Emission Dispatch (EED) problem is a multi-dimensional power system optimization problem that is the consequence of combining both the economic dispatch and emission dispatch objectives.In this paper, the Lambda Iteration Method and the Newton technique have been used to solve the EED problem.
- On three generator systems both solution techniques have been validated. A comparison of the outcomes from the two approaches has been carried out.

Page2

Introduction

EED: (Economic Emission Dispatch)

The major objective of today's huge, interconnected power systems is to distribute the load demand among participating generators as quickly as feasible.

Economic dispatch is the process of determining how much power each generator in a power system will produce while minimizing fuel costs and ensuring that different system constraints are met.

The system restrictions include ensuring system stability, running the generators within allowable parameters, and matching the power generation with the load.

When fossil fuels are burnt, toxic gasses are released, such as oxides of carbon, oxides of sulfur and oxides of nitrogen.

These gases pollute the atmosphere, which disturbs the ecological balance and causes global warming.

The environment has become inappropriate for the survival of living things as a result of the increased pollution caused by the increased energy production to fulfill the rising demand.

Page3:

It is necessary to reduce emissions and fuel costs, a process known as emission dispatch, in order to meet the demand for clean energy.

While minimizing the emissions, there is a need to satisfy the system constraints. When economic dispatch and emission dispatch problems are combined together it becomes as EED problem.

Problem formulation

Consider a system of N thermal-generating units connected to a single bus-bar serving the electrical load. The input for each unit is F_i. The output of each unit is P_{gi}. The problem can be divided into two phases:

a) Economic dispatch:
Transmission losses are neglected. The objective function F_T is equal to the total cost of all units. The problem is to minimize this function while satisfying constraints(given later).

$$F_T = F_1 + F_2 + ... + F_N \quad (1)$$
$$F_T = \Sigma_{i=1}^{N} F_i(P_{gi}) \quad (2)$$

The fuel cost for each unit(without valve point loading) is given as a second order polynomial:

$$F_i(P_{gi}) = a_i P_{gi}^2 + b_i P_{gi} + c_i \quad \text{Rs/hr} \quad (3)$$

Hence the problem can be summarized as:
$$Min\ F_T \quad (4)$$

(b) Emission Dispatch:
The objective E_T for the total emission from all units is given as:

$$E_T = E_1 + E_2 + ... + E_N \quad (5)$$
$$E_T = \Sigma_{i=1}^{N} E_i(P_{gi}) \quad (6)$$

The emission from each unit can be given as a second order polynomial as:

$$E_i(P_{gi}) = \alpha_i P_{gi}^2 + \beta_i P_{gi} + \gamma_i \quad \text{Kg/hr} \quad (7)$$

Page4:

Hence the problem can be summarized as:
$$Min\ E_T \quad (8)$$

System constraints:

The constraints considered in the case of minimizing economic and emission dispatch functions are:

(a) Equality constraints:

Total power generation must cover both power demand and power loss. This helps us formulate the equality constraint as:

$\Sigma_{i=1}^{N} P_{gi} = P_D + P_{LOSS}$ (9)

(b) Inequality constraints:

Each generator unit would have a lower and upper limit on the amount of power it can generate. These make up our inequality constraints. The constraints can be formulated as:

$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max}$ (for i = 1 to N) (10)

Algorithms

We initially planned to implement two algorithms:
- Lambda-iteration algorithm
- Particle Swarm Optimization algorithm

However, PSO proved to be a bit complex and obscure for us to implement in given time constraints, so we implemented these algorithms for the final report:
- Lambda-iteration algorithm
- Newton's method

First we convert the two objective functions into a single objective function by considering two weights w1 and w2 such that w1+w2=1 and the new objective function will be w1F1+w2F2. Since both objective functions are similar we will just get a new function similar to the objective

Page5:

:functions with new weighted coefficients. For the rest of this report we will assume that the weighted coefficients are being used.

Lambda-iteration algorithm

Using lagrange multipliers, define an augmented function:

$L(P_i,\lambda)=F(P_i) + \lambda(P_D - P_L - \Sigma_{i=1}^{n_g} P_i)$

Where λ is the lagrange multiplier. Now, in case of lagrange multiplier method:

$\frac{\partial L}{\partial Pi}$=0

$\frac{\partial F}{\partial Pi}$ +λ($\frac{\partial P_L}{\partial Pi}$-1)=0

We will not consider losses for now, so the equality becomes:

$\frac{\partial F}{\partial Pi} = \lambda$ - (i)

For optimal dispatch, we assume that the incremental cost of running each unit is equal
Hence

$\frac{\partial F_i}{\partial Pi} = \lambda$ for all i - (ii)

Hence, λ is the incremental cost. Also:

$F_i(P_{gi}) = a_iP_{gi}^2 + b_iP_{gi} + c_i$

Where a,b and c are the weighted coefficients and F is the weighted function combining economic and emission problems.
Hence, from (ii):

$2a_iP_{gi} + b_i = \lambda$

$P_{gi} = \frac{\lambda - bi}{2ai}$ - (iii)

The lambda iteration method follows the following steps:

1. Choose initial values of λ and Δλ *.*
2. Calculate all P_{gi} using (iii).
3. For each P check the inequality constraint:

   $P_{gi}^{min} \le P_{gi} \le P_{gi}^{max}$ (for i = 1 to N)

   If $P_{gi}^{max} < P_{gi}$ set $P_{gi} = P_{gi}^{max}$
4. Calculate:

Page6:

$\Delta P = \Sigma_{i=1}^{n_g} P_{ig} - P_D$

5. If ΔP is less than some specified value then end the algorithm and return the values of power of this iteration along with cost. Else move to step 6.
6. Increase or decrease the value of λ by Δλ accordingly and repeat from step 2.

