**UNIVERSITY OF ENGINEERING AND**

**TECHNOLOGY LAHORE**



**Assignment # 2**

**Economic Dispatch with Linear Programming**

**Course Title: Advanced Power System Operation and Control**

**Course Code: EE 641**

**Submitted to:**

**Dr. Muhammad Asghar Saqib**

**Assistant Professor**

**Submitted by:**

**Muhammad Shamaas**

**ID # 2018-MS-EE-4**

**Date of Submission: 3 March 2020**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**Economic Dispatch with Linear Programming**

**Problem Statement**

This study aims to determine the economic operating point for three generator units when delivering a total of 850 MW. The operating cost of each unit is specified as:

The operating limits of the units are:

The incremental cost rates of the units are:

We must minimize the total operating cost:

The Lagrange Function is:

At the economic operating point,

Using the incremental cost rates of the three units,

The exact solution is:

The solution will now be presented using iterative method of Linear Programing. This method uses piecewise linear cost functions for the three units. A sample case of 2 linear segments is presented to illustrate the method of lambda search using Linear Programming.

**Case Study: 2 Segments**

The cost functions are partitioned into two linear segments as follows:

The conditions for the segments are:

The corresponding incremental operating costs are

The piecewise linear cost functions are shown in the figure below.



**Iteration # 1**

This is not a valid solution.

**Iteration # 2**

This is not a valid solution.

**Iteration # 3**

This is not a valid solution.

**Iteration # 4**

This is a valid solution.

**Iteration # 5**

This is not a valid solution.

**Iteration # 6**

This is not a valid solution.

**Solution**

The corresponding operating cost is:

The operating points are shown in the figure below using circles.

****

This algorithm was implemented in MATLAB. The results are presented here for the cases of 1, 2, 3, 5, 10 and 50 segments.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Segments | Generator 1  (MW) | Generator 2  (MW) | Generator 3  (MW) | Total Cost  ($/h) | Lambda  ($/MWh) |
| 1 | 400 | 400 | 50 | 8227.870 | 9.0915 |
| 2 | 375 | 350 | 125 | 8195.369 | 9.1110 |
| 3 | 450 | 300 | 100 | 8204.105 | 9.0915 |
| 5 | 400 | 340 | 110 | 8195.206 | 9.0915 |
| 10 | 385 | 340 | 125 | 8194.554 | 9.1618 |
| 50 | 393 | 335 | 122 | 8194.357 | 9.1576 |
| Standard solution with lambda search | 393.2 | 334.6 | 122.2 | 8194.356 | 9.1480 |

**MATLAB Code**

clc;clear all;

%%

Ngen=3;

Pmax=[600 400 200];

Pmin=[150 100 050];

Pload=850;

divisions=50;

%%

for i=1:Ngen

range(i)=Pmax(i)-Pmin(i);

dP(i)=range(i)/divisions;

end

%%

for i=1:Ngen

for k=1:divisions

sPmin(i,k)=Pmin(i)+(k-1)\*dP(i);

sPmax(i,k)=Pmin(i)+k \*dP(i);

sFmin(i,k)=F(i,sPmin(i,k));

sFmax(i,k)=F(i,sPmax(i,k));

s (i,k)=(sFmax(i,k)-sFmin(i,k))/dP(i);

end

end

ordered\_s=sort(transpose(reshape(s,[],1)));

%%

Pgen=[0 0 0;0 0 0];

Pgen=[Pmin(1) Pmin(2) Pmin(3);Pmax(1) Pmax(2) Pmax(3)];

eps1=Pload;

eps2=Pload;

threshold=1;

iter=1;

maxiter=Ngen\*divisions;

TotalCost1=0;

TotalCost2=0;

found=0;

Ans=[];

OldCostMin=1e5;

Oldeps=Pload;

lamdaOld=0;

for n=1:maxiter

lamda=ordered\_s(n);

for rep=1:2

for i=1:Ngen

for k=1:divisions

if (s(i,k)<lamda)

if (k<divisions)

if (s(i,k+1)>lamda)

Pgen(2,i)=sPmax(i,k);

end

else

Pgen(2,i)=sPmax(i,k);

end

if (divisions==1)

Pgen(1,i)=sPmax(i,k);

end

end

if (s(i,k)>lamda)

if (k>1)

if (s(i,k-1)<lamda)

Pgen(1,i)=sPmin(i,k);

end

else

Pgen(1,i)=sPmin(i,k);

end

if (divisions==1)

Pgen(2,i)=sPmin(i,k);

end

end

if (s(i,k)==lamda)

Pgen(1,i)=sPmin(i,k);

Pgen(2,i)=sPmax(i,k);

t1=Pload-sum(Pgen(1,:))+Pgen(1,i);

t2=Pload-sum(Pgen(2,:))+Pgen(2,i);

if ((t1<=sPmax(i,k))&&(t1>=sPmin(i,k)))

Pgen(1,i)=t1;

end

if ((t2<=sPmax(i,k))&&(t2>=sPmin(i,k)))

Pgen(2,i)=t2;

end

end

Pgen;

eps1=abs(sum(Pgen(1,:))-Pload);

eps2=abs(sum(Pgen(2,:))-Pload);

eps\_his(n\*i\*k)=eps;

TotalCost1=0;

TotalCost2=0;

for m=1:Ngen

TotalCost1=TotalCost1+F(m,Pgen(1,m));

TotalCost2=TotalCost2+F(m,Pgen(2,m));

end

if(eps1==0)&&(TotalCost1<OldCostMin)

Ans=Pgen;

TotalCost1;

Oldeps=eps1;

OldCostMin=TotalCost1;

lamdaOld=lamda;

end

if(eps2==0)&&(TotalCost2<OldCostMin)

Ans=Pgen;

TotalCost2;

Oldeps=eps2;

OldCostMin=TotalCost2;

lamdaOld=lamda;

end

end

end

end

lamda;

Pgen;

%plot(Pgen,lamda,'o')

Pgen=[Pmin(1) Pmin(2) Pmin(3);Pmax(1) Pmax(2) Pmax(3)];

end

Pgen=Ans(2,:);

lamda=lamdaOld;

% hold on

% stairs([sPmin(1,1) sPmax(1,1)],[s(1,1),s(1,1)],'r')

% stairs([sPmin(1,2) sPmax(1,1)],[s(1,1),s(1,2)],'r')

% stairs([sPmin(1,2) sPmax(1,2)],[s(1,2),s(1,2)],'r')

%

% stairs([sPmin(2,1) sPmax(2,1)],[s(2,1),s(2,1)],'b')

% stairs([sPmin(2,2) sPmax(2,1)],[s(2,1),s(2,2)],'b')

% stairs([sPmin(2,2) sPmax(2,2)],[s(2,2),s(2,2)],'b')

%

% stairs([sPmin(3,1) sPmax(3,1)],[s(3,1),s(3,1)],'g')

% stairs([sPmin(3,2) sPmax(3,1)],[s(3,1),s(3,2)],'g')

% stairs([sPmin(3,2) sPmax(3,2)],[s(3,2),s(3,2)],'g')

% xlabel('Pgen (MW)');

% ylabel('dF/dPgen ($/MWh)');

% text([500],[9.5],'Unit 1');

% text([300],[9.15],'Unit 2');

% text([160],[9.6],'Unit 3');

% plot([50,600],[lamda,lamda])

% plot(Pgen,lamda,'o')

% plot(Pgen(1,:),[s(1,1) lamdaOld s(3,1)],'o')

TotalCost=0;

for i=1:Ngen

TotalCost=TotalCost+F(i,Pgen(1,i));

end

Pgen

TotalCost

lamda