Newton method

If we observe the lagrangian function we will observe that the aim for optimization is to make:

$\Delta L_x=0$

Hence this problem can be reduced to a problem of finding the correction that drives this function to 0. To make the function f(x) 0:

$f(x+\Delta x)=f(x)+f'(x)\Delta x=0$

Hence the adjustment for a given step comes to be $\Delta x=-[f'(x)]^{-1}f(x)$

Where $f(x)=\Delta L_x$

$\Delta x=-[\frac{\partial}{\partial x}\Delta L_x]^{-1}\Delta L_x$

$[\frac{\partial}{\partial x}\Delta L_x]^{-1}$ will be the inverse of the Hessian.

Coding

We coded the implementations in python.

Lambda-iteration

```
import pandas as pd
import numpy as np
n=3
a=np.array([0.06,0.065,0.07], dtype=float)
b=np.array([7.3,5.5,6.8], dtype=float)
c=np.array([700,500,300], dtype=float)
pmin=np.array([200,150,100], dtype=float)
pmax=np.array([450,350,225], dtype=float)
```



```
lam=float(input('Enter the value of lambda: '))
p=np.zeros(n)
dem=float(input('enter demand load: '))
dp=float(10)
e=.0001
it=0
while abs(dp)>e:
    tot=0
    for i in range(n):
```

```
        p[i]=(lam-b[i])/(2*a[i])
        if p[i]>pmax[i]:
            p[i]=pmax[i]
        elif p[i]<pmin[i]:
            p[i]=pmin[i]
        tot=tot+a[i]*p[i]*p[i]+b[i]*p[i]+c[i]
    dp=(dem-np.sum(p))/dem
    if abs(dp)>0:
        t=0
        for j in range(n):
            t=t+(1/2*a[j])
    lam=lam+dp/t
    it=it+1
for i in range(n):
    print(f'For generator: {i} power: {p[i]} cost: {tot} iterations(Only for Lambda-iteration): {it}')
```

Newton method

```
import pandas as pd
import numpy as np
n=3
a=np.array([0.06,0.065,0.07], dtype=float)
b=np.array([7.3,5.5,6.8], dtype=float)
c=np.array([700,500,300], dtype=float)
pmin=np.array([200,150,100], dtype=float)
pmax=np.array([450,350,225], dtype=float)
lam=float(input('Enter the value of lambda: '))
p=np.zeros(n)
dem=float(input('enter demand load: '))
dp=float(10)
e=.0001
L=np.zeros((n,),dtype=float)
```



```
dL=np.zeros((n+1,),dtype=float)
H=np.zeros((n+1,n+1),dtype=float)
dP=0
tot=0
ps=0
for i in range(n):
    ps=ps+p[i]
    t=2*a[i]*p[i]
    L[i]=b[i]+t
```

```
    dL[i]=L[i]-lam
    H[i][i]=2*a[i]
    H[i][n]=-1
    H[n][i]=-1
dL[n]=dem-ps
M=np.linalg.inv(H)
dP=-np.matmul(M,dL)
for i in range(n):
    p[i]=p[i]+dP[i]
    tot=tot+c[i]+b[i]*p[i]+a[i]*p[i]*p[i]
```

Simulation

For testing we consider 3 generators with weighted coefficients:
a=0.06,0.065,0.07
b=7.3,5.5,6.8
c=700,500,300
And the values of Pmin and Pmax:
Pmin=200,150,100
Pmax=450,350,225

We consider the initial value of λ as 2 and demand load as 1000. The value of e is 0.0001.

Results

Number of iterations taken by Lambda-iteration method=88

Generator number	P(Lambda iteration)	P(Newton)
0	424.9001779608388	353.41897233201576
1	350.0	340.07905138339925
2	225.0	306.50197628458494

Final cost for lambda iteration method=30395.43097298327
Final cost for Newton method=29622.441699604737

Conclusions

We observe that Newton's method gives better approximation for the dataset we chose in one step than lambda-iteration method does in many steps. Although it requires the calculation of the hessian which might not be feasible in all questions. Similarly the steps followed in the lambda-iteration method might not be feasible for all kinds of questions. So the methods should be applied whenever they are feasible. It’s also possible that the results might have been different in other situations. All in all we conclude that the two methods are easy to implement and it's not hard to see why they are considered traditional methods to solve the economic dispatch problem.

References

Comparative Analysis of Lambda Iteration Method and Particle Swarm Optimization for Economic Emission Dispatch Problem - Vipandeep Kour, Lakhwinder Singh

Economic load dispatch problem and MATLAB programming of different methods - Rahul Dogra and Nikita Gupta